Efficient Parallelizable Hashing Using Small Non-Compressing Primitives

Bart Mennink and Bart Preneel

Dept. Electrical Engineering, ESAT/COSIC, KU Leuven, and iMinds, Belgium
bart.mennink@esat.kuleuven.be, bart.preneel@esat.kuleuven.be

Abstract. A well-established method of constructing hash functions is to base them on non-compressing primitives, such as one-way functions or permutations. In this work we present S^r, an rn-to-n-bit compression function (for $r \geq 1$) making $2r-1$ calls to n-to-n-bit primitives (random functions or permutations). S^r compresses its inputs at a rate (the amount of message blocks per primitive call) up to almost 1/2, and it outperforms all existing schemes with respect to rate and/or the size of underlying primitives. For instance, instantiated with the 1600-bit permutation of NIST's SHA-3 hash function standard it offers about 800-bit security at a rate of almost 1/2, while SHA-3-512 itself achieves only 512-bit security at a rate of about 1/3. We prove that S^r achieves asymptotically optimal collision security against semi-adaptive adversaries up to almost $2^{n/2}$ queries and that it can be made preimage secure up to 2^n queries using a simple tweak.

Keywords. Hash function; Small Primitives; Collision Resistance; Preimage Resistance; Efficient; Parallelizable.

1 Introduction

For decades, the leading approach in hash function design has been blockcipher-based: a cipher E is employed in a certain mode to obtain a hash function that satisfies some security guarantees. This approach has been analyzed in detail [13,14,20,23,24,26,28,34,37,38,47], and is followed by numerous hash functions, including the well-known SHA-{0,1,2} and MD{4,5}. The recent trend is, however, predominantly permutation-based. Notably, three out of the five finalists in NIST's SHA-3 competition [36], including the eventual winner Keccak [8], are built on permutations, and also the recently started CAESAR authenticated encryption competition [15] received a wide range of permutation-based submissions [1, 2, 5, 10, 11, 17, 22, 25, 35, 43, 44]. Permutations do not require a key schedule (and particularly, there is no need to re-key, which could be quite expensive for some blockciphers) and are simpler to design and analyze. Additionally, the study of constructions starting from small domain random functions or permutations is highly relevant [19, 30]. Note that, furthermore, a small set of permutations can be easily generated from one blockcipher by fixing a handful of keys.

Consider a compression function $F : \{0,1\}^{M+n} \to \{0,1\}^n$ making d calls to an s-bit primitive f (a blockcipher, non-compressing function, or permutation). The efficiency of such a construction is commonly expressed in terms of a *rate*: $\frac{M}{ds}$, the number of message bits divided by the (scaled) number of primitive calls. Intuitively, a larger rate corresponds to less primitive calls per message compression and thus to a higher efficiency. The "scaling" is done by the term s in the denominator, and a construction with a larger underlying primitive has a larger value s and thus a lower rate.

Many blockcipher-based compression functions achieve a high rate. For instance, the classical $2n$-to-n-bit Davies-Meyer compression function $F(h, m) = E(m, h) \oplus h$ has rate 1. Double-length blockcipher-based compression function such as Tandem-DM [28] compress at a rate 1/2: they map $3n$ bits to $2n$ bits making 2 calls to an n-bit blockcipher.

For non-compressing primitives, which do not offer compression on their own, a high rate appears harder to achieve. One approach of designing a hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^n$

with optimal $n/2$-bit collision security is using non-compressing primitives of size significantly larger than n bits. This approach is for instance followed by the Sponge [9]: it iterates a permutation on $c+m$ bits, where c is the capacity and m equals the message block size.[1] Sponge functions make one primitive call per message block, have a rate $\frac{m}{c+m}$, and are proven secure up to $2^{c/2}$ queries [6]. (We remark that Sponge functions allow for variable output sizes, by making extra primitive calls in the squeezing phase and outputting m bits at a time. This approach, however, requires extra primitive calls, which influences the rate.) The new SHA-3 hashing standard is a Sponge using a 1600-bit permutation, and above computations apply. For example, SHA-3-256 offers 256-bit security by compressing at a rate of about 2/3 and SHA-3-512 offers 512-bit security by compressing at a rate of about 1/3. Note that, intuitively, one should be able to achieve about 800-bit security using a 1600-bit primitive.

Black et al. [12], however, showed that it is impossible to construct a secure n-bit hash or compression function using one call to an n-bit non-compressing primitive. More generally, for a function $F : \{0,1\}^{M+n} \to \{0,1\}^n$ making d calls to an s-bit primitive, collisions can be found in at most $2^{(ds-M)/(d+1)}$ queries, a bound commonly known as "Stam's bound" and proven in [41,46,48,49]. Stam's bound implies that a $2n$-to-n-bit function requires at least three n-bit primitive calls to achieve optimal collision resistance, hence such a function has rate at most 1/3. The problem of designing such a function has been well-studied [29, 31, 42, 45], and we highlight the Shrimpton-Stam compression function, which we will refer to as S^2:

$$S^2(x_0, x_1) = f_0(x_0) \oplus f_2(f_0(x_0) \oplus f_1(x_1)) \,.$$

The design is proven asymptotically optimally collision secure if f_0, f_1, f_2 are three independent n-bit random functions, or if they are instantiated as $f_i(x) = \pi_i(x) \oplus x$ for distinct permutations π_i [31]. It is, however, known to be insecure if one takes $f_0 = f_1 = f_2$ [31, 45]. This and other functions have a rate of 1/3 or worse, and improving it has turned out to be a very difficult theoretical problem.

1.1 Our Contributions

We introduce the family of compression functions $S^r : \{0,1\}^{rn} \to \{0,1\}^n$ for $r \geq 1$. For $r = 8$, the function S^8 is depicted in Fig. 1. The function makes $2r-1$ function calls to $2\lceil \log_2 r \rceil + 1$ distinct primitives. Our class of functions is a graphical generalization of the Shrimpton-Stam compression function S^2, but it offers a higher rate $\frac{r-1}{2r-1}$, approaching 1/2 for increasing values of r, and thus allowing for a more efficient throughput of data whilst achieving comparable collision security. This rate is in fact optimal, witnessed by Stam's bound which suggests that at least $2r-1$ function calls need to be made. Additionally, S^r is well-parallelizable, and generally benefits from the same advantages as tree-based hash functions.

1.2 Efficiency

Based on the SHA-3 permutation $\pi : \{0,1\}^{1600} \to \{0,1\}^{1600}$, our function S^r achieves almost[2] 800-bit security with a rate approaching 1/2. This is in sharp contrast with SHA-3-512 which only achieves 512-bit security by compressing at a rate of about 1/3. If we instantiate our function S^r using smaller versions of the SHA-3 permutation, for instance on 400 or 200 bits, we can still get a high security level of almost 200 or 100 bits, respectively, while hashing at a rate that approaches 1/2. This is of particular interest for lightweight cryptography, because S^r shows that approximately the same level of security can be achieved as comparable schemes, but using much smaller underlying primitives.

[1] The authors originally refer to m as the "rate," but in our terminology "rate" has a different meaning.

[2] S^r makes use of $p := 2\lceil \log_2 r \rceil + 1$ distinct primitives, so $\lceil \log_2 p \rceil$ bits of the input to π are reserved for domain separation.

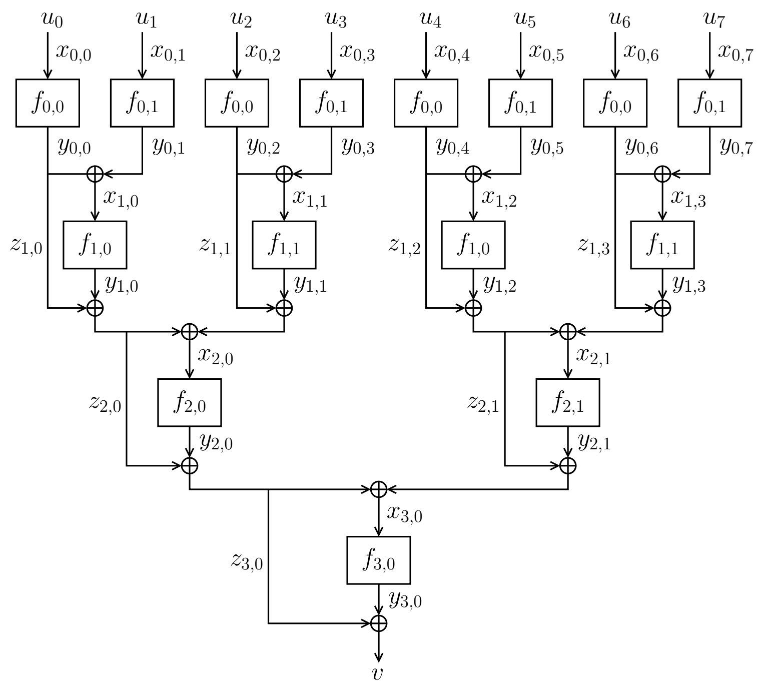


Fig. 1: Compression function $S^8 : \{0,1\}^{8n} \to \{0,1\}^n$ making 15 primitive calls. Here, $f_{j,b}$ (for $(j,b) \in (\{0,1,2\} \times \{0,1\}) \cup \{(3,0)\})$ are one-way functions, but these can be instantiated as $f_{j,b}(x) = \pi_{j,b}(x) \oplus x$ at no collision security loss.

We present a generic comparison of S^r in a Merkle-Damgård mode of operation (MD-S^r) [16,33] or in a Merkle tree (MT-S^r) [32] with Sponge functions [9], Grøstl [21], and MD6 [39] in Table 1. In this analysis (see App. A for the technical details) we aim for comparable $2^{n/2}$ collision security and adopt the design parameters accordingly. We also include a comparison for a specific set of parameters. We observe that S^r achieves comparable rate and efficiency, but using primitives that are a factor 2 to 4 smaller. However, the security analysis of S^r requires more distinct primitives than the other functions, and the proof is performed in a slightly different model (see Sect. 1.3).

1.3 Security

We prove that S^r, either based on random functions or random permutations, is collision secure up to about $2^{n/2}/n$ queries. In other words, S^r is asymptotically nearly optimally collision secure. In Table 2, we compare the rates and collision security guarantees of various instantiations of S^r, both for the general case and for $n = 512$. The proof is performed in a model where the adversary makes its queries layer-wise, which means that all queries to $f_{j-1,0}$ and $f_{j-1,1}$ must be made before all queries to $f_{j,0}$ and $f_{j,1}$, for $j = 1, \dots, \ell$. We also present a proof of $n/3$-bit security in the fully adaptive model and justify why it cannot be easily improved. This is in part as security proofs are known to become significantly harder when based on non-compressing primitives [30]. We conjecture that S^r *does* achieve optimal collision security. Additionally, for technical reasons we require distinctness of the $2\lceil\log_2 r\rceil + 1$ underlying primitives. This can be achieved by employing a single blockcipher for a fixed set of distinct keys. In Sect. 7 we show that it is non-trivial to reduce the number of distinct primitives in S^r.

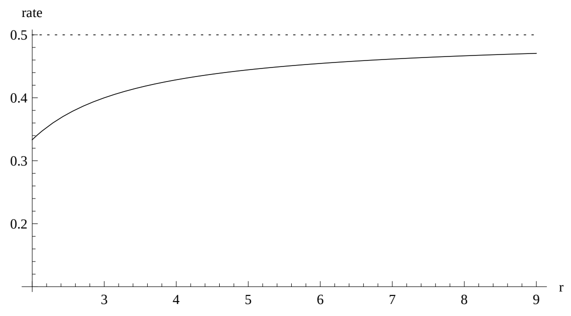
Next, we prove that S^2 based on random functions is preimage resistant up to $2^{2n/3}$ queries (solving an open problem of Shrimpton and Stam [45]). This result does not apply to the permutation-based setting: an attack on S^r proving tight $2^{n/2}$ security is derived for any

Table 1: Simplified comparison of MD-S^r with the Sponge function and Grøstl (first), and MT-S^r with MD6 (second). m denotes the message block size (its rate). See also App. A.

	general bounds				for $r=4$ and $n=m=512$			
function	rate	primitive size	# distinct primitives	collision security	rate	primitive size	# distinct primitives	collision security
MD-S^r	$\frac{r-1}{2r-1}$	n	$2\lceil\log_2 r\rceil+1$	$2^{n/2}/n$	3/7	512	5	2^{247}
Sponge [9]	$\frac{m}{n+m}$	$n+m$	1	$2^{n/2}$	1/2	1024	1	2^{256}
Grøstl [21]	1/2	$2n$	2	$2^{n/2}$	1/2	1024	2	2^{256}
MT-S^r	$\frac{r-1}{2r-1}\frac{M}{M-1}$	n	$2\lceil\log_2 r\rceil+1$	$2^{n/2}/n$	$3/7\frac{M}{M-1}$	512	5	2^{247}
MD6 [39]	$3/4\frac{M}{M-1}$	$4n$	1	$2^{n/2}$	$3/4\frac{M}{M-1}$	2048	1	2^{256}

Table 2: Rates of $S^r : \{0,1\}^{rn} \to \{0,1\}^n$ for various values of r, with on the right a supporting graph for $n = 512$.

function	rate	# distinct primitives	collision security
S^2	1/3	3	$2^{n/2}/n$
S^3	2/5	5	$2^{n/2}/n$
S^4	3/7	5	$2^{n/2}/n$
...	...	...	...
S^r	$\frac{r-1}{2r-1}$	$2\lceil\log_2 r\rceil+1$	$2^{n/2}/n$



$r \geq 2$. We also show that a simple tweak can make S^r optimally preimage secure. Formally, if we consider a hash or compression function F followed by a sufficiently strong finalization function G, the design is as collision secure as the weakest of both and as preimage secure as G (see also Lem. 1). The efficiency (rate) of the design is dominated by the rate of F. Concretely:

$$\mathcal{H}(M) = f \circ \text{MD-}S^r(M)\,,$$

where $G := f$ is a random function, is collision resistant up to about $2^{n/2}/n$ queries (the bound of S^r), preimage resistant up to about 2^n queries (the preimage security of f), and it has a rate of more or less $\frac{r-1}{2r-1}$. Hence, in this way, one combines the efficiency and collision security of S^r with the preimage security of f. The same result holds if $f(x) = \pi(x) \oplus x$ for a permutation π.[3] We remark that this trick does not apply to second preimage resistance.

