IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2024, No. 1, pp. 459–496. DOI:10.46586/tosc.v2024.i1.459-496

Tightening Leakage Resilience of the Suffix Keyed Sponge

Henk Berendsen and Bart Mennink

Radboud University, Nijmegen, The Netherlands
henk.berendsen@ru.nl, b.mennink@cs.ru.nl

Abstract. Lightweight cryptographic constructions are often optimized on multiple aspects that put the security bounds to the limit. In this respect, it is important to obtain security bounds that are tight and give an accurate and exact indication of the generic security. However, whereas for black-box security bounds it has become common practice to argue tightness of security bounds, for leakage resilience security bounds this is not the case. This is unfortunate, as for leakage resilience results, tightness is even more important as there is already a lossiness incurred in capturing the actual leakage by a theoretical model in the first place.
In this work, we consider the SuKS (Suffix Keyed Sponge) PRF construction and investigate tightness of the leakage resilience bound of Dobraunig and Mennink (ToSC 2019). We observe that, although their black-box security result is tight, their leakage resilience bound is *not* tight in their bounded leakage term λ. We observe that this is caused by the fact that parts of the security bound contain a term covering multicollisions *and* a term covering leakage, but an adversary is unable to combine both. We next consider improved security of the SuKS for two types of leakage: fixed position leakage, where the adversary directly learns the value of λ bits of a secret state, and Hamming weight leakage, where the Hamming weight of a fixed part of the state is leaked. For fixed position leakage, a very generous form of bounded leakage, we improve the original bound by making wise use of the multicollision limit function of Daemen et al. (ASIACRYPT 2017). For the more realistic setting of Hamming weight leakage, we structurally revisit the multicollision limit function analysis by including Hamming weight in the computation, a problem that is difficult on its own due to the non-uniform character of this type of leakage. In both cases, we improve and tighten the leakage resilience bound of Dobraunig and Mennink. The improved bound for the SuKS has immediate consequences for the leakage resilience of the NIST lightweight cryptography competition finalist ISAP v2, an authenticated encryption scheme that uses the SuKS internally.

Keywords: SuKS · PRF · leakage resilience · tightness · Hamming weight

1 Introduction

Symmetric cryptographic schemes find their use over a whole spectrum of applications, running from lightweight to high-end. However, when such a scheme is evaluated in a potentially hostile environment, which is for example the case for lightweight applications ran on smart cards and wearables, side-channel attacks are a serious threat. In such an attack, the attacker has additional information about the evaluation of the cryptographic scheme, such as runtime [Koc96], electromagnetic emanation [KA98], or power consumption [KJJ99], and can use this information to gain additional information about secret data processed in the algorithm. These attacks can be quite influential: even if the cryptographic scheme has a high level of generic proven security, its black-box security can be completely nullified due to side-channel attacks.

Received: 2023-11-23 Accepted: 2024-01-23 Published: 2024-03-01

One way to mitigate these side-channel attacks is at the implementation level by employing strong countermeasures [GP99,CJRR99,NRR06,NRS11]. An alternative solution is to use less efficient but stronger schemes that provably resist side-channel attacks, a direction known as leakage resilience. The seemingly most relevant approach is a combination of those two, where one designs a scheme such that it provably resists most of the side-channel attacks and only protection against the weaker types is needed. This approach is often referred to as a leveled implementation [PSV15].

One main problem with leakage resilient proofs, however, is that it is extremely hard to accurately upper bound the actual leakage. Instead, one often resorts to the bounded leakage model where one assumes that the amount of leakage (de facto, the entropy loss due to leakage) per primitive evaluation is bounded by λ. It is then important to "select" λ to closely match practical cases (see also Dobraunig et al. [DMP22]). Indeed, lightweight cryptographic schemes are optimized in various different dimensions (size, latency, power, etc.), and this optimization often has an impact on the security parameters that appear in the bound. A too lossy security bound then gives a false sense of insecurity.

However, beyond the seemingly inevitable lossiness in the bound of the actual physical leakage by λ,[1] proven security may also expose lossiness in the actual security bound, i.e., in the bound that argues what the security parameters should satisfy for the scheme to be secure. The necessity of tightness of security proofs, in particular for lightweight schemes, has been widely understood and acknowledged in the black-box setting before [DM20, LNS18, JN20, DDNT23, LMP17]. However, tightness analysis of a leakage resilience bound is typically not thoroughly investigated. More concretely, there are (to the best of our knowledge) no security analyses in the bounded leakage model that come with a tightness analysis with respect to the leakage term λ. This is unfortunate, as tightness of the leakage term is not less important than of the other terms, in particular as λ is already a bound on the actual physical leakage in the first place.

1.1 Suffix Keyed Sponge

In this work we focus on the suffix keyed sponge (SuKS), a message authentication code (MAC) function derived from the sponge hash function [BDPV07]. Just like the sponge function, the SuKS operates using a b-bit permutation p and keeps a b-bit state split into a c-bit inner part and an r-bit outer part. In its simplest form, the SuKS concatenates the (padded) message with the key, partitions it into blocks of r bits, and absorbs them into the state by adding the blocks into the outer part, interleaved with evaluations of p. Then, a tag is squeezed r bits at a time from the outer part, again interleaved with evaluations of p until a tag of required length is obtained.

The idea of this simple SuKS construction dates back to the original introduction of the sponge function by Bertoni et al. [BDPV07, BDPV11]. Dobraunig and Mennink [DM19b] considered the construction in generality, where the key may be blended into the state using an arbitrary function G, which in turn may affect more than just the r outer bits of the state, and where the tag squeezing rate t may likewise be larger than r so that the tag could be squeezed in one evaluation of p. This general construction is also depicted in Figure 1. Dobraunig and Mennink proved that this general construction is black-box secure as well as leakage resilient, provided the permutation p is assumed to be perfectly random, and provided p and G do not leak "too much" (refer to Section 2.4 for more formalism regarding bounded leakage).

As correctly mentioned by Dobraunig and Mennink, the SuKS construction is particularly interesting in the leakage setting: as the key is only processed "at the end", there is only a bounded amount of secret information processed in any full evaluation of the SuKS: all intermediate states while processing the message are non-secret and need not be

[1] Alternative bounding approaches exist [KR19], but the issue remains the same.

protected. The only secret values/states that require side-channel protection are the secret key, the output of the function G, and the input and output of the *final* permutation p. In addition, it turns out that by maintaining an inner part in parallel to processing G, the state size of G that would need to be protected can be as low as $2k$ bits (including the key input). This stands in sharp contrast with the typical hash-then-PRF approach where the input to the keyed finalization function is of size $3k$ bits. In part inspired by these factors, the SuKS was selected as MAC function in ISAP v2 [DEM+17,DEM+20,DEM+21], a finalist authenticated encryption scheme in the NIST lightweight cryptography competition [NIS19], that aimed at a design that is hard to break even with only simple side-channel protection.

Soon after the formalization of the general SuKS construction, Dobraunig and Mennink [DM20] analyzed tightness of their black-box security result, and they demonstrated that their bound is indeed tight by presenting attacks that matched the terms in the security bound up to a constant. The sophistication in these attacks lay in exploiting the multicollisions. For example, if μ evaluation of the SuKS lead to the same tag, the distinguisher can make inverse permutation queries for that tag and a guessed inner part and they succeed with probability $\mu/2^{b-t}$, instead of just $1/2^{b-t}$. If t is small enough relative to the adversarial complexity, large multicollisions on the t-bit tag may become plausible.

1.2 Tightening Leakage Resilience

However, despite the tightness analysis of Dobraunig and Mennink, tightness of the leakage resilience bound, and in particular of the leakage term λ, is unclear. In our work, we perform a detailed study of the leakage resilience bound of Dobraunig and Mennink [DM19b] and conclude that, perhaps surprisingly, the term is non-tight in λ! In detail, the leakage resilience security bound contains two terms involving leakage: one term which corresponds to guessing the secret part after G (noting that the attacker learns the corresponding inner part of the state), and one term which corresponds to guessing the secret part next to the tag (noting that the attacker learns the tag but not the truncated part). For each of these two terms, we were not able to find a tightness attack in λ. In a nutshell, the cause of this gap is that both security terms involve a term covering multicollisions *as well as* a term covering leakage, but leakages happen to annihilate potential multicollision gain. Stated differently, we conclude that the adversary may accelerate its attack by exploiting multicollisions, *or* by exploiting the additional leakage, *but not both*.

As second contribution, we improve the security bound of the SuKS under leakage. However, depending on the type of leakage, this is much harder than it intuitively seems. The reason is that the leakages must be taken into account *in the analysis of multicollisions*, and this analysis is non-trivial if elements are drawn non-uniformly. Daemen et al. [DMV17] already performed a technical analysis in case elements are drawn uniformly without replacement (a result also used by Dobraunig and Mennink [DM19b] for the analysis of the SuKS). Capturing other forms of non-uniform drawings is extremely involved.[2] Thus, we will restrict our focus to two types of (bounded and non-adaptive [FPS12]) leakage:

- Fixed position leakage (Section 5): in this case, the adversary learns λ bits of predetermined positions of the secret part of the state. This is one of the most generous settings for the adversary, as it results in actual secret bits being leaked. The model is still somewhat restrictive in the sense that the positions that leak are fixed before the start of the experiment, and thus only captures a certain type of attacks. For example, it captures typical types of probing attacks, where an adversary gains

[2] We remark that the same issue occurs for the alternative multicollision bounding approach based on the expected value, of Choi et al. [CLL19]. A naive multicollision bounding akin to Jovanovic et al. [JLM14, JLM+19] and Chakraborty et al. [CJN20] may still work but gives a very loose upper bound.

information about some (non-adaptively) selected computation variables [ISW03]. Here, we remark that the more general model of t-threshold probing security even allows repositioning the probes in-between queries [DDF14, DDF19], and these are not captured by our model.

In this leakage setting, the original multicollision limit function for distributions without replacement of Daemen et al. *does* carry over with small adjustments;

- Hamming weight leakage (Section 6): in this case, the adversary learns the weight of certain n bits of the secret part of the state. The positions of these n bits is assumed to be fixed (see also Section 7 for a discussion on generalizing this assumption). A logical choice would be $n = 8$, a typical processing unit, but we consider any possible value in our work, and for simplicity we specifically take $n = 7$ in our examples instead, as Hamming weight leakage of 7 bits can be encoded into 3 bits of leakage. This is a much more relevant case to study as Hamming weight more realistically models leakage [May00, MOP07, DMMS21]. As a matter of fact, experiments of Mayer-Sommer [May00] demonstrated a correlation between leakage and Hamming weights, even for simple power analysis, and many side-channel attacks in literature explicitly use the Hamming weight model to simulate leakage to adversaries [KJJ99, Mes00, BBD+13, BFG14].

  For this type of leakage, we derive a new bound on the multicollision limit function of Daemen et al. to cover (i) distribution without replacement and (ii) non-uniform Hamming weight leakage. Whereas Daemen et al. already showcased how to turn the multicollision limit function analysis to distributions without replacement, dealing with highly non-uniform outputs makes the analysis much more complex. This multicollision limit function bounding is included in Section 6.1.

Having settled these two choices of leakage function and having obtained bounds on the multicollision limit function for these leakages, we can finally apply our findings to the SuKS to obtain improved security bounds under non-adaptive fixed position or Hamming weight leakage. The obtained results are in the ideal permutation model and are stated in Theorem 2 and Theorem 4. The improvements are discussed in Section 5.2 and Section 6.4, where the bounds are mapped to the parameters of ISAP v2. In these sections, we demonstrate for example that if we apply our results to ISAP v2 with the parameter set corresponding to Ascon-p and with $\lambda = 3$ bits of leakage, the original bound guaranteed up to around 110-bit security, whereas our bound for fixed position leakage guarantees up to around 122-bit security and our bound for Hamming weight leakage up to around 118-bit security. Similar conclusions can be drawn for different values of λ. However, we wish to stress again that these conclusions only hold in the ideal permutation model and under the specific types of bounded non-adaptive leakage as analyzed in this work.

1.3 Outline

We start with discussing some preliminary material in Section 2. The SuKS construction is presented in Section 3, with the state-of-the-art bound in Theorem 1. Non-tightness of the leakage resilience bound is discussed in Section 4. We improve the bound under fixed position leakage, and map it to the parameters of ISAP v2 in Section 5. We analyze how Hamming weight leakage can be included in the computation of the multicollision limit function, improve the bound under this type of leakage, and map it to the parameters of ISAP v2 in Section 6. We conclude the work in Section 7.

2 Preliminaries

Let $m, n \in \mathbb{N}$ such that $m \leq n$. The set of n-bit permutations is denoted by $\mathsf{perm}(n)$. The set of n-bit strings is denoted by $\{0,1\}^n$, and the set of strings of arbitrary length by $\{0,1\}^*$. For $X \in \{0,1\}^n$, $\mathrm{left}_m(X)$ and $\mathrm{right}_m(X)$ denote respectively the m left-most and m right-most bits of the string X, and X_m denotes the mth bit of X. Let $\log(\cdot)$ be the binary logarithm. Let $\mathrm{HW}(X)$ be the $\lceil \log(n+1) \rceil$-bit representation of the Hamming weight of X, i.e., the number of bits in X equal to 1. For $Y_1, Y_2 \in \{0,1\}^*$, $Y_1 \| Y_2$ denotes the concatenation of Y_1 and Y_2. The mth falling factorial of n is denoted by $(n)_m = n(n-1)\cdots(n-m+1)$. For some event E, $\mathbf{Pr}(E)$ denotes the probability that E occurs. Let the random choice of an element s from a set S be denoted $s \xleftarrow{\$} S$.

2.1 Uniformity and Universality

The notion of uniformity of a function describes how likely it is that a certain input maps to a certain output; the notion of universality of a function describes how likely it is that this function gives the same output for distinct inputs. We adopt the notation and terminology of Dobraunig and Mennink [DM19b].

Definition 1 ($2^{-\delta}$-uniformity)**.** A function $G : \{0,1\}^k \times \{0,1\}^s \to \{0,1\}^s$ is $2^{-\delta}$-uniform if, for a randomly drawn $K \xleftarrow{\$} \{0,1\}^k$ and any $X, Y \in \{0,1\}^s$, δ is the largest real number such that

$$\mathbf{Pr}\left(G(K,X) = Y\right) \leq 2^{-\delta}\,.$$

Definition 2 ($2^{-\varepsilon}$-universality)**.** A function $G : \{0,1\}^k \times \{0,1\}^s \to \{0,1\}^s$ is $2^{-\varepsilon}$-universal if, for a randomly drawn $K \xleftarrow{\$} \{0,1\}^k$ and any distinct $X, X' \in \{0,1\}^s$, ε is the largest real number such that

$$\mathbf{Pr}\left(G(K,X) = G(K,X')\right) \leq 2^{-\varepsilon}\,.$$

Dobraunig and Mennink used both $2^{-\delta}$-uniformity and $2^{-\varepsilon}$-universality for the assumption that a function G is strongly protected, meaning it is $2^{-\delta}$-uniform and $2^{-\varepsilon}$-universal even under internal leakage.

2.2 Distinguishing Advantage

We define an adversary $\mathcal{A}$ which has access to one of two oracles $\mathcal{O}$ and $\mathcal{P}$. The adversary outputs a decision bit b after interacting with the oracle. Let $\mathcal{A}$ output $b = 1$ when it decides it interacted with $\mathcal{O}$, and $b = 0$ otherwise. Let $\mathcal{A}^{\mathcal{O}}$ and $\mathcal{A}^{\mathcal{P}}$ denote respectively that the adversary interacted with $\mathcal{O}$ and $\mathcal{P}$. The distinguishing advantage of $\mathcal{A}$ is then defined as

$$\Delta_{\mathcal{A}}\left(\mathcal{O}\,;\,\mathcal{P}\right) = \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{\mathcal{O}}\right) - \mathbf{Pr}\left(1 \leftarrow \mathcal{A}^{\mathcal{P}}\right)\,.$$

2.3 PRF Security

Let $b, k, t \in \mathbb{N}$ and $m \in \mathbb{N} \cup \{*\}$. Let $F^p : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^t$ be a function that internally uses a permutation $p \in \mathsf{perm}(b)$ and let $R_{m,t} : \{0,1\}^m \to \{0,1\}^t$ be a uniformly random function. The pseudorandom function (PRF) security of F against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}_F^{\mathrm{prf}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left(F_K^p, p\,;\; R_{m,t}, p\right)\,,$$

with $p \xleftarrow{\$} \mathsf{perm}(b)$ and $K \xleftarrow{\$} \{0,1\}^k$. The adversary has query access to either F instantiated with p and K, or to $R_{m,t}$. In both cases, the adversary can also query p itself in both directions.

