Bachelor's Thesis Computing Science

Radboud University Nijmegen

Multi-Preimage Resistance of The Sponge Construction

Author:
Aaditya Kondhia
s1096819

First supervisor/assessor:
Bart Mennink

Second supervisor
Charlotte Lefevre

Second assessor:
Joan Daemen

June 12, 2026

Abstract

The cryptographic sponge is a widely used framework for hash function design. In the ideal permutation model, the sponge construction has been proven indifferentiable from a random oracle up to the birthday bound in the capacity. Any attack whose complexity remains below this bound is expected to achieve security comparable to that of a random oracle with respect to standard properties such as collision, preimage, and second-preimage resistance.

In this thesis, we investigate the multi-preimage resistance of the cryptographic sponge. In particular, we study two scenarios: one in which inner collisions are used and one in which they are not. We show that involving inner collisions does not yield a more efficient multi-preimage attack. When inner collisions are excluded, we derive tight bounds on the adversary's advantage.

More specifically, we show that in the absence of inner collisions, the probability of finding x preimages decreases exponentially with x. In contrast, when inner collisions are used, the security is bounded by the complexity of constructing an inner collision state. As a result, increasing x strengthens security only while direct multi-preimage search remains the dominant attack strategy. Once inner collision-based attacks become cheaper, the security is effectively capped by the inner collision-based attack's complexity.

Contents

Chapter 1

Introduction

The sponge construction of Bertoni et al. [1] is a popular choice for modern cryptographic hash functions based on the random permutation $\mathcal{P}$ model. It works on a b-bit state which consists of an inner c-bit part, known as the capacity, and an outer r-bit part, known as the rate, where $b = r + c$. The sponge construction has two phases, namely the absorbing phase and the squeezing phase. In the absorbing phase, input is padded into r-bit blocks which is fed iteratively into the sponge. In each iteration, an input block is XORed into the outer r-bits of the state followed by an update to the b-bit. Once all input blocks have been absorbed, the squeezing phase commences. This phase does not deal with any input, rather in each iteration the outer r-bits are extracted and the state passes through $\mathcal{P}$ to squeeze out the next r-bits. The output is taken by concatenating all extracted r-bits in order. This is called the digest of the sponge.

The developers of PHOTON [2] proposed a generalization of this construction where a higher rate $r' \geq r$ is used during the squeezing phase. In that case, we have $b = r + c = r' + c'$. This is depicted in Figure 2.1.

The sponge found quick adoption due to its simplicity and its ability to offer variable length digests. Another main cause was that the developers proved the security of sponge construction in the indifferentiability framework [3]. This implies that no adversary with an attack complexity less than $2^{c/2}$ can differentiate the sponge from a random oracle, which allows the sponge to be used in applications that were proven secure in the random oracle model. It also implies that finding collisions, preimages, and second preimages for the sponge is not easier than for a random oracle.

Andreeva et al. [4] proved this and demonstrated that for a sponge that outputs a digest of n-bits, finding collisions requires at least

$$q \approx \min\{2^{c/2}, 2^{n/2}\} \tag{1.1}$$

work, and finding preimages or second preimages requires at least

$$q \approx \min\{2^{c/2}, 2^n\} \tag{1.2}$$

work. These bounds have directly influenced the parameter choices of many sponge-based hash functions. Most notably, the SHA-3 family of hash functions that sets the capacity c to twice the digest length n, for a fixed set of values of n.

The collision security bound of (1.1) is tight as proven by the indifferentiability bound of Bertoni et al. [6], where a collision for a sponge with fixed n-bit output can be obtained by finding a c-bit inner collision or an n-bit output collision.

For preimage security, for certain values of c and n, the bound of (1.2) is not tight. This is mainly caused by the fact that, unlike for second preimage security, the final state before squeezing cannot always be easily found. In the original introduction of the sponge construction in 2007[1], it was claimed that a preimage attack can only be found in $\max\{2^{n-r'}, 2^{c/2}\}$ work. This bound is tight, but has only recently been proven [5].

For x-preimage security of the sponge construction, tight bounds have not yet been proven. In this thesis, we try to prove tight bounds for x-preimage resistance against attacks that do and do not use inner collisions, for any adversary that can make at most q queries. This paper is motivated by and closely follows the tight preimage resistance of the sponge paper [5].

The following chapters explain the necessary knowledge to follow the thesis, followed by attacks on relevant cryptographic primitives along with their bounds. Then, a detailed security proof is conducted whose conclusions are finally summarized.

Chapter 2

Preliminaries

2.1 Notation

We write $M \xleftarrow{\$} S$ for the experiment of choosing a random element from the distribution S and calling it M. When S is a finite set, it is given the uniform distribution. The concatenation of the strings M and $M^{'}$ is denoted by $M||M^{'}$.

For a b-bit string s, where $b \in \mathbb{N}$ and $0 \leq x \leq y \leq b-1$, $s[x:y]$ denotes the substring of s from the bits of position x to y. We denote $\text{inner}_x(s) = s[b-x:b-1]$ and $\text{outer}_x(s) = s[0:x-1]$. $Perm(b)$ denotes the set of functions from $\{0,1\}^b \to \{0,1\}^b$. For a permutation $\mathcal{P} \xleftarrow{\$} Perm(b)$ and for $i \in \mathbb{N}$, $\mathcal{P}^i$ means the i^{th} iteration of $\mathcal{P}$.

A hash function family is a function $H : K \times M \to Y$ where H is a hash function, K is a set of fixed keys, M is a set of chosen plaintexts to be encrypted, and Y is a set whose elements are called digests that are produced by H. K and Y are finite nonempty sets, M and Y are sets of strings. We insist that digest $Y = \{0,1\}^n$ for some $n \in \mathbb{N}$ and $n > 0$. n is called the hash length of H. Let $m \in \mathbb{N}$ be the length of an input message, then a hash function is denoted as $H : K \times \{0,1\}^m \to \{0,1\}^n$.