1.4 Outline

We present our family of functions S^r based on functions $f_{j,b}$ in Sect. 2. Next, we give a security analysis of S^r: the model is introduced in Sect. 3, collision resistance is analyzed in Sect. 4, and preimage resistance in Sect. 5. In Sect. 6, we discuss the effect of instantiating the underlying primitives $f_{j,b}$ using permutations $\pi_{j,b}$. The work is concluded in Sect. 7.

[3] These findings seem to violate the preimage bound of Rogaway and Steinberger [41], but note that their bound *does not apply*: the construction $\mathcal{H}$ has a high preimage degeneracy.

S^r for $r = 2^\ell$ with $\ell \geq 0$

```
procedure $S^r(u_0, \ldots, u_{r-1})$
    for $i = 0, \ldots, 2^\ell - 1$ do
        $x_{0,i} \leftarrow u_i$
        $y_{0,i} \leftarrow f_{0,i \bmod 2}(x_{0,i})$
        $z_{0,i} \leftarrow 0$
    for $j = 1, \ldots, \ell$ and $i = 0, \ldots, 2^{\ell-j} - 1$ do
        $x_{j,i} \leftarrow (y \oplus z)_{j-1,2i} \oplus (y \oplus z)_{j-1,2i+1}$
        $y_{j,i} \leftarrow f_{j,i \bmod 2}(x_{j,i})$
        $z_{j,i} \leftarrow (y \oplus z)_{j-1,2i}$
    return $v \leftarrow (y \oplus z)_{\ell,0}$
```

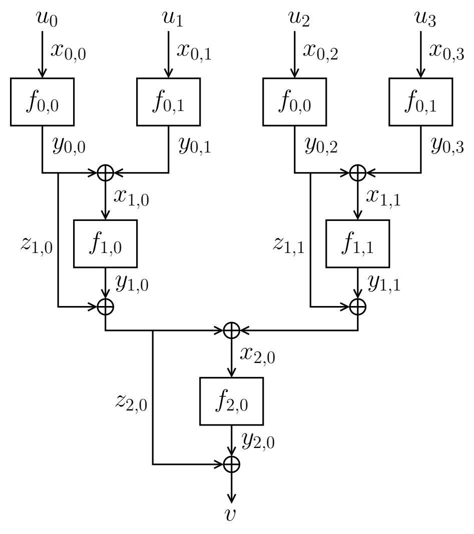


Fig. 2: S^r for $r = 2^\ell$ with $\ell \geq 0$ and an illustration of S^4. Here, we write $(y \oplus z)_{j,i} = y_{j,i} \oplus z_{j,i}$.

2 Hash Function Proposal S^r

Throughout, r and ℓ always denote integral parameters. We consider $n \in \mathbb{N}$ and put $N = 2^n$. For simplicity, we first introduce S^r for r being a power of two. Next, we generalize it to arbitrary $r \geq 1$.

2.1 S^r for $r = 2^\ell$

Write $r = 2^\ell$ with $\ell \geq 0$, and let $f_{j,b} : \{0,1\}^n \to \{0,1\}^n$ be one-way functions for $(j, b) \in (\{0, \ldots, \ell-1\} \times \{0,1\}) \cup \{(\ell, 0)\}$. A description of $S^r : \{0,1\}^{rn} \to \{0,1\}^n$ is given in Fig. 2 together with an illustration of S^4. S^r makes in total $2r - 1$ primitive calls, which is optimal with respect to Stam's bound. These calls are made to in total $2\ell + 1$ distinct primitives.

The description of S^r can informally be described by the following two steps. First, the inputs $u_0, \ldots, u_{r-1}$ are "processed" using functions $f_{0,0}, f_{0,1}$. Second, for $j = 1, \ldots, \ell$ and $i = 0, \ldots, 2^{\ell-j} - 1$, at position (j, i) the function S^r proceeds as follows: given the outcomes of the rounds at positions $(j-1, 2i)$ and $(j-1, 2i+1)$, the primitive $f_{j,i \bmod 2}$ is evaluated on input of the XOR of these, and its output is XORed with the outcome of round $(j-1, 2i)$. Eventually, the output of S^r is the value obtained after the last step (for $j = \ell$). The feed-forwards in the evaluation are necessary: absence of them would allow an adversary to find a collision in, say, $x_{1,0}$ — typically found in about $2^{n/4}$ queries — in order to obtain a collision for S^r. They also prevent trivial attacks where, e.g., the left and right input halves are swapped.

2.2 S^r for Arbitrary r

The description of S^r for arbitrary $r \geq 1$ is given in Fig. 3 together with an illustration of S^3. The generalized S^r, indeed, also makes $2r - 1$ primitive calls (to in total $2\lceil \log_2 r \rceil + 1$ distinct primitives). Although this description is significantly more complex than the one of Fig. 2, the intuition is rather simple and we give it for $r = 3$.

To define S^3, we first consider S^4 (see the illustration of Fig. 2). Now, S^3 only has inputs (u_0, u_1, u_2) and no u_3, and therefore the "fork" that processes inputs u_2 and u_3 in S^4 only gets u_2. We can then simply discard the two corresponding calls $f_{0,0}$ and $f_{0,1}$ and define u_2 to be the input to $f_{1,1}$. The resulting function matches the illustration of S^3 in Fig. 3. In general, if the function has r input blocks, where $2^{\ell-1} < r \leq 2^\ell$, the idea is to have $2^{\ell-1}$ input blocks

S^r for $r \geq 1$ with $\ell = \lceil \log_2 r \rceil$

```
1: procedure $S^r(u_0, \ldots, u_{r-1})$
2:     for $i = 0, \ldots, 2(r - 2^{\ell-1}) - 1$ do
3:         $x_{0,i} \leftarrow u_i$
4:         $y_{0,i} \leftarrow f_{0,i \bmod 2}(x_{0,i})$
5:         $z_{0,i} \leftarrow 0$
6:     for $i = 0, \ldots, r - 2^{\ell-1} - 1$ do
7:         $x_{1,i} \leftarrow (y \oplus z)_{0,2i} \oplus (y \oplus z)_{0,2i+1}$
8:         $y_{1,i} \leftarrow f_{1,i \bmod 2}(x_{j,i})$
9:         $z_{1,i} \leftarrow (y \oplus z)_{0,2i}$
10:    for $i = r - 2^{\ell-1}, \ldots, 2^{\ell-1} - 1$ do
11:        $x_{1,i} \leftarrow u_{i+r-2^{\ell-1}}$
12:        $y_{1,i} \leftarrow f_{1,i \bmod 2}(x_{1,i})$
13:        $z_{1,i} \leftarrow 0$
14:    for $j = 2, \ldots, \ell$ and $i = 0, \ldots, 2^{\ell-j} - 1$ do
15:        $x_{j,i} \leftarrow (y \oplus z)_{j-1,2i} \oplus (y \oplus z)_{j-1,2i+1}$
16:        $y_{j,i} \leftarrow f_{j,i \bmod 2}(x_{j,i})$
17:        $z_{j,i} \leftarrow (y \oplus z)_{j-1,2i}$
18:    return $v \leftarrow (y \oplus z)_{\ell,0}$
```

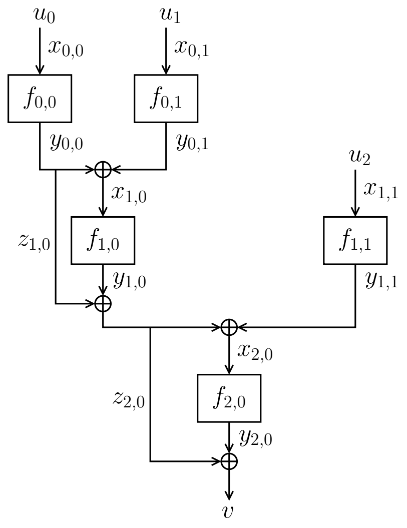


Fig. 3: S^r for $r \geq 1$ with $\ell = \lceil \log_2 r \rceil$ and an illustration of S^3.

right before the second layer. This means that in the first layer, $2(r - 2^{\ell-1})$ input blocks have to be processed to obtain $r - 2^{\ell-1}$ blocks $x_{1,i}$. These are then appended with the remaining $r - 2(r - 2^{\ell-1}) = 2^\ell - r$ compression function input blocks to obtain $2^{\ell-1}$ input values to the second layer. From this point onwards, the function description is the same as in Fig. 2.

3 Security Model

For two sets $S, T \subseteq \{0,1\}^n$, we denote $S \oplus T = \{s \oplus t \mid (s,t) \in S \times T\}$. For $n \in \mathbb{N}$, we denote by $\mathsf{Func}(n)$ the set of all functions $\{0,1\}^n \to \{0,1\}^n$, and by $\mathsf{Perm}(n)$ its subset of all permutations on n bits. We consider the security of S^r in the ideal model, where its underlying primitives, denoted as a set $\mathcal{P}$, are considered to be randomly drawn from $\mathsf{Func}(n)$. (Later on, we consider S^r where its underlying functions $f_{j,b}$ are instantiated as $f_{j,b}(x) = \pi_{j,b}(x) \oplus x$, and in this case we consider S^r based on ideal permutations from $\mathsf{Perm}(n)$.) We consider adversaries A that have unbounded computational power and query access to these random primitives $\mathcal{P}$, and their complexities are solely measured by the number of queries they make to their oracles. We assume that the adversarial queries are stored in a query history $\mathcal{Q}$. We require that $\mathcal{Q}$ always contains the queries necessary for the evaluation of the mounted collision or preimage attack.

Collision Resistance. Adversary A finds a collision for S^r if it obtains two distinct tuples $\underline{u} = (u_0, \ldots, u_{r-1})$, $\underline{u}' = (u'_0, \ldots, u'_{r-1})$ that satisfy $S^r(\underline{u}) = S^r(\underline{u}')$. The advantage of a collision-finding adversary A is defined as

$$\mathsf{Adv}^{\mathsf{col}}_{S^r}[A] = \mathbb{P}\left[\mathcal{P} \xleftarrow{\$} \mathsf{Func}(n)^{2\lceil \log_2 r \rceil + 1},\ \underline{u}, \underline{u}' \leftarrow A^{\mathcal{P}}\ :\ \underline{u} \neq \underline{u}' \ \wedge\ S^r(\underline{u}) = S^r(\underline{u}')\right].$$

For a set of adversaries $\mathcal{A}$, we define by $\mathsf{Adv}^{\mathsf{col}}_{S^r}[\mathcal{A}]$ the maximum advantage of any adversary $A \in \mathcal{A}$.

Preimage Resistance. We consider preimage security for every range point (also known as everywhere preimage resistance [40]). Prior to making any query to its oracles, A is given a range value $v \in \{0,1\}^n$, and A succeeds in finding a preimage for v if it detects a $\underline{u}$ satisfying

$S^r(\underline{u}) = v$. The success probability of A is then maximized over all possible chosen range values. The advantage of an everywhere preimage-finding adversary A is defined as

$$\mathsf{Adv}^{\mathsf{epre}}_{S^r}[A] = \max_{v \in \{0,1\}^n} \mathbb{P}\left[\mathcal{P} \xleftarrow{\$} \mathsf{Func}(n)^{2\lceil \log_2 r \rceil + 1},\ \underline{u} \leftarrow A^{\mathcal{P}}(v)\ :\ S^r(\underline{u}) = v\right].$$

For a set of adversaries $\mathcal{A}$, we define by $\mathsf{Adv}^{\mathsf{epre}}_{S^r}[\mathcal{A}]$ the maximum advantage of any adversary $A \in \mathcal{A}$.

Composition. Evidently, equivalent definitions can be given for functions $F : \{0,1\}^s \to \{0,1\}^n$ based on a different amount of primitives $\mathcal{P}$ from $\mathsf{Func}(n)$, or even on different primitives. We present the following useful lemma regarding the collision and preimage security of $G \circ F$ for any hash or compression function F and any sufficiently strong finalization function G based on $\mathcal{P}$. One can think of F being S^r or even MD-S^r and G an element from $\mathsf{Func}(n)$.

Lemma 1. *Consider $G \circ F$ based on random primitives $\mathcal{P}$. Then, for any adversary A,*

$$\mathsf{Adv}^{\mathsf{col}}_{G\circ F}[A] \leq \mathsf{Adv}^{\mathsf{col}}_{G}[A] + \mathsf{Adv}^{\mathsf{col}}_{F}[A], \qquad \mathsf{Adv}^{\mathsf{epre}}_{G\circ F}[A] \leq \mathsf{Adv}^{\mathsf{epre}}_{G}[A].$$

Proof. For $G \circ F$, denote the input to F as u, the input to G (the output of F) as v and the output of G as w. In order to find a collision for $G \circ F$, the adversary needs to find two different u, u' with $w = w'$. Clearly, if the intermediate values v, v' are distinct, such a collision implies a collision for G, and otherwise it implies a collision for F. Next, for preimage resistance, consider a given range value w, and assume the adversary finds a preimage u. Then, $F(u)$ is a preimage of w under G. □

4 Collision Security of S^r

In this section we analyze the collision resistance of S^r. First, in Sect. 4.1, we consider the case of $r = 2^\ell$ (with security proofs in Sects. 4.2-4.3). Next, in Sect. 4.4 we show how the result generalizes to arbitrary values of r.

4.1 S^r for $r = 2^\ell$

We derive the following result for the collision resistance of S^r for $r = 2^\ell$ with $\ell \geq 0$ (Fig. 2).

