SECURITY ANALYSIS OF SPONGE CONSTRUCTIONS

Zarif Khudoykulov
PhD, Tashkent University of Information
Technologies named after Muhammad al-Khwarizmi,
zarif.khudoykulov@tuit.uz

Abstract

The sponge construction is a versatile cryptographic framework supporting keyless applications like hashing and keyed applications such as MACs and stream ciphers, with security reliant on the capacity c and the robustness of the permutation f. This paper classifies attacks into generic (e.g., collision, preimage, length extension, meet-in-the-middle) and primary (e.g., differential, linear) types, detailing their mechanisms, goals, and security bounds for both modes. Generic attacks are bounded by c, while primary attacks exploit f's structural weaknesses to achieve lower complexity. We explore the role of key length k in security bounds for keyed modes, including potential adjustments like $\min(2^{c/2}, 2^{k})$ for certain attacks, and emphasize the hermetic sponge strategy to mitigate vulnerabilities. The analysis provides insights into designing secure sponge-based systems.

Keywords: Sponge construction, cryptographic attacks, hash functions, keyed modes, security bounds.

1. Introduction

Secure Hash Algorithms (SHAs) are cryptographic functions that convert data of any size into a fixed-length bit string, ensuring data integrity and authenticity. Traditional hash functions like MD4, MD5, SHA-1, and SHA-2 use the Merkle–Damgård construction. MD4, introduced in 1989, was soon replaced by the more secure MD5. Later, SHA-1 was standardized by NIST in 1995, followed by SHA-2, which offers improved security and six fixed output lengths (e.g., SHA-256, SHA-512). However, as attacks against MD5 and SHA-1 became practical, the need for a stronger standard grew [3].

To develop a robust successor, NIST launched a public competition, resulting in the selection of Keccak as the winner among five finalists (BLAKE, JH, Grøstl, Skein, and Keccak). Standardized as SHA-3, Keccak uses a sponge construction, which differs from the Merkle–Damgård design by enabling better resistance to certain attacks and supporting more flexible output. SHA-3 not only mirrors SHA-2's fixed-length outputs but also introduces SHAKE128 and SHAKE256, offering variable-length digests for broader cryptographic applications.

The sponge construction, introduced by Bertoni et al., is a versatile framework for building cryptographic primitives [1]. After preprocessing, the input message is divided into equal-

sized blocks, typically denoted as p_i. In the sponge construction, two key parameters—bit rate (r) and capacity (c)—define how the underlying permutation function f operates. These parameters control how much of the internal state is exposed and how much remains hidden for security. Given an input of length N and a desired output length d, the sponge function—denoted as $Z = \text{sponge}[f, r](N, d)$—produces an output Z of d bits (Figure 1).

The function f is applied repeatedly to the internal state during two phases: absorbing and squeezing. In the absorbing phase, message blocks are mixed into the state using the bit-rate portion. In the squeezing phase, the function f is continuously applied to the state, and r-bit chunks are extracted until the required d-bit output is obtained. This iterative process allows the sponge construction to flexibly generate both fixed and variable-length outputs while maintaining strong cryptographic properties. The security of sponge functions depends on the capacity c, which limits generic attacks to approximately $2^{c/2}$ queries, and the absence of structural distinguishers in f, which could enable more efficient primary attacks.