The resource complexity of the adversary is measured by the number of queries q to the construction (F_K^p or $R_{m,t}$), and the number of primitive queries N to the permutation p. The time complexity of the adversary, such as the time spent comparing the results of the queries to find collisions, is not taken into account.

2.4 NALR-PRF Security

PRF security is not well suited to describe security in the case of leakage resilience. We follow the adoption of PRF security in the ideal permutation model to non-adaptive leakage by Dobraunig and Mennink [DM19b]. We reuse the definitions of Section 2.3. Let $\lambda \in \mathbb{N}$ and let $\mathcal{L}$ be a class of leakage functions $\{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda$, that on input of two b-bit strings output at most λ bits of leakage. For $L \in \mathcal{L}$, $\lceil F_K^p \rceil_L$ is a leaky version of F_K^p which evaluates F_K^p as usual, but additionally outputs the value $L(X,Y)$ for *each permutation evaluation* $p(X) = Y$ incurred in the evaluation. The NALR-PRF security of F with respect to $\mathcal{L}$ is defined as

$$\mathbf{Adv}_{F,\mathcal{L}}^{\text{nalr-prf}}(\mathcal{A}) = \max_{L \in \mathcal{L}} \Delta_{\mathcal{A}} \left(\lceil F_K^p \rceil_L, F_K^p, p \,;\, \lceil F_K^p \rceil_L, R_{m,t}, p \right) .$$

Here, in addition to the queries the adversary can make in the case of PRF security, the adversary can also make queries to $\lceil F_K^p \rceil_L$ regardless of whether they interact with F_K^p or $R_{m,t}$. In our case, F_K^p will be the SuKS (see Section 3), and in this construction, only the final call to p operates on secret information and all earlier calls occur before the key is blended into the state. Therefore, in our case, we will consider that in each query to F_K^p only the final call to p leaks. The leakage function L is independent of p, meaning that it does not internally evaluate p or p^{-1}.

Note that we consider non-adaptive leakage: we define the advantage as the maximum taken over all leakage functions $L \in \mathcal{L}$ and this leakage function stays the same throughout the experiment. This also means that it always returns the same leakage for the same input.

The set $\mathcal{L}$ can, a priori, be any set of functions of the form $\{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda$. However, this set should closely match the type of leakage the adversary can obtain. Later, we will restrict ourselves to two particular sets of leakage functions, namely functions $\mathcal{L}_{\text{fixed}}$ that leak on fixed positions, and functions $\mathcal{L}_{\text{Hamming}}$ that leak the Hamming weight of part of a state.

We finally remark that the SuKS internally also uses a function G next to the permutation p. However, we do not consider leakage incurred in evaluations of the function G because we assume that G is strongly protected (similar to [BKP+18, BPPS17, GPPS19] but then for the case of universal hashing).

2.5 Multicollisions

The maximum size of a multicollision can be upper bounded by the multicollision limit function formalized by Daemen et al. [DMV17]. Dobraunig and Mennink defined this function as follows to reason about the security of the SuKS [DM19b]:

Definition 3. Let $q, b, s \in \mathbb{N}$ such that $s \leq b$. Consider the experiment of throwing q balls uniformly at random in 2^{b-s} bins, and denote by μ the maximum number of balls in any single bin. The multicollision limit function $\mu_{b-s,s}^q$ is defined as the smallest natural number x that satisfies

$$\mathbf{Pr}\left(\mu > x\right) \leq \frac{x}{2^s} .$$

The right hand side of the inequality is mostly inspired by the particular applications where the multicollision limit function is used. In fact, in sponge-based schemes we are often concerned with an attacker that can obtain a multicollision on a part of the b-bit

state, say $b - s$ bits. If it finds a μ-collision of this kind, it can speed-up the guessing of the remaining s bits by a factor μ and we can argue that its success probability is at most $\mu N/2^s$, where N is the total number of attempts. By imposing that $\mu \leq \mu^q_{b-s,s}$ except with probability $\mu^q_{b-s,s}/2^s$, we obtain an accumulated probability bound of $(\mu^q_{b-s,s}N + \mu^q_{b-s,s})/2^s$. A more detailed treatment of this definition and a comparison with alternative approaches [JLM14, JLM+19, CLL19, CJN20] is given in [Men23, Section 4.2].

The multicollision limit function of Definition 3 can be bounded using probability theory. In particular, Daemen et al. [DMV17, Section 6.5] demonstrated that the value x satisfies the inequality

$$\frac{2^b e^{-\gamma}\gamma^x}{(x-\gamma)x!} \leq 1 \tag{1}$$

with $\gamma = q/2^{b-s}$, from which x can be determined numerically (see also [Men23, Appendix A]). This analysis, however, assumes that the balls are thrown with replacement.

In sponge-based constructions, however, we are not concerned with a random transformation, but rather a random b-bit permutation: for each ball a b-bit value is selected *without* replacement, and placed in one of the 2^{b-s} bins depending on certain $b - s$ bits of the ball. Note that, in this case, balls are not thrown into the bins *with* replacement (so above definition and reasoning does not carry over verbatim), neither are they thrown *without* replacement, as still 2^s balls can end in a certain bin. To capture this case, Daemen et al. [DMV17, Section 6.6] have extended their analysis and showed that if balls are thrown into bins according to a certain distribution that is "reasonably close" to random, i.e., according to a distribution D such that any thrown ball ends up in any bin with a probability p that satisfies

$$|p - 2^{-(b-s)}| \leq 0.1 \cdot 2^{-(b-s)}\,, \tag{2}$$

then the corresponding multicollision limit function $\mu^{D,q}_{b-s,s}$ satisfies $\mu^{D,q}_{b-s,s} \leq \mu^{2q}_{b-s,s}$. The proof of this considers two experiments, one corresponding to the first multicollision limit function and the other one corresponding to the second multicollision function, and argues that each bin in the former experiment is at most as full as in the latter experiment.

Looking ahead to Section 6.1, we will take inspiration from their analysis but will have to deal with an additional difficulty, namely that we consider Hamming weight leakage that gives a very biased distribution D that does not easily fit the above reasoning.

3 The Suffix Keyed Sponge

3.1 Construction

The suffix keyed sponge (SuKS), depicted in Figure 1, is a MAC function formalized by Dobraunig and Mennink [DM19b]. Let $b, c, k, r, s, t \in \mathbb{N}$ such that $b = r + c$ and $k, s, t \leq b$. Let $p \in \mathsf{perm}(b)$ be a permutation and let $G : \{0,1\}^k \times \{0,1\}^s \to \{0,1\}^s$ be a function. The SuKS takes as input an arbitrarily long plaintext P and a k-bit long secret key K, and produces as output a t-bit long tag T.

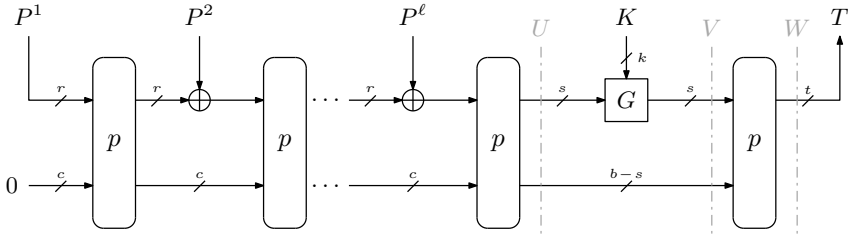

Figure 1: The suffix keyed sponge

The state of the SuKS consists of an r-bit outer part and c-bit inner part. Just like the sponge function, P is injectively padded, split into r-bit blocks $P^1, P^2, \ldots, P^\ell$ and absorbed by XORing these blocks with the outer part of the state. The permutation p is evaluated on the state each time a block has been absorbed. Then, the outer s bits of the state are transformed by the function G with the secret key K as input. Finally, p is evaluated on the state one more time, and the outer t bits of the resulting state form the tag T.

The states V and W are secret because they occur after K has been absorbed into the SuKS state. Therefore, only the function G and last evaluation of the permutation p are vulnerable to leakage. However, we do not take leakage from G into account because it is assumed to be strongly protected.

3.2 Security Bounds

Dobraunig and Mennink have given a PRF and a NALR-PRF bound on the security of the SuKS, respectively in Section 5 and 6 of their article formalizing the SuKS [DM19b]. We repeat these bounds in Theorem 1. Note that we have amended the third term of the PRF bound by removing an error present in the superscript parameter of the multicollision limit function.[3] Looking ahead, we have moved the second term of the NALR-PRF bound to the end of the bound to highlight the similarities between the two security bounds.

Theorem 1. *Let F be the suffix keyed sponge described in Section 3.1 based on a random permutation $p \xleftarrow{\$} \mathsf{perm}(b)$ and function $G : \{0,1\}^k \times \{0,1\}^s \to \{0,1\}^s$. Assuming that G is $2^{-\delta}$-uniform and $2^{-\varepsilon}$-universal, it holds for any adversary $\mathcal{A}$ with access to $q \geq 2$ construction queries and $N \leq 2^{b-1}$ primitive queries that*

$$\mathbf{Adv}_F^{\mathrm{prf}}(\mathcal{A}) \leq \frac{2N^2}{2^c} + \frac{\mu_{b-s,s}^{2(N-q)} \cdot N}{2^{\min\{\delta,\varepsilon\}}} + \frac{\mu_{t,b-t}^{2q} \cdot N}{2^{b-t}}\,. \tag{3}$$

When additionally assuming that G is strongly protected, and that $\mathcal{A}$ receives the output of a non-adaptive leakage function $L \in \mathcal{L}$ which leaks at most λ bits of information for each evaluation of the permutation p, it holds that

$$\mathbf{Adv}_{F,\mathcal{L}}^{\text{nalr-prf}}(\mathcal{A}) \leq \frac{2N^2}{2^c} + \frac{\mu_{b-s,s}^{2(N-q)} \cdot N}{2^{\min\{\delta,\varepsilon\} - \mu_{s,b-s}^{2(N-q)}\lambda}} + \frac{\mu_{t,b-t}^{2q} \cdot N}{2^{b-t-\lambda}} + \frac{\mu_{s,b-s}^{2(N-q)}}{2^{b-s}}\,. \tag{4}$$

4 Non-Tightness Under Leakage

In [DM20], Dobraunig and Mennink argued tightness of the black-box security bound of (3). However, it is yet unclear whether the leakage resilience bound is tight. In this

[3] To wit, the third term of the PRF bound had multicollision term $\mu_{t,b-t}^{q}$ instead, thus not taking into account the doubling of the number of balls to account for the non-uniform drawing as outlined in Section 2.5.

section, we show that the NALR-PRF security bound on the SuKS given in (4) is, as a matter of fact, not tight. We focus on the second and third term of the bound of (4), because they contain the leakage term λ. (The fourth term is collateral damage coming from multicollisions on the leakage value.)

For both terms, we first describe how an attack without leakage (thus, in the black-box setting) would be mounted, and subsequently discuss the difficulties in adding the λ into the attack. For simplicity, the third term is discussed first, in Section 4.1, and the second term in Section 4.2. We recap in Section 4.3.

4.1 Non-Tightness of Leakage in Third Term

We have combined elements from two attacks of [DM20, Sections 4.1, 5.1] to form an attack which exploits multicollisions on the tag to mount a forgery. We first describe this attack in the non-leaky setting.

(1) Make q construction queries with distinct plaintexts P_i to get the corresponding tags T_i, and make primitive queries on these same plaintexts to get the corresponding states before key absorption U_i.

(2) Find a multicollision of size μ for the tag T; that is, μ values i such that T_i is equal to some $T^\star$. Let $\mathcal{S}$ be the set containing these μ plaintexts.

(3) Additionally, find a set $\mathcal{S}'$ such that for all elements $P_i \in \mathcal{S}$, $\mathcal{S}'$ contains a different element P_i' with the same value for the outer part of the state U, i.e., $\mathrm{left}_s(U_i) = \mathrm{left}_s(U_i')$.

(4) For varying $Z_j \in \{0,1\}^{b-t}$, make N inverse primitive queries $p^{-1}(T^\star \| Z_j)$. If the result of one of these queries is of the form $Y \| \mathrm{right}_{b-s}(U_i)$ for some i in the multicollision, $\mathrm{left}_s(V_i)$ is likely equal to Y. Note that $\mathrm{left}_s(V_i) = \mathrm{left}_s(V_i')$ because $\mathrm{left}_s(U_i)$ and $\mathrm{left}_s(U_i')$ collide.

(5) If V_i has been found for some i in the multicollision, compute

$$T' = \mathrm{left}_t(p(\mathrm{left}_s(V_i) \| \mathrm{right}_{b-s}(U_i'))) \,.$$

Then, (P_i', T') is a valid forgery.

Note that this attack matches the third term of the PRF security bound on the SuKS given in (3). For each inverse primitive query $p^{-1}(T^\star \| Z_j)$, the attacker has a probability of $\frac{\mu}{2^{b-t}}$ to correctly guess one of the values W_i in the multicollision and recover the corresponding V_i. Because the attacker makes N such queries and the size of the largest multicollision is bounded by $\mu_{t,b-t}^{2q}$, the attack matches the bound of $(\mu_{t,b-t}^{2q} \cdot N)/2^{b-t}$.

We remark that it is possible that, in step (4), the result of a query is of the form $Y \| \mathrm{right}_{b-s}(U_i)$ such that $Y \neq \mathrm{left}_s(V_i)$. However, the probability of this event occurring is negligible, unless $b - t > s$, and in that case the second term of (3) would be dominant, not the third term.

The attacker could exploit leakage to decrease the amount of bits of W (the state at tag squeezing) which need to be guessed in step (4). Consider for example the following leakage function which leaks the λ right-most bits of W:

$$\begin{aligned} &L_p^{\mathrm{right}} : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda \,, \\ &L_p^{\mathrm{right}}(V, W) = \mathrm{right}_\lambda(W) \,. \end{aligned} \tag{5}$$

With this leakage function, the attacker only needs to guess $b - t - \lambda$ bits of W instead of $b - t$ bits, which is reflected in the denominator of the third term of (4). However, this

attack improvement does not work. The problem lies in the numerator of this term. The attack without leakage exploits the multicollision for T by being able to match μ values W_i with each guess. With the above leakage function, the leaked value does not have to be the same for every W_i, meaning it may not be possible to match μ values with one guess when exploiting leakage. However, the numerator of this term does not change with respect to the numerator of its counterpart in the PRF bound to account for this problem. Therefore, assuming that the attacker uses the leakage function defined in (5), the third term of the NALR-PRF bound is not tight.

4.2 Non-Tightness of Leakage in Second Term

We first describe the attack of [DM20, Section 5.1] in the non-leaky setting, which exploits multicollisions on the inner part of the state during key absorption to mount a forgery.

(1) Make primitive queries on distinct plaintexts P_i to get the corresponding states before key absorption U_i.

(2) Find a multicollision of μ plaintexts P_i such that for all corresponding states U_i, $\mathrm{right}_{b-s}(U_i)$ is equal to some value $U^\star$. Let $\mathcal{S}$ be the set containing these μ plaintexts.

(3) Make construction queries to compute the tags T_i corresponding to the plaintexts $P_i \in \mathcal{S}$.

(4) Additionally, find a set $\mathcal{S}'$ such that for all elements $P_i \in \mathcal{S}$, $\mathcal{S}'$ contains a different element P_i' with the same value for the outer part of the state U, i.e., $\mathrm{left}_s(U_i) = \mathrm{left}_s(U_i')$.

