Published Date: - 01-05-2025
Page No: - 190-194

OPPORTUNITIES OF SPONGE CONSTRUCTION IN CRYPTOGRAPHIC SYSTEMS: DESIGN SETTINGS AND SECURITY CONSIDERATIONS

Zarif Khudoykulov
PhD, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Uzbekistan

ABSTRACT

Sponge construction has emerged as a versatile and unified structure for designing cryptographic primitives such as hash functions, authenticated encryption with associated data (AEAD), pseudorandom number generators (PRNG), and stream ciphers. This paper analyzes the configurable parameters of sponge construction, including state size, capacity, and rate, and discusses their security implications. A comparative study of existing sponge-based cryptosystems is provided, and general recommendations for design settings are presented to ensure resistance against known attacks while optimizing performance for various applications. The significance of sponge construction in enabling lightweight, secure, and flexible cryptographic solutions is explored, particularly in the context of modern demands such as IoT, embedded systems, and post-quantum resistance.

KEYWORDS

Sponge construction, hash function, security consideration, attacks, design guidelines.

INTRODUCTION

Cryptographic constructions are foundational to secure communications, digital signatures, authentication mechanisms, and secure key management. Hash functions, AEAD ciphers, and PRNGs are critical in ensuring confidentiality, integrity, and availability in modern cryptographic protocols. Traditional hash constructions such as Merkle-Damgård and Davies-Meyer schemes have been widely adopted but exhibit structural weaknesses like poor parallelism and vulnerability to specific types of attacks (e.g., length extension or collision attacks) [1].

AEAD constructions based on block ciphers such as AES-GCM, while effective, often face challenges in low-resource environments due to computational overhead and implementation complexity [2]. Stream ciphers like RC4 and ChaCha provide flexibility but lack a unified structure to extend to other cryptographic functions.

The sponge construction, introduced by Bertoni et al., unifies multiple cryptographic functionalities under a single abstraction [3]. It underpins the SHA-3 standard [6] and enables robust AEAD schemes like Ascon [4] and Ketje [5]. Due to its configurability and simplicity, it is ideal for constrained environments such as IoT devices, RFID systems, and embedded cryptographic modules. Sponge's ability to absorb arbitrary-length input and squeeze an output of arbitrary length makes it a unique and scalable design for diverse cryptographic needs.

2. Sponge Construction: Architecture and Functionalities

Sponge construction is based on a simple yet powerful principle. It operates on a fixed-size internal state divided into two parts: the rate (r) and the capacity (c). The construction

alternates between an absorbing phase, where input is mixed into the state, and a squeezing phase, where output is extracted (Figure 1).

State (b): Total size of the internal state (b = r + c). It determines the memory footprint and affects performance and security.

Rate (r): Number of bits processed per iteration. A higher rate improves performance but reduces security.

Capacity (c): Bits reserved for internal secrecy and protection against attacks. Higher capacity increases resistance to collision, preimage, and state recovery attacks.

Permutation function (l): A cryptographically secure transformation applied to the state after each absorption/squeezing step.

Padding rules (pad): padding is a crucial aspect of sponge construction because it ensures that input messages of arbitrary length are properly aligned to the block size used during the absorbing phase. The padding rule must satisfy the multi-rate padding property, meaning no two distinct messages can be padded into the same final input to the sponge function (pad10^* 1).

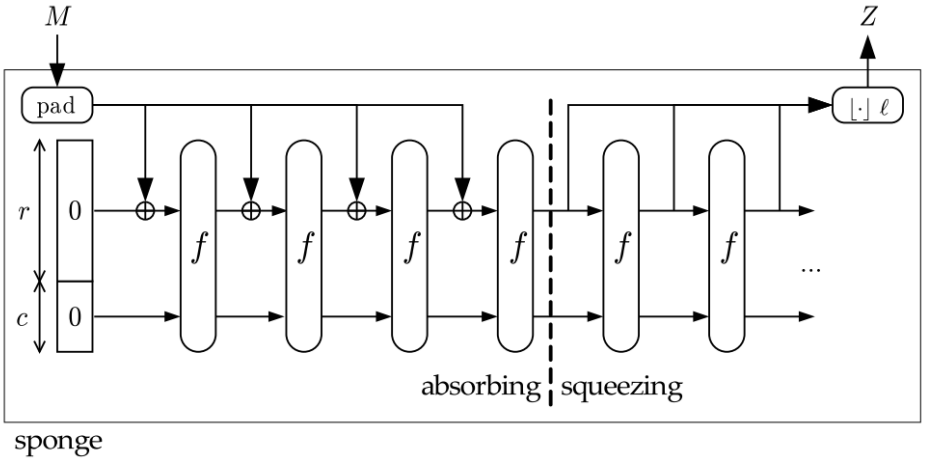

Figure 1. The sponge construction [3]