2.2 Sponge Construction

Let $b, c, r, c', r', n \in \mathbb{N}$ be with $b = r + c = r' + c'$. Let $\mathcal{P} \in Perm(b)$ be a cryptographic permutation. Let pad be an injective padding function that maps a message $M \in \{0,1\}^*$ into k blocks of r-bits such that the last block is non-zero. The padding appends a single 1 followed by a uniquely determined number $j \geq 0$ of 0 bits such that $\text{pad}(M) = M||1||0^j$ and $\text{pad}(M) \bmod r = 0$. The sponge construction can generate an arbitrary length output denoted by n. We define a variable $\ell = \lceil n/r' \rceil$ which is the

number of permutations needed to be able to squeeze out a digest of length n.

Let $M \in \{0,1\}^*$ be an input message. The sponge construction instantiated with the permutation $\mathcal{P}$, denoted by $H^{\mathcal{P}} : \{0,1\}^* \to \{0,1\}^n$, is now defined as follows.

- M is padded in k blocks $M_1||\ldots||M_k$ using $\text{pad}(M)$.
- The state is initialized as 0^b.
- Absorbing phase: for each iteration $i = 1, \ldots, k$, the state is updated as $S \leftarrow P(S \oplus (M_i||0^c))$.
- Squeezing phase: for each iteration $i = 1, \ldots, \ell$, the outer $r'-$bits of S are extracted as $Z_i \leftarrow outer_{r'}(S)$ and the state is updated as $S \leftarrow P(S)$.
- The digest is taken as $Z \leftarrow (Z_1||\ldots||Z_\ell)[0 : n-1]$.

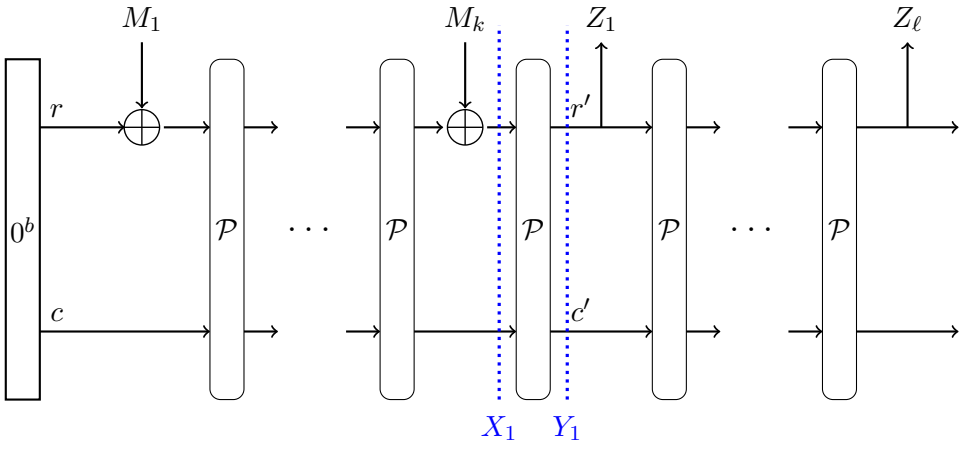

Figure 2.1: Sponge construction

2.3 Security Model

Consider an adversary $\mathcal{A}$, which is a probabilistic algorithm that has access to a permutation $\mathcal{P}$ sampled uniformly at random. $\mathcal{A}$ can make queries to $\mathcal{P}$ denoted by $\mathcal{P} \xleftarrow{\$} Perm(b)$. $\mathcal{A}$ is restricted only be the number of queries denoted by q to $\mathcal{P}$ and $\mathcal{P}^{-1}$. The collection of queries that $\mathcal{A}$ has made is called $\mathcal{Q}$ and is an ordered list of tuples of the form $(X, Y, d) \in \{0,1\}^b \times \{0,1\}^b \times \{\text{fwd}, \text{inv}\}$, where $\mathcal{P}(X) = Y$ and d denotes the query

direction. For $i \in \mathbb{N}$, we use $\mathcal{Q}_i$ to denote the set of first i queries the adversary has made. We can assume without loss of generality that $\mathcal{A}$ never repeats queries.

Everywhere Preimage Resistance Consider any image $Z \in \{0,1\}^n$ of length n and consider an adversary $\mathcal{A}$ that can query $\mathcal{P}$ and wants to output a message M such that $H^{\mathcal{P}}(M) = Z$.

Definition 1. Let $b, n, q \in \mathbb{N}$, consider the sponge construction $H^{\mathcal{P}}$ described earlier. For any adversary $\mathcal{A}$, we define its everywhere preimage resistance as:

$$\mathrm{Adv}^{ePre}_{H^{\mathcal{P}}}(\mathcal{A}) = \max_{Z \in \{0,1\}^n} \Pr\left[\mathcal{P} \xleftarrow{\$} Perm(b);\, M \leftarrow \mathcal{A}^{\mathcal{P}}(Z) : H^{\mathcal{P}}(M) = Z\right]$$

We define $\mathrm{Adv}^{ePre}_{H^{\mathcal{P}}}(q)$ as the maximum advantage over all adversaries making at most q queries.

Everywhere Collision Resistance Consider the adversary $\mathcal{A}$ that can query $\mathcal{P}$ and wants to output messages M and M' such that $H^{\mathcal{P}}(M) = H^{\mathcal{P}}(M')$.

Definition 2. Let $b, q \in \mathbb{N}$, consider the sponge construction $H^{\mathcal{P}}$ described earlier. For any adversary $\mathcal{A}$, we define its everywhere collision resistance as:

$$\mathrm{Adv}^{eCol}_{H^{\mathcal{P}}}(\mathcal{A}) = \max \Pr\left[\mathcal{P} \xleftarrow{\$} Perm(b);\, (M, M') \leftarrow \mathcal{A}^{\mathcal{P}} : H^{\mathcal{P}}(M) = H^{\mathcal{P}}(M')\right]$$

We define $\mathrm{Adv}^{eCol}_{H^{\mathcal{P}}}(q)$ as the maximum advantage over all adversaries making at most q queries.