(5) Find the state after key absorption V_i for some i by making N different guesses Z_j for the outer part of V_i. Verify each guess by checking whether $\mathrm{left}_t(p(Z_j \| U^\star))$ equals T_i for some i. Note that $\mathrm{left}_s(V_i) = \mathrm{left}_s(V_i')$ because $\mathrm{left}_s(U_i)$ and $\mathrm{left}_s(U_i')$ collide.

(6) If V_i has been found for some i in the multicollision, compute

$$T' = \mathrm{left}_t(p(\mathrm{left}_s(V_i) \| \mathrm{right}_{b-s}(U_i'))) \,.$$

Then (P_i', T') is a valid forgery.

Because G is assumed to be $2^{-\delta}$-uniform, the attacker has a probability of $\frac{\mu}{2^\delta}$ to correctly guess one of the values V_i for each primitive query $p(Z_j \| U^\star)$. Because the attacker makes N such queries and the size of the largest multicollision is bounded by $\mu_{b-s,s}^{2(N-q)}$, the attack matches the bound of $(\mu_{b-s,s}^{2(N-q)} \cdot N)/2^\delta$.

This bound is almost the same as the second term of the PRF security bound on the SuKS given in (3); the only difference is the denominator being 2^δ instead of $2^{\min\{\delta,\varepsilon\}}$. We were unable to find an equally powerful attack which exploits the $2^{-\varepsilon}$-universality of G, meaning that (to the best of our knowledge) the bound is tight only if $\delta \leq \varepsilon$. However, for usual instantiations this is the case (e.g., assuming $k = s$, we have $\delta = k$ and $\varepsilon = \infty$ for XORing and $\delta = \varepsilon = k$ for a random function). Therefore, we will not consider attacks exploiting the $2^{-\varepsilon}$-universality of G.

We remark that it is possible that, in step (5), for some Z_j, $\mathrm{left}_t(p(Z_j \| U^\star))$ equals T_i for some i, but Z_j is not equal to the outer part of V_i. However, the probability of this event occurring is negligible, unless $s > b - t$, in which case the third term of (3) would be dominant, not the second term.

The attacker could exploit the combination of multicollisions and leakage to learn a large part of $\mathrm{left}_s(V_i)$ for all i. Suppose that in step (4) of the attack, the attacker finds

for each element $P_i \in \mathcal{S}$ a multicollision of size μ' instead of a single collision for $\text{left}_s(U_i)$. The attacker could use a leakage function which leaks different parts of $\text{left}_s(V_i)$ depending on the value of the inner part of U. Assuming that within each multicollision of size μ', each element has a distinct value for the inner part of U, the attacker would learn $\mu'\lambda$ bits of $\text{left}_s(V_i)$ for each $P_i \in \mathcal{S}$. Because μ' is bounded by the multicollision limit function $\mu_{s,b-s}^{2(N-q)}$ and G is $2^{-\delta}$-uniform, the probability of the attacker guessing one value V_i is $1/2^{\delta-\mu_{s,b-s}^{2(N-q)}\lambda}$, which is reflected in the denominator of the second term.

The exploitation of both leakage and multicollisions leads to multiple problems, however. Similarly to the problem described in Section 4.1, the attacker may not be able to match all μ values V_i corresponding to the plaintexts $P_i \in \mathcal{S}$ with one guess because the $\mu'\lambda$ leaked bits may not be equal for each V_i. Furthermore, because the leakage function leaks different bits depending on the value of $\text{right}_{b-s}(U)$, which is distinct for each plaintext within the multicollisions of size μ', the bit positions of $\text{left}_s(V_i)$ which leak may be different for each i.

Indeed, consider for the sake of example the following leakage function which leaks bits in a fixed position, namely the λ left-most bits of V:

$$\begin{aligned} &L_p^{\text{left}} : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda\,, \\ &L_p^{\text{left}}(V,W) = \text{left}_\lambda(V)\,. \end{aligned} \tag{6}$$

With this leakage function, the first problem diminishes because there are only λ leaked bits which may not be equal instead of $\mu'\lambda$ bits, and the second problem does not apply.

4.3 Recap

Neither the attack of Section 4.1 nor that of Section 4.2 reaches tightness with respect to (4), and the main cause is that the attacker cannot exploit multicollisions and leakage separately: they need to be analyzed jointly. In the upcoming sections, we consider a tightened NALR-PRF bound for the SuKS for two specific types of leakage: fixed position leakage in Section 5 and Hamming weight leakage in Section 6.

In these tightened bounds, we adapt the multicollision limit functions such that the attacker can exploit multicollisions and leakage simultaneously. This does decrease the size of the multicollisions the attacker can find, as seen in Section 5.2 and Section 6.4, where we respectively apply the ISAP v2 parameters to the bound for fixed position leakage and Hamming weight leakage.

5 Improved Bound for Leakage in Fixed Positions

We give a tightened NALR-PRF bound on the security of the SuKS when assuming leakage functions which leak bits in fixed positions. Formally, we restrict ourselves to the following leakage set $\mathcal{L}_{\text{fixed}}$ of functions of the form $\{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda$:

$$\mathcal{L}_{\text{fixed}} := \{(X,Y) \mapsto Z_{i_1}\|Z_{i_2}\|\cdots\|Z_{i_\lambda} \mid Z \in \{X,Y\} \wedge i_1 < i_2 < \cdots < i_\lambda \in \{1,2,\ldots,b\}\}\,.$$

We are now ready to state the NALR-PRF security bound of the SuKS in case of fixed position leakage.

Theorem 2. *Let F be the suffix keyed sponge described in Section 3.1 based on a random permutation $p \xleftarrow{\$} \mathsf{perm}(b)$ and function $G : \{0,1\}^k \times \{0,1\}^s \to \{0,1\}^s$. Assume that G is $2^{-\delta}$-uniform, $2^{-\varepsilon}$-universal and strongly protected.*

Let $\lambda \in \mathbb{N}$ such that $\lambda \le b$. Let $\mathcal{A}$ be an adversary who receives the output of a leakage function $L \in \mathcal{L}_{\text{fixed}}$ which leaks λ fixed bits of a secret suffix keyed sponge state. It holds for

any such adversary $\mathcal{A}$ with access to $q \geq 2$ construction queries and $N \leq 2^{b-1}$ primitive queries that:

$$\mathbf{Adv}^{\text{nalr-prf}}_{F,\mathcal{L}_{\text{fixed}}}(\mathcal{A}) \leq \frac{2N^2}{2^c} + \frac{\mu^{2(N-q)}_{b-s+\lambda,s-\lambda} \cdot N}{2^{\min\{\delta,\varepsilon\}-\lambda}} + \frac{\mu^{2q}_{t+\lambda,b-t-\lambda} \cdot N}{2^{b-t-\lambda}} \,. \tag{7}$$

Note that there are two changes with respect to the original NALR-PRF bound of (4) given in Theorem 1: the second and third term have changed, and the fourth term has disappeared. Regarding the second and third term: the parameters in the subscript of the multicollision limit function in the numerator have changed. Recall the problem described in Section 4 which prohibited the attacker from matching all elements in the multicollision with a single guess for a secret value, because these elements could have different leakage values. By changing the parameters of the multicollision limit function, the λ leaked bits are included in the multicollision, which eliminates this problem. For the second term, additionally, the multicollision limit function in the exponent of the denominator has disappeared because of the assumption that the leakage function always leaks bits in the same position, instead of different positions dependent on its input. This also means that the fourth term in Theorem 1, basically coming from bounding a multicollision on the leakage (this term corresponds to the right hand side of the inequality of Definition 3), disappears.

Note that the bound is tight in general, i.e., the leakage function (5) that makes the third term tight is in $\mathcal{L}_{\text{fixed}}$, and so is the leakage function (6) that makes the second term tight. Attacks that match the terms in the bound are given in Section 5.1.

Proof (of Theorem 2). The proof follows from the simple observation that the security of the SuKS under fixed position leakage of λ bits is *equivalent* to a black-box setting of the SuKS, but where the adversary knows $b - s + \lambda$ bits of the input to the last permutation (instead of $b - s$ bits) and $t + \lambda$ bits of its output (instead of t bits). Thus, the black-box bound of (3) of Theorem 1 carries over *almost immediately*: the only difference is in the denominator of the second term. Here, it should be observed that $\min\{\delta, \varepsilon\}$ regards the s leftmost bits of the state and not the $s - \lambda$ leftmost bits, and that by revealing λ more bits the term δ gets decreased by λ. The bound is then marginally simplified by noting that

$$\min\{\delta - \lambda, \varepsilon\} \geq \min\{\delta, \varepsilon\} - \lambda \,.$$

A more theoretical approach towards proving Theorem 2 would be to look at the proof of Dobraunig and Mennink [DM19b, Section 6.3], and observe that only the analysis of the bad events changes:

- $\mathsf{mc}_{\mathsf{tag}}$ and $\mathsf{coll}_{\mathsf{cp\text{-}out}}$ change by considering multicollisions at a state of $t + \lambda$ bits. For $\mathsf{coll}_{\mathsf{cp\text{-}out}}$, an additional change is in the observation that for each evaluation, the entropy loss due to leakage is already taken into account by virtually considering tags of size $t + \lambda$ bits;
- $\mathsf{mc}_{\mathsf{right}}$, $\mathsf{coll}_{\mathsf{cc}}$, and $\mathsf{coll}_{\mathsf{cp\text{-}in}}$ change by considering multicollisions at a state of $b - s + \lambda$ bits. For $\mathsf{coll}_{\mathsf{cp\text{-}in}}$, an additional change is in the observation that for each evaluation $G(K, \text{left}_s(U_i))$ exactly λ bits of leakage are known, instead of $\mu^{2(N-q)}_{s,b-s}$ pieces of leakage, a value analyzed by $\mathsf{mc}_{\mathsf{left}}$;
- Due to the previous point, bad event $\mathsf{mc}_{\mathsf{left}}$ has become redundant, and disappears.

□

5.1 Matching Attacks

We adapt the attacks given in Section 4.1 and Section 4.2 such that they exploit fixed position leakage and match terms of the NALR-PRF bound given in (7).

5.1.1 Attack Matching the Third Term

We use the leakage function L_p^{right} given in (5), which is repeated below:

$$L_p^{\text{right}} : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda ,$$
$$L_p^{\text{right}}(V, W) = \text{right}_\lambda(W) .$$

The attack is performed as follows. The attack is very similar to that in Section 4.1, the only differences being in the explicit inclusion of the leakage values in steps (1), (2), and (4).

(1) Make q construction queries with distinct plaintexts P_i to get the corresponding tags T_i and leakage values L_i, and make primitive queries on these same plaintexts to get the corresponding states before key absorption U_i.

(2) Find a multicollision of size μ for the tag and leakage value; that is, μ values i such that T_i is equal to some $T^\star$ and L_i is equal to some $L^\star$. Let $\mathcal{S}$ be the set containing these μ plaintexts.

(3) Additionally, find a set $\mathcal{S}'$ such that for all elements $P_i \in \mathcal{S}$, $\mathcal{S}'$ contains a different element P_i' with the same value for the outer part of the state U, i.e., $\text{left}_s(U_i) = \text{left}_s(U_i')$.

(4) For varying $Z_j \in \{0,1\}^{b-t-\lambda}$, make N inverse primitive queries $p^{-1}(T^\star \| Z_j \| L^\star)$. If the result of one of these queries is of the form $Y \| \text{right}_{b-s}(U_i)$ for some i in the multicollision, $\text{left}_s(V_i)$ is likely equal to Y. Note that $\text{left}_s(V_i) = \text{left}_s(V_i')$ because $\text{left}_s(U_i)$ and $\text{left}_s(U_i')$ collide.

(5) If V_i has been found for some i in the multicollision, compute

$$T' = \text{left}_t(p(\text{left}_s(V_i) \| \text{right}_{b-s}(U_i'))) .$$

Then, (P_i', T') is a valid forgery.

For each inverse primitive query $p^{-1}(T^\star \| Z_j \| L^\star)$, the attacker has a probability of $\frac{\mu}{2^{b-t-\lambda}}$ to correctly guess one of the values W_i in the multicollision and recover the corresponding V_i. Because the attacker makes N such queries and the size of the largest multicollision is bounded by $\mu^{2q}_{t+\lambda,b-t-\lambda}$, where in comparison to the previous attack now also the leakage is taken into account, the attack matches the bound of $(\mu^{2q}_{t+\lambda,b-t-\lambda} \cdot N)/2^{b-t-\lambda}$.

5.1.2 Attack Matching the Second Term

We use the leakage function L_p^{left} given in (6), which is repeated below:

$$L_p^{\text{left}} : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda ,$$
$$L_p^{\text{left}}(V, W) = \text{left}_\lambda(V) .$$

The attack is performed as follows. The attack differs from that in Section 4.2 in how the attacker exploits multicollisions on the inner part of the state. In the attack in Section 4.2, the attacker only has to find the largest multicollision on $\text{right}_{b-s}(U_i)$. However, when also exploiting leakage, the attacker must also find within the largest multicollision on $\text{right}_{b-s}(U_i)$ the largest set of plaintexts that collide on the leakage value, as described in step (3).

(1) Make primitive queries on distinct plaintexts P_i to get the corresponding states before key absorption U_i.

(2) Find a multicollision of plaintexts P_i such that for all corresponding states U_i, $\text{right}_{b-s}(U_i)$ is equal to some value $U^\star$.

(3) Make construction queries to compute the tags T_i and leakage values L_i corresponding to the plaintexts P_i in the multicollision. Then, within this multicollision, find another multicollision of μ plaintexts P_i such that the corresponding L_i are all equal to some $L^\star$. Let $\mathcal{S}$ be the set containing these μ plaintexts.

(4) Additionally, find a set $\mathcal{S}'$ such that for all elements $P_i \in \mathcal{S}$, $\mathcal{S}'$ contains a different element P_i' with the same value for the outer part of the state U, i.e., $\text{left}_s(U_i) = \text{left}_s(U_i')$.

(5) Find the state after key absorption V_i for some i by making N different guesses Z_j for the outer part of V_i, with $Z_j \in \{0,1\}^{s-\lambda}$. Verify each guess by checking whether $\text{left}_t(p(L^\star \| Z_j \| U^\star))$ equals T_i for some i. Note that $\text{left}_s(V_i) = \text{left}_s(V_i')$ because $\text{left}_s(U_i)$ and $\text{left}_s(U_i')$ collide.

(6) If V_i has been found for some i in the multicollision, compute

$$T' = \text{left}_t(p(\text{left}_s(V_i) \| \text{right}_{b-s}(U_i'))) \,.$$

Then (P_i', T') is a valid forgery.

The attacker has a probability of $\frac{\mu}{2^{\delta-\lambda}}$ to correctly guess one of the values V_i for each primitive query $p(L^\star \| Z_j \| U^\star)$. Because the attacker makes N such queries and the size of the largest multicollision is bounded by $\mu^{2(N-q)}_{b-s+\lambda,s-\lambda}$, where in comparison to the previous attack now also the leakage is taken into account, the attack matches the bound of $(\mu^{2(N-q)}_{b-s+\lambda,s-\lambda} \cdot N)/2^{\delta-\lambda}$. This bound differs slightly from the second term in its denominator. However, as explained in Section 4.2, this does not make a difference for usual instantiations of the SuKS.

5.2 Application to ISAP v2

The improvement in the bound becomes apparent when looking at the SuKS when instantiated with the parameters of ISAP v2 [DEM+17, DEM+20, DEM+21]. It runs on two different parameter sets, namely $(b, c, r, k) = (400, 256, 144, 128)$ (corresponding to instantiation with Keccak-f[400]) and $(b, c, r, k) = (320, 256, 64, 128)$ (corresponding to instantiation with Ascon-p), with in both cases $s = t = k$. We stress that in below application, we *only* take these parameter sets and analyze the generic bound of Theorem 2 for these parameters, but that the bound still only holds in the ideal permutation model and for the specific type of bounded non-adaptive leakage.