2.1 Duplex Construction

The duplex construction is a variant that allows simultaneous input/output operations. It maintains the same structure as the sponge but provides better support for incremental input/output, essential for online encryption and streaming modes. AEAD schemes using duplex mode, like Ascon and Ketje, offer secure and authenticated data streams while managing associated data efficiently (Figure 2).

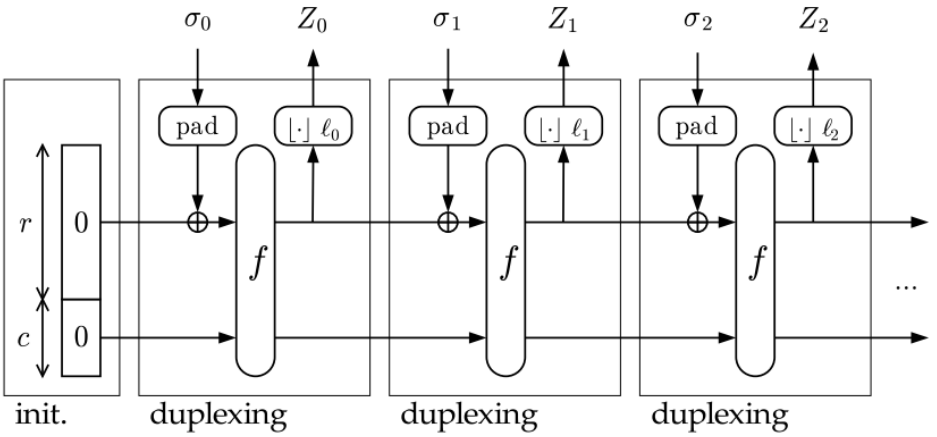

Figure 2. The duplex construction [3]

2.2 Inner Functions

The permutation function f is the core of the sponge. The permutation function is the core transformation applied to the internal state during each absorption or squeezing phase. Its design is critical to the security and efficiency of the cryptographic system. The structure of inner functions in sponge constructions typically follows a round-based design combining non-linear substitution layers (S-boxes), linear mixing or permutation layers, addition of round constants, and operations like modular addition, rotation, and XOR (ARX) to achieve strong diffusion and cryptographic strength.

The purpose of the inner function in sponge constructions is to ensure diffusion and confusion across the internal state, making each output bit a complex function of all previous inputs and providing resistance against various cryptanalytic attacks. Security expectations from inner functions in sponge constructions include invertibility, strong diffusion to prevent fixed points and differential biases, resistance to collision, preimage, and second preimage attacks, and the ability to produce an avalanche effect to ensure high unpredictability of output bits. Examples include Keccak-f in SHA-3 (1600-bit state), Xoodoo in Xoodyak (384-bit state), and the Ascon-p permutation (320-bit state).

3. Comparative Analysis of Sponge-Based Cryptosystems

Here's a concise comparative analysis of popular sponge-based cryptosystems, focusing on their architecture, internal parameters, and application types (Table 1).

Table 1. Comparative Analysis of Sponge-Based Cryptosystems

Scheme	Type	State Size (b)	Rate (r)	Capacity (c)	Permutation Function	Applications
Keccak	Hash	1600	1088	512	Keccak-f [b]	SHA-3
Ascon	AEAD, Hash	320	64 128	256 192	Ascon-p	NIST LWC
Ketje	AEAD	200, 400, 800, 1600	16, 32, 128, 256	184, 368, 672, 1344	Keccak-f [b]	CAESAR
Xoodyak	AEAD	384	192	192	Xoodoo	NIST LWC
PHOTON	Hash	100, 144, 196, 256, 288	20, 16, 36, 32, 32	80, 128, 160, 224, 256	AES like permutation	Hashing, AEAD

Each of these systems adjusts state and capacity to balance between efficiency and resistance to cryptanalysis. Keccak provides extremely high flexibility with large states, while Xoodyak and Ascon are optimized for minimal memory overhead and fast software/ hardware implementations [7].