Theorem 1. *Let $r = 2^\ell$ with $\ell \geq 0$. Let $\mathcal{A}^{\mathsf{lw}}(q)$ denote the set of all adversaries that make at most q queries and make those layer-wise (all queries to $f_{j-1,b}$ are made before all queries to $f_{j,b'}$ (for $j = 1, \ldots, \ell$ and b, b' arbitrary)). Then, for any positive integer value $\tau \geq 2$,*

$$\mathsf{Adv}^{\mathsf{col}}_{S^r}[\mathcal{A}^{\mathsf{lw}}(q)] \leq \frac{2(\tau^\ell q)^2}{N} + 2N\left(\frac{e(\tau^\ell q)^2}{N}\right)^\tau.$$

The proof is presented in Sect. 4.2. At a very high level, it is performed in a recursive manner: we demonstrate that a collision for S^r either happens in the last round, or that "something happened at an earlier stage." In more detail, we first claim that associated to every query $(x_{j,b}, y_{j,b})$ to $f_{j,b}$ there are at most τ^j possible values $z_{j,b}$, for some threshold value τ. Then, the adversary wins in either of the following two cases: (i) it finds a collision assuming that our claim holds, or (ii) it breaks the claim. Putting $\tau = n^{1/\ell}$ and recalling $N = 2^n$, we find that for any $\varepsilon > 0$,

$$\mathsf{Adv}^{\mathsf{col}}_{S^r}[\mathcal{A}^{\mathsf{lw}}(N^{1/2}/n^{1+\varepsilon})] \leq \frac{2}{n^{2\varepsilon}} + 2\left(\frac{2^\ell e}{n^{2\varepsilon}}\right)^{n^{1/\ell}},$$

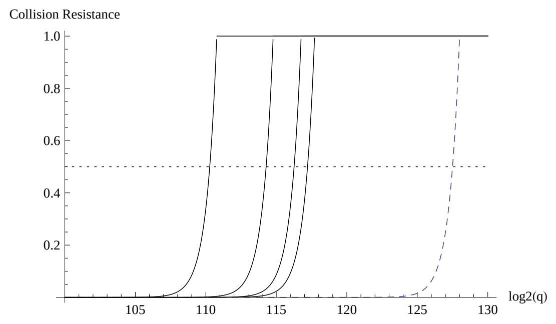


Fig. 4: The function $\mathsf{Adv}^{\mathsf{col}}_{S^r}[\mathcal{A}^{\mathsf{lw}}(q)]$ of Thm. 1 for $n = 256$ for $\ell = 16, 8, 4, 2$ (from left to right), in comparison with the trivial bound $q(q + 1)/2^n$ (dashed line).

which approaches 0 for $n \to \infty$. For various ℓ, the bound of Thm. 1 is depicted in Fig. 4.

Theorem 1 is restricted to adversaries that make their queries layer-wise (hence the "lw" in $\mathcal{A}^{\mathsf{lw}}(q)$). Intuitively, this does not limit the impact of the security proof: the best way for an adversary to find a collision is to make queries to $f_{j,b}$ for increasing values of j in such a way to obtain a maximal yield (the number of inputs to S^r that can be evaluated using the queries made by the adversary). In fact, Shrimpton and Stam [45] already pointed out that for S^2 it is fair to just consider adversaries that query their oracles sequentially (top layer first, bottom layer next). Unfortunately, the proof of Thm. 1 cannot straightforwardly be generalized to the fully adaptive case due to a complicated technicality: a query to $f_{j,b}$ influences all possible lower-level feed-forward values. Nevertheless, using a simple tweak it is possible to generalize Thm. 1 to adaptive security up to about $2^{n/3}$ queries:

Theorem 2. *Let $r = 2^\ell$ with $\ell \geq 0$. Let $\mathcal{A}(q)$ denote the set of all adversaries that make at most q queries. Then, for any positive integer value $\tau \geq 2$,*

$$\mathsf{Adv}^{\mathsf{col}}_{S^r}[\mathcal{A}(q)] \leq \frac{2(\tau^\ell q)^2}{N} + 2N\left(\frac{e(\tau^\ell q)^2}{N}\right)^\tau + \frac{2\tau^{2\ell}q^3}{N}\,.$$

The proof is given in Sect. 4.3. We conjecture that this result can be improved to approximately $2^{n/2}$ collision secure (in the fully adaptive model).

4.2 Proof of Theorem 1

We consider the security of $S^r : \{0,1\}^{rn} \to \{0,1\}^n$, for $r = 2^\ell$ with $\ell \geq 0$, based on $2\ell + 1$ functions $\big\{f_{j,b} \mid (j, b) \in (\{0, \ldots, \ell - 1\} \times \{0, 1\}) \cup \{(\ell, 0)\}\big\}$ randomly drawn from $\mathsf{Func}(n)$. The focus is on adversaries that make all queries to $f_{j-1,b}$ before all queries to $f_{j,b'}$.

We consider any adversary A that has query access to its oracles $\mathcal{P}$ and makes q queries. These queries are stored in a query history $\mathcal{Q}$ as indexed tuples of the form $(x^k_{j,b}, y^k_{j,b})$, where k is the query index (omitted if irrelevant) and (j, b) refers to the oracle index. For $q \geq 0$, by $\mathcal{Q}_q$ we define the query history after q queries.

Associated to each query $(x_{j,b}, y_{j,b})$ is a multiset $\mathcal{Z}_{j,b}$ of all possible feed-forward values $z_{j,b}$ occurring for this query. E.g., for a query $(x_{0,0}, y_{0,0})$ we have $\mathcal{Z}_{0,0} = \{0\}$, and for an additional query $(x_{1,0}, y_{1,0})$ for which also $(x_{0,1}, y_{0,0} \oplus x_{1,0})$ to $f_{0,1}$ exists, $\mathcal{Z}_{1,0} = \{y_{0,0}\}$. Abusing notation, we sometimes refer to the query and its feed-forward set as $(x_{j,b}, y_{j,b}, \mathcal{Z}_{j,b})$ or simply $(x, y, \mathcal{Z})_{j,b}$. We recall notation $(y \oplus z)_{j,b} = y_{j,b} \oplus z_{j,b}$. Note that $\mathcal{Z}_{j,b}$ is independent of the position at which $x_{j,b}$ may occur in S^r: it may occur at position $(j, b + 2\lambda)$ for $\lambda \in \{0, \ldots, 2^{\ell-j-1} - 1\}$, but for every such position its corresponding feed-forward set is the same.

Denote by $\mathsf{col}S^r(\mathcal{Q}_q)$ the event that A finds two distinct evaluations of S^r satisfying $S^r(u_0, \ldots, u_{r-1}) = S^r(u'_0, \ldots, u'_{r-1})$. We write $x_{j,i}, y_{j,i}, z_{j,i}$ for all intermediate values corresponding to the first evaluation and $x'_{j,i}, y'_{j,i}, z'_{j,i}$ for all values of the second evaluation. By definition:

$$\mathsf{Adv}^{\mathsf{col}}_{S^r}[A] = \mathbb{P}\left[\mathsf{col}S^r(\mathcal{Q}_q)\right] . \tag{1}$$

For the analysis of $\mathbb{P}\left[\mathsf{col}S^r(\mathcal{Q}_q)\right]$ we introduce two helping events. Here, let $\tau \geq 2$ be any integer value.

$$\begin{aligned}
\mathsf{eA}(\mathcal{Q})_{j,b} :\ & \exists\, (x,y,\mathcal{Z})_{j,b}, (x',y',\mathcal{Z}')_{j,b} \in \mathcal{Q}_q \text{ such that} \\
& \qquad\qquad x_{j,b} \neq x'_{j,b} \wedge y_{j,b} \oplus y'_{j,b} \in \mathcal{Z}_{j,b} \oplus \mathcal{Z}'_{j,b} \,; \\
\mathsf{eB}(\mathcal{Q})_j :\ & \max_{z\in\{0,1\}^n} \left| \left\{ \begin{array}{l} (x,y,\mathcal{Z})_{j-1,0}, (x,y,\mathcal{Z})_{j-1,1} \in \mathcal{Q}_q \mid \\ \quad y_{j-1,0} \oplus y_{j-1,1} \oplus z \in \mathcal{Z}_{j-1,0} \oplus \mathcal{Z}_{j-1,1} \end{array} \right\} \right| > \tau^j .
\end{aligned}$$

We simply write $\mathsf{eA}(\mathcal{Q})_j = \mathsf{eA}(\mathcal{Q})_{j,0} \cup \mathsf{eA}(\mathcal{Q})_{j,1}$ and $\mathsf{eX}(\mathcal{Q}) = \bigcup_j \mathsf{eX}(\mathcal{Q})_j$ for $\mathsf{X} = \mathsf{A}, \mathsf{B}$. We furthermore write $\mathsf{e}(\mathcal{Q}) = \mathsf{eA}(\mathcal{Q}) \cup \mathsf{eB}(\mathcal{Q})$. First, in Lem. 2, we demonstrate that finding a collision is at least as hard as finding a solution for $\mathsf{eA}(\mathcal{Q}_q)$.

Lemma 2. $\mathsf{col}S^r(\mathcal{Q}_q) \Rightarrow \mathsf{eA}(\mathcal{Q}_q)$.

Proof. The proof is by contradiction. Assume $\neg\mathsf{eA}(\mathcal{Q}_q)$, and suppose an adversary makes all queries for the computation of S^r on input of two different vectors $(u_0, \ldots, u_{r-1})$ and $(u'_0, \ldots, u'_{r-1})$. By construction:

$$S^r(u_0, \ldots, u_{r-1}) = (y \oplus z)_{\ell,0} = (y' \oplus z')_{\ell,0} = S^r(u'_0, \ldots, u'_{r-1}) .$$

First, assume $x_{\ell,0} \neq x'_{\ell,0}$. Then, the collision forms a valid solution to $\mathsf{eA}(\mathcal{Q}_q)_{\ell,0}$, contradicting our assumption. Next, assume $x_{\ell,0} = x'_{\ell,0}$. Then also $y_{\ell,0} = y'_{\ell,0}$ and thus $z_{\ell,0} = z'_{\ell,0}$. By construction, this implies $(y \oplus z)_{\ell-1,0} = (y' \oplus z')_{\ell-1,0}$ and $(y \oplus z)_{\ell-1,1} = (y' \oplus z')_{\ell-1,1}$. Note that S^r without the ℓth layer corresponds to two parallel independent $S^{r/2}$ evaluations: one with inputs $(u_0, \ldots, u_{r/2-1})$ and output $(y \oplus z)_{\ell-1,0}$ and one with inputs $(u_{r/2}, \ldots, u_{r-1})$ and output $(y \oplus z)_{\ell-1,1}$. Given that the collision for S^r is non-trivial, it implies a non-trivial collision of either of the two $S^{r/2}$'s. Consider the $S^{r/2}$ with the non-trivial collision, and apply the same reasoning using $\mathsf{eA}(\mathcal{Q}_q)_{\ell-1,b}$. Here, $b = 0$ iff the non-trivial collision is in the left half. At some point one indeed ends up with a distinct pair $x_{j,i} \neq x'_{j,i}$ for some $j = \ell, \ldots, 0$, as $(u_0, \ldots, u_{r-1}) \neq (u'_0, \ldots, u'_{r-1})$. □

Therefore, we obtain for (1):

$$\mathbb{P}\left[\mathsf{col}S^r(\mathcal{Q}_q)\right] \leq \mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)\right] \leq \mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right] .$$

A bound on this probability is derived in Lem. 3.

Lemma 3. $\mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right] \leq \dfrac{2(\tau^\ell q)^2}{N} + 2N\left(\dfrac{e(\tau^\ell q)^2}{N}\right)^\tau$.

Proof. Recall the notation $\mathsf{e}(\mathcal{Q}_q) = \mathsf{eA}(\mathcal{Q}_q) \cup \mathsf{eB}(\mathcal{Q}_q)$. By basic probability theory:

$$\mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right] \leq \sum_{j=0}^{\ell} \mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)_j \cap \neg\mathsf{eB}(\mathcal{Q}_q)_j\right] + \sum_{j=1}^{\ell} \mathbb{P}\left[\mathsf{eB}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-1}\right], \tag{2}$$

noting that $\mathsf{eB}(\mathcal{Q}_q)_0$ is false by construction. We consider both probabilities separately.

We recall that A makes all queries to $f_{j-1,b}$ before all queries to $f_{j,b'}$. This particularly means that, at the point of making queries to $f_{j,b}$, the sets $\mathcal{Z}_{j,b}$ are fixed (by all previous queries) and remain unchanged. In more detail, we regularly use the following observation for any query $(x,y,\mathcal{Z})_{j,b}$ to $f_{j,b}$: $\neg\mathsf{eB}(\mathcal{Q}_q)_j \Rightarrow |\mathcal{Z}_{j,b}| \leq \tau^j$.

$\mathsf{eA}(\mathcal{Q}_q)_j$**.** Assume $\neg\mathsf{eB}(\mathcal{Q}_q)_j$ holds. Consider any b and any two distinct queries $(x,y,\mathcal{Z})_{j,b}$ and $(x',y',\mathcal{Z}')_{j,b}$ to $f_{j,b}$ (at most $\binom{q}{2}$ choices). These queries render a solution if $y_{j,b} \oplus y'_{j,b} \in \mathcal{Z}_{j,b} \oplus \mathcal{Z}'_{j,b}$. By $\neg\mathsf{eB}(\mathcal{Q}_q)_j$, we have $|\mathcal{Z}_{j,b}|, |\mathcal{Z}'_{j,b}| \leq \tau^j$. Consequently, the two queries complete the collision with probability at most $\frac{\tau^{2j}}{N}$. Summing over all queries to $f_{j,b}$, and both choices of b, we obtain:

$$\mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)_j \cap \neg\mathsf{eB}(\mathcal{Q}_q)_j\right] \leq 2\frac{(\tau^j q)^2}{2N} = \frac{(\tau^j q)^2}{N}\,.$$

$\mathsf{eB}(\mathcal{Q}_q)_j$**.** Assume $\neg\mathsf{e}(\mathcal{Q}_q)_{j-1}$ holds. Consider any $z \in \{0,1\}^n$. By virtue of $\neg\mathsf{eB}(\mathcal{Q}_q)_{j-1}$, any query $(x,y,\mathcal{Z})_{j-1,0}$ has $|\mathcal{Z}_{j-1,0}| \leq \tau^{j-1}$. Similar for any query $(x,y,\mathcal{Z})_{j-1,1}$.