We stick to the instantiation using the Ascon-p-based parameters, and assume leakage of $\lambda = 3$ for the sake of example. We assume that G is strong and that $\min\{\delta, \varepsilon\} = k$. The original bound of Theorem 1 is of the simplified form

$$\mathbf{Adv}^{\text{nalr-prf}}_{F,\mathcal{L}}(\mathcal{A}) \leq \frac{2N^2}{2^{256}} + \frac{\mu^{2(N-q)}_{192,128} \cdot N}{2^{128-\mu^{2(N-q)}_{128,192}\lambda}} + \frac{\mu^{2q}_{128,192} \cdot N}{2^{189}} + \frac{\mu^{2(N-q)}_{128,192}}{2^{192}} \,. \tag{8}$$

If we set the number of balls in the multicollision limit functions to $2(N-q), 2q \leq 2^{129}$, we can use the script of Mennink [Men23, Appendix A] to estimate $\mu^{2(N-q)}_{192,128} \approx 5$, $\mu^{2(N-q)}_{128,192} \approx 80$, and $\mu^{2q}_{128,192} \approx 80$, and conclude that the bound becomes meaningless. The reason is that the multicollision limit function in the denominator of the second term of (8) grows quite large for N approaching the maximum. Taking a perhaps more realistic bounding for the

number of balls for *this (and only this)* multicollision function, $2(N-q) \leq 2^{65}$, we can estimate *that (and only that)* multicollision function as $\mu^{2(N-q)}_{128,192} \approx 5$ and obtain:

$$\mathbf{Adv}^{\text{nalr-prf}}_{F,\mathcal{L}_{\text{fixed}}}(\mathcal{A}) \leq \frac{2N^2}{2^{256}} + \frac{5N}{2^{113}} + \frac{80N}{2^{189}} + \frac{5}{2^{192}}\,. \tag{9}$$

If we restrict our focus to fixed position leakage, the improved bound of Theorem 2 is of the simplified form

$$\mathbf{Adv}^{\text{nalr-prf}}_{F,\mathcal{L}_{\text{fixed}}}(\mathcal{A}) \leq \frac{2N^2}{2^{256}} + \frac{\mu^{2(N-q)}_{195,125} \cdot N}{2^{125}} + \frac{\mu^{2q}_{131,189} \cdot N}{2^{189}}\,. \tag{10}$$

Using the script of Mennink to estimate $\mu^{2(N-q)}_{195,125} \approx 5$ and $\mu^{2q}_{131,189} \approx 50$, we obtain:

$$\mathbf{Adv}^{\text{nalr-prf}}_{F,\mathcal{L}_{\text{fixed}}}(\mathcal{A}) \leq \frac{2N^2}{2^{256}} + \frac{5N}{2^{125}} + \frac{50N}{2^{189}}\,. \tag{11}$$

Clearly, (11) improves over (9) in both its second and third term (the fourth term of (9) is negligible in the first place). A comparable improvement is obtained for the Keccak-f[400]-based parameter set. We do stress, however, that this instantiation is *only* done for the parameter sets

6 Improved Bound for Hamming Weight Leakage

In this section, we analyze how leakage of the Hamming weight of a value can be incorporated into the multicollision limit function. We remark that the Hamming weight is a logical type of leakage to consider. In particular, it is a reasonably realistic modeling of leakage [DMMS21, MOP07, May00]. In addition, the amount of information the attacker learns, as well as the size of the largest multicollision, depends on the leakage value. Consider leaking the Hamming weight w of an n-bit value X, with $n \in \mathbb{N}$ and $w \in \{0,\ldots,n\}$. The amount of possible n-bit values with Hamming weight w is $\binom{n}{w}$. Therefore, the difficulty of guessing X and the amount of n-bit values with the same Hamming weight as X depends on the value of w. Unfortunately, due to the non-uniform character, the analysis of leakage resilience in the Hamming weight setting becomes much more delicate.

We will consider $n \in \mathbb{N}$ such that $n \leq b$, and assume that the Hamming weight of n predetermined bits are leaked. Let $\lambda = \lceil \log(n+1) \rceil$. We define the following leakage set $\mathcal{L}_{\text{Hamming}}$ of functions of the form $\{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda$:

$$\mathcal{L}_{\text{Hamming}} := \{(X,Y) \mapsto \text{HW}(Z_{i_1} \| \cdots \| Z_{i_n}) \mid Z \in \{X,Y\} \wedge i_1 < \cdots < i_n \in \{1,\ldots,b\}\}\,.$$

We first model the idea of multicollisions covering Hamming weight leakage in Section 6.1. The section also includes a main theorem relating the novel multicollision limit function to the original one of Definition 3: Theorem 3. The proof of this theorem is given in Section 6.5. We derive an improved NALR-PRF security bound for the SuKS, and matching attacks for this bound, in Section 6.2 and Section 6.3 respectively.

6.1 Modeling Hamming Weight Into Multicollisions

6.1.1 Balls-and-Bins Problem

To model the Hamming weight leakage in a balls-and-bins problem, the balls and bins have to be redefined. We will consider a setting where the balls are b-bit values, and the attacker is trying to find collisions for a part a secret SuKS state, i.e., V or W. Without loss of generality, we consider it to learn the value of $\text{left}_r(W)$ and $\text{HW}(\text{right}_n(W))$. Hence,

we consider a bin for each possible combination of these values. Because the Hamming weight leakage function's codomain contains $n+1$ different values, there are $2^r \cdot (n+1)$ bins.

In the original balls-and-bins problem of the multicollision limit function, the balls are thrown with replacement according to a uniform distribution. In our redefined balls-and bins problem, the balls are thrown according to a non-uniform distribution denoted by $D_{\mathsf{HW\text{-}nr}}$. The distribution is non-uniform due to the balls being thrown without replacement, and due to the non-uniform distribution of the Hamming weight itself, which was explained at the start of Section 6.

Using the new balls-and-bins problem, we can now define a multicollision limit function which incorporates the following leakage function (the selection of which is without loss of generality):

$$\begin{aligned} &L_p^{\text{HW-right}} : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda, \\ &L_p^{\text{HW-right}}(V,W) = \text{HW}(\text{right}_n(W))\,. \end{aligned} \tag{12}$$

6.1.2 Multicollisions with Hamming Weight

In the Hamming weight leakage setting, the leakage value w which results in the largest multicollision on $\text{left}_r(W)$ and $\text{HW}(\text{right}_n(W))$ is likely to be a value which occurs frequently, such as $w = \lfloor \frac{n}{2} \rfloor$. However, with such a value for w it is more difficult to guess the actual value of $\text{right}_n(W)$ than with a value which occurs less often, such as $w = 0$. Therefore, the largest multicollision is not necessarily the multicollision which leads to the optimal attack; to obtain a tight bound in the Hamming weight leakage setting, we must analyze the size of the largest multicollision separately for each possible leakage value $w \in \{0, \ldots, n\}$.

In order to only analyze multicollisions on specific values for w, we define new distributions derived from $D_{\mathsf{HW\text{-}nr}}$. For $w \in \{0, \ldots, n\}$, let $D_{\mathsf{HW\text{-}nr}}(w)$ be a distribution such that balls are thrown in bins according to $D_{\mathsf{HW\text{-}nr}}$, but only the balls falling in the bins corresponding to the leakage value w are counted; the bins which do not correspond to the leakage value w can be considered bottomless, such that balls fall through them instead of in them. The maximum amount of balls in a single bin μ for the balls-and-bins problem where the balls are thrown according to $D_{\mathsf{HW\text{-}nr}}(w)$ can be described by

$$\mu_{r',c'}^{q,D_{\mathsf{HW\text{-}nr}}(w)}\,,$$

with $w \in \{0, \ldots, n\}$, $r' = r + \log(n+1)$ and $c' = c - \log(n+1)$.

According to the definition of the multicollision limit function in Section 2.5, we get from the subscript parameter r' that there are $2^{r'} = 2^{r+\log(n+1)} = 2^r \cdot (n+1)$ bins, and from the subscript parameter c' that we define this multicollision limit function as the smallest x satisfying:

$$\mathbf{Pr}\left(\mu > x\right) \leq \frac{x}{2^{c'}} = \frac{x}{2^{c-\log(n+1)}} = \frac{x(n+1)}{2^c}\,,$$

with μ the maximum amount of balls in any single bin.

6.1.3 Relation to (Uniform) Multicollision Limit Function

Due to the non-uniformity of the Hamming weight leakage function, the multicollision limit function defined in Section 6.1.2 is hard to compute. Therefore, we will prove that it is upper bounded by another multicollision limit function which uniformly distributes the balls over the bins (cf., Section 2.5):

Theorem 3. *Let $b, c, n, q, r \in \mathbb{N}$ and $w \in \{0, \ldots, n\}$ such that $b = r + c$, $1 \leq n \leq c$, $r \geq 1$ and $q \leq 2^b$. Let $r' = r + \log(n+1)$ and $c' = c - \log(n+1)$. Then $\mu_{r',c'}^{q, D_{\mathsf{HW}\text{-}nr}(w)} \leq \mu_{r',c'}^{\alpha(w)q}$ for $\alpha(w) = \max\left\{1, \left\lceil \binom{n}{w} \frac{e^2(n+1)}{2^n} \right\rceil\right\}$.*

The proof is postponed to Section 6.5.

6.2 Improved NALR-PRF Security Bound

We are now ready to state the NALR-PRF security bound of the SuKS in case of Hamming weight leakage.

Theorem 4. *Let F be the suffix keyed sponge described in Section 3.1 based on a random permutation $p \xleftarrow{\$} \mathsf{perm}(b)$ and function $G : \{0,1\}^k \times \{0,1\}^s \to \{0,1\}^s$, with parameters $b - s, t \geq 1$. Assume that G is $2^{-\delta}$-uniform, $2^{-\varepsilon}$-universal and strongly protected.*

Let $n, \lambda \in \mathbb{N}$ such that $1 \leq n \leq \min\{s, b-t\}$ and $\lambda = \lceil \log(n+1) \rceil$. For $w \in \{0, \ldots, n\}$, let $\alpha(w) = \max\left\{1, \left\lceil \binom{n}{w} \frac{e^2(n+1)}{2^n} \right\rceil\right\}$. Let $\mathcal{A}$ be an adversary who receives the output of a leakage function $L \in \mathcal{L}_{\mathrm{Hamming}}$ which leaks a λ-bit encoding of the Hamming weight of n bits of a secret suffix keyed sponge state. It holds for any such adversary $\mathcal{A}$ with access to $q \geq 2$ construction queries and $N \leq 2^{b-1}$ primitive queries that:

$$\mathbf{Adv}^{\text{nalr-prf}}_{F,\mathcal{L}_{\mathrm{Hamming}}}(\mathcal{A}) \leq \frac{2N^2}{2^c} + \max_w \frac{\mu_{b-s',s'}^{\alpha(w)(N-q)} \cdot N}{\binom{n}{w} 2^{\min\{\delta,\varepsilon\}-n}} + \max_w \frac{\mu_{t',b-t'}^{\alpha(w)q} \cdot N}{\binom{n}{w} 2^{b-t-n}}\,, \tag{13}$$

where $s' = s - \log(n+1)$ and $t' = t + \log(n+1)$.

Note that, just like Theorem 2, the second and third term have changed with respect to the original NALR-PRF bound given in Theorem 1. In these two terms, Theorem 3 has been applied so that the multicollision limit functions used in them take the non-uniform character of Hamming weight leakage into account. Furthermore, the fourth term has disappeared (the reason for its disappearance being the same as in Theorem 2). Also, this bound is tight in general, i.e., one can select two leakage functions from $\mathcal{L}_{\mathrm{Hamming}}$ similar to (5) and (6) that make the terms tight. Attacks that match the terms in the bound are given in Section 6.3.

Proof (of Theorem 4). The extension of the proof of Dobraunig and Mennink [DM19b, Section 6.3] to the setting of Hamming weight leakage is less straightforward. In particular, we cannot simply reduce security to the black-box setting as we did for Theorem 2. On the upside, the proof is still an adaptation of [DM19b, Section 6.3] where the difference is *only* in the bad event analysis.

In a nutshell, like in the proof of Theorem 2 we operate on multicollisions at a state of $t' = t + \log(n+1)$ or $b - s' = b - s + \log(n+1)$ bits. Then, for any adversarial attempt to set a bad event, the adversary has to judge based on the Hamming weight, and this gives a bias. For example, for guessing the output of a permutation, the adversary has to guess $b - t - n$ bits (with probability $1/2^{b-t-n}$) *and* the correct string matching the Hamming weight w (with probability $1/\binom{n}{w}$). Each guess may be for a different w, and the success probability is amplified by a multicollision limit function that depends on w in the number of balls (due to Theorem 3), in our case $\mu_{b-s',s'}^{\alpha(w)(N-q)}$ or $\mu_{t',b-t'}^{\alpha(w)q}$. In the worst case, the adversary restricts to leakage values w for which this probability is optimal. This observation results in the $\max_w$ in front of the second and third term.

In detail, consider any w. We can observe the following changes in the bad event analysis of [DM19b, Section 6.3]:

- cap considers inner part collisions before keying the construction. The event remains unchanged and contributes the following to the bound:

$$\frac{2(N-q)^2}{2^c}\,; \tag{14}$$

- $\mathsf{mc}_{\mathsf{tag}}$ and $\mathsf{coll}_{\mathsf{cp\text{-}out}}$ both get indexed by w. Both events then consider multicollisions at a state of $t' = t + \log(n+1)$ bits with $\alpha(w)q$ drawings.
  - For $\mathsf{mc}_{\mathsf{tag}}(w)$, for any w, this yields an updated probability bound of $\frac{\mu^{\alpha(w)q}_{t',b-t'}}{2^{b-t'}}$;
  - For $\mathsf{coll}_{\mathsf{cp\text{-}out}}(w)$, the analysis of forward queries $(+, X_j, Y_j)$ in [DM19b, Section 6.3] remains and gives $\frac{2q}{2^b}$ per query. For inverse queries $(-, X_j, Y_j)$, the value Y_j fixes the tag value as well as an n-bit string with Hamming weight w at certain fixed positions (w.l.o.g., at the unknown part). By $\neg\mathsf{mc}_{\mathsf{tag}}(w)$, there are at most $\mu^{\alpha(w)q}_{t',b-t'}$ earlier construction queries for the same tag and with the same Hamming weight at those positions. The value Y_j is then equal to W_i for any of those construction queries with probability at most $\frac{\mu^{\alpha(w)q}_{t',b-t'}}{\binom{n}{w}2^{b-t-n}}$ per query.

  Any of the at most $N-q$ attempts in $\mathsf{coll}_{\mathsf{cp\text{-}out}}$ in fact fixes w, and this means we can simply maximize over w, to get that these two bad events contribute the following to the bound:

$$\max_w \left(\frac{\mu^{\alpha(w)q}_{t',b-t'}}{2^{b-t'}} + \frac{\mu^{\alpha(w)q}_{t',b-t'} \cdot (N-q)}{\binom{n}{w}2^{b-t-n}} \right) + \frac{2q(N-q)}{2^b}\,; \tag{15}$$

- $\mathsf{mc}_{\mathsf{right}}$, $\mathsf{coll}_{\mathsf{cc}}$ and $\mathsf{coll}_{\mathsf{cp\text{-}in}}$ get indexed by w. These three events then consider multicollisions at a state of $b - s' = b - s + \log(n+1)$ bits with $\alpha(w)(N-q)$ drawings.
  - For $\mathsf{mc}_{\mathsf{right}}(w)$, for any w, this yields an updated probability bound of $\frac{\mu^{\alpha(w)(N-q)}_{b-s',s'}}{2^{s'}}$;
  - For $\mathsf{coll}_{\mathsf{cp\text{-}in}}(w)$, for forward queries $(+, X_j, Y_j)$, the value X_j fixes the inner part $\mathrm{right}_{b-s}(U_i)$ as well as an n-bit string with Hamming weight w at certain fixed positions (w.l.o.g., at the unknown part). By $\neg\mathsf{mc}_{\mathsf{right}}(w)$, there are at most $\mu^{\alpha(w)(N-q)}_{b-s',s'}$ earlier primitive queries for the same $\mathrm{right}_{b-s}(U_i)$ and with the same Hamming weight at those positions. The value X_j is then equal to U_i for any of those construction queries with probability at most $\frac{\mu^{\alpha(w)(N-q)}_{b-s',s'}}{\binom{n}{w}2^{\delta-n}}$ per query.