4. Security Considerations and Attacks

Sponge security depends on the quality of the permutation function and the chosen capacity size. Common attack vectors include:

Collision Attacks: The birthday bound dictates that to resist 2^(n/2) collision attacks, the capacity must be at least 2n bits.

Preimage Attacks: Similar to collision resistance, achieving 2^n preimage resistance requires a capacity of at least n bits.

Length Extension: Sponge construction naturally resists length extension attacks due to its absorption mechanism.

State Recovery Attacks: Attackers try to reconstruct the internal state. A high-capacity value mitigates these attacks.

Side-channel Attacks: Physical attacks like power analysis target the implementation. Countermeasures include masking, constant-time operations, and differential power analysis (DPA) resistance.

Differential and Linear Cryptanalysis: Analyzed against the permutation function; well-designed permutations (like Keccak-f, Ascon-p) have strong resistance due to avalanche effect and round diffusion.

Security proofs often rely on the indifferentiability of the sponge from a random oracle, under the assumption that the permutation behaves ideally. For AEAD modes, proofs are based on indistinguishability from an ideal authenticated encryption scheme.

5. Design Guidelines and Tolerance to Attacks

Different applications demand different security/performance trade-offs. Below are general guidelines [8]:

Hash Functions:

Target security: 256-bit collision resistance.

Recommended capacity: ≥ 512 bits.

Example: Keccak with b = 1600, r = 1088, c = 512.

AEAD for Constrained Devices:

Target security: 128-bit confidentiality and integrity.

Capacity: ≥ 256 bits.

Example: Ascon with b = 320, r = 64, c = 256.

Stream Ciphers / PRNGs:

Target: high throughput, moderate security.

Capacity: ≥ 128 bits for resistance to distinguishing attacks.

Example: Xoodyak with b = 384, r = 192, c = 192.

Proof of Security Levels:

Let c be the capacity.

Resistance to collision: up to 2^(c/2).

Resistance to preimage: up to 2^c.

Tolerance to adversary data processing: up to 2^r messages safely (depending on padding and domain separation).

A good security margin is choosing c such that:

For 128-bit security: c = 256.

For 256-bit security: c = 512. This ensures resistance to both quantum and classical attacks.

Sponge constructions offer a versatile framework for cryptographic functions, balancing security, efficiency, and flexibility. By adhering to robust design principles—such as selecting

secure permutations, tuning r and c and implementing proper padding—sponge-based systems can resist a wide array of attacks.

CONCLUSION

Sponge construction offers a modern, unified framework suitable for a wide range of cryptographic functions. Its parameter-driven design enables fine-tuning of performance and security, especially in lightweight and high-assurance applications. With strong theoretical foundations and proven deployments (e.g., SHA-3, Ascon), sponge construction is likely to dominate next-generation cryptographic system design. Future work includes optimizing hardware implementations, analyzing post-quantum security aspects, and developing new inner functions for constrained environments.

REFERENCES

1. Kundu R., Dutta A. Cryptographic Hash Functions and Attacks-A Detailed Study //International Journal of Advanced Research in Computer Science. – 2020. – Т. 11. – №. 2.
2. Kampanakis P. et al. Practical challenges with AES-GCM and the need for a new cipher //The Third NIST Workshop on Block Cipher Modes of Operation. – 2023.
3. Guido B. et al. Cryptographic sponge functions //2011-STMicroelectronics NXP Semiconductors, Version 0.1 January 14. – 2011.
4. Dobraunig C. et al. Ascon v1. 2: Lightweight authenticated encryption and hashing //Journal of Cryptology. – 2021. – Т. 34. – С. 1-42.
5. Guido B. et al. Caesar submission: K v2 //Ketjev2-doc2. 0. pdf. – 2014.
6. Bertoni G. et al. Keccak specifications //Submission to nist (round 2). – 2009. – Т. – №. 30. – С. 320-337.
7. Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2016). Xoodyak. https://keccak.team/xoodyak.html
8. Andreeva E. et al. Security of keyed sponge constructions using a modular proof approach //International Workshop on Fast Software Encryption. – Berlin, Heidelberg : Springer Berlin Heidelberg, 2015. – С. 364-384.