Everywhere Multi-Collision Resistance Consider the adversary $\mathcal{A}$ that can query $\mathcal{P}$ and wants to output $x > 2$ messages such that $H^{\mathcal{P}}(M_1) = \cdots = H^{\mathcal{P}}(M_x)$.

Definition 3. Let $b, q \in \mathbb{N}$, consider the sponge construction $H^{\mathcal{P}}$ described earlier. For any adversary $\mathcal{A}$, we define its everywhere multi-collision resistance as:

$$\mathrm{Adv}^{x\text{-}eCol}_{H^{\mathcal{P}}}(\mathcal{A}) = \Pr\left[\begin{array}{l}\mathcal{P} \xleftarrow{\$} Perm(b);\, (M_1, \ldots, M_x) \leftarrow \mathcal{A}^{\mathcal{P}} : \\ (M_1 \neq \ldots \neq M_k) \wedge (H^{\mathcal{P}}(M_1) = \cdots = H^{\mathcal{P}}(M_x))\end{array}\right]$$

We define $\mathrm{Adv}^{x\text{-}eCol}_{H^{\mathcal{P}}}(q)$ as the maximum advantage over all adversaries making at most q queries.

Everywhere Multi-Preimage Resistance Consider any image $Z \in \{0,1\}^n$ of length n and consider the adversary $\mathcal{A}$ that can query $\mathcal{P}$ and wants to output $x > 1$ messages such that $H^{\mathcal{P}}(M_1) = \cdots = H^{\mathcal{P}}(M_x) = Z$.

Definition 4. Let $b, n, q \in \mathbb{N}$, consider the sponge construction $H^{\mathcal{P}}$ described earlier. For any adversary $\mathcal{A}$, we define its everywhere multi-preimage resistance as:

$$\mathrm{Adv}^{x\text{-}ePre}_{H^{\mathcal{P}}}(\mathcal{A}) = \max_{Z \in \{0,1\}^n} \Pr\left[\begin{array}{l} \mathcal{P} \xleftarrow{\$} Perm(b);\ (M_1, \ldots, M_x) \leftarrow \mathcal{A}^{\mathcal{P}}(Z) : \\ (M_1 \neq \ldots \neq M_x) \wedge (H^{\mathcal{P}}(M_1) = \cdots = H^{\mathcal{P}}(M_x) = Z) \end{array}\right]$$

We define $\mathrm{Adv}^{x\text{-}ePre}_{H^{\mathcal{P}}}(q)$ as the maximum advantage over all adversaries making at most q queries.

Chapter 3

Attack

3.1 Random Oracle

Before analyzing the security of the sponge construction, it is useful to recall the security properties of an ideal hash function. The random oracle model provides such a reference point. A random oracle behaves as a perfectly random function that returns an independent uniformly distributed digest for every new query while remaining consistent on repeated queries.

The security of hash constructions is often evaluated by comparing their resistance to generic attacks with the corresponding resistance of a random oracle. Any attack in a hash construction that significantly outperforms the best known generic attack in the random oracle model may indicate a structural weakness in the construction. Consequently, the random oracle serves as a useful baseline for understanding the security level that can realistically be expected from a hash function.

Let $\mathcal{R} : \{0,1\}^* \to \{0,1\}^n$ be a random oracle where $n \in \mathbb{N}$ is the fixed length of the digest. Let $\mathcal{A}$ be any adversary using the best known attack making at most q queries to $\mathcal{R}$. The following summarizes the best known generic attacks and their corresponding success probabilities.

- for $\mathcal{A}$ trying to find a preimage, they carry out the brute force attack. Let $y \in \{0,1\}^n$ be a known hash of $\mathcal{R}$. To find a match, $\mathcal{A}$ needs to try 2^n inputs to find a preimage with high probability. This advantage over all adversaries against $\mathcal{R}$ is given by
  $\mathrm{Adv}_{\mathcal{R}}^{ePre}(\mathcal{A}) \leq \frac{q}{2^n}$,
  where the security is guaranteed up to $\approx 2^n$ queries.

- for $\mathcal{A}$ trying to find $x > 1$ preimages, they carry out the brute force attack. Let $y \in \{0,1\}^n$ be a known hash of $\mathcal{R}$. $\mathcal{A}$ keeps attempting distinct queries to $\mathcal{R}$ until they get x preimages. After $q \leq 2^n$ queries,

the advantage over all adversaries against $\mathcal{R}$ is given by
$\mathrm{Adv}_{\mathcal{R}}^{x-ePre}(\mathcal{A}) \leq \frac{1}{x!}\left(\frac{q}{2^n}\right)^x \leq \left(\frac{eq}{x2^n}\right)^x$,
where Stirling's approximation $x! \leq (\frac{x}{e})^x$ is applied, and the security is guaranteed up to $\approx \frac{x2^n}{e}$ queries.

- for $\mathcal{A}$ trying to find a collision, they carry out the birthday attack. Finding a collision is bound by the birthday bound over the parameter n. This advantage over all adversaries against $\mathcal{R}$ is given by
$\mathrm{Adv}_{\mathcal{R}}^{eCol}(\mathcal{A}) \leq \frac{q^2}{2^n}$,
where the security is guaranteed up to $\approx 2^{n/2}$ queries.

- for $\mathcal{A}$ trying to find $x > 2$ collisions, they carry out the generalized birthday attack. Finding x collisions is bound by the generalized birthday bound over the parameters n, x. This advantage over all adversaries against $\mathcal{R}$ is given by
$\mathrm{Adv}_{\mathcal{R}}^{x-eCol}(\mathcal{A}) \leq \frac{q^x}{2^{(x-1)n}}$,
where the security is guaranteed up to $\approx 2^{(x-1)n/x}$ queries.

These bounds provide the benchmark against which the security of sponge construction is compared. The goal of the subsequent analysis is to determine whether the sponge construction achieves security levels comparable to those of a random oracle for multi-preimage resistance.