Without loss of generality (by symmetry) consider a new query $(x,y,\mathcal{Z})_{j-1,0}$. This adds a solution to $\mathsf{eB}(\mathcal{Q}_q)_j$ with probability at most $\tau^{2(j-1)}q/N$, and any hit adds at most τ^{j-1} values (by $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-1}$). More than τ^j solutions are added with probability at most

$$\binom{q}{\frac{\tau^j}{\tau^{j-1}}}\left(\frac{\tau^{2(j-1)}q}{N}\right)^{\frac{\tau^j}{\tau^{j-1}}} \leq \left(\frac{e\tau^{2j-3}q^2}{N}\right)^{\tau} \leq \left(\frac{e(\tau^j q)^2}{N}\right)^{\tau}.$$

Summing over all N values z, we obtain:

$$\mathbb{P}\left[\mathsf{eB}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-1}\right] \leq N\left(\frac{e(\tau^j q)^2}{N}\right)^{\tau}.$$

Conclusion of proof. From (2), we obtain:

$$\begin{aligned}\mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right] &\leq \sum_{j=0}^{\ell}\frac{(\tau^j q)^2}{N} + \sum_{j=1}^{\ell} N\left(\frac{e(\tau^j q)^2}{N}\right)^{\tau} = \frac{q^2}{N}\sum_{j=0}^{\ell}\tau^{2j} + N\left(\frac{eq^2}{N}\right)^{\tau}\sum_{j=1}^{\ell}\tau^{2\tau j}\\ &\leq \frac{2(\tau^\ell q)^2}{N} + 2N\left(\frac{e(\tau^\ell q)^2}{N}\right)^{\tau}.\end{aligned}$$

Here, we use that $\sum_{j=0}^{\ell} x^j = \frac{x^{\ell+1}-1}{x-1} \leq \frac{x}{x-1}x^\ell \leq 2x^\ell$ for $x \geq 2$. □

4.3 Proof of Theorem 2

The proof of security against adaptive adversaries follows the proof of Thm. 1 in Sect. 4, but differs in various aspects. First of all, we add the following event $\mathsf{eC}(\mathcal{Q})$:

$$\begin{aligned}\mathsf{eC}(\mathcal{Q})_{j,b}:\ \exists\,(x,y,\mathcal{Z})^{k}_{j-1,0}, (x,y,\mathcal{Z})^{k'}_{j-1,1}, (x,y)^{k''}_{j,b} \in \mathcal{Q}_q \text{ such that}\\ \max\{k,k'\} > k'' \wedge y_{j-1,0} \oplus y_{j-1,1} \oplus x_{j,b} \in \mathcal{Z}_{j-1,0} \oplus \mathcal{Z}_{j-1,1}\,.\end{aligned}$$

We define $\mathsf{eC}(\mathcal{Q})_j$ and $\mathsf{eC}(\mathcal{Q})$ similar as before and write $\mathsf{e}(\mathcal{Q}) = \mathsf{eA}(\mathcal{Q}) \cup \mathsf{eB}(\mathcal{Q}) \cup \mathsf{eC}(\mathcal{Q})$. $\mathsf{eC}(\mathcal{Q})$ essentially covers the case that somewhere in the evaluation of S^r a fork $(y \oplus z)_{j-1,0} \oplus (y \oplus z)_{j-1,1} = x_{j,b}$ is completed by an upper-level query. It could essentially also be the case that $k'' > k, k'$ but a new query results in a fresh element in $\mathcal{Z}_{j-1,0}$, therewith rendering a hit, but in this case the query would invalidate $\mathsf{eC}(\mathcal{Q})$ in the first place (for an earlier value of j). Intuitively, assuming $\neg\mathsf{eC}(\mathcal{Q}_q)$, we can indeed consider the adversary to make its queries layer-wise.

Lemma 2 still holds, and

$$\mathbb{P}\left[\mathsf{col}S^r(\mathcal{Q}_q)\right] \leq \mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)\right] \leq \mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right]\,.$$

A bound on this probability is derived in Lem. 4. It is similar to Lem.3.

Lemma 4. $\mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right] \leq \frac{2(\tau^\ell q)^2}{N} + 2N\left(\frac{e(\tau^\ell q)^2}{N}\right)^\tau + \frac{2\tau^{2\ell}q^3}{N}$.

Proof. Recall the notation $\mathsf{e}(\mathcal{Q}_q) = \mathsf{eA}(\mathcal{Q}_q) \cup \mathsf{eB}(\mathcal{Q}_q) \cup \mathsf{eC}(\mathcal{Q}_q)$. Write $\mathsf{eBC}(\mathcal{Q}_q) = \mathsf{eB}(\mathcal{Q}_q) \cup \mathsf{eC}(\mathcal{Q}_q)$. By basic probability theory:

$$\begin{aligned}\mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right] \leq &\sum_{j=0}^{\ell} \mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)_j \cap \neg\mathsf{eBC}(\mathcal{Q}_q)_j \cap \cap_{j'=1}^{j-1} \neg\mathsf{e}(\mathcal{Q}_q)_{j'}\right] + \\ &\sum_{j=1}^{\ell} \mathbb{P}\left[\mathsf{eB}(\mathcal{Q}_q)_j \cap \cap_{j'=1}^{j-1} \neg\mathsf{e}(\mathcal{Q}_q)_{j'}\right] + \mathbb{P}\left[\mathsf{eC}(\mathcal{Q}_q)_j \cap \cap_{j'=1}^{j-1} \neg\mathsf{e}(\mathcal{Q}_q)_{j'}\right].\end{aligned} \tag{3}$$

Indeed, $\mathsf{e}(\mathcal{Q}_q)$ should be triggered for some j. Therefore, we consider any j, assume $\mathsf{e}(\mathcal{Q}_q)_{j'}$ has not been triggered for any $j' < j$, and consider the probability that a query for this specific value of j triggers $\mathsf{e}(\mathcal{Q})_j$. This event can then be further divided into a success for $\mathsf{eA}(\mathcal{Q}_q)_j$, $\mathsf{eB}(\mathcal{Q}_q)_j$, or $\mathsf{eC}(\mathcal{Q}_q)_j$.

$\mathsf{eA}(\mathcal{Q}_q)_j$. Assume $\neg\mathsf{eBC}(\mathcal{Q}_q)_j \cap \cap_{j'=1}^{j-1} \neg\mathsf{e}(\mathcal{Q}_q)_{j'}$ holds. Consider any b. The equation of $\mathsf{eA}(\mathcal{Q}_q)_j$ could get satisfied in two ways: (i) via a query to $f_{j,b}$, or (ii) via a query $(x, y)_{j',b'}$ for $j' < j$ that results in a new value in $\mathcal{Z}_{j,b}$ for any older query $(x, y)_{j,b}$. However, in case (ii) the query to $f_{j',b'}$ triggered $\mathsf{eC}(\mathcal{Q})_{j'+1}$, which is impossible by assumption. The remaining analysis is the same as in Lem. 3, and we obtain:

$$\mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)_j \cap \neg\mathsf{eBC}(\mathcal{Q}_q)_j \cap \cap_{j'=1}^{j-1} \neg\mathsf{e}(\mathcal{Q}_q)_{j'}\right] \leq \frac{(\tau^j q)^2}{N}.$$

$\mathsf{eB}(\mathcal{Q}_q)_j$. Assume $\cap_{j'=1}^{j-1} \neg\mathsf{e}(\mathcal{Q}_q)_{j'}$ holds. Consider any $z \in \{0, 1\}^n$. The equation of $\mathsf{eB}(\mathcal{Q}_q)_j$ could get satisfied in two ways: (i) via a query $(x, y, \mathcal{Z})_{j-1,0}$ or $(x, y, \mathcal{Z})_{j-1,1}$, or (ii) via a query $(x, y, \mathcal{Z})_{j',b'}$ for $j' < j - 1$ that results in a new value in either $\mathcal{Z}_{j-1,0}$ for any older query $(x, y)_{j-1,0}$ or $\mathcal{Z}_{j-1,1}$ for any older query $(x, y)_{j-1,1}$. However, in case (ii) the query to $f_{j',b'}$ triggered $\mathsf{eC}(\mathcal{Q})_{j'+1}$, which is impossible by assumption. Therefore, it suffices to consider the case a fresh query to $f_{j-1,0}$ or $f_{j-1,1}$ makes the equation satisfied. The remaining analysis is the same as in Lem. 3, and we obtain:

$$\mathbb{P}\left[\mathsf{eB}(\mathcal{Q}_q)_j \cap \cap_{j'=1}^{j-1} \neg\mathsf{e}(\mathcal{Q}_q)_{j'}\right] \leq N\left(\frac{e(\tau^j q)^2}{N}\right)^\tau.$$

$\mathsf{eC}(\mathcal{Q}_q)_j$. Assume $\cap_{j'=1}^{j-1} \neg\mathsf{e}(\mathcal{Q}_q)_{j'}$ holds. Similar to $\mathsf{eB}(\mathcal{Q}_q)_j$, by assumption the equation of $\mathsf{eC}(\mathcal{Q}_q)_j$ could only be triggered via a query $(x, y, \mathcal{Z})_{j-1,0}$ or $(x, y, \mathcal{Z})_{j-1,1}$. Any query $(x, y, \mathcal{Z})_{j-1,0}$ has $|\mathcal{Z}_{j-1,0}| \leq \tau^{j-1}$, due to $\neg\mathsf{eB}(\mathcal{Q}_q)_{j-1}$, and similar for any $(x, y, \mathcal{Z})_{j-1,1}$.

Consider the $\max\{k, k'\}$th query. There are at most q^2 choices for the other two queries, and it adds a solution to $\mathsf{eC}(\mathcal{Q}_q)_{j,b}$ with probability at most $\tau^{2(j-1)}q^2/N$. Summing over all queries, we eventually find:

$$\mathbb{P}\left[\mathsf{eC}(\mathcal{Q}_q)_j \cap \cap_{j'=1}^{j-1} \neg\mathsf{e}(\mathcal{Q}_q)_{j'}\right] \leq \frac{\tau^{2j} q^3}{N}.$$

Conclusion of proof. The proof is now completed via (3), as in Lem. 3. □

4.4 S^r for Arbitrary r

The previous analysis carries over to the generalized S^r (Fig. 3) almost verbatim, with the difference that we take $\ell = \lceil \log_2 r \rceil$. The only technical change lies in the sets $\mathcal{Z}_{j,b}$ associated to the queries: a query to $f_{j,b}$ may occur in an evaluation of S^r at position $(j, b + 2\lambda)$ for $\lambda \in \{0, \ldots, 2^{\ell-j-1} - 1\}$, and due to the asymmetric character of S^r it may have two different feed-forward sets. For the proof of Lem. 3 this concretely means that we need to consider two feed-forward sets associated with every query. This affects the bound as follows: regarding $\mathsf{eA}(\mathcal{Q}_q)_j$ we end up with bound $\frac{(2\tau^j q)^2}{N}$. For $\mathsf{eB}(\mathcal{Q})_j$, a collision is found with probability at most $(2\tau^{j-1})^2 q/N$ and any hit adds at most $2\tau^{j-1}$ values. Using $\tau \geq 2$, this results in the same bound for $\mathsf{eB}(\mathcal{Q})_j$:

$$\mathbb{P}\left[\mathsf{eB}(\mathcal{Q}_q)_j \cap \neg \mathsf{e}(\mathcal{Q}_q)_{j-1}\right] \leq N \left(\frac{e(2^3 \tau^{2j-3} q)^2}{N}\right)^{\tau} \leq N \left(\frac{e(\tau^j q)^2}{N}\right)^{\tau} .$$

Hence, as a direct corollary of Thm. 1 and Thm. 2 (to which the same reasoning applies), we find:

Corollary 1. *Let $r \geq 1$ with $\ell = \lceil \log_2 r \rceil$. Let $\mathcal{A}^{\mathsf{lw}}(q)$ be as in Thm. 1, and $\mathcal{A}(q)$ as in Thm. 2. Then, for any positive integer value $\tau \geq 2$,*

$$\mathsf{Adv}^{\mathsf{col}}_{S^r}[\mathcal{A}^{\mathsf{lw}}(q)] \leq \frac{8(\tau^\ell q)^2}{N} + 2N \left(\frac{e(\tau^\ell q)^2}{N}\right)^{\tau} ,$$
$$\mathsf{Adv}^{\mathsf{col}}_{S^r}[\mathcal{A}(q)] \leq \frac{8(\tau^\ell q)^2}{N} + 2N \left(\frac{e(\tau^\ell q)^2}{N}\right)^{\tau} + \frac{8\tau^{2\ell} q^3}{N} .$$

The asymptotic behavior of the bounds remains the same.

5 Preimage Security of S^r

Theorem 1 trivially implies preimage security up to the birthday bound. For $r = 1, 2$, we derive the following result in the fully adaptive model. This result particularly solves an open problem of Shrimpton and Stam [45], namely to prove $2n/3$-bit preimage security of their design (optimal w.r.t. the bounds of Rogaway and Steinberger [41]).

Theorem 3. *Let $r \in \{1, 2\}$. Let $\mathcal{A}(q)$ be as in Thm. 2. Then, for any positive integer value $\tau \geq 2$,*

$$\mathsf{Adv}^{\mathsf{epre}}_{S^1}[\mathcal{A}(q)] \leq \frac{q}{N}, \qquad \mathsf{Adv}^{\mathsf{epre}}_{S^2}[\mathcal{A}(q)] \leq \frac{\tau q}{N} + \frac{q^3}{N^2} + (N+2)\left(\frac{2eq^2}{\tau N}\right)^{\tau/2} .$$

The proof is given in Sect. 5.1. Similar to Thm. 1, we can put $\tau = N^{1/3}$, and find that for any $\varepsilon > 0$, $\mathsf{Adv}^{\mathsf{epre}}_{S^2}[\mathcal{A}(N^{2/3}/n^{\varepsilon})]$ approaches 0 for $n \to \infty$. Unfortunately, the proof cannot be easily generalized to larger r: the threshold value τ^ℓ starts exploding for $\ell \geq 2$.