  Any of the at most $N-q$ attempts in $\mathsf{coll}_{\mathsf{cp\text{-}in}}$ in fact fixes w, and this means we can simply maximize over w, to get that these two bad events contribute the following to the bound:

$$\max_w \left(\frac{\mu^{\alpha(w)(N-q)}_{b-s',s'}}{2^{s'}} + \frac{\mu^{\alpha(w)(N-q)}_{b-s',s'} \cdot (N-q)}{\binom{n}{w}2^{\delta-n}} \right)\,; \tag{16}$$

  Finally, for $\mathsf{coll}_{\mathsf{cc}}(w)$, we consider construction-construction collisions: the leakage and also the indexing of w does not matter, and this bad event contributes the following to the bound (as in [DM19b, Section 6.3]):

$$\max_w \frac{\mu^{\alpha(w)(N-q)}_{b-s',s'} \cdot q/2}{2^\varepsilon} + \frac{q^2}{2^b}\,; \tag{17}$$

- For the same reason as in the proof of Theorem 2, bad event $\mathsf{mc}_{\mathsf{left}}$ has become redundant, and disappears.

If we sum (14), (15), (16) and (17), and simplify some terms, we obtain

$$\frac{2N^2}{2^c} + \max_w \frac{\mu_{b-s',s'}^{\alpha(w)(N-q)} \cdot N}{\binom{n}{w} 2^{\min\{\delta,\varepsilon\}-n}} + \max_w \frac{\mu_{t',b-t'}^{\alpha(w)q} \cdot N}{\binom{n}{w} 2^{b-t-n}}$$

as claimed. □

6.3 Matching Attacks

We adapt the attacks given in Section 4.1 and Section 4.2 such that they exploit Hamming weight leakage and match terms of the NALR-PRF bound given in (13).

6.3.1 Attack Matching the Third Term

We use the leakage function $L_p^{\text{HW-right}}$ given in (12), which is repeated below:

$$L_p^{\text{HW-right}} : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda,$$
$$L_p^{\text{HW-right}}(V,W) = \text{HW}(\text{right}_n(W)) .$$

The attack is performed as follows. Like the attack in Section 5.1.1, the only addition of this attack with respect to the attack in Section 4.1 is the explicit inclusion of the leakage values in steps (1), (2), and (4).

(1) Make q construction queries with distinct plaintexts P_i to get the corresponding tags T_i and leakage values L_i, and make primitive queries on these same plaintexts to get the corresponding states before key absorption U_i.

(2) Find the optimal multicollision of size μ for the tag and leakage value; that is, μ values i such that T_i is equal to some $T^\star$, L_i is equal to some $L^\star$ and $\mu/\binom{n}{w}$ is maximal, where w is the decimal representation of $L^\star$. Let $\mathcal{S}$ be the set containing these μ plaintexts.

(3) Additionally, find a set $\mathcal{S}'$ such that for all elements $P_i \in \mathcal{S}$, $\mathcal{S}'$ contains a different element P_i' with the same value for the outer part of the state U, i.e., $\text{left}_s(U_i) = \text{left}_s(U_i')$.

(4) For varying $Z_j \in \{0,1\}^{b-t}$ such that $\text{HW}(\text{right}_n(Z_j)) = L^\star$, make N inverse primitive queries $p^{-1}(T^\star \| Z_j)$. If the result of one of these queries is of the form $Y \| \text{right}_{b-s}(U_i)$ for some i in the multicollision, $\text{left}_s(V_i)$ is likely equal to Y. Note that $\text{left}_s(V_i) = \text{left}_s(V_i')$ because $\text{left}_s(U_i)$ and $\text{left}_s(U_i')$ collide.

(5) If V_i has been found for some i in the multicollision, compute

$$T' = \text{left}_t(p(\text{left}_s(V_i) \| \text{right}_{b-s}(U_i'))) .$$

Then, (P_i', T') is a valid forgery.

For each inverse primitive query $p^{-1}(T^\star \| Z_j)$, the attacker guesses one of the values W_i in the multicollision and recovers the corresponding V_i with probability $\mu/\left(\binom{n}{w} 2^{b-t-n}\right)$. Because the attacker makes N such queries and the size of the optimal multicollision is bounded by $\mu_{t',b-t'}^{\alpha(w)q}$ with $t' = t + \log(n+1)$, where in comparison to the previous attack now also the leakage is taken into account, the attack matches the bound of $(\mu_{t',b-t'}^{\alpha(w)q} \cdot N)/\left(\binom{n}{w} 2^{b-t-n}\right)$ for the value of w that maximizes the bound.

6.3.2 Attack Matching the Second Term

We use the leakage function $L_p^{\text{HW-left}}$ defined below:

$$\begin{aligned} &L_p^{\text{HW-left}} : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda, \\ &L_p^{\text{HW-left}}(V, W) = \text{HW}(\text{left}_n(V)) \,. \end{aligned} \tag{18}$$

The attack is performed as follows. The attack differs from that in Section 4.2 in how the attacker exploits multicollisions on the inner part of the state, similarly to Section 5.1.2: besides finding the largest multicollision $\text{right}_{b-s}(U_i)$, the attacker must also find within this multicollision the optimal set of plaintexts that collide on the leakage value, as described in step (3).

(1) Make primitive queries on distinct plaintexts P_i to get the corresponding states before key absorption U_i.

(2) Find a multicollision of plaintexts P_i such that for all corresponding states U_i, $\text{right}_{b-s}(U_i)$ is equal to some value $U^\star$.

(3) Make construction queries to compute the tags T_i and leakage values L_i corresponding to the plaintexts P_i in the multicollision. Then, within this multicollision, find another multicollision of μ plaintexts P_i such that the corresponding L_i are all equal to some $L^\star$ and $\mu/\binom{n}{w}$ is maximal, where w is the decimal representation of $L^\star$. Let $\mathcal{S}$ be the set containing these μ plaintexts.

(4) Additionally, find a set $\mathcal{S}'$ such that for all elements $P_i \in \mathcal{S}$, $\mathcal{S}'$ contains a different element P_i' with the same value for the outer part of the state U, i.e., $\text{left}_s(U_i) = \text{left}_s(U_i')$.

(5) Find the state after key absorption V_i for some i by making N different guesses Z_j for the outer part of V_i, with $Z_j \in \{0,1\}^s$ such that $\text{HW}(\text{left}_n(Z_j)) = L^\star$. Verify each guess by checking whether $\text{left}_t(p(Z_j \| U^\star))$ equals T_i for some i. Note that $\text{left}_s(V_i) = \text{left}_s(V_i')$ because $\text{left}_s(U_i)$ and $\text{left}_s(U_i')$ collide.

(6) If V_i has been found for some i in the multicollision, compute

$$T' = \text{left}_t(p(\text{left}_s(V_i) \| \text{right}_{b-s}(U_i'))) \,.$$

Then (P_i', T') is a valid forgery.

The attacker has a probability of $\mu / \left(\binom{n}{w} 2^{\delta-n}\right)$ to correctly guess one of the values V_i for each primitive query $p(Z_j \| U^\star)$. Because the attacker makes N such queries and the size of the optimal multicollision is bounded by $\mu_{b-s',s'}^{\alpha(w)(N-q)}$ with $s' = s - \log(n+1)$, where in comparison to the previous attack now also the leakage is taken into account, the attack matches the bound of $(\mu_{b-s',s'}^{\alpha(w)(N-q)} \cdot N) / \left(\binom{n}{w} 2^{\delta-n}\right)$. This bound differs slightly from the second term in its denominator. However, as explained in Section 4.2, this does not make a difference for usual instantiations of the SuKS.

6.4 Application to ISAP v2

Like in Section 5.2, we look at the SuKS instantiated with the Ascon-p-based ISAP v2 parameters $(b, c, r, k) = (320, 256, 64, 128)$ with $s = t = k$ to demonstrate the improvement in the bound. Again, we stress that, even though we analyze the generic bound of Theorem 4 for specific parameter sets, the bound remains to hold only in the ideal permutation model and for the specific type of bounded non-adaptive leakage.

We use the same assumptions as in Section 5.2: we assume that the amount of leakage is bounded by $\lambda = 3$, that G is strong and that $\min\{\delta, \varepsilon\} = k$. Furthermore, we again set the number of balls in the multicollision limit function to $2(N-q), 2q \leq 2^{129}$ to obtain the same bound from Theorem 1 as given in (9). For convenience, we repeat that bound below.

$$\mathbf{Adv}^{\text{nalr-prf}}_{F,\mathcal{L}}(\mathcal{A}) \leq \frac{2N^2}{2^{256}} + \frac{5N}{2^{113}} + \frac{80N}{2^{189}} + \frac{5}{2^{192}}\,.$$

If we restrict our focus to Hamming weight leakage, the improved bound of Theorem 4 is of the form

$$\mathbf{Adv}^{\text{nalr-prf}}_{F,\mathcal{L}_{\text{Hamming}}}(\mathcal{A}) \leq \frac{2N^2}{2^{256}} + \max_w \frac{\mu^{\alpha(w)(N-q)}_{b-s',s'} \cdot N}{\binom{n}{w} 2^{121}} + \max_w \frac{\mu^{\alpha(w)q}_{t',b-t'} \cdot N}{\binom{n}{w} 2^{185}}\,. \tag{19}$$

To further simplify the bound, we must specify the amount of bits n for which the adversary learns the Hamming weight. We choose $n = 7$, which is the largest value n such that all possible leakage values can be encoded in $\lambda = 3$ bits, and therefore the optimal value for the adversary in this case.

Additionally, we must find for the second and third term separately the value w which maximizes the term. We know that $N-q, q \leq 2^{128}$ and that $\alpha(w) = \max\left\{1, \left\lceil \binom{n}{w} \frac{e^2(n+1)}{2^n} \right\rceil\right\}$. We have extended the script of Mennink [Men23, Appendix A] such that it calculates the value of $\mu^{\alpha(w)(N-q)}_{b-s',s'} / \binom{n}{w}$ and $\mu^{\alpha(w)q}_{t',b-t'} / \binom{n}{w}$ for each $w \in \{0, \ldots, n\}$. This script is given in Appendix A. Using this script, we find that the value $w = 0$ maximizes both terms. For $w = 0$, the bound is of the simplified form

$$\mathbf{Adv}^{\text{nalr-prf}}_{F,\mathcal{L}_{\text{Hamming}}}(\mathcal{A}) \leq \frac{2N^2}{2^{256}} + \frac{5N}{2^{121}} + \frac{45N}{2^{185}}\,. \tag{20}$$

We can observe that this bound also improves over (9) (repeated above), noting that the second term is dominant and the third term is inferior to the other terms. A possible explanation for this perhaps surprising behavior may be that the bounding of $\alpha(w)$ is a bit loose for the general case, but for a typical use case as the SuKS, this does not appear to be a problem.

6.5 Proof of Theorem 3

The proof is inspired by that of Daemen et al. [DMV17, Section 6.6], but the analysis is significantly different due to the fact that Hamming weight leakage is non-uniform.

Consider two ball-and-bins experiments:

(1) We throw $\alpha(w)q$ balls into $2^{r'}$ bins according to a uniform distribution with replacement.

(2) We throw q balls into $2^{r'}$ bins according to the distribution $D_{\mathsf{HW\text{-}nr}}(w)$ without replacement.

Note that the maximum number of balls in any single bin in Experiment 1 and Experiment 2 are bounded by $\mu^{\alpha(w)q}_{r',c'}$, and $\mu^{q,D_{\mathsf{HW\text{-}nr}}(w)}_{r',c'}$ respectively. Let X^{exp1}_i and X^{exp2}_i denote the number of balls in bin i in the respective experiments, for $1 \leq i \leq 2^{r'}$. Let μ^{exp1} and μ^{exp2} denote the highest number of balls in any single bin in the respective experiments.

To prove Theorem 3, we will apply the same strategy as used by Daemen et al. [DMV17, Section 6.6]. First, we prove that $\mu^{\alpha(w)q}_{r',c'}$ has some threshold t as lower bound. We can assume that $t > 0$, because it can only be the case that $\mu^{\alpha(w)q}_{r',c'} = 0$ if $q = 0$, and then

Theorem 3 trivially holds. By the pigeonhole principle, there must be a bin in Experiment 1 containing at least $\left\lceil \frac{\alpha(w)q}{2^{r'}} \right\rceil$ balls. Hence, for $t = \left\lceil \frac{\alpha(w)q}{2^{r'}} \right\rceil$, $\mu_{r',c'}^{\alpha(w)q} \geq t$ holds.

Then, we will prove that, for all $y \geq t$,

$$\mathbf{Pr}\left(\mu^{\text{exp1}} > y\right) \geq \mathbf{Pr}\left(\mu^{\text{exp2}} > y\right) . \tag{21}$$

This is a useful result because of the following lemma:

Lemma 1. *If* $\boldsymbol{Pr}\left(\mu^{\text{exp1}} > y\right) \geq \boldsymbol{Pr}\left(\mu^{\text{exp2}} > y\right)$ *for all* $y \geq t$, *then* $\mu_{r',c'}^{q,D_{\text{HW-nr}}(w)} \leq \mu_{r',c'}^{\alpha(w)q}$.

Proof.

(1) Assume that $\mathbf{Pr}\left(\mu^{\text{exp1}} > y\right) \geq \mathbf{Pr}\left(\mu^{\text{exp2}} > y\right)$ for all $y \geq t$.

(2) From (1) it follows that $\mathbf{Pr}\left(\mu^{\text{exp1}} > y\right) \geq \mathbf{Pr}\left(\mu^{\text{exp2}} > y\right)$ holds for $y = \mu_{r',c'}^{\alpha(w)q}$.

(3) By definition (see Section 2.5), $\mu_{r',c'}^{\alpha(w)q}$ is the smallest number x such that $\mathbf{Pr}\left(\mu^{\text{exp1}} > x\right) < \frac{x}{2^{c'}}$.

(4) From (2) and (3) it follows that $\mathbf{Pr}\left(\mu^{\text{exp1}} > x\right) \geq \mathbf{Pr}\left(\mu^{\text{exp2}} > x\right)$.

(5) From (3) and (4) it follows that $\mathbf{Pr}\left(\mu^{\text{exp2}} > x\right) < \frac{x}{2^{c'}}$.

(6) By definition (see Section 6.1.2), $\mu_{r',c'}^{q,D_{\text{HW-nr}}(w)}$ is the smallest number x' such that $\mathbf{Pr}\left(\mu^{\text{exp2}} > x'\right) < \frac{x'}{2^{c'}}$.

(7) From (5) and (6) it follows that $x' \leq x$.

(8) From (3), (6) and (7) we can conclude that $\mu_{r',c'}^{q,D_{\text{HW-nr}}(w)} \leq \mu_{r',c'}^{\alpha(w)q}$. □

Because of Lemma 1, Theorem 3 follows if we prove that (21) holds for all $y \geq t$. This in turn can be proven by showing that

$$\mathbf{Pr}\left(X_i^{\text{exp1}} > y\right) \geq \mathbf{Pr}\left(X_i^{\text{exp2}} > y\right) \tag{22}$$

holds for all $y \geq t$ and for all bins i; if each bin i in Experiment 1 is at least as likely to contain more than y balls as each bin i in Experiment 2, then certainly the single bin with the most balls in Experiment 1 is at least as likely to contain more than y balls as the single bin with the most balls in Experiment 2.

In turn, (22) can be proven by demonstrating that

$$\mathbf{Pr}\left(X_i^{\text{exp1}} = y\right) \geq \mathbf{Pr}\left(X_i^{\text{exp2}} = y\right) \tag{23}$$

holds for all $y \geq t$ and for all bins i; if each bin i in Experiment 1 is at least as likely to contain y balls as each bin i in Experiment 2, for all $y \geq t$, then each bin i in Experiment 2 is also at least as likely to contain more than y balls as each bin in Experiment 2.