3.1.1 Security lower bound

Bertoni et al. [6] proved that the sponge is indifferentiable from a random oracle up to the bound $\mathbf{Adv}_{\mathcal{H}^{\mathcal{P}}}^{\mathrm{indiff}}(q) \leq \frac{q(q+1)}{2^{c+1}}$. This means that the sponge behaves like a random oracle up to this bound. In the case of x-preimage resistance, we obtain

$$\mathbf{Adv}_{\mathcal{H}^{\mathcal{P}}}^{x-\mathrm{ePre}}(q) \leq \mathbf{Adv}_{\mathcal{H}^{\mathcal{P}}}^{\mathrm{indiff}}(q) + \mathbf{Adv}_{\mathcal{R}}^{x-\mathrm{ePre}}(q)$$

Plugging in the known bounds, we obtain the following state-of-the-art bound for x-preimage resistance of the sponge:

$$\mathbf{Adv}_{\mathcal{H}^{\mathcal{P}}}^{x-\mathrm{ePre}}(q) \leq \frac{q(q+1)}{2^{c+1}} + \left(\frac{eq}{x2^n}\right)^x \tag{3.1}$$

3.2 Multi-Preimage Attack on the Sponge

In the following attacks, we first assume that x preimages of Z exist. The attack is chosen based on a case distinction depending on the values of n, c.

- Case $n \leq c/2$. The adversary picks a distinct message M and queries $H^{\mathcal{P}}(M)$. Then $H^{\mathcal{P}}(M) = Z$ with a probability of $1/2^n$ as there are

2^n possible results for this query. They keep picking distinct messages individually for each preimage, until x distinct preimages M_i for $i = 1, \ldots, x$ are found. After $q \approx \frac{x2^n}{e}$ queries, the adversary has found x preimages with high probability.

- Case $n > c/2$. The attack consists of two sequential parts and involves finding inner collisions. In the attack, an adversary first needs to generate x messages that collide to the same state S_t. Once such a state has been found, they carry out a 1-preimage attack from the state S_t for the digest Z.

  To find x colliding messages, the adversary starts by picking a message block M_1 and queries $\mathcal{P}(0^b \oplus (M_1||0^c))$ and keeps track of the b-bit state S_1. Then, they keep picking subsequent message blocks M_i and tracking b-bit states S_i for $i = 2, \ldots$ until eventually, one message block M_u whose permutation results in a b-bit state $S_u = \mathcal{P}(M_u \oplus \ldots (\mathcal{P}(M_1||0^c))$, where $inner_c(S_u)$ has appeared before. This happens after $\approx 2^{c/2}$ queried message blocks, where an inner c-bit collision is found with high probability.

  Let the first occurrence of the appeared state S_u whose inner c-bits match be S_t where $t < u$. $inner_c(S_u) = inner_c(S_t)$ but $outer_r(S_u) \neq outer_r(S_t)$. The adversary can get a full state collision for free from S_u by generating a message block $M_{free} = (outer_r(S_t) \oplus outer_r(S_u))$ such that they get $S_t = S_u \oplus (M_{free}||0^c)$. Thus, they get a collision to the state S_t that has appeared before.

  In order to generate multi-collisions, first, the message blocks between the first occurrence of S_t and S_u are taken and absorbed after which M_{free} is XORed to get S_t. Repeating this message sequence $x - 1$ more times generates x colliding messages. This attack is known as the cyclic attack which is seen in the repeating message sequence to generate collisions.

  Let prefix $P = M_1|| \ldots ||M_t$ be the chain of message blocks that result in the first occurrence of S_t, collision block $C = M_{t+1}|| \ldots ||M_u||M_{free}$ be the chain of message blocks between the first and next occurrence of S_t. Then, x colliding messages to S_t are created as follows:

  $$M_i = P||C^{i-1} \quad \text{for } i = 1, \ldots, x$$

  Note that after M_{free}, the cycle is closed at the XOR stage without applying the permutation, i.e., $S_u \oplus (M_{free}||0^c) = S_t$. When re-entering

the cycle, M_{t+1} is first absorbed as follows:
$\mathcal{P}(S_u \oplus (M_{free}||0^c) \oplus (M_{t+1}||0^c)) = \mathcal{P}(S_t \oplus (M_{t+1}||0^c)) = S_{t+1}$.

From the state S_t, the adversary needs to find a message
$M_{pre} = M_{p1}|| \ldots ||M_{pk}$ such that $\mathcal{P}(M_k \oplus \ldots (\mathcal{P}(S_t \oplus (M_1||0^c)))) = Z$.

First the adversary finds a state Y_1 before the first squeeze, which satisfies the following:

$$Z_i = \begin{cases} outer_{r'}(Y_i) & \text{for } i = 2, ..., \ell - 1 \\ outer_{n-(\ell-1)r'}(Y_i) & \text{for } i = \ell \end{cases}$$

The adversary knows that Z_1 is equal to the outer r'-bits of Y_1 giving them $outer_{r'}(Y_1) = Z_1$. Then, the adversary guesses $inner_c(Y_1)$ such that the previous equations hold. The probability of finding such a state Y_1 depends on the probability of the above equation holding true and is as follows:

$$\left(\frac{1}{2^{r'}}\right)^{\ell-2} \cdot \frac{1}{2^{n-(\ell-1)r'}} = \frac{1}{2^{n-r'}}$$

Based on the above probability, the adversary has a valid state Y_1 after $\approx 2^{n-r'}$ queries with high probability.