We remark that S^r (or MD-S^r) for $r \geq 2$ can be made preimage resistant up to N queries by adding one single primitive call at the end of its evaluation (now only for S^r, a similar claim for MD-S^r was already made in Sect. 1):

Theorem 4. *Let $r \geq 1$ with $\ell = \lceil \log_2 r \rceil$. Suppose $\mathcal{P} = \big\{f_{j,b} \mid (j,b) \in (\{0, \ldots, \ell-1\} \times \{0,1\}) \cup \{(\ell, 0)\}\big\} \cup \{f\} \xleftarrow{\$} \mathsf{Func}(n)^{2\ell+2}$. Let $\mathcal{A}(q)$ be as in Thm. 2. Then,*

$$\mathsf{Adv}^{\mathsf{epre}}_{f \circ S^r}[\mathcal{A}(q)] \leq \frac{q}{N} .$$

Proof. The proof directly follows from Lem. 1 and Thm. 3, noting $S^1 = f$. □

5.1 Proof of Theorem 3

S^1 is equal to $f_{0,0}$ and the result is trivial. We proceed with S^2. We employ the same conventions and notations as in the proof of Sect. 4. As before, we consider any adversary A that has query access to its oracles $\mathcal{P}$ and makes q queries. Recall that $\mathcal{Z}_{0,b} = \{0\}$ for $b = 0, 1$. In fact, we only need $\mathcal{Z}_{1,0}$ and therefore we will discard the notion entirely (and write everything explicitly).

Let $v \in \{0,1\}^n$ be the challenged range value. Denote by $\mathsf{pre}S^2(\mathcal{Q}_q)$ the event that A finds an evaluation of S^2 satisfying $S^2(u_0, u_1) = v$. By definition:

$$\mathsf{Adv}^{\mathsf{epre}}_{S^2}[A] = \mathbb{P}\left[\mathsf{pre}S^2(\mathcal{Q}_q)\right] . \tag{4}$$

For the analysis of $\mathbb{P}\left[\mathsf{pre}S^2(\mathcal{Q}_q)\right]$ we introduce four helping events. Here, let $\tau \geq 2$ be any integer value.

$$\begin{aligned}
\mathsf{eA}(\mathcal{Q})_{j,b} :\ & \exists \text{ distinct } (x,y)_{j,b}, (x',y')_{j,b}, (x'',y'')_{j,b} \in \mathcal{Q}_q \text{ such that} \\
& \qquad\qquad y_{j,b} = y'_{j,b} = y''_{j,b} \text{ or } (x \oplus y)_{j,b} = (x' \oplus y')_{j,b} = (x'' \oplus y'')_{j,b} \,; \\
\mathsf{eB}(\mathcal{Q}) :\ & \max_{z \in \{0,1\}^n} \left|\left\{(x,y)_{0,0}, (x,y)_{0,1} \in \mathcal{Q}_q \;\middle|\; y_{0,0} \oplus y_{0,1} = z\right\}\right| > \tau \,; \\
\mathsf{eC}(\mathcal{Q}) :\ & \left|\left\{(x,y)_{0,1}, (x,y)_{1,0} \in \mathcal{Q}_q \;\middle|\; y_{0,1} \oplus (x \oplus y)_{1,0} = v\right\}\right| > \tau \,; \\
\mathsf{eD}(\mathcal{Q}) :\ & \left|\left\{(x,y)_{0,0}, (x,y)_{1,0} \in \mathcal{Q}_q \;\middle|\; y_{0,0} \oplus y_{1,0} = v\right\}\right| > \tau \,.
\end{aligned}$$

We write $\mathsf{eA}(\mathcal{Q}) = \mathsf{eA}(\mathcal{Q})_{0,0} \cup \mathsf{eA}(\mathcal{Q})_{0,1} \cup \mathsf{eA}(\mathcal{Q})_{1,0}$, $\mathsf{eBCD}(\mathcal{Q}) = \mathsf{eB}(\mathcal{Q}) \cup \mathsf{eC}(\mathcal{Q}) \cup \mathsf{eD}(\mathcal{Q})$, and $\mathsf{e}(\mathcal{Q}) = \mathsf{eA}(\mathcal{Q}) \cup \mathsf{eBCD}(\mathcal{Q})$. In Lem. 5, we demonstrate that finding a preimage assuming $\neg\mathsf{eBCD}(\mathcal{Q}_q)$ happens with probability at most $\frac{\tau q}{N}$.

Lemma 5. $\mathbb{P}\left[\mathsf{pre}S^2(\mathcal{Q}_q) \cap \neg\mathsf{eBCD}(\mathcal{Q}_q)\right] \leq \dfrac{\tau q}{N}$.

Proof. Assume $\neg\mathsf{eBCD}(\mathcal{Q}_q)$. We make a distinction among queries made to $f_{0,0}$, $f_{0,1}$, and $f_{1,0}$.

Starting with a query $(x,y)_{1,0}$ to $f_{1,0}$, it renders a preimage for S^2 if $y_{1,0} \oplus y_{0,0} = v$ for some older queries $(x,y)_{0,0}, (x,y)_{0,1}$ satisfying $y_{0,0} \oplus y_{0,1} = x_{1,0}$. By $\neg\mathsf{eB}(\mathcal{Q}_q)$, there are at most τ such solutions. Consequently, the query results in a preimage with probability at most $\frac{\tau}{N}$.

Next, for a query $(x,y)_{0,0}$ to $f_{0,0}$, it results in a preimage if $y_{1,0} \oplus y_{0,0} = v$ and $y_{0,0} \oplus y_{0,1} = x_{1,0}$ for some older queries $(x,y)_{0,1}, (x,y)_{1,0}$. By $\neg\mathsf{eC}(\mathcal{Q}_q)$, there are at most τ solutions to $y_{1,0} \oplus v = y_{0,1} \oplus x_{1,0}$. Consequently, the query results in a preimage with probability at most $\frac{\tau}{N}$.

Finally, for a query $(x,y)_{0,1}$ to $f_{0,1}$, it gives a preimage if $y_{0,0} \oplus y_{0,1} = x_{1,0}$ for some older queries $(x,y)_{0,0}, (x,y)_{1,0}$ satisfying $y_{1,0} \oplus y_{0,0} = v$. By $\neg\mathsf{eD}(\mathcal{Q}_q)$, there are at most τ such solutions. Consequently, the query results in a preimage with probability at most $\frac{\tau}{N}$.

Summing over all queries, we obtain our bound. □

Therefore, we obtain for (4):

$$\mathbb{P}\left[\mathsf{pre}S^2(\mathcal{Q}_q)\right] \leq \frac{\tau q}{N} + \mathbb{P}\left[\mathsf{eBCD}(\mathcal{Q}_q)\right] \leq \frac{\tau q}{N} + \mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right] .$$

A bound on $\mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right]$ is derived in Lem. 6.

Lemma 6. $\mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right] \leq \dfrac{q^3}{N^2} + (N+2)\left(\dfrac{2eq^2}{\tau N}\right)^{\tau/2}$.

Proof. By basic probability theory:

$$\mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right] \le \mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)\right] + \mathbb{P}\left[\mathsf{eB}(\mathcal{Q}_q) \cap \neg\mathsf{eA}(\mathcal{Q}_q)\right] + \mathbb{P}\left[\mathsf{eC}(\mathcal{Q}_q) \cap \neg\mathsf{eA}(\mathcal{Q}_q)\right] + \mathbb{P}\left[\mathsf{eD}(\mathcal{Q}_q) \cap \neg\mathsf{eA}(\mathcal{Q}_q)\right]. \tag{5}$$

We consider the four probabilities separately.

$\mathsf{eA}(\mathcal{Q}_q)$. Consider any j, b and any three distinct queries $(x, y)_{j,b}$, $(x', y')_{j,b}$, and $(x'', y'')_{j,b}$ to $f_{j,b}$ (at most $\binom{q}{3}$ choices). These queries render a solution if $y_{j,b} = y'_{j,b} = y''_{j,b}$ or $(x \oplus y)_{j,b} = (x' \oplus y')_{j,b} = (x'' \oplus y'')_{j,b}$, which happens with probability at most $\frac{2}{N^2}$. Summing over all queries to $f_{j,b}$, and all choices of j, b (three in total), we obtain:

$$\mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)\right] \le \frac{q^3}{N^2}.$$

$\mathsf{eB}(\mathcal{Q}_q)$. Assume $\neg\mathsf{eA}(\mathcal{Q}_q)$ holds. Consider any $z \in \{0,1\}^n$. Without loss of generality (by symmetry) consider a new query $(x, y)_{0,1}$. This adds a solution to $\mathsf{eB}(\mathcal{Q}_q)$ with probability at most q/N, and any hit adds at most 2 solutions (by $\neg\mathsf{eA}(\mathcal{Q}_q)$). More than τ solutions are added with probability at most $\binom{q}{\tau/2}\left(\frac{q}{N}\right)^{\tau/2} \le \left(\frac{2eq^2}{\tau N}\right)^{\tau/2}$. Summing over all N values z, we obtain:

$$\mathbb{P}\left[\mathsf{eB}(\mathcal{Q}_q) \cap \neg\mathsf{eA}(\mathcal{Q}_q)\right] \le N\left(\frac{2eq^2}{\tau N}\right)^{\tau/2}.$$

$\mathsf{eC}(\mathcal{Q}_q)$ and $\mathsf{eD}(\mathcal{Q}_q)$. The analysis is similar to $\mathsf{eB}(\mathcal{Q}_q)$ except that there is no need to sum over all values z.

Conclusion of proof. The proof is now completed via (5). □

6 Instantiation Using Permutations

In this section, we discuss the effect of instantiating S^r with random permutations $\pi_{j,b}$ instead of random functions $f_{j,b}$. The most basic and well-established way is to set $f_{j,b}(x) = \pi_{j,b}(x) \oplus x$, and we consider S^r with its underlying primitives transformed this way. For ease of presentation, we focus on S^r for $r = 2^\ell$ with $\ell \ge 0$, but the findings carry over to the general setting.

For the top layer of S^r, the feed-forward of x in $f_{0,b}$ is necessary: in absence of it, it suffices for an adversary to find a collision with these calls eliminated and to just make r inverse calls afterwards. However, for the remaining evaluations of $f_{j,b}$ (with $j \ge 1$), the feed-forward x is pointless: it simply corresponds to reflecting S^r along its vertical axis.[4] Thus, we focus on S^r with $f_{0,b}(x) = \pi_{0,b}(x) \oplus x$ (for $b \in \{0,1\}$) and $f_{j,b}(x) = \pi_{j,b}(x)$ (for $(j, b) \in (\{1, \ldots, \ell-1\} \times \{0,1\}) \cup \{(\ell, 0)\}$), where $\left\{\pi_{j,b} \mid (j,b) \in (\{0, \ldots, \ell-1\} \times \{0,1\}) \cup \{(\ell,0)\}\right\} \xleftarrow{\$} \mathsf{Perm}(n)^{2\ell+1}$. A formal description is given in Fig. 5.

Starting with collision resistance, we transform Thm. 1 to the permutation-based setting.

Theorem 5. *Let $r = 2^\ell$ with $\ell \ge 0$. Suppose $\mathcal{P} = \left\{\pi_{j,b} \mid (j,b) \in (\{0, \ldots, \ell-1\} \times \{0,1\}) \cup \{(\ell,0)\}\right\} \xleftarrow{\$} \mathsf{Perm}(n)^{2\ell+1}$. Let $\mathcal{A}^{\mathsf{lw}}(q)$ be as in Thm. 1. Then, for any positive integer value $\tau \ge 3$,*

$$\mathsf{Adv}^{\mathsf{col}}_{S^r}[\mathcal{A}^{\mathsf{lw}}(q)] \le \frac{4(\tau^\ell q)^2}{N-q} + 8N\left(\frac{e(\tau^\ell q)^2}{N-q}\right)^{\tau^{1/2}-1}.$$

[4] In more detail, in S^r of Fig. 2, the feed-forward $(y \oplus z)_{j-1,2i}$ in round (j, i) would be replaced by $(y \oplus z)_{j-1,2i+1}$.

```
S^r for r = 2^ℓ with ℓ ≥ 0
1: procedure $S^r(u_0, \ldots, u_{r-1})$
2:     for $i = 0, \ldots, 2^\ell - 1$ do
3:         $x_{0,i} \leftarrow u_i$
4:         $y_{0,i} \leftarrow \pi_{0,i \bmod 2}(x_{0,i})$
5:         $z_{0,i} \leftarrow x_{0,i}$
6:     for $j = 1, \ldots, \ell$ and $i = 0, \ldots, 2^{\ell-j} - 1$ do
7:         $x_{j,i} \leftarrow (y \oplus z)_{j-1,2i} \oplus (y \oplus z)_{j-1,2i+1}$
8:         $y_{j,i} \leftarrow \pi_{j,i \bmod 2}(x_{j,i})$
9:         $z_{j,i} \leftarrow (y \oplus z)_{j-1,2i}$
10:    return $v \leftarrow (y \oplus z)_{\ell,0}$
```

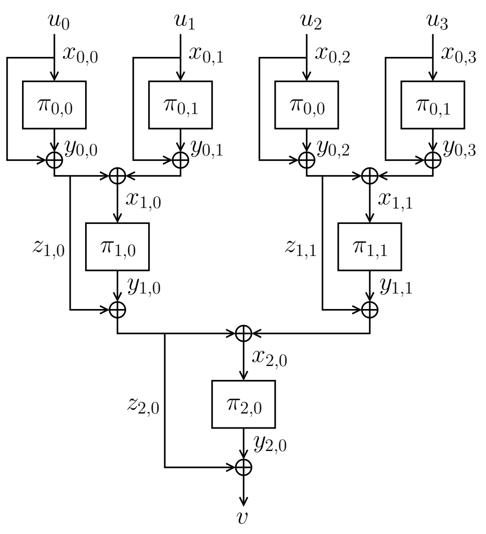


Fig. 5: Alternative permutation-based description of S^r of Fig. 2 and an illustration of S^4.