Therefore, to show that Theorem 3 holds, it remains to be proven that (23) holds for all $y \geq t$ and for all bins i. We first determine these two probabilities $\mathbf{Pr}\left(X_i^{\text{exp1}} = y\right)$ and $\mathbf{Pr}\left(X_i^{\text{exp2}} = y\right)$.

6.5.1 Probability in Experiment 1

The probability of a single ball falling in bin i is $2^{-r'}$, and y balls need to fall in this bin. The probability of a single ball ending up in any bin except bin i is $1-2^{-r'}$, and this needs to occur for the remaining $\alpha(w)q-y$ balls. Finally, there are $\binom{\alpha(w)q}{y}$ ways to choose the y balls which fall in bin i.

We thus have

$$\mathbf{Pr}\left(X_i^{\text{exp1}}=y\right)=\binom{\alpha(w)q}{y}(2^{-r'})^y(1-2^{-r'})^{\alpha(w)q-y}\,. \tag{24}$$

6.5.2 Probability in Experiment 2

Since there are $(n+1)\cdot 2^r$ bins in total, there are 2^r bins per Hamming weight value $0,\ldots,n$. To model this, we define the Hamming weight value of bin i as $j:=i \bmod (n+1)$.

For each bin i, there are $\binom{n}{j}$ n-bit values with the correct Hamming weight, the r left-most bits are fixed to one value and there are 2^{b-r-n} possible values for the remaining $b-r-n$ bits. Therefore, there are a total of $\binom{n}{j}2^{b-r-n}=\binom{n}{j}2^{c-n}$ balls which belong to bin i and $2^b-\binom{n}{j}2^{c-n}$ balls which do not belong to bin i.

By definition of $D_{\mathsf{HW\text{-}nr}}(w)$ (see Section 6.1.2) and because $y\geq t>0$, we know that $\mathbf{Pr}\left(X_i^{\text{exp2}}=y\right)=0$ for all i such that $j\neq w$. For the remaining bins, we make a case distinction on y:

(1) We consider the case that y is greater than $\binom{n}{j}2^{c-n}$, i.e., the amount of balls that fit in bin i. In this case, the probability of bin i containing y balls is zero.

(2) We consider the case that y is less than or equal to the amount of balls that fit in bin i. In this case, y balls need to fall in bin i and $q-y$ in the other bins, and there are $\binom{q}{y}$ ways to choose the y balls which fall in bin i.

Because the balls are sampled without replacement, the probabilities change with each ball thrown, since the total amount of balls to sample from and the amount of space left in one of the bins decrease with each ball thrown. The falling factorial is used in (25) to account for this. We thus have

$$\mathbf{Pr}\left(X_i^{\text{exp2}}=y\right)=\begin{cases}0 & \text{if } j\neq w \text{ or } y>\binom{n}{j}2^{c-n}\,,\\ \binom{q}{y}\frac{\left(\binom{n}{j}2^{c-n}\right)_y\left(2^b-\binom{n}{j}2^{c-n}\right)_{q-y}}{(2^b)_q} & \text{if } j=w \text{ and } y\leq\binom{n}{j}2^{c-n}\,.\end{cases} \tag{25}$$

6.5.3 Proving the Inequality (23)

Now, we will show that (23) holds, i.e., that $\mathbf{Pr}\left(X_i^{\text{exp1}}=y\right)\geq\mathbf{Pr}\left(X_i^{\text{exp2}}=y\right)$ for all bins i. Note that this inequality would trivially hold in the first case of (25); in this case $\mathbf{Pr}\left(X_i^{\text{exp2}}=y\right)$ is equal to 0, while $\mathbf{Pr}\left(X_i^{\text{exp1}}=y\right)$ is always at least 0. Therefore, it only remains to prove the inequality in the second case of (25). By substituting the values of (24) and (25) into (23), we get the following inequality:

$$\binom{\alpha(w)q}{y}(2^{-r'})^y(1-2^{-r'})^{\alpha(w)q-y}\overset{?}{\geq}\binom{q}{y}\frac{\left(\binom{n}{j}2^{c-n}\right)_y\left(2^b-\binom{n}{j}2^{c-n}\right)_{q-y}}{(2^b)_q}\,. \tag{26}$$

We prove that (26) holds by using the assumptions in Theorem 3, namely $1\leq n\leq c$ and $r\geq 1$. In the remainder of the proof, we will use the following three lemmas:

Lemma 2. *Let $A, B, n \in \mathbb{N}$. Assume that $A \geq B \geq n$. Then, it holds that*

$$\frac{(A)_n}{(B)_n} \geq \left(\frac{A}{B}\right)^n .$$

Lemma 3. $\left(1 - \frac{1}{x}\right)^x \geq e^{-\frac{x}{x-1}}$ *for all* $x \in \mathbb{R}^+ \setminus \{1\}$.

Lemma 4. $e^{\frac{2^x}{2^x-1}} \leq e^2$ *on the interval* $[1, \infty)$.

The proofs of these lemmas are given in Appendix B.

6.5.4 Proving the Inequality (26)

Using that $\binom{n}{r} = \frac{n!}{r!(n-r)!} = \frac{(n)_r}{r!}$, we get:

$$\frac{(\alpha(w)q)_y}{y!}(2^{-r'})^y(1-2^{-r'})^{\alpha(w)q-y} \overset{?}{\geq} \frac{(q)_y}{y!}\frac{\left(\binom{n}{j}2^{c-n}\right)_y\left(2^b-\binom{n}{j}2^{c-n}\right)_{q-y}}{(2^b)_q},$$
$$\frac{(\alpha(w)q)_y}{(q)_y}(2^{-r'})^y(1-2^{-r'})^{\alpha(w)q-y} \overset{?}{\geq} \frac{\left(\binom{n}{j}2^{c-n}\right)_y\left(2^b-\binom{n}{j}2^{c-n}\right)_{q-y}}{(2^b)_q}.$$

Using that $b = r' + c' \implies -r' = c' - b$, we get:

$$\frac{(\alpha(w)q)_y}{(q)_y}(2^{c'-b})^y(1-2^{c'-b})^{\alpha(w)q-y} \overset{?}{\geq} \frac{\left(\binom{n}{j}2^{c-n}\right)_y\left(2^b-\binom{n}{j}2^{c-n}\right)_{q-y}}{(2^b)_q},$$
$$\frac{(\alpha(w)q)_y}{(q)_y}\left(\frac{2^{c'}}{2^b}\right)^y\left(1-\frac{2^{c'}}{2^b}\right)^{\alpha(w)q-y} \overset{?}{\geq} \frac{\left(\binom{n}{j}2^{c-n}\right)_y\left(2^b-\binom{n}{j}2^{c-n}\right)_{q-y}}{(2^b)_y\,(2^b-y)_{q-y}},$$
$$\frac{(\alpha(w)q)_y}{(q)_y}\left(\frac{2^{c'}}{2^b}\right)^y\left(\frac{2^b}{2^b}-\frac{2^{c'}}{2^b}\right)^{\alpha(w)q-y} \overset{?}{\geq} \frac{\left(\binom{n}{j}2^{c-n}\right)_y}{(2^b)_y}\frac{\left(2^b-\binom{n}{j}2^{c-n}\right)_{q-y}}{(2^b-y)_{q-y}},$$
$$\frac{(\alpha(w)q)_y}{(q)_y}\left(\frac{2^{c'}}{2^b}\right)^y\left(\frac{2^b-2^{c'}}{2^b}\right)^{\alpha(w)q-y} \overset{?}{\geq} \frac{\left(\binom{n}{j}2^{c-n}\right)_y}{(2^b)_y}\frac{\left(2^b-\binom{n}{j}2^{c-n}\right)_{q-y}}{(2^b-y)_{q-y}},$$
$$\frac{(\alpha(w)q)_y}{(q)_y}\left(\frac{2^{c'}}{2^b}\right)^y\left(\frac{2^b-2^{c'}}{2^b}\right)^{\alpha(w)q}\left(\frac{2^b}{2^b-2^{c'}}\right)^y \overset{?}{\geq} \frac{\left(\binom{n}{j}2^{c-n}\right)_y}{(2^b)_y}\frac{\left(2^b-\binom{n}{j}2^{c-n}\right)_{q-y}}{(2^b-y)_{q-y}},$$
$$\frac{(\alpha(w)q)_y}{(q)_y}\left(\frac{2^{c'}}{2^b-2^{c'}}\right)^y\frac{\left(2^b\right)_y}{\left(\binom{n}{j}2^{c-n}\right)_y}\frac{\left(2^b-y\right)_{q-y}}{\left(2^b-\binom{n}{j}2^{c-n}\right)_{q-y}} \overset{?}{\geq} \left(\frac{2^b}{2^b-2^{c'}}\right)^{\alpha(w)q}.$$

The following properties hold:

- $\alpha(w)q \geq q$ because $\alpha(w) = \max\left\{1, \left\lceil \binom{n}{w}\frac{e^2(n+1)}{2^n}\right\rceil\right\}$.
- $2^b \geq \binom{n}{j}2^{c-n}$ since $2^b > 2^c = 2^n \cdot 2^{c-n} = \left(\sum_{k=0}^{n}\binom{n}{k}\right)\cdot 2^{c-n} > \binom{n}{j}2^{c-n}$,
- $2^b - y \geq 2^b - \binom{n}{j}2^{c-n}$ because $y \leq \binom{n}{j}2^{c-n}$.

Using these properties and the fact that $\frac{(A)_x}{(B)_x} \geq \left(\frac{A}{B}\right)^x$ if $A \geq B \geq x$ (see Lemma 2), we get:

$$\left(\frac{\alpha(w)q}{q}\right)^y\left(\frac{2^{c'}}{2^b-2^{c'}}\right)^y\left(\frac{2^b}{\binom{n}{j}2^{c-n}}\right)^y\left(\frac{2^b-y}{2^b-\binom{n}{j}2^{c-n}}\right)^{q-y} \overset{?}{\geq} \left(\frac{2^b}{2^b-2^{c'}}\right)^{\alpha(w)q},$$

$$\alpha(w)^y \left(\frac{2^{c'}}{2^b-2^{c'}}\right)^y \left(\frac{2^b}{\binom{n}{j}2^{c-n}}\right)^y \left(\frac{2^b-\binom{n}{j}2^{c-n}}{2^b-y}\right)^y \stackrel{?}{\geq} \left(\frac{2^b}{2^b-2^{c'}}\right)^{\alpha(w)q} \left(\frac{2^b-\binom{n}{j}2^{c-n}}{2^b-y}\right)^q,$$
$$\left(\alpha(w)\frac{2^{c'}}{2^b-2^{c'}}\frac{2^b}{\binom{n}{j}2^{c-n}}\frac{2^b-\binom{n}{j}2^{c-n}}{2^b-y}\right)^y \stackrel{?}{\geq} \left(\frac{2^b}{2^b-2^{c'}}\right)^{\alpha(w)q} \left(\frac{2^b-\binom{n}{j}2^{c-n}}{2^b-y}\right)^q.$$

Using that $b = r' + c'$ and $b = r + c$, we get:

$$\left(\alpha(w)\frac{1}{2^{r'}-1}\frac{1}{\binom{n}{j}2^{-r-n}}\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right)^y \stackrel{?}{\geq} \left(\frac{1}{1-2^{-r'}}\right)^{\alpha(w)q} \left(\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right)^q,$$
$$\left(\frac{\alpha(w)}{2^{r'}-1}\frac{1}{\binom{n}{j}2^{-(r+n)}}\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right)^y \stackrel{?}{\geq} \left(\frac{1^{\alpha(w)}}{(1-2^{-r'})^{\alpha(w)}}\right)^q \left(\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right)^q,$$
$$\left(\frac{\alpha(w)}{2^{r'}-1}\frac{2^{r+n}}{\binom{n}{j}}\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right)^y \stackrel{?}{\geq} \left(\frac{1-\binom{n}{j}2^{-r-n}}{(1-2^{-r'})^{\alpha(w)}\left(1-\frac{y}{2^b}\right)}\right)^q. \tag{27}$$

To further simplify the inequality (27), we first need to show that the base of the left hand side is at least 1:

$$\frac{\alpha(w)}{2^{r'}-1}\frac{2^{r+n}}{\binom{n}{j}}\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}} \stackrel{?}{\geq} 1. \tag{28}$$

Recall that y is the amount of balls in bin i, so $y \geq 0$. We know that the left hand side of (28) is minimal when $y = 0$. Therefore, if we can prove that (28) holds for $y = 0$, we know that it must hold for every possible value of y. We substitute $y = 0$ in (28):

$$\frac{\alpha(w)}{2^{r'}-1}\frac{2^{r+n}}{\binom{n}{j}}\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{0}{2^b}} \stackrel{?}{\geq} 1,$$
$$\frac{\alpha(w)}{2^{r'}-1}\frac{2^{r+n}}{\binom{n}{j}}\left(1-\binom{n}{j}2^{-r-n}\right) \stackrel{?}{\geq} 1,$$
$$\frac{\alpha(w)}{2^{r'}-1}\left(\frac{2^{r+n}}{\binom{n}{j}}-\frac{\binom{n}{j}}{\binom{n}{j}}\right) \stackrel{?}{\geq} 1,$$
$$\frac{\alpha(w)}{2^{r'}-1}\frac{2^{r+n}-\binom{n}{j}}{\binom{n}{j}} \stackrel{?}{\geq} 1,$$
$$\frac{\alpha(w)}{2^{r'}-1} \stackrel{?}{\geq} \frac{\binom{n}{j}}{2^{r+n}-\binom{n}{j}},$$
$$\alpha(w) \stackrel{?}{\geq} \frac{2^{r'}\binom{n}{j}-\binom{n}{j}}{2^{r+n}-\binom{n}{j}},$$
$$\alpha(w) \stackrel{?}{\geq} \frac{2^{r+\log(n+1)}\binom{n}{j}-\binom{n}{j}}{2^{r+n}-\binom{n}{j}},$$
$$\alpha(w) \stackrel{?}{\geq} \frac{2^r(n+1)\binom{n}{j}-\binom{n}{j}}{2^{r+n}-\binom{n}{j}}.$$

Therefore, the base of the left hand side of (27) is at least 1 under the *condition* that:

$$\alpha(w) \geq \frac{2^r(n+1)\binom{n}{j}-\binom{n}{j}}{2^{r+n}-\binom{n}{j}}. \tag{29}$$

We will show that this condition holds later in the proof.

Now, it suffices to show that (27) holds for $y = t = \left\lceil \frac{\alpha(w)q}{2^{r'}} \right\rceil$ as left hand side exponent instead of for all $y \geq t$. This is because the left hand side of (27) is minimal for $y = t$ if the base of the left hand side is at least 1, and we have just shown that this is the case if $\alpha(w) \geq \frac{2^r(n+1)\binom{n}{j} - \binom{n}{j}}{2^{r+n} - \binom{n}{j}}$.