Starting from S_t, the adversary computes $Y_0^{\rightarrow} = \mathcal{P}(S_t \oplus (M_{p1}||0^c))$ for q different values of M_{p1}. Starting from the state Y_1 found in the first part of the attack, the adversary computes $Y_0^{\leftarrow} = \mathcal{P}^{-1}(\mathcal{P}^{-1}(Y_1) \oplus (M_{pk}||0^c))$ for q different values of M_{pk}. If $q \approx 2^{c/2}$, there is a high probability that there will be two values of M_{p1} and M_{pk} such that

$$inner_c(Y_0^{\rightarrow} \oplus Y_0^{\leftarrow}) = 0^c$$

Let $M_{chain} := outer_r(Y_0^{\rightarrow} \oplus Y_0^{\leftarrow})$. Define the preimage as the unique message M_{pre} such that $\text{pad}(M_{pre}) = M_{p1}||M_{chain}||M_{pk}$. We remark that if $r \leq \frac{c}{2}$, one will need multiple message blocks for both forward and the inverse parts to perform $q \approx 2^{c/2}$ evaluations, but the attack works in a comparable way.

Finally, the messages are generated by appending the resulting M_{pre} from the preimage attack to each of the colliding messages to obtain

x preimages of Z. The equation is as follows:

$$M_i = P||C^{i-1}||M_{pre} \quad \text{for } i = 1, \ldots, x$$

In total, in this case the attack requires $q \approx 2^{c/2} + (2^{n-r'} + 2^{c/2}) \approx 2^{\frac{c}{2}+1} + 2^{n-r'}$ evaluations.

Thus, total cost of the entire attack is $\min\{\frac{x2^n}{e}, (2^{\frac{c}{2}+1} + 2^{n-r'})\}$ which is also called the security upper bound.

Chapter 4

Proof

4.1 Setup

Let $Z \in \{0,1\}^n$ be any digest, and $Z = Z_1||Z_2||\dots||Z_\ell$ with $|Z_i| = r'$ for $i \in \{1,\dots,\ell-1\}$ and $|Z_\ell| = s \leq r'$. Let $\mathcal{A}$ be any adversary as defined in Section 2.3. We use the graph representation for its query history $\mathcal{Q}$ as introduced in Section 2.3.
Let Z_i be as follows:

$$Z_i := \begin{cases} \{Y_i \in \{0,1\}^b \,|\, outer_{r'}(Y_i) = Z_i\} & \text{for } i \in \{1,...,\ell-1\} \\ \{Y_i \in \{0,1\}^b \,|\, outer_s(Y_i) = Z_i\} & \text{for } i = \ell \end{cases} \tag{4.1}$$

The goal of $\mathcal{A}$ is to find x preimages M_i for $i = 1,\dots,x$ which implies the event x-PRE($\mathcal{Q}$):

$$x\text{-PRE}(\mathcal{Q}) : \mathcal{Q} \text{ defines } x \text{ distinct paths } 0^b \xrightarrow{M_{i\_1}} \dots \xrightarrow{M_{i\_(k-1)}} Y_{i\_0} \xrightarrow{M_{i\_k}} Y_{i\_1} \to \dots \to Y_{i\_\ell}$$
$$\text{such that } Y_j \in Z_j \text{ for } i = 1,\dots,x \text{ and } j = 1,\dots,\ell$$

In the case of minimal injective padding presented in Section 2.2, finding a preimage corresponds to x-PRE($\mathcal{Q}$) with the restriction that the last message block is not zero. In such case, x preimages found by $\mathcal{A}$ is the unique message M_i such that $\text{pad}(M_i) = M_{i\_1}||\dots||M_{i\_k}$ for $i = 1,\dots,x$.

4.2 Logic

First, we introduce the following event:

$$\text{INNER}(\mathcal{Q}) : \exists (X, Y, \text{fwd}), (X', Y', \text{inv}) \in \mathcal{Q} \cup \{(, 0^b, \text{fwd})\}$$
$$\text{such that } inner_c(Y) = inner_c(X')$$

Then, we separate the event x-PRE($\mathcal{Q}$) as the disjoint union of the following

two events:

$$x\text{-PRE}(\mathcal{Q}) \iff (x\text{-PRE}(\mathcal{Q}) \wedge \neg\text{INNER}(\mathcal{Q})) \vee (x\text{-PRE}(\mathcal{Q}) \wedge \text{INNER}(\mathcal{Q}))$$

We will consider dedicated trigger points for both events. Let $S = \{Y_1 \,|\, \mathcal{P}^{i-1}(Y_1) \in Z_i \text{ for all } i \in \{1, \ldots, \ell\}\} \subseteq Z_1$. We define the set S_{fwd} and the multiset S_{inv} as follows:

$$S_{\text{fwd}} = \{\mathcal{P}^{-1}(Y_1) \,|\, Y_1 \in S\}$$
$$S_{\text{inv}} = \{Y_1, \mathcal{P}(Y_1), \ldots, \mathcal{P}^{\ell-1}(Y_1) \,|\, Y_1 \in S\}$$

Intuitively, S_{inv} includes all the states $Y_1, \ldots, Y_\ell$ appearing in paths which set x-PRE($\mathcal{Q}$) if discovered, while S_{fwd} captures all values X_1 from which such path $Y_1 \to \cdots \to Y_\ell$ starts. These trigger points can be repeated in S_{inv} when some values Z_i are colliding. We now introduce the following events:

- x-BADFWD($\mathcal{Q}$) : $\exists (X_i, Y_i, \text{fwd}) \in \mathcal{Q}$ for $i = 1, \ldots, x$ such that $X_i \in S_{\text{fwd}}$ and $\forall i \neq j : X_i \neq X_j$
- BADUNION($\mathcal{Q}$) : $\exists (X, Y, \text{d}) \in \mathcal{Q}$ such that $(\text{d} = \text{inv} \wedge (Y \in S_{\text{inv}}) \vee (\text{d} = \text{fwd} \wedge (X \in S_{\text{fwd}} \vee X \in S_{\text{inv}}))$

For $(x\text{-PRE}(\mathcal{Q}) \wedge \neg\text{INNER}(\mathcal{Q}))$ to be set, since no inner collisions occur, each successful preimage must be obtained via a distinct forward query $\mathcal{P}(X_i)$ with $X_i \in S_{\text{fwd}}$. Hence, the adversary must obtain x distinct forward hits as such. Thus, $(x\text{-PRE}(\mathcal{Q}) \wedge \neg\text{INNER}(\mathcal{Q}))$ implies x-BADFWD($\mathcal{Q}$).