The proof is in the same spirit as the one of Thm. 1 but is technically more demanding and is included in App. B. Theorem 2 (for fully adaptive adversaries) and Cor. 1 (for arbitrary $r \geq 1$) generalize in a similar way.

Unfortunately, the preimage result of Thm. 3 does not carry over to the permutation-based case: a preimage for S^2 can be found in approximately $2^{n/2}$ queries [31, 45]. In Thm. 6 we generalize the attack to the case $r = 2^\ell$ with $\ell \geq 1$; it generalizes to arbitrary $r \geq 2$ the obvious way.

Theorem 6. *Let $r = 2^\ell$ with $\ell \geq 1$. Suppose $\mathcal{P} = \{\pi_{j,b} \mid (j,b) \in (\{0,\ldots,\ell-1\} \times \{0,1\}) \cup \{(\ell,0)\}\} \xleftarrow{\$} \mathsf{Perm}(n)^{2\ell+1}$. Let $\mathcal{A}((2r-1)q)$ denote the set of all adversaries that make at most $(2r-1)q$ queries. Then,*

$$\mathsf{Adv}^{\mathsf{epre}}_{S^r}[\mathcal{A}((2r-1)q)] \geq \frac{q^2}{N}\,.$$

Proof. Let v be any given range value. We consider the following adversary. First, for $k = 1, \ldots, q$, it randomly selects a tuple $(u_0, \ldots, u_{r/2-1})^k$, computes the left half of S^r up to $(y \oplus z)^k_{\ell-1,0}$, and queries $x^k_{\ell,0} \leftarrow \pi^{-1}_{\ell,0}(v \oplus (y \oplus z)^k_{\ell-1,0})$. Next, for $k' = 1, \ldots, q$, it randomly selects a tuple $(u_{r/2+1}, \ldots, u_r)^{k'}$ and computes the right half of S^r up to $(y \oplus z)^{k'}_{\ell-1,1}$. A preimage for S^r is found if there exist $k, k' \in \{1, \ldots, q\}$ such that

$$(y \oplus z)^k_{\ell-1,0} \oplus x^k_{\ell,0} = (y \oplus z)^{k'}_{\ell-1,1}\,.$$

This happens with probability at least $\frac{q^2}{N}$. □

Nevertheless, the trick of applying a postprocessing $\pi(x) \oplus x$ to S^r to get 2^n preimage security (cf. Thm. 4) still applies.

7 Conclusion

Our generalized S^r compression function design achieves high efficiency, approaching rate 1/2 using primitives of state size. The function can be used in a Merkle-Damgård mode of operation or in a Merkle tree. Compared with recent designs such as Sponge functions, Grøstl, and MD6, S^r achieves asymptotically the same collision security and offers comparable rates, but using primitives that are at least twice as small, saving a significant amount of computational overhead. However, we acknowledge that S^r uses more distinct primitives. Depending on the application, S^r may be more suitable than the other schemes, and it complements well to these designs.

S^r can be securely instantiated using non-compressing one-way functions or permutations, but our targeted generality comes at a technical price: the asymptotic $n/2$-bit collision security is only proven in a setting where the adversary is limited to making its queries layer-wise. Although we present a proof in the fully adaptive model up to $2^{n/3}$ queries, we expect this bound to be non-optimal and conjecture (almost) optimal collision security of S^r in the adaptive model.

The proof of S^r requires $2\lceil\log_2 r\rceil + 1$ distinct primitives: one primitive for the last layer, and two for every but last layer. While this requirement has a partly technical cause, it is far from trivial to analyze S^r with less distinct primitives. For instance, in the permutation-based setting (see Fig. 5), S^2 and S^3 are insecure if $\pi_{\ell,0} = \pi_{\ell-1,1}$: putting $x_{\ell-1,1} = \pi^{-1}_{\ell-1,1}((y \oplus z)_{\ell-1,0})$ yields hash value 0, for arbitrary $(y \oplus z)_{\ell-1,0}$. It is unclear how these observations generalize to larger S^r (based on one-way functions or permutations), and this remains an interesting open research problem.

Acknowledgments. This work was supported in part by the Research Fund KU Leuven, OT/13/071, and in part by the Research Council KU Leuven: GOA TENSE (GOA/11/007). Bart Mennink is a Postdoctoral Fellow of the Research Foundation – Flanders (FWO).

References

[1] Alizadeh, J., Aref, M., Bagheri, N.: Artemia v1 (2014), submission to CAESAR competition

[2] Andreeva, E., Bilgin, B., Bogdanov, A., Luykx, A., Mendel, F., Mennink, B., Mouha, N., Wang, Q., Yasuda, K.: PRIMATEs v1 (2014), submission to CAESAR competition

[3] Andreeva, E., Mennink, B., Preneel, B.: Security reductions of the second round SHA-3 candidates. In: ISC 2010. Lecture Notes in Computer Science, vol. 6531, pp. 39–53. Springer, Heidelberg (2010)

[4] Andreeva, E., Mennink, B., Preneel, B.: The parazoa family: Generalizing the sponge hash functions. International Journal of Information Security 11(3), 149–165 (2012)

[5] Aumasson, J., Jovanovic, P., Neves, S.: NORX v1 (2014), submission to CAESAR competition

[6] Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the indifferentiability of the sponge construction. In: EUROCRYPT 2008. Lecture Notes in Computer Science, vol. 4965, pp. 181–197. Springer, Heidelberg (2008)

[7] Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sufficient conditions for sound tree and sequential hashing modes. Cryptology ePrint Archive, Report 2009/210 (2009)

[8] Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The KECCAK sponge function family (2011), submission to NIST's SHA-3 competition

[9] Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge functions (ECRYPT Hash Function Workshop 2007)

[10] Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., Van Keer, R.: Ketje v1 (2014), submission to CAESAR competition

[11] Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., Van Keer, R.: Keyak v1 (2014), submission to CAESAR competition

[12] Black, J., Cochran, M., Shrimpton, T.: On the impossibility of highly-efficient blockcipher-based hash functions. In: EUROCRYPT 2005. Lecture Notes in Computer Science, vol. 3494, pp. 526–541. Springer, Heidelberg (2005)

[13] Black, J., Rogaway, P., Shrimpton, T.: Black-box analysis of the block-cipher-based hash-function constructions from PGV. In: CRYPTO 2002. Lecture Notes in Computer Science, vol. 2442, pp. 320–335. Springer, Heidelberg (2002)

[14] Black, J., Rogaway, P., Shrimpton, T., Stam, M.: An analysis of the blockcipher-based hash functions from PGV. Journal of Cryptology 23(4), 519–545 (2010)

[15] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness (April 2014), `http://competitions.cr.yp.to/caesar.html`

[16] Damgård, I.: A design principle for hash functions. In: CRYPTO '89. Lecture Notes in Computer Science, vol. 435, pp. 416–427. Springer, Heidelberg (1990)

[17] Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1 (2014), submission to CAESAR competition

[18] Dodis, Y., Reyzin, L., Rivest, R., Shen, E.: Indifferentiability of permutation-based compression functions and tree-based modes of operation, with applications to MD6. In: FSE 2009. Lecture Notes in Computer Science, vol. 5665, pp. 104–121. Springer, Heidelberg (2009)

[19] Dodis, Y., Steinberger, J.P.: Domain extension for MACs beyond the birthday barrier. In: EUROCRYPT 2011. Lecture Notes in Computer Science, vol. 6632, pp. 323–342. Springer, Heidelberg (2011)

[20] Duo, L., Li, C.: Improved collision and preimage resistance bounds on PGV schemes. Cryptology ePrint Archive, Report 2006/462 (2006)

[21] Gauravaram, P., Knudsen, L., Matusiewicz, K., Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.: Grøstl – a SHA-3 candidate (2011), submission to NIST's SHA-3 competition

[22] Gligoroski, D., Mihajloska, H., Samardjiska, S., Jacobsen, H., El-Hadedy, M., Jensen, R.: π-Cipher v1 (2014), submission to CAESAR competition

[23] Hirose, S.: Some plausible constructions of double-block-length hash functions. In: FSE 2006. Lecture Notes in Computer Science, vol. 4047, pp. 210–225. Springer, Heidelberg (2006)

[24] Jetchev, D., Özen, O., Stam, M.: Collisions are not incidental: A compression function exploiting discrete geometry. In: TCC 2012. Lecture Notes in Computer Science, vol. 7194, pp. 303–320. Springer, Heidelberg (2012)

[25] Kavun, E., Lauridsen, M., Leander, G., Rechberger, C., Schwabe, P., Yalçın, T.: Prøst v1 (2014), submission to CAESAR competition

[26] Knudsen, L., Lai, X., Preneel, B.: Attacks on fast double block length hash functions. Journal of Cryptology 11(1), 59–72 (1998)

[27] Knudsen, L., Rechberger, C., Thomsen, S.: The Grindahl hash functions. In: FSE 2007. Lecture Notes in Computer Science, vol. 4593, pp. 39–57. Springer, Heidelberg (2007)

[28] Lai, X., Massey, J.: Hash function based on block ciphers. In: EUROCRYPT '92. Lecture Notes in Computer Science, vol. 658, pp. 55–70. Springer, Heidelberg (1992)

[29] Lee, J., Kwon, D.: Security of single-permutation-based compression functions. Cryptology ePrint Archive, Report 2009/145 (2009)

[30] Maurer, U.M., Tessaro, S.: Domain extension of public random functions: Beyond the birthday barrier. In: CRYPTO 2007. Lecture Notes in Computer Science, vol. 4622, pp. 187–204. Springer, Heidelberg (2007)

[31] Mennink, B., Preneel, B.: Hash functions based on three permutations: A generic security analysis. In: CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 330–347. Springer, Heidelberg (2012)

[32] Merkle, R.: Protocols for public key cryptosystems. In: IEEE Symposium on Security and Privacy. pp. 122–134. IEEE Computer Society Press (1980)

[33] Merkle, R.: One way hash functions and DES. In: CRYPTO '89. Lecture Notes in Computer Science, vol. 435, pp. 428–446. Springer, Heidelberg (1990)

[34] Meyer, C., Schilling, M.: Secure program load with manipulation detection code. In: Proc. Securicom. pp. 111–130 (1988)

[35] Morawiecki, P., Gaj, K., Homsirikamol, E., Matusiewicz, K., Pieprzyk, J., Rogawski, M., Srebrny, M., Wójcik, M.: ICEPOLE v1 (2014), submission to CAESAR competition

[36] National Institute for Standards and Technology: Announcing request for candidate algorithm nominations for a new cryptographic hash algorithm (SHA3) family (November 2007)

[37] Peyrin, T., Gilbert, H., Muller, F., Robshaw, M.: Combining compression functions and block cipher-based hash functions. In: ASIACRYPT 2006. Lecture Notes in Computer Science, vol. 4284, pp. 315–331. Springer, Heidelberg (2006)

[38] Preneel, B., Govaerts, R., Vandewalle, J.: Hash functions based on block ciphers: A synthetic approach. In: CRYPTO '93. Lecture Notes in Computer Science, vol. 773, pp. 368–378. Springer, Heidelberg (1993)

[39] Rivest, R., Agre, B., Bailey, D.V., Crutchfield, C., Dodis, Y., Fleming, K.E., Khan, A., Krishnamurthy, J., Lin, Y., Reyzin, L., Shen, E., Sukha, J., Sutherland, D., Tromer, E., Yin, Y.L.: The MD6 hash function – A proposal to NIST for SHA-3 (2008), submission to NIST's SHA-3 competition

[40] Rogaway, P., Shrimpton, T.: Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In: FSE 2004. Lecture Notes in Computer Science, vol. 3017, pp. 371–388. Springer, Heidelberg (2004)

[41] Rogaway, P., Steinberger, J.: Security/efficiency tradeoffs for permutation-based hashing. In: EUROCRYPT 2008. Lecture Notes in Computer Science, vol. 4965, pp. 220–236. Springer, Heidelberg (2008)

[42] Rogaway, P., Steinberger, J.P.: Constructing cryptographic hash functions from fixed-key blockciphers. In: CRYPTO 2008. Lecture Notes in Computer Science, vol. 5157, pp. 433–450. Springer, Heidelberg (2008)

[43] Saarinen, M.: CBEAM r1 (2014), submission to CAESAR competition

[44] Saarinen, M.: STRIBOB r1 (2014), submission to CAESAR competition

[45] Shrimpton, T., Stam, M.: Building a collision-resistant compression function from non-compressing primitives. In: ICALP (2) 2008. Lecture Notes in Computer Science, vol. 5126, pp. 643–654. Springer, Heidelberg (2008)
[46] Stam, M.: Beyond uniformity: Better security/efficiency tradeoffs for compression functions. In: CRYPTO 2008. Lecture Notes in Computer Science, vol. 5157, pp. 397–412. Springer, Heidelberg (2008)
[47] Stam, M.: Blockcipher-based hashing revisited. In: FSE 2009. Lecture Notes in Computer Science, vol. 5665, pp. 67–83. Springer, Heidelberg (2009)
[48] Steinberger, J.: Stam's collision resistance conjecture. In: EUROCRYPT 2010. Lecture Notes in Computer Science, vol. 6110, pp. 597–615. Springer, Heidelberg (2010)
[49] Steinberger, J., Sun, X., Yang, Z.: Stam's conjecture and threshold phenomena in collision resistance. In: CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 384–405. Springer, Heidelberg (2012)
[50] Wu, H.: The Hash Function JH (2011), submission to NIST's SHA-3 competition

A Comparison of S^r With Known Permutation Based Hash Functions

We present technical support for Table 1 of Sect. 1. In more detail, we compute the rates and primitive sizes of Sponge functions, Grøstl, and MD6, and of S^r when in a Merkle-Damgård mode of operation or in a Merkle tree. As these functions are all defined for different modes of operation and different parameters, separate treatments are required. In our comparison, we target hash functions $\mathcal{H} : \{0,1\}^* \to \{0,1\}^n$ with $n/2$-bit security; we adopt the parameters for the specific designs alike, and derive the rates and the primitive sizes.