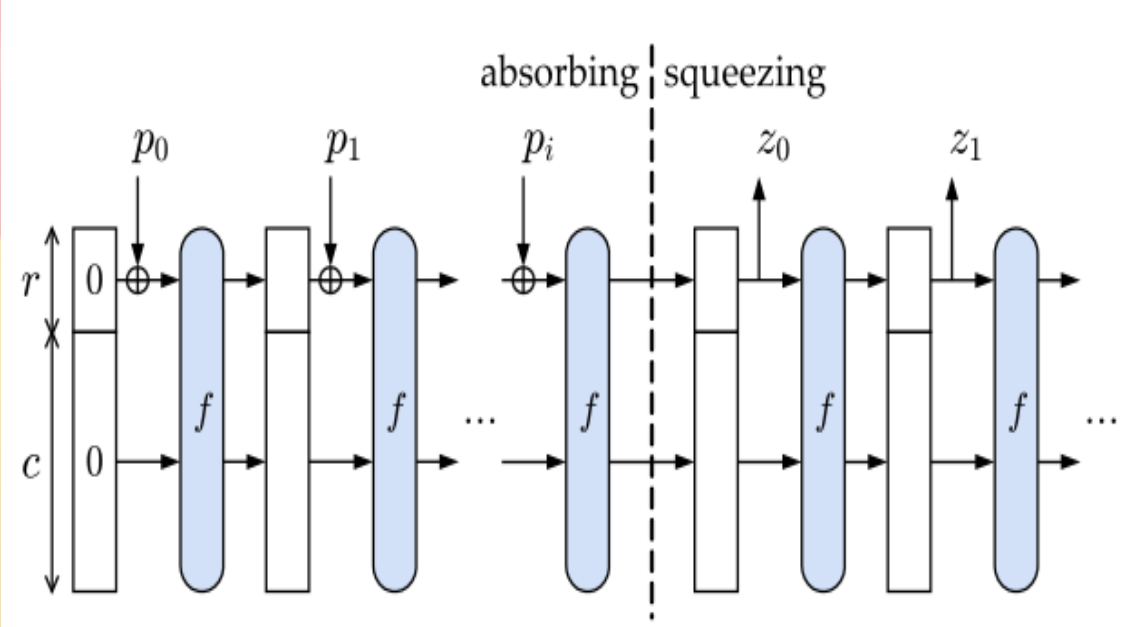

Figure 1. The general structure of Sponge construction

The sponge construction is a versatile cryptographic framework used for both keyed (e.g., MACs, stream ciphers, authenticated encryption) and keyless (e.g., hash functions) applications. Attacks on sponge constructions can exploit either the structure of the sponge itself (generic attacks) or weaknesses in the underlying permutation or transformation f (primary attacks). Both types apply to keyed and keyless modes, with different goals (e.g., collisions for hashing, key recovery for keyed modes) [4].

We provide detailed descriptions of each attack type, their goals, mechanisms, and security bounds for both keyed and keyless modes. We also discuss design strategies, particularly the hermetic sponge strategy [1], to ensure robust security.

2. Classification of Attacks

2.1 Generic Attacks

Collision attacks. A collision attack aims to find two distinct inputs $m \neq m'$ that produce the same output, such as a hash in keyless mode or a tag in keyed mode, compromising the integrity of applications like digital signatures or MACs. In both modes, the attacker queries the sponge function, which processes inputs via the absorbing phase (XORing inputs into the bitrate r-bits and applying the permutation f) and generates outputs during the squeezing phase. The attack succeeds by causing an inner collision, where two different input sequences yield the same internal state, requiring approximately $2^{c/2}$ queries due to the birthday paradox applied to the c-bit capacity. In keyless mode, collision resistance is $\min(2^{n/2}, 2^{c/2})$, where n is the output length, ensuring standard hash function security (e.g., 2^{128} for $n = 256, c \geq 512$); in keyed mode, the bound is $2^{c/2}$, as the secret key does not increase state entropy beyond c. However, structural distinguishers in f, such as high-probability differential characteristics, can enable primary attacks with lower complexity (e.g., 2^{50} queries), underscoring the need for a hermetic f to maintain the $2^{c/2}$ bound [1].

Preimage Attacks. A preimage attack aims to find an input m for a given output h, critical for keyless hashing (e.g., password recovery) but less relevant in keyed modes due to the secret key. The attacker tries inputs to match h, with the sponge absorbing inputs and squeezing n-bit outputs. The generic bound is $\min(2^n, 2^c)$, as the output length n or state entropy c limits the search; for example, $n = 256$, $c = 512$ yields 2^{256}. In keyed mode, preimage attacks (e.g., finding an input for a tag) follow the same bound, and key length k does not affect it unless key recovery (2^k) precedes, which is a separate attack. Primary attacks exploiting algebraic distinguishers in f can reduce complexity to 2^t (variable t represents the complexity, in terms of queries or computational effort, required to execute a primary attack), requiring a robust f.

Second Preimage Attack. A second preimage attack seeks a different input $m' \neq m$ producing the same output as $H(m)$, undermining integrity in keyless hashing or enabling forgery in keyed modes. The attacker searches for an inner collision with m's state, requiring $2^{c/2}$ queries due to the c-bit capacity. In keyless mode, the bound is $\min(2^n, 2^{c/2})$, achieving 2^n for $c \geq 2n$; in keyed mode, it's $2^{c/2}$, as the key (length k) does not alter the state's entropy,

and $\min(2^{c/2}, 2^k)$ is not standard since k affects key recovery, not collisions. Differential distinguishers in f can lower complexity to 2^t, necessitating a hermetic f.

Distinguishing Attacks. A distinguishing attack aims to differentiate the sponge from a random function, weakening protocols in keyless mode or detecting patterns in keyed mode outputs (e.g., keystreams). The attacker queries the sponge, seeking inner collisions or non-random patterns, detectable after $2^{c/2}$ queries due to the c-bit state's birthday bound. In both modes, the bound is $2^{c/2}$, and key length k does not affect it, as distinguishing relies on state collisions, not key guessing, making $\min(2^{c/2}, 2^k)$ irrelevant. Linear distinguishers in f can reduce complexity to $1/\epsilon^2$, where ϵ is the bias, requiring a robust f [1].

