

# The Summation-Truncation Hybrid: Reusing Discarded Bits for Free

Aldo Gunsing and Bart Mennink

Digital Security Group, Radboud University, Nijmegen, The Netherlands  
aldo.gunsing@ru.nl, b.mennink@cs.ru.nl

**Abstract.** A well-established PRP-to-PRF conversion design is truncation: one evaluates an  $n$ -bit pseudorandom permutation on a certain input, and truncates the result to  $a$  bits. The construction is known to achieve tight  $2^{n-a/2}$  security. Truncation has gained popularity due to its appearance in the GCM-SIV key derivation function (ACM CCS 2015). This key derivation function makes four evaluations of AES, truncates the outputs to  $n/2$  bits, and concatenates these to get a  $2n$ -bit subkey. In this work, we demonstrate that truncation is wasteful. In more detail, we present the Summation-Truncation Hybrid (STH). At a high level, the construction consists of two parallel evaluations of truncation, where the truncated  $(n - a)$ -bit chunks are not discarded but rather summed together and appended to the output. We prove that STH achieves a similar security level as truncation, and thus that the  $n - a$  bits of extra output is rendered for free. In the application of GCM-SIV, the current key derivation can be used to output  $3n$  bits of random material, or it can be reduced to three primitive evaluations. Both changes come with no security loss.

**Keywords:** PRP-to-PRF, Truncation, Sum of permutations, Efficiency, GCM-SIV

## 1 Introduction

The vast majority of symmetric cryptographic schemes is built upon a pseudorandom permutation, such as AES [21]. Such a function gets as input a key and bijectively transforms its input to an output such that the function is hard to distinguish from random if an attacker has no knowledge about the key. The approach is natural: the design as well as the analysis of pseudorandom permutations has faced ample research. Yet, in many encryption modes [3], message authentication codes [4, 10, 18, 65], authenticated encryption schemes [24, 29, 49] and other applications of pseudorandom permutations [45], the underlying primitive is only used *in forward direction*. Here, one does not make use of the invertibility of the permutation, and even stronger: the fact that one uses a pseudorandom permutation instead of a pseudorandom function comes at a security penalty.

A prominent example of this is the Wegman-Carter nonce-based message authentication code from 1981 [18, 65]:

$$\text{WC}^{F,H}(\nu, m) = F(\nu) \oplus H(m),$$

where  $F$  is a pseudorandom function transforming a nonce  $\nu$  to an  $n$ -bit output and  $H$  a universal hash function transforming an arbitrary length message  $m$  to an  $n$ -bit output. Provided that  $F$  and  $H$  are sufficiently strong and the nonce is never repeated, this construction is known to achieve  $2^n$  security [65]. However, given the thorough understanding in pseudorandom permutation design, Shoup suggested to instantiate Wegman-Carter using a pseudorandom permutation  $P$ , leading to a construction now known as the Wegman-Carter-Shoup construction [61]:

$$\text{WCS}^{P,H}(\nu, m) = P(\nu) \oplus H(m).$$

This construction, however, is known to only achieve approximately  $2^{n/2}$  birthday bound security in the size of  $P$  [11, 47, 61]. This bound may be fine for sufficiently large pseudorandom permutations like the AES, but with the use of legacy ciphers and with the rise of lightweight pseudorandom permutations [1, 2, 15, 16, 23, 28, 35, 43, 60, 66] whose widths could get down to 64 or even 32 bits, birthday attacks are a practical threat as recently demonstrated by McGrew [48] and Bhargavan and Leurent [12].

This and comparable examples (e.g., counter mode encryption [3] and GCM authenticated encryption [49]) showcase the value and need for pseudorandom functions. Unfortunately, we have little understanding in how to design dedicated pseudorandom functions, the only two notable exceptions to date being SURF [9] and AES-PRF [52] (see also Section 1.1.3). With respect to generic constructions, the well-established PRP-PRF switch dictates that a pseudorandom permutation behaves like a pseudorandom function up to the birthday bound,  $2^{n/2}$  where  $n$  is the primitive width [6, 8, 19, 25, 34, 36]. This switch allows one to obtain a pseudorandom function by simply taking a pseudorandom permutation, but yet, it incurs a loss in the security bound that is comparable to the loss in moving from Wegman-Carter to Wegman-Carter-Shoup.

## 1.1 Beyond Birthday Bound PRP-to-PRF Conversion

Various methods to transform a PRP into a PRF have been proposed that achieve security beyond the birthday bound on the block size of the underlying primitive. This work will mostly be concerned with two of them: the sum of permutations and truncation.

**1.1.1 Sum of Permutations.** The sum of two independent  $n$ -bit permutations  $P_1, P_2$ ,

$$\text{SoP}^{P_1, P_2}(x) = P_1(x) \oplus P_2(x), \tag{1}$$

was first introduced by Bellare et al. [7]. Closely following this introduction, Lucks [46] proved around  $2^{2n/3}$  security and Bellare and Impagliazzo [5] around  $2^n/n$  security. An intensive line of research of Patarin [56–58] yielded around optimal  $2^n$  security, up to constant, following the mirror theory. Dai et al. [22]

proved around  $2^n$  security using their rather compact and elegant chi-squared method.

The two independent permutations can be simulated using a single one through domain separation [5, 46]:

$$\text{SoSP}^P(x) = P(x\|0) \oplus P(x\|1). \quad (2)$$

The scheme achieves a similar level of security as SoP [22, 57].