Note that for $\mathcal{H}$ we can define the rate similarly as in Sect. 1, namely as $\frac{M}{ds}$, where M denotes the total length of the message in bits, and d denotes the number of calls to the underlying s-bit non-compressing primitive f. We simplify the analysis by assuming that the message is always of full length (ignoring additional primitive calls due to padding, length strengthening, and final transformations). In the remainder of this section, we derive the rates and primitive sizes for the above-mentioned functions. A further comparison is given in Sect. 1.

Sponge Functions. Sponge functions [9] have a state of $c + m$ bits, where c is the capacity and m the message block size (see footnote 1). Sponge functions are collision resistant up to $2^{c/2}$ queries [6]. Hence, $n/2$-bit security means that we take $c = n$. Its compression function $F : \{0,1\}^{n+2m} \to \{0,1\}^{n+m}$ makes one primitive call to a $(n + m)$-bit permutation. On input of a message of M blocks of m bits, it makes M primitive calls, and thus has rate $\frac{m}{n+m}$ using primitives on $n+m$ bits. The same reasoning applies to Keccak (as it is a Sponge function) [8], Grindahl [27], JH [50], and parazoa functions [4].

Grøstl. Grøstl [21] is a Merkle-Damgård function. It has a state size of l bits and collision security is proven up to $2^{l/4}$ queries [3] (hence, we consider $l = 2n$). It employs a compression function $F : \{0,1\}^{2l} \to \{0,1\}^l$ making two primitive calls to two distinct l-bit permutations. On input of a message of M blocks of l bits, it makes $2M$ primitive calls, and thus has a rate of $1/2$ using primitives on $2n$ bits.

MD6. MD6 [39] is a tree-based hash function with output size 16 words of $w = 64$ bits (we write the output size as $n = 16w$). Collision security is proven up to $2^{n/2}$ queries [18]. It employs a compression function $F : \{0,1\}^{4n} \to \{0,1\}^n$ making one primitive call to a $4n$-bit permutation. (In fact, F and its underlying permutation get 25 additional words of input, but these are ignored for the sake of simplicity. Taking these into account leads to a worse rate.) On input of a message of M blocks of n bits (w.l.o.g. $M = 4^\alpha$ for some α), it makes $\frac{M-1}{3}$ primitive calls, and thus has a rate of $3/4\frac{M}{M-1}$ using primitives on $4n$ bits.

S^r in MD or MT. For the comparison with Sponge functions and Grøstl, we consider S^r in a Merkle-Damgård mode of operation (MD-S^r) [16,33], and for the comparison with MD6 we consider it in a Merkle tree (MT-S^r) [32]. Both MD-S^r and MT-S^r preserve collision resistance [7,16,18,33] when correctly padded and have about $2^{n/2}$ collision resistance. First consider MD-S^r. On input of a message of M blocks of n bits (w.l.o.g. $M = (r-1)\alpha$ for some α), it makes $\frac{(2r-1)M}{r-1}$ primitive calls, and thus has a rate of $\frac{r-1}{2r-1}$ using primitives on n bits. Next consider MT-S^r. On input of a message of M blocks of n bits (w.l.o.g. $M = r^\alpha$ for some α), it makes $\frac{M-1}{r-1}$ primitive calls, and thus has a rate of $\frac{r-1}{2r-1}\frac{M}{M-1}$ using primitives on n bits.

B Proof of Theorem 5

We consider the security of $S^r : \{0,1\}^{rn} \to \{0,1\}^n$, for $r = 2^\ell$ with $\ell \geq 0$, based on $2\ell+1$ functions $\{\pi_{j,b} \mid (j,b) \in (\{0,\ldots,\ell-1\} \times \{0,1\}) \cup \{(\ell,0)\}\}$ randomly drawn from $\mathsf{Perm}(n)$. See Fig. 5. In the proof we consider adversaries that make all queries to $\pi_{j-1,b}$ before all queries to $\pi_{j,b'}$.

The proof shows similarities with the proof of Thm. 1 but is more technical. As before, we associate to each query $(x_{j,b}, y_{j,b})$ a multiset $\mathcal{Z}_{j,b}$ of all possible feed-forward values $z_{j,b}$ occurring for this query. The only difference is that now for queries $(x_{0,b}, y_{0,b})$ we have $\mathcal{Z}_{0,b} = \{x_{0,b}\}$. We recall that $(x_{j,b}, y_{j,b}, \mathcal{Z}_{j,b}) = (x, y, \mathcal{Z})_{j,b}$, and again write $(y \oplus z)_{j,b} = y_{j,b} \oplus z_{j,b}$ and similarly $(x \oplus y)_{j,b} = x_{j,b} \oplus y_{j,b}$.

In the proof we employ four helping events. Here, let $\tau \geq 3$ be any integer value.

$$\begin{aligned}
\mathsf{eA}(\mathcal{Q})_{j,b} :\ & \exists\, (x,y,\mathcal{Z})_{j,b}, (x',y',\mathcal{Z}')_{j,b} \in \mathcal{Q}_q \text{ such that} \\
& \qquad\qquad x_{j,b} \neq x'_{j,b} \wedge y_{j,b} \oplus y'_{j,b} \in \mathcal{Z}_{j,b} \oplus \mathcal{Z}'_{j,b}\,; \\
\mathsf{eB}(\mathcal{Q})_{j,b} :\ & \max_{z \in \{0,1\}^n} \left| \left\{ \begin{array}{l} (x,y,\mathcal{Z})_{j-1,0}, (x,y,\mathcal{Z})_{j,b} \in \mathcal{Q}_q \mid \\ \quad y_{j,b} \neq z \wedge y_{j,b} \oplus y_{j-1,0} \oplus z \in \mathcal{Z}_{j,b} \oplus \mathcal{Z}_{j-1,0} \end{array} \right\} \right| > \tau^{j+1/2}\,; \\
\mathsf{eC}(\mathcal{Q})_{j} :\ & \max_{z \in \{0,1\}^n \setminus \{0\}} \left| \left\{ \begin{array}{l} (x,y,\mathcal{Z})_{j-1,0}, (x',y',\mathcal{Z}')_{j-1,0} \in \mathcal{Q}_q \mid \\ \quad y_{j-1,0} \oplus y'_{j-1,0} \oplus z \in \mathcal{Z}_{j-1,0} \oplus \mathcal{Z}'_{j-1,0} \end{array} \right\} \right| > \tau^{j}\,; \\
\mathsf{eD}(\mathcal{Q})_{j} :\ & \max_{z \in \{0,1\}^n} \left| \left\{ \begin{array}{l} (x,y,\mathcal{Z})_{j-1,0}, (x,y,\mathcal{Z})_{j-1,1} \in \mathcal{Q}_q \mid \\ \quad y_{j-1,0} \oplus y_{j-1,1} \oplus z \in \mathcal{Z}_{j-1,0} \oplus \mathcal{Z}_{j-1,1} \end{array} \right\} \right| > \tau^{j}\,.
\end{aligned}$$

For $\mathsf{X},\mathsf{Y} \in \{\mathsf{A},\ldots,\mathsf{D}\}$, we simply write $\mathsf{eX}(\mathcal{Q})_j = \mathsf{eX}(\mathcal{Q})_{j,0} \cup \mathsf{eX}(\mathcal{Q})_{j,1}$, $\mathsf{eX}(\mathcal{Q}) = \bigcup_j \mathsf{eX}(\mathcal{Q})_j$, and $\mathsf{eXY}(\mathcal{Q}) = \mathsf{eX}(\mathcal{Q}) \cup \mathsf{eY}(\mathcal{Q})$. We furthermore write $\mathsf{e}(\mathcal{Q}) = \mathsf{eA}\cdots\mathsf{D}(\mathcal{Q})$. Note that $\mathsf{eA}(\mathcal{Q})_{j,b}$ is as in Thm. 1, and $\mathsf{eD}(\mathcal{Q})_j$ replaces $\mathsf{eB}(\mathcal{Q})_j$.

The threshold values for $\mathsf{eB}(\mathcal{Q})_{j,b}$ on one hand and $\mathsf{eCD}(\mathcal{Q})_j$ on the other hand differ. This has a technical origin, and we present a brief and informal explanation. Write the two bounds as $\tau\mathsf{B}_j$ and $\tau\mathsf{CD}_j$. As will become clear in the proof of Lem. 7, any adversarial query adds at most $\tau\mathsf{B}_{j-1}$ solutions to $\mathsf{eC}(\mathcal{Q})_j$ and $\mathsf{eD}(\mathcal{Q})_j$, and at most $\tau\mathsf{CD}_j$ solutions to $\mathsf{eB}(\mathcal{Q})_{j,b}$. Hence, in order for our proof to make sense, we require $\tau\mathsf{B}_j > \tau\mathsf{CD}_j > \tau\mathsf{B}_{j-1}$, which justifies the choice of $\tau\mathsf{B}_j = \tau^{j+1/2}$ and $\tau\mathsf{CD}_j = \tau^j$.

The definition $\mathsf{col}S^r(\mathcal{Q}_q)$ of the proof of Thm. 1 carries over to the permutation-based setting, as well as Lem. 2. We obtain:

$$\mathsf{Adv}^{\mathsf{col}}_{S^r}[A] = \mathbb{P}\left[\mathsf{col}S^r(\mathcal{Q}_q)\right] \leq \mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)\right] \leq \mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right]\,.$$

A bound on this probability is derived in Lem. 7.

Lemma 7. $\mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right] \leq \dfrac{4(\tau^\ell q)^2}{N-q} + 8N\left(\dfrac{e(\tau^\ell q)^2}{N-q}\right)^{\tau^{1/2}-1}.$

Proof. By basic probability theory:

$$\mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right] \leq \sum_{j=0}^{\ell} \mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right]\,, \tag{6}$$

where $\neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1} = \neg\mathsf{e}(\mathcal{Q}_q)_{j-2} \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-1}$, and $\mathsf{eBCD}(\mathcal{Q}_q)_0$ is false by construction. We further split up this probability as follows:

$$\begin{aligned}
\mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] \leq\ & \mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)_j \cap \neg\mathsf{eBCD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] + \\
& \mathbb{P}\left[\mathsf{eBCD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] \\
\leq\ & \mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)_j \cap \neg\mathsf{eBCD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] + \\
& \mathbb{P}\left[\mathsf{eB}(\mathcal{Q}_q)_j \cap \neg\mathsf{eCD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] + \\
& \mathbb{P}\left[\mathsf{eCD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] \qquad (7) \\
\leq\ & \mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)_j \cap \neg\mathsf{eBD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] + \\
& \mathbb{P}\left[\mathsf{eB}(\mathcal{Q}_q)_j \cap \neg\mathsf{eCD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] + \\
& \mathbb{P}\left[\mathsf{eC}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] \\
& \mathbb{P}\left[\mathsf{eD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] ,
\end{aligned}$$

We consider these probabilities separately. Recall that A makes all queries to $\pi_{j-1,b}$ before all queries to $\pi_{j,b'}$ (for $j = 1, \ldots, \ell$ and b, b' arbitrary). Throughout the proof, by "for any $(x, y, z)_{j,b}$" we mean "for any $(x, y, \mathcal{Z})_{j,b}$ and any $z_{j,b} \in \mathcal{Z}_{j,b}$." Hence, any tuple $(x, y, \mathcal{Z})_{j,b}$ corresponds to $|\mathcal{Z}_{j,b}|$ tuples $(x, y, z)_{j,b}$.

$\mathsf{eA}(\mathcal{Q}_q)_j$. Assume $\neg\mathsf{eBD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}$ holds. Consider any b. The analysis depends on whether the query is forward or inverse. First, consider a *forward* query $y_{j,b} \leftarrow \pi_{j,b}(x_{j,b})$. We say that it renders a solution if it makes $y_{j,b} \oplus y'_{j,b} \in \mathcal{Z}_{j,b} \oplus \mathcal{Z}'_{j,b}$ satisfied for any existing query $(x', y', \mathcal{Z}')_{j,b}$. By $\neg\mathsf{eD}(\mathcal{Q}_q)_j$, we have $|\mathcal{Z}_{j,b}|, |\mathcal{Z}'_{j,b}| \leq \tau^j$ (these sets are fixed once $x_{j,b}$ is fixed). Consequently, the query completes a collision with probability at most $\frac{\tau^{2j}q}{N-q}$. Next, consider an *inverse* query $x_{j,b} \leftarrow \pi^{-1}_{j,b}(y_{j,b})$. It renders a solution if for some $(x, y, z)_{j-1,0}$, $(x, y, z)_{j-1,1}$, and $(x', y', z')_{j,b}$:

$$\begin{aligned}
x_{j,b} &= (y \oplus z)_{j-1,0} \oplus (y \oplus z)_{j-1,1} , \\
y_{j,b} &= (y \oplus z)_{j-1,0} \oplus (y \oplus z)'_{j,b} .
\end{aligned}$$

(Indeed, in this case $z_{j,b} = (y \oplus z)_{j-1,0}$.) By $\neg\mathsf{eB}(\mathcal{Q}_q)_j$ for $z := y_{j,b}$ (by condition, we have $y'_{j,b} \neq y_{j,b}$), there are at most $\tau^{j+1/2}$ solutions $(x, y, z)_{j-1,0}$ and $(x', y', z')_{j,b}$ to the second equation (using $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-1}$). By $\neg\mathsf{eD}(\mathcal{Q}_q)_{j-1}$, there are also at most $\tau^{j-1}q$ solutions for $(x, y, z)_{j-1,1}$, and a collision is thus triggered with probability at most $\frac{\tau^{2j-1/2}q}{N-q}$. Summing over all *forward and inverse* queries to $\pi_{j,b}$, and both choices of b, we obtain:

$$\mathbb{P}\left[\mathsf{eA}(\mathcal{Q}_q)_j \cap \neg\mathsf{eBD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] \leq \frac{2(\tau^j q)^2}{N-q} .$$