Key Recovery Attack (Keyed Only). A key recovery attack seeks to deduce the secret key or state in keyed modes (e.g., MACs, stream ciphers), enabling decryption or forgery. The attacker queries the sponge, analyzing outputs to infer the c-bit state or k-bit key, with generic bounds of $2^{c/2}$ (collision-based state recovery), 2^c (exhaustive state recovery), or 2^k (key guessing). The effective bound is $\min(2^{\frac{c}{2}}, 2^c, 2^k)$, as a short key (e.g., $k < c/2$) makes guessing easier; for example, $c = 256$, $k = 80$ yields 2^{80}. Structural distinguishers (e.g., linear correlations) can reduce complexity to 2^t, underscoring the need for a hermetic f [5].

Forgery Attack (Keyed Only). A forgery attack aims to produce a valid output (e.g., MAC tag T of length n-bits) without the key, compromising authenticity, where the attacker queries the sponge, manipulating inputs to cause inner collisions or guess the tag, with the generic bound being $\min(2^{c/2}, 2^n)$ queries since an inner collision in the c-bit capacity requires $2^{c/2}$ queries, but guessing an n-bit tag needs only 2^n attempts if $n < c/2$, while a more conservative bound considering the total state size $b = r + c$ can be $2^{(c+r)/2}$, as the bitrate r determines the rate of output extraction (e.g., squeezing the tag in $\lceil n/r \rceil$ iterations). For example, with $c = 256$ and $n = 64$, the bound drops to 2^{64} due to guessing, weakening the intended 2^{128}, and the key length k does not affect this bound, as forgery depends on state collisions or tag guessing, not key recovery (2^k), making $\min(2^{c/2}, 2^k)$ inappropriate unless key recovery precedes, which is a separate attack. Differential distinguishers in f can further lower complexity to 2^t, necessitating a robust f, and to ensure security, the tag length n should be at least $c/2$ (e.g., $n \geq 128$ for $c = 256$) [5].

Length Extension Attack. A length extension attack targets keyless sponge modes (e.g., hashing) by exploiting the incremental absorbing phase, where an attacker, given a hash $H(m)$ of a message m, appends data to reconstruct the state and compute $H(m \,||\, m')$ without knowing m, potentially undermining integrity in applications like HMAC. The attacker uses the bitrate r-bits to extend the input, applying f to update the state, with the generic bound being $2^{c/2}$ queries to find an inner collision that aligns the extended state, as the capacity c limits state entropy; in keyed modes, this attack is mitigated if the key is properly absorbed

with domain separation, but if the key is prepended like a message, the bound remains $2^{c/2}$ unless key recovery (2^k) occurs. The attack's feasibility depends on knowing the message length and r, with complexity reduced to 2^t if f has differential distinguishers, requiring a hermetic f and proper key handling (e.g., using a nonce or salt) to prevent extension.

Meet-in-the-Middle Attack. A meet-in-the-middle attack targets sponge constructions by splitting the computation into forward and backward phases to recover the state or key, enabling preimage attacks in keyless modes (e.g., finding m for H(m)) or key recovery in keyed modes (e.g., MACs), potentially undermining confidentiality or authenticity. The attacker computes forward from the initial state (e.g., with the key) and backward from the output (e.g., a tag or hash) over half the iterations, matching intermediate states to find a collision, requiring $2^{c/2}$ queries due to the c-bit capacity's birthday bound, or $2^{k/2}$ for key recovery if $k < c$; in keyless mode, the preimage bound is $\min(2^n, 2^{c/2})$, where n is the output length, while in keyed mode, the effective bound is $\min(2^{c/2}, 2^{k/2})$ though the bitrate r affects iteration counts but not the core complexity. For example, with $c = 256$, $k = 128$, the attack requires 2^{64} queries for key recovery, violating 2^{128}, and structural distinguishers in f (e.g., invertible rounds) can reduce complexity to 2^t, necessitating a hermetic f with non-invertible properties to maintain the generic bound [2].

2.1 Primary Attacks

Differential-Based Attack. A differential-based attack exploits high-probability differential characteristics in f, where input differences lead to predictable output differences, enabling collisions in keyless mode or forgery/ state recovery in keyed mode. The attacker uses differential paths to manipulate the state during absorbing, achieving inner collisions or state predictions with complexity 2^k, where 2^{-k} is the differential probability; for example, $k = 50$ yields 2^{50} queries, violating $2^{c/2}$. The key length k (key size) does not affect this bound, as the attack targets f's weaknesses, not the key, emphasizing the need for a hermetic f.