Likewise, for $(x\text{-PRE}(\mathcal{Q}) \wedge \text{INNER}(\mathcal{Q}))$ to be set, the adversary must ever make a query that either appears in a path $Y_1 \to \cdots \to Y_\ell$ in a query direction or guesses a state X_1 which sets Y_1 from which the ealier path follows. In other words, it must ever query a value in S_{fwd} or S_{inv}. Hence, $(x\text{-PRE}(\mathcal{Q}) \wedge \text{INNER}(\mathcal{Q}))$ implies $(\text{BADUNION}(\mathcal{Q}) \wedge \text{INNER}(\mathcal{Q}))$. More formally:

$$x\text{-PRE}(\mathcal{Q}) \wedge \neg\text{INNER}(\mathcal{Q}) \implies x\text{-BADFWD}(\mathcal{Q}) \tag{4.2}$$
$$x\text{-PRE}(\mathcal{Q}) \wedge \text{INNER}(\mathcal{Q}) \implies \text{BADUNION}(\mathcal{Q}) \wedge \text{INNER}(\mathcal{Q}) \tag{4.3}$$

From (4.2) and (4.3), we logically obtain,

$$x\text{-PRE}(\mathcal{Q}) \implies x\text{-BADFWD}(\mathcal{Q}) \vee (\text{BADUNION}(\mathcal{Q}) \wedge \text{INNER}(\mathcal{Q}))$$

and thus

$$\begin{aligned}\mathbf{Pr}(x\text{-PRE}(\mathcal{Q})) \leq\, & \mathbf{Pr}(x\text{-BADFWD}(\mathcal{Q})) \\ & + \min\{\mathbf{Pr}(\text{BADUNION}(\mathcal{Q})) \,,\, \mathbf{Pr}(\text{INNER}(\mathcal{Q}))\}\end{aligned} \tag{4.4}$$

4.3 Probability Computation

We upper bound the three probabilities of (4.4), starting with $\mathbf{Pr}(\text{INNER}(\mathcal{Q}))$ in Lemma 1, then $\mathbf{Pr}(\text{BADUNION}(\mathcal{Q}))$ in Lemma 2, and finally $\mathbf{Pr}(x\text{-BADFWD}(\mathcal{Q}))$ in Lemma 3.

Lemma 1. We have

$$\mathbf{Pr}(\text{INNER}(\mathcal{Q})) \leq \frac{q(q+1)}{2} \tag{4.5}$$

Proof of Lemma 1. The following proof is taken from the tight preimage resistance of the sponge paper [5]. We index the queries by the query number, i.e., the i^{th} query is denoted by (X^i, Y^i, d^i). $\text{INNER}(\mathcal{Q})$ translates to the fact that either there is an inner collision between the set of forward or inverse queries, or that the output of an inverse query inner collides with 0^b. More formally, this implies that there exists $i \in \{1, \ldots, q\}$ such that one of the following two events happens:

$$\begin{aligned}
\text{HIT}_i^{\text{fwd}} &: (X^i, Y^i, \text{fwd}) \in \mathcal{Q} \text{ and} \\
&\qquad \text{inner}_c(Y^i) \in \{\text{inner}_c(X^1), \ldots, \text{inner}_c(X^{i-1})\} \\
\text{HIT}_i^{\text{inv}} &: (X^i, Y^i, \text{inv}) \in \mathcal{Q} \text{ and} \\
&\qquad \text{inner}_c(X^i) \in \{\text{inner}_c(Y^1), \ldots, \text{inner}_c(Y^{i-1}), \text{inner}_c(0^b)\}
\end{aligned}$$

By basic probability theory,

$$\begin{aligned}
\mathbf{Pr}(\text{INNER}(\mathcal{Q})) &\leq \sum_{i=1}^{q} \mathbf{Pr}(\text{INNER}(\mathcal{Q}_i) \wedge \neg\text{INNER}(\mathcal{Q}_{i-1})) \\
&\leq \sum_{i=1}^{q} \mathbf{Pr}\left(\text{HIT}_i^{\text{fwd}}(\mathcal{Q}_i) \vee \text{HIT}_i^{\text{inv}}(\mathcal{Q}_i)) \,\middle|\, \neg\text{INNER}(\mathcal{Q}_{i-1})\right)
\end{aligned}$$

For any i, the query is either in forward direction or in inverse direction. so it can only set one of the two events. The response at the i^{th} query is uniformly drawn from a set of size at least $2^b - q$, among which at most $i2^r$ elements set $\text{HIT}_i^{\text{fwd}}(\mathcal{Q}_i)$ or $\text{HIT}_i^{\text{inv}}(\mathcal{Q}_i)$. Thus:

$$\mathbf{Pr}(\text{HIT}_i^{\text{fwd}}(\mathcal{Q}_i) \vee \text{HIT}_i^{\text{inv}}(\mathcal{Q}_i)) \leq \frac{i2^r}{2^b - q}$$

Then, as $q \leq 2^{b-1}$,

$$\mathbf{Pr}(\text{INNER}(\mathcal{Q})) \leq \sum_{i=1}^{q} \frac{2i}{2^c} \leq \frac{q(q+1)}{2^c}$$

Lemma 2. We have

$$\mathbf{Pr}(\text{BADUNION}(\mathcal{Q})) \leq \frac{4q}{2^n} + \frac{4\ell q}{2^{n-r'}} \tag{4.6}$$