$\mathsf{eB}(\mathcal{Q}_q)_j$. Assume $\neg\mathsf{eCD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}$ holds. Consider any $z \in \{0,1\}^n$ and any b. By the layer-wise character of A, $\mathsf{eB}(\mathcal{Q}_q)_{j,b}$ can only be satisfied by queries to $\pi_{j,b}$. The analysis depends on whether the query is forward or inverse. First, consider a *forward* query $y_{j,b} \leftarrow \pi_{j,b}(x_{j,b})$. There exists at most q queries $(x, y, \mathcal{Z})_{j-1,0}$. By $\neg\mathsf{eD}(\mathcal{Q}_q)_{j-1}$, we have $|\mathcal{Z}_{j-1,0}| \leq \tau^{j-1}$. Additionally, by $\neg\mathsf{eD}(\mathcal{Q}_q)_j$, we have $|\mathcal{Z}_{j,b}| \leq \tau^j$ (this set is fixed once $x_{j,b}$ is fixed). Therefore, the query adds a solution to $\mathsf{eB}(\mathcal{Q}_q)_{j,b}$ with probability at most $\frac{\tau^{2j-1}q}{N-q}$, and any hit adds at most τ^j values (by $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-1}$). Next, consider an *inverse* query $x_{j,b} \leftarrow \pi^{-1}_{j,b}(y_{j,b})$ (and assume $y_{j,b} \neq z$). It renders a solution if for some $(x, y, z)_{j-1,0}$, $(x, y, z)_{j-1,1}$, and $(x', y', z')_{j-1,0}$:

$$\begin{aligned}
x_{j,b} &= (y \oplus z)_{j-1,0} \oplus (y \oplus z)_{j-1,1} , \\
y_{j,b} &= (y \oplus z)_{j-1,0} \oplus (y' \oplus z')_{j-1,0} \oplus z .
\end{aligned}$$

(Indeed, in this case $z_{j,b} = (y \oplus z)_{j-1,0}$.) By $\neg\mathsf{eC}(\mathcal{Q}_q)_j$ for $\overline{z} := y_{j,b} \oplus z \neq 0$, there are at most τ^j solutions $(x, y, z)_{j-1,0}$ and $(x', y', z')_{j-1,0}$ to the second equation (using $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-2}$). By $\neg\mathsf{eD}(\mathcal{Q}_q)_{j-1}$, there are also at most $\tau^{j-1}q$ solutions for $(x, y, z)_{j-1,1}$, and a solution is thus obtained with probability at most $\frac{\tau^{2j-1}q}{N-q}$. Any hit adds at most τ^j values (by $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-1}$).

More than $\tau^{j+1/2}$ solutions are added with probability at most

$$\binom{q}{\tau^{j+1/2}/\tau^j}\left(\frac{\tau^{2j-1}q}{N-q}\right)^{\tau^{j+1/2}/\tau^j} \leq \left(\frac{e(\tau^j q)^2}{N-q}\right)^{\tau^{1/2}} .$$

Summing over all N choices of z, and both choices of b, we obtain:

$$\mathbb{P}\left[\mathsf{eB}(\mathcal{Q}_q)_j \cap \neg\mathsf{eCD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] \leq 2N\left(\frac{e(\tau^j q)^2}{N-q}\right)^{\tau^{1/2}} .$$

$\boldsymbol{\mathsf{eC}(\mathcal{Q}_q)_j}$**.** Assume $\neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}$ holds. Consider any $z \in \{0,1\}^n\backslash\{0\}$. Note that, by $\mathsf{eC}(\mathcal{Q}_q)_{j-1}$, there exist at most τ^{j-1} solutions in case of $y_{j-1,0} = y'_{j-1,0}$. Hence, in order to find more than τ^j solutions, the adversary needs to find more than $\tau^j - \tau^{j-1}$ solutions with $y_{j-1,0} \neq y'_{j-1,0}$, and we focus on this problem. The analysis depends on whether the query is forward or inverse. First, consider a *forward* query $y_{j-1,0} \leftarrow \pi_{j-1,0}(x_{j-1,0})$. There exist at most q other tuples $(x', y', \mathcal{Z}')_{j-1,0}$, and by $\neg\mathsf{eD}(\mathcal{Q}_q)_{j-1}$ we have $|\mathcal{Z}_{j-1,0}|, |\mathcal{Z}'_{j-1,0}| \leq \tau^{j-1}$ (these sets are fixed once $x_{j-1,0}$ is fixed). Therefore, the query adds a solution to $\mathsf{eC}(\mathcal{Q}_q)_j$ with probability at most $\frac{\tau^{2(j-1)}q}{N-q}$, and any hit adds at most τ^{j-1} values (by $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-1}$). Next, consider an *inverse* query $x_{j-1,0} \leftarrow \pi^{-1}_{j-1,0}(y_{j-1,0})$. It renders a solution if for some $(x, y, z)_{j-2,0}$, $(x, y, z)_{j-2,1}$, and $(x', y', z')_{j-1,0}$:

$$\begin{aligned} x_{j-1,0} &= (y \oplus z)_{j-2,0} \oplus (y \oplus z)_{j-2,1} , \\ y_{j-1,0} &= (y \oplus z)_{j-2,0} \oplus (y' \oplus z')_{j-1,0} \oplus z . \end{aligned}$$

(Indeed, in this case $z_{j-1,0} = (y \oplus z)_{j-2,0}$.) We make a distinction between the cases $y_{j-1,0} \oplus z = y'_{j-1,0}$ and $y_{j-1,0} \oplus z \neq y'_{j-1,0}$; an adversary may succeed in both cases. In the former case, the values $y_{j-1,0}$ and z fix $y'_{j-1,0}$ (recall $y_{j-1,0} \neq y'_{j-1,0}$), and by $\neg\mathsf{eD}(\mathcal{Q}_q)_{j-1}$, we have $|\mathcal{Z}'_{j-1,0}| \leq \tau^{j-1}$. By $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-2}$, there is one solution $(x, y, z)_{j-2,0}$ to the second equation. By $\neg\mathsf{eD}(\mathcal{Q}_q)_{j-2}$, there are also at most $\tau^{j-2}q$ solutions for $(x, y, z)_{j-2,1}$, and a solution is thus obtained with probability at most $\frac{\tau^{2j-3}q}{N-q}$. Any hit adds at most τ^{j-1} values (by $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-2}$). Next, we consider the general case of $y_{j-1,0} \oplus z \neq y'_{j-1,0}$. By $\neg\mathsf{eB}(\mathcal{Q}_q)_{j-1}$ for $\overline{z} := y_{j-1,0} \oplus z$ (for which we thus have $y'_{j-1,0} \neq y_{j-1,0} \oplus z$), there are at most $\tau^{j-1/2}$ solutions $(x, y, z)_{j-2,0}$ and $(x', y', z')_{j-1,0}$ to the second equation (using $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-2\cap j-1}$). By $\neg\mathsf{eD}(\mathcal{Q}_q)_{j-2}$, there are also at most $\tau^{j-2}q$ solutions for $(x, y, z)_{j-2,1}$, and a solution is thus obtained with probability at most $\frac{\tau^{2j-5/2}q}{N-q}$. Any hit adds at most $\tau^{j-1/2}$ values (by $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-2}$). Concluding the inverse case, a hit is found with probability at most $\frac{\tau^{2j-3}q}{N-q} + \frac{\tau^{2j-5/2}q}{N-q} \leq \frac{\tau^{2(j-1)}q}{N-q}$ (where inequality holds as $1 + \tau^{1/2} \leq \tau$ for $\tau \geq 3$) and any hit adds at most $\tau^{j-1} + \tau^{j-1/2}$ values.

More than $\tau^j - \tau^{j-1}$ solutions are added with probability at most

$$\binom{q}{(\tau^j - \tau^{j-1})/(\tau^{j-1} + \tau^{j-1/2})}\left(\frac{\tau^{2(j-1)}q}{N-q}\right)^{(\tau^j - \tau^{j-1})/(\tau^{j-1} + \tau^{j-1/2})} \leq \left(\frac{e(\tau^j q)^2}{N-q}\right)^{\tau^{1/2}-1} .$$

Here, we again use that $1 + \tau^{1/2} \leq \tau$ and that $\tau^j - \tau^{j-1} \geq \tau^{j-1}$. Summing over all $N - 1 \leq N$ choices of z, we obtain:

$$\mathbb{P}\left[\mathsf{eC}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] \leq N\left(\frac{e(\tau^j q)^2}{N-q}\right)^{\tau^{1/2}-1} .$$

$\mathbf{eD}(\mathcal{Q}_q)_j$**.** Assume $\neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}$ holds. Consider any $z \in \{0,1\}^n$. Without loss of generality (by symmetry) consider a new query $(x, y, \mathcal{Z})_{j-1,0}$. The analysis depends on whether the query is forward or inverse. First, consider a *forward* query $y_{j-1,0} \leftarrow \pi_{j-1,0}(x_{j-1,0})$. There exist at most q other tuples $(x, y, \mathcal{Z})_{j-1,1}$, and by $\neg\mathsf{eD}(\mathcal{Q}_q)_{j-1}$ we have $|\mathcal{Z}_{j-1,0}|, |\mathcal{Z}_{j-1,1}| \le \tau^{j-1}$ (these sets are fixed once $x_{j-1,0}$ is fixed). Therefore, the query adds a solution to $\mathsf{eD}(\mathcal{Q}_q)_j$ with probability at most $\frac{\tau^{2(j-1)}q}{N-q}$, and any hit adds at most τ^{j-1} values (by $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-1}$). Next, consider an *inverse* query $x_{j-1,0} \leftarrow \pi^{-1}_{j-1,0}(y_{j-1,0})$. It renders a solution if for some $(x, y, z)_{j-2,0}$, $(x, y, z)_{j-2,1}$, and $(x, y, z)_{j-1,1}$:

$$\begin{aligned} x_{j-1,0} &= (y \oplus z)_{j-2,0} \oplus (y \oplus z)_{j-2,1}\,, \\ y_{j-1,0} &= (y \oplus z)_{j-2,0} \oplus (y \oplus z)_{j-1,1} \oplus z\,. \end{aligned}$$

(Indeed, in this case $z_{j-1,0} = (y \oplus z)_{j-2,0}$.) We make a distinction between the cases $y_{j-1,0} \oplus z = y_{j-1,1}$ and $y_{j-1,0} \oplus z \ne y_{j-1,1}$; an adversary may succeed in both cases. In the former case, the values $y_{j-1,0}$ and z fix $y_{j-1,1}$, and by $\neg\mathsf{eD}(\mathcal{Q}_q)_{j-1}$, we have $|\mathcal{Z}_{j-1,1}| \le \tau^{j-1}$. By $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-2}$, there is one solution $(x, y, z)_{j-2,0}$ to the second equation. By $\neg\mathsf{eD}(\mathcal{Q}_q)_{j-2}$, there are also at most $\tau^{j-2}q$ solutions for $(x, y, z)_{j-2,1}$, and a solution is thus obtained with probability at most $\frac{\tau^{2j-3}q}{N-q}$. Any hit adds at most τ^{j-1} values (by $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-2}$). Next, we consider the general case of $y_{j-1,0} \oplus z \ne y_{j-1,1}$. By $\neg\mathsf{eB}(\mathcal{Q}_q)_{j-1}$ for $\overline{z} := y_{j-1,0} \oplus z$ (for which we thus have $y_{j-1,1} \ne y_{j-1,0} \oplus z$), there are at most $\tau^{j-1/2}$ solutions $(x, y, z)_{j-2,0}$ and $(x, y, z)_{j-1,1}$ to the second equation (using $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-2\cap j-1}$). By $\neg\mathsf{eD}(\mathcal{Q}_q)_{j-2}$, there are also at most $\tau^{j-2}q$ solutions for $(x, y, z)_{j-2,1}$, and a solution is thus obtained with probability at most $\frac{\tau^{2j-5/2}q}{N-q}$. Any hit adds at most $\tau^{j-1/2}$ values (by $\neg\mathsf{eA}(\mathcal{Q}_q)_{j-2}$). Concluding the inverse case, a hit is found with probability at most $\frac{\tau^{2j-3}q}{N-q} + \frac{\tau^{2j-5/2}q}{N-q} \le \frac{\tau^{2(j-1)}q}{N-q}$ and any hit adds at most $\tau^{j-1} + \tau^{j-1/2}$ values.

More than τ^j solutions are added with probability at most

$$\binom{q}{\tau^j/(\tau^{j-1}+\tau^{j-1/2})} \left(\frac{\tau^{2(j-1)}q}{N-q}\right)^{\tau^j/(\tau^{j-1}+\tau^{j-1/2})} \le \left(\frac{e(\tau^j q)^2}{N-q}\right)^{\tau^{1/2}-1}.$$

Summing over all N choices of z, we obtain:

$$\mathbb{P}\left[\mathsf{eD}(\mathcal{Q}_q)_j \cap \neg\mathsf{e}(\mathcal{Q}_q)_{j-2\cap j-1}\right] \le N\left(\frac{e(\tau^j q)^2}{N-q}\right)^{\tau^{1/2}-1}.$$

Conclusion of proof. From (6) and (7), and simplifying the above bounds, we obtain:

$$\begin{aligned} \mathbb{P}\left[\mathsf{e}(\mathcal{Q}_q)\right] &\le \sum_{j=0}^{\ell} \frac{2(\tau^j q)^2}{N-q} + \sum_{j=1}^{\ell} 4N\left(\frac{e(\tau^j q)^2}{N-q}\right)^{\tau^{1/2}-1} \\ &= \frac{2q^2}{N-q}\sum_{j=0}^{\ell}\tau^{2j} + 4N\left(\frac{eq^2}{N-q}\right)^{\tau^{1/2}-1}\sum_{j=1}^{\ell}\tau^{2(\tau^{1/2}-1)j} \\ &\le \frac{4(\tau^\ell q)^2}{N-q} + 8N\left(\frac{e(\tau^\ell q)^2}{N-q}\right)^{\tau^{1/2}-1}. \end{aligned}$$

Here, we use that $\sum_{j=0}^{\ell} x^j = \frac{x^{\ell+1}-1}{x-1} \le \frac{x}{x-1}x^\ell \le 2x^\ell$ for $x \ge 2$. □