This final demonstration of (27) is quite elaborate, and has been deferred to Appendix C to improve the readability of this section. The result of this demonstration is that the following *condition* on $\alpha(w)$ must be satisfied for (27) to hold:

$$\alpha(w) \geq \binom{n}{j} \frac{e^2(n+1)}{2^n} \,. \tag{30}$$

6.5.5 Proving Conditions (29) and (30)

Now, finally, to show that (26) holds, it remains to be proven that $\alpha(w)$, defined to be $\alpha(w) = \max\left\{1, \left\lceil \binom{n}{w} \frac{e^2(n+1)}{2^n} \right\rceil\right\}$ in the statement of Theorem 3, satisfies the two conditions given in (29) and (30). We will prove a slightly stronger claim, namely that the following holds:

$$\frac{2^r(n+1)\binom{n}{j} - \binom{n}{j}}{2^{r+n} - \binom{n}{j}} \overset{\star}{\leq} \binom{n}{j} \frac{e^2(n+1)}{2^n} \overset{\star\star}{\leq} \max\left\{1, \left\lceil \binom{n}{w} \frac{e^2(n+1)}{2^n} \right\rceil\right\} .$$

Inequality $\overset{\star}{\leq}$. The inequality can be simplified as follows:

$$\begin{aligned}
\frac{2^r(n+1)\binom{n}{j} - \binom{n}{j}}{2^{r+n} - \binom{n}{j}} &\overset{?}{\leq} \binom{n}{j} \frac{e^2(n+1)}{2^n} \,, \\
\frac{2^r(n+1) - 1}{2^{r+n} - \binom{n}{j}} &\overset{?}{\leq} \frac{e^2(n+1)}{2^n} \,, \\
2^{r+n}(n+1) - 2^n &\overset{?}{\leq} 2^{r+n}e^2(n+1) - \binom{n}{j} e^2(n+1) \,, \\
-2^n &\overset{?}{\leq} 2^{r+n}e^2(n+1) - \binom{n}{j} e^2(n+1) - 2^{r+n}(n+1) \,, \\
2^n &\overset{?}{\geq} \binom{n}{j} e^2(n+1) + 2^{r+n}(n+1) - 2^{r+n}e^2(n+1) \,, \\
2^n &\overset{?}{\geq} (n+1)\left(\binom{n}{j} e^2 + 2^{r+n} - 2^{r+n}e^2\right) \,, \\
\frac{2^n}{n+1} &\overset{?}{\geq} \binom{n}{j} e^2 + 2^{r+n}(1 - e^2) \,.
\end{aligned} \tag{31}$$

Recall the assumptions $r \geq 1$ and $n \geq 1$. We know that the term $2^{r+n}(1 - e^2)$ is negative since 2^{r+n} is positive and $1 - e^2$ is negative. Therefore, the right-hand side of the above inequality (31) is largest when $r = 1$. It follows that we can substitute $r = 1$ in (31), because if it holds for $r = 1$, then it must certainly hold for all $r \geq 1$:

$$\begin{aligned}
\frac{2^n}{n+1} &\overset{?}{\geq} \binom{n}{j} e^2 + 2^{1+n}(1 - e^2) \,, \\
\frac{2^n}{n+1} &\overset{?}{\geq} \binom{n}{j} e^2 + 2^n(2(1 - e^2)) \,,
\end{aligned}$$

$$\frac{2^n}{n+1} \stackrel{?}{\geq} \binom{n}{j} e^2 + 2^n(2-2e^2)\,.$$

We use that $\binom{n}{j} < \sum_{k=0}^{n} \binom{n}{k} = 2^n$ to replace $\binom{n}{j}$ with 2^n:

$$\frac{2^n}{n+1} \stackrel{?}{\geq} 2^n e^2 + 2^n(2-2e^2)\,,$$
$$\frac{2^n}{n+1} \stackrel{?}{\geq} 2^n(2-e^2)\,.$$

Because $n \geq 1$, the left hand side of this inequality is positive. Since $2 \leq e^2$, the right-hand side of this inequality is negative. Therefore, the inequality holds.

Inequality $\stackrel{\star\star}{\leq}$. Now, it only remains to be shown that the second inequality on $\alpha(w)$ is satisfied. Hence, we must show that $\binom{n}{j}\frac{e^2(n+1)}{2^n}$ is less than or equal to $\alpha(w)$. Using that $j = w$, we get:

$$\binom{n}{w}\frac{e^2(n+1)}{2^n} \stackrel{?}{\leq} \max\left\{1, \left\lceil \binom{n}{w}\frac{e^2(n+1)}{2^n} \right\rceil \right\}\,,$$
$$\binom{n}{w}\frac{e^2(n+1)}{2^n} \stackrel{?}{\leq} \max\left\{1, \binom{n}{w}\frac{e^2(n+1)}{2^n} \right\}\,.$$

Because the left-hand side of this inequality is one of the operands of the max operator on the right-hand side, this inequality trivially holds.

7 Conclusion

To the best of our knowledge, there had not been any earlier demonstration of tightness of a leakage resilience analysis. Our result tightens the leakage resilience bound for the SuKS under non-adaptive fixed position leakage and under non-adaptive Hamming weight leakage in the ideal permutation model. For the case of fixed position leakage, it would be interesting (though non-trivial) to investigate the power that an adversary has if it can adapt the leakage positions in-between queries. This would further close the gap between our theoretical analysis and typical probing attack models [ISW03,DDF14,DDF19]. Likewise, despite its high level of technicality, the eventual tight security analysis of the SuKS under Hamming weight leakage is still reasonably simplistic in the sense that we only considered leakage for *one part* of the state. It may be possible, though highly non-trivial, to extend the analysis to a setting where the Hamming weight of *multiple parts* of the state leak, or where the adversary can change the positions of these parts in-between queries.

We remark that, despite our restricted focus to the SuKS, the analysis has many more applications. First off, the SuKS is used in the NIST lightweight cryptography competition [NIS19] finalist authenticated encryption scheme ISAP v2 [DEM+17,DEM+20, DEM+21]. More broadly seen, the analysis of earlier analyses of sponge-/duplex-based cryptographic schemes [DMV17,DM19a,Men23] highly depends on the multicollision limit function, and our findings in case of Hamming weight leakage, and in particular Theorem 3, directly apply to this setting.

Acknowledgments

We want to thank Vahid Jahandideh and Damian Vizár for their valuable help and feedback. Bart Mennink is supported by the Netherlands Organisation for Scientific Research (NWO) under grant VI.Vidi.203.099.

References

[BBD+13] Sonia Belaïd, Luk Bettale, Emmanuelle Dottax, Laurie Genelle, and Franck Rondepierre. Differential Power Analysis of HMAC SHA-2 in the Hamming Weight Model. In Pierangela Samarati, editor, *SECRYPT 2013 - Proceedings of the 10th International Conference on Security and Cryptography, Reykjavík, Iceland, 29-31 July, 2013*, pages 230–241. SciTePress, 2013.

[BDPV07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. Ecrypt Hash Workshop 2007, May 2007.

[BDPV11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Cryptographic sponge functions, January 2011.

[BFG14] Sonia Belaïd, Pierre-Alain Fouque, and Benoît Gérard. Side-Channel Analysis of Multiplications in GF(2^{128}) - Application to AES-GCM. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 306–325. Springer, 2014.

[BKP+18] Francesco Berti, François Koeune, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Ciphertext integrity with misuse and leakage: Definition and efficient constructions with symmetric primitives. In Jong Kim, Gail-Joon Ahn, Seungjoo Kim, Yongdae Kim, Javier López, and Taesoo Kim, editors, *Proceedings of the 2018 on Asia Conference on Computer and Communications Security, AsiaCCS 2018, Incheon, Republic of Korea, June 04-08, 2018*, pages 37–50. ACM, 2018.

[BPPS17] Francesco Berti, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. On leakage-resilient authenticated encryption with decryption leakages. *IACR Trans. Symmetric Cryptol.*, 2017(3):271–293, 2017.

[CJN20] Bishwajit Chakraborty, Ashwin Jha, and Mridul Nandi. On the Security of Sponge-type Authenticated Encryption Modes. *IACR Trans. Symmetric Cryptol.*, 2020(2):93–119, 2020.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.

[CLL19] Wonseok Choi, ByeongHak Lee, and Jooyoung Lee. Indifferentiability of Truncated Random Permutations. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 175–195. Springer, 2019.

[DDF14] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying Leakage Models: From Probing Attacks to Noisy Leakage. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 -*

33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings, volume 8441 of *Lecture Notes in Computer Science*, pages 423–440. Springer, 2014.

[DDF19] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying Leakage Models: From Probing Attacks to Noisy Leakage. *J. Cryptol.*, 32(1):151–177, 2019.

[DDNT23] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Suprita Talnikar. Tight Multi-User Security Bound of DbHtS. *IACR Trans. Symmetric Cryptol.*, 2023(1):192–223, 2023.

[DEM+17] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, and Thomas Unterluggauer. ISAP - towards side-channel secure authenticated encryption. *IACR Trans. Symmetric Cryptol.*, 2017(1):80–105, 2017.

[DEM+20] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas, and Thomas Unterluggauer. Isap v2.0. *IACR Trans. Symmetric Cryptol.*, 2020(S1):390–416, 2020.

[DEM+21] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas, and Thomas Unterluggauer. ISAP v2. Final Round Submission to NIST Lightweight Cryptography, 2021.

[DM19a] Christoph Dobraunig and Bart Mennink. Leakage Resilience of the Duplex Construction. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 225–255. Springer, 2019.

[DM19b] Christoph Dobraunig and Bart Mennink. Security of the Suffix Keyed Sponge. *IACR Trans. Symmetric Cryptol.*, 2019(4):223–248, 2019.

[DM20] Christoph Dobraunig and Bart Mennink. Tightness of the Suffix Keyed Sponge Bound. *IACR Trans. Symmetric Cryptol.*, 2020(4):195–212, 2020.

[DMMS21] Sébastien Duval, Pierrick Méaux, Charles Momin, and François-Xavier Standaert. Exploring Crypto-Physical Dark Matter and Learning with Physical Rounding Towards Secure and Efficient Fresh Re-Keying. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(1):373–401, 2021.

[DMP22] Christoph Dobraunig, Bart Mennink, and Robert Primas. Leakage and Tamper Resilient Permutation-Based Cryptography. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 859–873. ACM, 2022.

[DMV17] Joan Daemen, Bart Mennink, and Gilles Van Assche. Full-State Keyed Duplex with Built-In Multi-user Support. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 606–637. Springer, 2017.

[FPS12] Sebastian Faust, Krzysztof Pietrzak, and Joachim Schipper. Practical Leakage-Resilient Symmetric Cryptography. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*, pages 213–232. Springer, 2012.

[GP99] Louis Goubin and Jacques Patarin. DES and Differential Power Analysis (The "Duplication" Method). In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 1999.

[GPPS19] Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Towards lightweight side-channel security and the leakage-resilience of the duplex sponge. *IACR Cryptol. ePrint Arch.*, page 193, 2019.

[ISW03] Yuval Ishai, Amit Sahai, and David A. Wagner. Private Circuits: Securing Hardware against Probing Attacks. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.

[JLM14] Philipp Jovanovic, Atul Luykx, and Bart Mennink. Beyond $2^{c/2}$ Security in Sponge-Based Authenticated Encryption Modes. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 85–104. Springer, 2014.

[JLM+19] Philipp Jovanovic, Atul Luykx, Bart Mennink, Yu Sasaki, and Kan Yasuda. Beyond Conventional Security in Sponge-Based Authenticated Encryption Modes. *J. Cryptol.*, 32(3):895–940, 2019.

[JN20] Ashwin Jha and Mridul Nandi. Tight Security of Cascaded LRW2. *J. Cryptol.*, 33(3):1272–1317, 2020.

[KA98] Markus G. Kuhn and Ross J. Anderson. Soft Tempest: Hidden Data Transmission Using Electromagnetic Emanations. In David Aucsmith, editor, *Information Hiding, Second International Workshop, Portland, Oregon, USA, April 14-17, 1998, Proceedings*, volume 1525 of *Lecture Notes in Computer Science*, pages 124–142. Springer, 1998.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[Koc96] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[KR19] Yael Tauman Kalai and Leonid Reyzin. A Survey of Leakage-Resilient Cryptography. Cryptology ePrint Archive, Paper 2019/302, 2019. https://eprint.iacr.org/2019/302.

[LMP17] Atul Luykx, Bart Mennink, and Kenneth G. Paterson. Analyzing Multi-key Security Degradation. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 575–605. Springer, 2017.

[LNS18] Gaëtan Leurent, Mridul Nandi, and Ferdinand Sibleyras. Generic Attacks Against Beyond-Birthday-Bound MACs. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 306–336. Springer, 2018.

[May00] Rita Mayer-Sommer. Smartly Analyzing the Simplicity and the Power of Simple Power Analysis on Smartcards. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, volume 1965 of *Lecture Notes in Computer Science*, pages 78–92. Springer, 2000.

[Men23] Bart Mennink. Understanding the Duplex and Its Security. *IACR Trans. Symmetric Cryptol.*, 2023(2):1–46, 2023.

[Mes00] Thomas S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, volume 1965 of *Lecture Notes in Computer Science*, pages 238–251. Springer, 2000.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[NIS19] NIST. Lightweight Cryptography, February 2019. https://csrc.nist.gov/Projects/Lightweight-Cryptography.

[NRR06] Svetla Nikova, Christian Rechberger, and Vincent Rijmen. Threshold Implementations Against Side-Channel Attacks and Glitches. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006, Proceedings*, volume 4307 of *Lecture Notes in Computer Science*, pages 529–545. Springer, 2006.

[NRS11] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. *J. Cryptol.*, 24(2):292–321, 2011.

[PSV15] Olivier Pereira, François-Xavier Standaert, and Srinivas Vivek. Leakage-Resilient Authentication and Encryption from Symmetric Cryptographic Primitives. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 96–108. ACM, 2015.

A Python Script for Maximizing (19)

```
import sys
import math

# Function from script of Mennink [Men23, Appendix A]
def FindMinX(b,L):
    # Stores the minimum
    MinX = sys.maxsize;

    # Finds the minimum
    x = 0;
    while MinX == sys.maxsize:
        x = x+1;
        if x > 2**L:
            if ( b*math.log(2)-2**(L) + x*math.log(2**(L))
                <= math.log(x-2**(L)) + math.log(math.factorial(x)) ):
                MinX = x;
    # Returns the minimum
    return MinX;

def AlphaW(n, w):
    return max(1, math.ceil(math.comb(n, w) * (math.e**2 * (n+1)) / 2**n))

def FindMaxW(b, r, n):
    # Stores the maximum and corresponding value of w
    max = 0
    MaxW = 0

    # Finds the maximum and corresponding value of w
    for w in range(0, n+1):
        M = AlphaW(n, w) * 2**128
        L = math.log(M / 2**r, 2)
        if FindMinX(b, L) / math.comb(n, w) > max:
            max = FindMinX(b, L) / math.comb(n, w)
            MaxW = w
    # Returns maximum and corresponding value for w
    return max, MaxW

if __name__ == "__main__":
    n = 7
    b = 320
    s = t = 128
    sp = s - math.log(n+1, 2)
    tp = t + math.log(n+1, 2)
    print("Second term: w={1} gives maximum value {0}".format(*FindMaxW(b, b-sp, n)))
    print("Third term: w={1} gives maximum value {0}".format(*FindMaxW(b, tp, n)))
```

B Proofs of Lemmas 2, 3, and 4

B.1 Proof of Lemma 2

In order to prove Lemma 2, we need the following lemma:

Lemma 5. *Let $A, B, n \in \mathbb{N}$. Assume that $A \geq B > n$. Then, it holds that*

$$\frac{A-n}{B-n} \geq \frac{A}{B}\,.$$

Proof. The inequality can be simplified as follows:

$$\begin{aligned} \frac{A-n}{B-n} &\geq \frac{A}{B}\,, \\ AB - nB &\geq AB - nA\,, \\ -nB &\geq -nA\,, \\ B &\leq A\,. \end{aligned}$$

Because we assumed that $A \geq B$, the above inequality holds. □

With Lemma 5 proven, we restate and prove Lemma 2:

Lemma 2. *Let $A, B, n \in \mathbb{N}$. Assume that $A \geq B \geq n$. Then, it holds that*

$$\frac{(A)_n}{(B)_n} \geq \left(\frac{A}{B}\right)^n\,.$$

Proof. We prove by induction on n that Lemma 2 holds for all $n \in \mathbb{N}$ with $n \leq B$:

Base case.
$\frac{(A)_0}{(B)_0} \geq \left(\frac{A}{B}\right)^0$ because $\frac{(A)_0}{(B)_0} = \frac{1}{1} = 1 = \left(\frac{A}{B}\right)^0$.

Inductive case.