Proof of Lemma 2. We first note that if $\ell = 1$, $|S_{\text{inv}}| = 2^{c'} \leq 2^{b-n}$, and the result will be meaningless as BADUNION($\mathcal{Q}$) will depend on finding a state in S_{fwd} as finding a path state in S_{inv} is already guaranteed. Thus, we focus on the case $\ell \geq 2$. By basic probability, we obtain the following

$$\begin{aligned}\mathbf{Pr}(\text{BADUNION}(\mathcal{Q})) = \\ &\sum_{y=1}^{2^{c'}} \mathbf{Pr}(\text{BADUNION}(\mathcal{Q}) \,|\, |S_{\text{fwd}}| = y) \cdot \mathbf{Pr}(|S_{\text{fwd}}| = y) + \\ &\sum_{y=1}^{2^{c'}} \mathbf{Pr}(\text{BADUNION}(\mathcal{Q}) \,|\, |S_{\text{inv}}| = \ell y) \cdot \mathbf{Pr}(|S_{\text{fwd}}| = \ell y)\end{aligned}$$

From equations (14) and (20) of the tight preimage of the sponge paper [5], we have that

$$\mathbf{Pr}(\text{BADFWD}(\mathcal{Q})) = \sum_{y=1}^{2^{c'}} \mathbf{Pr}(\text{BADUNION}(\mathcal{Q}) \,|\, |S_{\text{fwd}}| = y) \cdot \mathbf{Pr}(|S_{\text{fwd}}| = y)$$

$$\mathbf{Pr}(\text{BADINV}(\mathcal{Q})) = \sum_{y=1}^{2^{c'}} \mathbf{Pr}(\text{BADUNION}(\mathcal{Q}) \,|\, |S_{\text{inv}}| = \ell y) \cdot \mathbf{Pr}(|S_{\text{inv}}| = \ell y)$$

Then,

$$\mathbf{Pr}(\text{BADUNION}(\mathcal{Q})) = \mathbf{Pr}(\text{BADFWD}(\mathcal{Q})) + \mathbf{Pr}(\text{BADINV}(\mathcal{Q})) \tag{4.7}$$

Taking the results from equations (13) and (19) from the tight preimage of the sponge paper [5], we have that

$$\mathbf{Pr}(\text{BADFWD}(\mathcal{Q})) \leq \frac{4q}{2^n} \tag{4.8}$$

and

$$\mathbf{Pr}(\text{BADINV}(\mathcal{Q})) \leq \frac{4\ell q}{2^{n-r'}} \tag{4.9}$$

Finally, plugging (4.8) and (4.9) into (4.7) we get

$$\mathbf{Pr}(\text{BADUNION}(\mathcal{Q})) \leq \frac{4q}{2^n} + \frac{4\ell q}{2^{n-r'}}$$

Lemma 3. We have

$$\mathbf{Pr}(x\text{-BADFWD}(\mathcal{Q})) \leq \left(\frac{4q}{2^n}\right)^x \tag{4.10}$$

Proof of Lemma 3.
In this proof, we prove the bound using inductive reasoning. We have the following base case whose bound is taken from the tight sponge preimage paper [5].

$$\mathbf{Pr}(1\text{-BADFWD}(\mathcal{Q})) \leq \frac{4q}{2^n} \tag{4.11}$$

The following is the inductive step, which is reasoned as follows. If the i^{th} query triggers x-BADFWD($\mathcal{Q}$), then it means that after $i-1$ queries x-BADFWD($\mathcal{Q}$) was not set. Also, it is necessary that $(x-1)$-BADFWD($\mathcal{Q}$) must have been set after $i-1$ queries for x-BADFWD($\mathcal{Q}$) to be set at the i^{th} query.

$$\begin{aligned}
&\mathbf{Pr}(x\text{-BADFWD}(\mathcal{Q})) \\
&= \sum_{i=1}^{q} \mathbf{Pr}(x\text{-BADFWD}(\mathcal{Q}_i) \wedge (x-1)\text{-BADFWD}(\mathcal{Q}_{i-1}) \wedge \neg x\text{-BADFWD}(\mathcal{Q}_{i-1})) \\
&\leq \sum_{i=1}^{q} \mathbf{Pr}(x\text{-BADFWD}(\mathcal{Q}_i) \,|\, (x-1)\text{-BADFWD}(\mathcal{Q}_{i-1}) \wedge \neg x\text{-BADFWD}(\mathcal{Q}_{i-1})) \\
&\qquad \times \mathbf{Pr}((x-1)\text{-BADFWD}(\mathcal{Q}_{i-1}))
\end{aligned} \tag{4.12}$$

We start by bounding the following event
$\mathbf{Pr}(x\text{-BADFWD}(\mathcal{Q}_i) \,|\, (x-1)\text{-BADFWD}(\mathcal{Q}_{i-1}) \wedge \neg x\text{-BADFWD}(\mathcal{Q}_{i-1}))$
There are a total of 2^b states that can be queried to find a winning state. There are x winning states, where a winning state is an element of S_{fwd}. A winning state is found when a queried state yields a path which squeezes out the digest. Each queried state is tried without replacement. If it is a winning state, then the total number of winning states and candidate states decreases by 1 element. If it is not a winning state, then winning states stay the same while candidate states decrease by 1 element. The event is conditioned under $x-1$ winning states already being found in $i-1$ queries. This means that finding the x^{th} winning state happens with probability $\frac{1}{2^b-(i-1)}$.

But let us take a look at the probability of there existing only 1 winning state. In this case, given the adversary finds the preimage at the q^{th} query,

they succeed in finding the preimage with probability $\frac{1}{2^b-(q-1)}$.