Linear-Based Attack. A linear-based attack exploits linear correlations in f, where input and output bits satisfy a linear equation with bias ϵ, enabling distinguishing n keyless mode or state/ key recovery in keyed mode. The attacker queries the sponge, checking output correlations, with complexity $1/\epsilon^2$; for example, $\epsilon = 2^{-20}$ yields 2^{40} queries. The key length k does not influence this bound, as the attack exploits f's structure, not the key, requiring a nonlinear f to ensure security matches $2^{c/2}$.

Algebraic Attack. An algebraic attack exploits low-degree polynomial equations in f, enabling preimage/ collision attacks in keyless mode or state/key recovery in keyed mode. The attacker models f with polynomials and solves them, with complexity $O(v^d)$, where d is the degree and v is variables; for example, degree-2, 50 variables yield 2^{50} operations. The key

length k does not affect this bound, as the attack targets f's algebraic structure, necessitating a high-degree f.

CICO-Based Attack. A CICO-based attack exploits easy solutions to constrained-input constrained-output problems in f, enabling collisions/ preimages in keyless mode or state recovery/ forgery in keyed mode. The attacker finds inputs satisfying input/ output constraints, with complexity 2^t, where $t < k + m$ (constrained bits); for example, 50-bit constraints solvable in 2^{30} queries. The key length k (key size) does not influence this bound, as the attack exploits f's weaknesses, requiring a robust f.

Cycle-Based Attack. A cycle-based attack exploits short cycles or fixed points in f's cycle structure, enabling collisions/ distinguishing in keyless mode or state recovery/ forgery in keyed mode. The attacker forces the state into predictable cycles, with complexity $O(L)$, where, L is the cycle length; for example, 2^{50} fixed points yield 2^{50} queries. The key length k does not affect this bound, as the attack targets f's structure, necessitating a random-like f.

3. Security Bounds and Design Implications

Generic attacks are bounded by c: collisions, second preimage, distinguishing, forgery and length extension at $2^{c/2}$; preimage at $\min(2^n, 2^c)$; and key recovery at $\min(2^{c/2}, 2^c, 2^k)$ and meet-in-the-middle at $\min(2^{c/2}, 2^{k/2})$ for keyed modes or $\min(2^n, 2^{c/2})$ for preimages in keyless modes. Primary attacks depend on distinguisher strength (e.g., 2^k, $1/\epsilon^2$), violating $2^{c/2}$ if weaker. The key length k affects key recovery but not collisions, forgery, or distinguishing, where $\min(2^{c/2}, 2^k)$ is inappropriate unless key recovery precedes. The hermetic sponge strategy [1] designs f with high diffusion, nonlinearity, and no distinguishers, ensuring security matches generic bounds. For hash functions, $c \geq 2n$; for keyed modes, $k \geq c/2$.

Recommendations for Enhancing Security:

Capacity Selection: To achieve n-bit security against collision attacks, set the capacity c to at least 2n bits.

Permutation Function Design: Use permutation functions with strong diffusion and non-linearity to resist MitM and other cryptanalytic attacks.

Key Management: In keyed sponge constructions, ensure that keys are integrated securely into the state and that key management practices prevent leakage.

Padding Schemes: Employ padding schemes that prevent ambiguity and ensure that different messages produce distinct padded inputs.

Domain Separation: Use domain separation techniques to prevent cross-protocol attacks and to distinguish between different uses of the sponge function.

Conclusion

Sponge constructions are secure if c is large and f lacks distinguishers. Generic attacks are limited by c, while primary attacks exploit f's weaknesses, requiring a hermetic f. The key length k influences key recovery bounds but not collisions or forgery, where c dominates. Future work includes analyzing new distinguishers and optimizing f.

References

1. Guido B. et al. Cryptographic sponge functions //2011-STMicroelectronics NXP Semiconductors, Version 0.1 January 14. – 2011.
2. Al-Odat Z., Khan S. Constructions and attacks on hash functions //2019 International Conference on Computational Science and Computational Intelligence (CSCI). – IEEE, 2019. – С. 139-144.
3. Berman I. et al. Multi-collision resistant hash functions and their applications //Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part II 37. – Springer International Publishing, 2018. – С. 133-161.
4. Dong X. et al. Generic MITM attack frameworks on sponge constructions //Annual International Cryptology Conference. – Cham: Springer Nature Switzerland, 2024. – С. 3-37.
5. Bertoni G. et al. On the security of the keyed sponge construction //Symmetric Key Encryption Workshop. – 2011. – Т. 2011.