For all $k \in \mathbb{N}$ such that $k+1 \leq B$, we have to prove that:

$$\frac{(A)_{k+1}}{(B)_{k+1}} \geq \left(\frac{A}{B}\right)^{k+1} \text{ if } A \geq B\,.$$

We take as induction hypothesis (IH):

$$\frac{(A)_k}{(B)_k} \geq \left(\frac{A}{B}\right)^k \text{ if } A \geq B\,.$$

Now, we can prove the inductive case as follows:

$$\begin{aligned} \frac{(A)_{k+1}}{(B)_{k+1}} &= \frac{A(A-1)\cdots(A-k)}{B(B-1)\cdots(B-k)} \\ &= \frac{(A)_k}{(B)_k} \cdot \frac{A-k}{B-k} \\ &\overset{\text{IH}}{\geq} \left(\frac{A}{B}\right)^k \cdot \frac{A-k}{B-k} \\ &\overset{\text{Lemma 5}}{\geq} \left(\frac{A}{B}\right)^k \cdot \frac{A}{B} \\ &= \left(\frac{A}{B}\right)^{k+1}\,. \end{aligned}$$

Therefore, we have proven by mathematical induction that Lemma 2 holds. □

B.2 Proof of Lemma 3

In order to prove Lemma 3, we need the following lemma:

Lemma 6. $1 + x \leq e^x$ *for all* $x \in \mathbb{R}$.

Proof. Let $f(x) = e^x - (1 + x)$. If $f(x) \geq 0$ for all $x \in \mathbb{R}$, then Lemma 6 holds. First, we find the first and second derivative of $f(x)$:

$$f'(x) = \frac{\mathrm{d}}{\mathrm{d}x}\left[e^x - (1+x)\right] = \frac{\mathrm{d}}{\mathrm{d}x}\left[e^x\right] - \frac{\mathrm{d}}{\mathrm{d}x}\left[1+x\right] = e^x - 1\,,$$
$$f''(x) = \frac{\mathrm{d}}{\mathrm{d}x}\left[e^x - 1\right] = \frac{\mathrm{d}}{\mathrm{d}x}\left[e^x\right] - \frac{\mathrm{d}}{\mathrm{d}x}\left[1\right] = e^x\,.$$

Then, we find the critical points of $f(x)$, i.e., the values x where $f'(x) = 0$. Since $f'(x) = e^x - 1$, we know that $f'(x) = 0$ if and only if $x = 0$. Using the second derivative test, we find that f has a local minimum at $x = 0$ because $f''(0) = e^0 = 1 > 0$. Since $x = 0$ is the only critical point of f, it follows that $x = 0$ is the global minimum of f.

Since $f(0) \geq 0$ and 0 is the value for x such that $f(x)$ is minimal, it certainly holds that $f(x) \geq 0$ for all $x \in \mathbb{R}$. □

With Lemma 6 proven, we restate and prove Lemma 3:

Lemma 3. $\left(1 - \frac{1}{x}\right)^x \geq e^{-\frac{x}{x-1}}$ *for all* $x \in \mathbb{R}^+ \setminus \{1\}$.

Proof. We first prove that $1 - \frac{1}{x} \geq e^{-\frac{1}{x-1}}$ for all $x \in \mathbb{R}^+ \setminus \{1\}$. Let $y = \frac{1}{x}$. We have for all $y \in \mathbb{R}^+ \setminus \{1\}$ that:

$$1 - y = \frac{1}{\frac{1}{1-y}} = \frac{1}{\frac{1-y+y}{1-y}} = \frac{1}{\frac{1-y}{1-y} + \frac{y}{1-y}} = \frac{1}{1 + \frac{y}{1-y}}\,. \tag{32}$$

Let $z = \frac{y}{1-y}$. Using that $1 + z \leq e^z$ for all $z \in \mathbb{R}$ (see Lemma 6), we get:

$$\frac{1}{1 + \frac{y}{1-y}} = \frac{1}{1+z} \geq \frac{1}{e^z} = e^{-z} = e^{-\frac{y}{1-y}}\,. \tag{33}$$

It follows from (32) and (33) that:

$$1 - y \geq e^{-\frac{y}{1-y}}\,. \tag{34}$$

Substituting $\frac{1}{x}$ for y in (34) gives:

$$1 - \frac{1}{x} \geq e^{-\frac{\frac{1}{x}}{1-\frac{1}{x}}} = e^{-\frac{1}{x\left(1-\frac{1}{x}\right)}} = e^{-\frac{1}{x-1}}\,. \tag{35}$$

With (35), we have proven that $1 - \frac{1}{x} \geq e^{-\frac{1}{x-1}}$ holds for all $x \in \mathbb{R}^+ \setminus \{1\}$. Now, raising both sides to the power of x gives:

$$\left(1 - \frac{1}{x}\right)^x \geq \left(e^{-\frac{1}{x-1}}\right)^x = e^{-\frac{x}{x-1}}\,. \tag{36}$$

It follows from (36) that Lemma 3 is proven. □

B.3 Proof of Lemma 4

We restate Lemma 4:

Lemma 4. $e^{\frac{2^x}{2^x-1}} \leq e^2$ *on the interval* $[1, \infty)$.

Proof. Let $f(x) = \frac{2^x}{2^x-1}$. We show that the inequality holds by proving that $f(x) \leq 2$ on the interval $[1, \infty)$. We first show that $f(x)$ is decreasing for all $x \in [1, \infty)$, which is the case if the derivative of $f(x)$ is less than 0 for all $x \in [1, \infty)$.

We start by finding the derivative of $f(x)$. Using the quotient rule and that $\frac{\mathrm{d}}{\mathrm{d}x}[2^x] = 2^x \ln(2)$, we get:

$$\begin{aligned} f'(x) &= \frac{(2^x-1)\frac{\mathrm{d}}{\mathrm{d}x}[2^x] - 2^x \frac{\mathrm{d}}{\mathrm{d}x}[2^x-1]}{(2^x-1)^2} \\ &= \frac{(2^x-1)\frac{\mathrm{d}}{\mathrm{d}x}[2^x] - 2^x\left(\frac{\mathrm{d}}{\mathrm{d}x}[2^x] - \frac{\mathrm{d}}{\mathrm{d}x}[1]\right)}{(2^x-1)^2} \\ &= \frac{(2^x-1)(2^x\ln(2)) - 2^x(2^x\ln(2))}{(2^x-1)^2} \\ &= \frac{(2^x-1-2^x)(2^x\ln(2))}{(2^x-1)^2} \\ &= -\frac{2^x\ln(2)}{(2^x-1)^2}\,. \end{aligned}$$

Because both $2^x \ln(2)$ and $(2^x-1)^2$ are positive on the interval $[1, \infty)$, it follows that for all $x \in [1, \infty)$, $f'(x)$ is negative and $f(x)$ is decreasing. Therefore, on the interval $[1, \infty)$, $f(x)$ is maximal for $x = 1$. Thus, to prove that $f(x) \leq 2$, it suffices to show that $f(1) \leq 2$, which holds because $f(1) = \frac{2^1}{2^1-1} = 2$. Therefore, we have proven that Lemma 4 holds. □

C Proving the Inequality (27)

We show in detail that (27) holds for $y = \left\lceil \frac{\alpha(w)q}{2^{r'}} \right\rceil$ as left-hand side exponent. In more detail, we will show that for this value:

$$\left(\frac{\alpha(w)}{2^{r'}-1}\,\frac{2^{r+n}}{\binom{n}{j}}\,\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right)^{\left\lceil \frac{\alpha(w)q}{2^{r'}} \right\rceil} \geq \left(\frac{1-\binom{n}{j}2^{-r-n}}{(1-2^{-r'})^{\alpha(w)}\left(1-\frac{y}{2^b}\right)}\right)^q .$$

Note that the derivation holds under the condition that $\alpha(w)$ is greater than or equal to $\binom{n}{j}\frac{e^2(n+1)}{2^n}$. This condition is taken into account in the rest of the proof of Theorem 3.

$$\left(\frac{\alpha(w)}{2^{r'}-1}\,\frac{2^{r+n}}{\binom{n}{j}}\,\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right)^{\left\lceil\frac{\alpha(w)q}{2^{r'}}\right\rceil} \overset{?}{\gtrless} \left(\frac{1-\binom{n}{j}2^{-r-n}}{(1-2^{-r'})\alpha(w)\left(1-\frac{y}{2^b}\right)}\right)^{q},$$

$$\left(\frac{\alpha(w)}{2^{r'}-1}\,\frac{2^{r+n}}{\binom{n}{j}}\,\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right)^{\frac{\alpha(w)q}{2^{r'}}} \overset{?}{\gtrless} \left(\frac{1-\binom{n}{j}2^{-r-n}}{(1-2^{-r'})\alpha(w)\left(1-\frac{y}{2^b}\right)}\right)^{q},$$

$$\left(\left(\frac{\alpha(w)}{2^{r'}-1}\,\frac{2^{r+n}}{\binom{n}{j}}\,\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right)^{\frac{\alpha(w)q}{2^{r'}}}\right)^{\frac{2^{r'}}{\alpha(w)q}} \overset{?}{\gtrless} \left(\left(\frac{1-\binom{n}{j}2^{-r-n}}{(1-2^{-r'})\alpha(w)\left(1-\frac{y}{2^b}\right)}\right)^{q}\right)^{\frac{2^{r'}}{\alpha(w)q}},$$

$$\frac{\alpha(w)}{2^{r'}-1}\,\frac{2^{r+n}}{\binom{n}{j}}\,\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}} \overset{?}{\gtrless} \left(\frac{1-\binom{n}{j}2^{-r-n}}{(1-2^{-r'})\alpha(w)\left(1-\frac{y}{2^b}\right)}\right)^{\frac{2^{r'}}{\alpha(w)}},$$

$$\log\left(\frac{\alpha(w)}{2^{r'}-1}\,\frac{2^{r+n}}{\binom{n}{j}}\,\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right) \overset{?}{\gtrless} \log\left(\left(\frac{1-\binom{n}{j}2^{-r-n}}{(1-2^{-r'})\alpha(w)\left(1-\frac{y}{2^b}\right)}\right)^{\frac{2^{r'}}{\alpha(w)}}\right),$$

$$\log\left(\frac{\alpha(w)}{2^{r'}-1}\right)+\log\left(\frac{2^{r+n}}{\binom{n}{j}}\right)+\log\left(\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right) \overset{?}{\gtrless} \frac{2^{r'}}{\alpha(w)}\log\left(\frac{1-\binom{n}{j}2^{-r-n}}{(1-2^{-r'})^{\alpha(w)}\left(1-\frac{y}{2^b}\right)}\right),$$

$$\log\left(\frac{\alpha(w)}{2^{r'}-1}\right)+\log\left(\frac{2^{r+n}}{\binom{n}{j}}\right)+\log\left(\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right) \overset{?}{\gtrless} \frac{2^{r'}}{\alpha(w)}\log\left(1-\binom{n}{j}2^{-r-n}\right)-\frac{2^{r'}}{\alpha(w)}\log\left(\left(1-2^{-r'}\right)^{\alpha(w)}\left(1-\frac{y}{2^b}\right)\right),$$

$$\log\left(\frac{\alpha(w)}{2^{r'}-1}\right)+\log\left(\frac{2^{r+n}}{\binom{n}{j}}\right)+\log\left(\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right) \overset{?}{\geq} \frac{2^{r'}}{\alpha(w)}\log\left(1-\binom{n}{j}2^{-r-n}\right)-2^{r'}\log\left(1-2^{-r'}\right)-\frac{2^{r'}}{\alpha(w)}\log\left(1-\frac{y}{2^b}\right),$$

$$\log\left(\frac{\alpha(w)}{2^{r'}-1}\right)+\log\left(\frac{2^{r+n}}{\binom{n}{j}}\right)+2^{r'}\log\left(1-2^{-r'}\right) \overset{?}{\gtrless} \left(\frac{2^{r'}}{\alpha(w)}-1\right)\log\left(1-\binom{n}{j}2^{-r-n}\right)-\left(\frac{2^{r'}}{\alpha(w)}-1\right)\log\left(1-\frac{y}{2^b}\right),$$

$$\log\left(\frac{\alpha(w)}{2^{r'}-1}\right)+\log\left(\frac{2^{r+n}}{\binom{n}{j}}\right)+2^{r'}\log\left(1-2^{-r'}\right)\overset{?}{\geq}\left(\frac{2^{r'}}{\alpha(w)}-1\right)\left(\log\left(1-\binom{n}{j}2^{-r-n}\right)-\log\left(1-\frac{y}{2^b}\right)\right),$$
$$\log\left(\frac{\alpha(w)}{2^{r'}-1}\right)+\log\left(\frac{2^{r+n}}{\binom{n}{j}}\right)+2^{r'}\log\left(1-2^{-r'}\right)\overset{?}{\geq}\left(\frac{2^{r'}}{\alpha(w)}-1\right)\left(\log\left(\frac{1-\binom{n}{j}2^{-r-n}}{1-\frac{y}{2^b}}\right)\right).$$

We know that $y\leq\binom{n}{j}2^{c-n}$, from which it follows that $1-\frac{y}{2^b}\geq 1-\binom{n}{j}2^{-r-n}$. Therefore, the second term on the right-hand side is at most $\log(1)$:

$$\log\left(\frac{\alpha(w)}{2^{r'}-1}\right)+\log\left(\frac{2^{r+n}}{\binom{n}{j}}\right)+2^{r'}\log\left(1-2^{-r'}\right)\overset{?}{\geq}\left(\frac{2^{r'}}{\alpha(w)}-1\right)\log(1),$$
$$\log\left(\frac{\alpha(w)}{2^{r'}-1}\right)+\log\left(\frac{2^{r+n}}{\binom{n}{j}}\right)+2^{r'}\log\left(1-2^{-r'}\right)\overset{?}{\geq}0,$$
$$\log\left(\frac{\alpha(w)}{2^{r'}-1}\right)+r+n-\log\left(\binom{n}{j}\right)+2^{r'}\log\left(1-2^{-r'}\right)\overset{?}{\geq}0,$$
$$\log(\alpha(w))\overset{?}{\geq}\log\left(2^{r'}-1\right)+\log\left(\binom{n}{j}\right)-2^{r'}\log\left(1-2^{-r'}\right)-r-n,$$
$$2^{\log(\alpha(w))}\overset{?}{\geq}2^{\log\left(2^{r'}-1\right)+\log\left(\binom{n}{j}\right)-2^{r'}\log\left(1-2^{-r'}\right)-r-n},$$
$$\alpha(w)\overset{?}{\geq}\frac{(2^{r'}-1)\binom{n}{j}}{(1-2^{-r'})^{2^{r'}}2^{r+n}}.$$

Using that $2^{r'}-1<2^{r'}$, we can replace $2^{r'}-1$ by $2^{r'}$ on the right-hand side of the inequality:

$$\alpha(w)\overset{?}{\geq}\frac{2^{r'}\binom{n}{j}}{(1-2^{-r'})^{2^{r'}}2^{r+n}},$$
$$\alpha(w)\overset{?}{\geq}\frac{(n+1)2^{r}\binom{n}{j}}{(1-2^{-r'})^{2^{r'}}2^{r+n}},$$
$$\alpha(w)\overset{?}{\geq}\binom{n}{j}\frac{n+1}{\left(1-\frac{1}{2^{r'}}\right)^{2^{r'}}2^{n}}.$$

We assumed that $r \geq 1$. Because $r' = r + \log(n+1) > r$, it follows that $r' > 1$ and that $2^{r'} > 2$. Therefore, we can use that $\left(1 - \frac{1}{x}\right)^x \geq e^{-\frac{x}{x-1}}$ for all $x \in \mathbb{R}^+ \setminus \{1\}$ (see Lemma 3) by substituting $x = 2^{r'}$ to get:

$$\alpha(w) \stackrel{?}{\geq} \binom{n}{j} \frac{n+1}{e^{-\frac{2^{r'}}{2^{r'}-1}} 2^n},$$

$$\alpha(w) \stackrel{?}{\geq} \binom{n}{j} \frac{e^{\frac{2^{r'}}{2^{r'}-1}}(n+1)}{2^n}.$$

Using that $e^{\frac{2^x}{2^x-1}} \leq e^2$ on the interval $[1, \infty)$ (see Lemma 4) and that $r' > 1$, we can substitute $x = r'$ to get:

$$\alpha(w) \stackrel{?}{\geq} \binom{n}{j} \frac{e^2(n+1)}{2^n}.$$