Clearly, $\frac{1}{2^b-(i-1)} \leq \frac{1}{2^b-(q-1)}$ for $i \leq q$. Consequently, the probability of obtaining the x^{th} distinct hit is upper bounded by the probability of a fresh single hit. Therefore, we upper bound the event of finding the x^{th} preimage given $x-1$ found preimages with a fresh single hit event. So,

$$\begin{aligned}\mathbf{Pr}(x\text{-BADFWD}(\mathcal{Q}_i) \,|\, (x-1)\text{-BADFWD}(\mathcal{Q}_{i-1}) \wedge \neg x\text{-BADFWD}(\mathcal{Q}_{i-1})) \\ \leq \mathbf{Pr}(1\text{-BADFWD}(\mathcal{Q}_q) \,|\, \neg 1\text{-BADFWD}(\mathcal{Q}_{q-1}))\end{aligned} \tag{4.13}$$

Plugging (4.11) into (4.13). Since conditioned on no previous occurrence of 1-BADFWD, the q^{th} query is the attempt when the probability of setting 1-BADFWD is highest. Hence, its success probability at this attempt is bounded by the case $q = 1$.

$$\begin{aligned}\mathbf{Pr}(x\text{-BADFWD}(\mathcal{Q}_i) \,|\, (x-1)\text{-BADFWD}(\mathcal{Q}_{i-1}) \wedge \neg x\text{-BADFWD}(\mathcal{Q}_{i-1})) \\ \leq \frac{4}{2^n}\end{aligned} \tag{4.14}$$

Then plugging (4.14) into (4.12).

$$\begin{aligned}\mathbf{Pr}(x\text{-BADFWD}(\mathcal{Q})) &\leq \sum_{i=1}^{q} \frac{4}{2^n} \times \mathbf{Pr}((x-1)\text{-BADFWD}(\mathcal{Q})) \\ &\leq \frac{4q}{2^n} \times \mathbf{Pr}((x-1)\text{-BADFWD}(\mathcal{Q}))\end{aligned} \tag{4.15}$$

Applying the inductive step (4.12) to (4.15), followed by the same proof recursively $x-2$ times until we get,

$$\mathbf{Pr}(x\text{-BADFWD}(\mathcal{Q})) \leq \left(\frac{4q}{2^n}\right)^{x-1} \times \mathbf{Pr}(1\text{-BADFWD}(\mathcal{Q})) \tag{4.16}$$

Applying the base case (4.11) to (4.16).

$$\begin{aligned}\mathbf{Pr}(x\text{-BADFWD}(\mathcal{Q})) &\leq \left(\frac{4q}{2^n}\right)^{x-1} \times \frac{4q}{2^n} \\ &\leq \left(\frac{4q}{2^n}\right)^{x}\end{aligned}$$

which completes the proof.

4.4 Result

From (4.4), Lemma 1, Lemma 2, Lemma 3, we obtain

$$\mathbf{Pr}(x\text{-PRE}(\mathcal{Q})) \leq \left(\frac{4q}{2^n}\right)^x + \min\left\{\frac{4q}{2^n} + \frac{4\ell q}{2^{n-r'}}\ ,\ \frac{q(q+1)}{2^c}\right\} \tag{4.17}$$

and this completes the proof of Theorem 1.

Chapter 5

Conclusions

The main contribution of this thesis is the extension of the preimage analysis of the sponge construction [5] to the setting of x-preimage resistance in the ideal permutation model. By decomposing the success event into executions with and without inner collisions, we derived an upper bound on the probability of finding x distinct preimages of a target digest Z.

$$\mathbf{Adv}^{x\text{-ePre}}_{\mathcal{H}^{\mathcal{P}}} \leq \left(\frac{4q}{2^n}\right)^x + \min\left\{\frac{4q}{2^n} + \frac{4\ell q}{2^{n-r'}}\ ,\ \frac{q(q+1)}{2^c}\right\}$$

Our security bound matches the security lower bound in (3.1) up to a factor of $\frac{4x}{e}$. It shows that, in the absence of inner collisions, the probability of obtaining x distinct preimages decreases exponentially with x, yielding a security level comparable to a random oracle.

Comparing the proof bound with the best known attack shows that the dominant attack strategy is not a direct x-preimage search but rather the construction of an inner-state collision followed by state recovery. This happens when x gets too large such that the security bound is capped by the latter. Thus, the effective security of the construction is governed by

$$\min\left\{\frac{x2^n}{e}, \left(2^{\frac{c}{2}+1} + 2^{n-r'}\right)\right\}$$

These findings provide a first step towards a theoretical characterization of multi-preimage resistance for sponge constructions and suggest that the techniques used in tight single-preimage analysis can be extended to reason about multiple preimages as well.

The proven bound we obtain is not a tight bound. This gap stems from the inductive argument used to extend the result from one preimage to x preimages. In particular, the proof bounds the probability of obtaining an additional preimage by the same tight 1-preimage bound at every induction

step. Although this gives the desired exponential decrease in the success probability, it does not exploit the combinatorial structure for x successful preimages. A tighter analysis would likely require a technique that treats the x preimages jointly and recovers the combinatorial factor. Whether such an analysis is possible in the ideal-permutation model remains an open problem.

Chapter 6

Bibliography

1. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge functions. Ecrypt Hash Workshop 2007 (May 2007)

2. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON Family of Lightweight Hash Functions. In: Rogaway, P. (ed.) Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6841, pp. 222-239. Springer (2011), https://doi.org/10.1007/978-3-642-22792-9_13

3. Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In: Naor, M. (ed.) Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings. Lecture Notes in Computer Science, vol. 2951, pp. 21-39. Springer (2004), https://doi.org/10.1007/978-3-540-24638-1_2

4. Andreeva, E., Mennink, B., Preneel, B.: Security Reductions of the Second Round SHA-3 Candidates. In: Burmester, M., Tsudik, G., Magliveras, S.S., Ilic, I. (eds.) Information Security - 13th International Conference, ISC 2010, Boca Raton, FL, USA, October 25-28, 2010, Revised Selected Papers. Lecture Notes in Computer Science, vol. 6531, pp. 39-53. Springer (2010), https://doi.org/10.1007/978-3-642-18178-8_5

5. Charlotte Lefevre and Bart Mennink. Tight preimage resistance of the sponge construction. IACR Cryptol. ePrint Arch., page 734, 2022.

6. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the Indifferentiability of the Sponge Construction. In: Smart, N.P. (ed.) Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic

Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings. Lecture Notes in Computer Science, vol. 4965, pp. 181-197. Springer (2008), https://doi.org/10.1007/978-3-540-78967-3_11