A generalization worth describing is the CENC construction of Iwata [37]. CENC offers a tradeoff between counter mode and the sum of permutations. It is determined by a parameter  $w \geq 1$  and uses  $P(x\|0)$  to mask  $w$  subsequent blocks  $P(x\|1), \dots, P(x\|w)$ . Iwata proved  $2^{2n/3}$  security [37]. Iwata et al. [38] argued that, in fact, optimal  $2^n$  security of CENC directly follows from Patarin’s mirror theory. Bhattacharya and Nandi [14] re-confirmed this bound using the chi-squared method.

**1.1.2 Truncation.** The idea of truncation consists of simply discarding part of the output of an  $n$ -bit permutation  $P$ :

$$\text{Trunc}_a^P(x) = \text{left}_a(P(x)), \quad (3)$$

where  $0 \leq a \leq n$ . The idea dates back to Hall et al. [34], who proved  $2^{n-a/2}$  security for a specific selection of parameters  $a$ . Bellare and Impagliazzo [5] and Gilboa and Gueron [26] improved the scope of the proof to tight  $2^{n-a/2}$  security for all parameter choices. The first documented solution for the problem, however, dates back to 1978, when Stam [62], derived it in a non-cryptographic context. (See also Gilboa et al. [27].) Bhattacharya and Nandi [13] recently transformed Stam’s analysis to the chi-squared method and derived the identical  $2^{n-a/2}$  bound. Mennink [50] considered a general treatment of truncation with pre- and post-processing and related the generalized scheme with historical results of Stam from 1986 [63].

**1.1.3 Other Approaches.** We briefly elaborate on two more recent approaches on beyond birthday bound secure PRP-to-PRF conversion. Cogliati and Seurin [20] introduced Encrypted Davies-Meyer:

$$\text{EDM}^{P_1, P_2}(x) = P_2(P_1(x) \oplus x), \quad (4)$$

where  $P_1$  and  $P_2$  are two  $n$ -bit permutations. They proved security up to around  $2^{2n/3}$ . Dai et al. [22] proved security of the construction up to around  $2^{3n/4}$  using the chi-squared method and Mennink and Neves [51] proved security up to around  $2^n/n$  using the mirror theory.

Mennink and Neves [51] proposed its dual version Encrypted Davies-Meyer Dual:

$$\text{EDMD}^{P_1, P_2}(x) = P_2(P_1(x)) \oplus P_1(x). \quad (5)$$

They proved that  $\text{EDMD}^{P_1, P_2}$  is at least as secure as  $\text{SoP}^{P_1, P_2}$ . In other words, the construction is known to achieve around  $2^n$  security. Mennink and Neves [52] subsequently used the construction to design a dedicated PRF based on the AES [21]. Bernstein’s SURF [9], dating back to 1997, follows the same idea.

## 1.2 Truncation in GCM-SIV

GCM is a well-established authenticated encryption scheme [40, 49]. It follows the nonce-based encrypt-then-MAC paradigm, where encryption is performed in counter mode and the associated data and ciphertext are subsequently authenticated using the GHASH universal hash function.

GCM is vulnerable to nonce misuse attacks. Gueron and Lindell introduced GCM-SIV, a nonce misuse resistant authenticated encryption scheme. Several variants of GCM-SIV exist [29, 32, 33, 39], and we will focus on the most recent one. It follows the nonce misuse resistant SIV mode of Rogaway and Shrimpton [59] and uses individual ingredients of GCM. In the context of this work, we are particularly interested in the key derivation function of GCM-SIV [33]. This key derivation function is based on an  $(n = 128)$ -bit block cipher  $E$  and it derives either 256 bits of key material (if  $E$  is instantiated with AES-128) or 384 bits of key material (if  $E$  is instantiated with AES-256) based on key  $k$  and nonce  $\nu$  as follows:

$$\begin{cases} \text{left}_{n/2}(E_k(\nu\|0)) \parallel \cdots \parallel \text{left}_{n/2}(E_k(\nu\|3)), & \text{for 256-bit subkey,} \\ \text{left}_{n/2}(E_k(\nu\|0)) \parallel \cdots \parallel \text{left}_{n/2}(E_k(\nu\|5)), & \text{for 384-bit subkey.} \end{cases} \quad (6)$$

This key derivation was in fact introduced in a follow-up version of GCM-SIV [33] after weaknesses were discovered in the original mechanism [55]. The key derivation of (6) has actually been disputed over time. Iwata and Seurin [41] advocated for the sum of permutations instead, and Bose et al. [17] noted that one can even just leave block ciphers in, as bijectivity in the key derivation function do not matter in the bigger picture of GCM-SIV. Despite this disputation, GCM-SIV enjoys strong support from the practical community. GCM-SIV is considered for standardization by the IETF-CFRG [30, 31] and NIST [54]. Therefore, it is a legitimate question to investigate the exact behavior of the key derivation function within GCM-SIV.

## 1.3 Summation-Truncation Hybrid

Besides the difference in security guaranteed between truncation and the sum of permutations,  $2^{n-a/2}$  versus  $2^n$ , the former has another drawback:  $n - a$  bits are truncated *and simply discarded*. We will demonstrate that this practice is wasteful: one can make more economical use of the discarded randomness *without any sacrifice of security!*

Before heading to the main contribution, let us first consider what we can do with the discarded part of truncation *if we focused on a single truncation call*. In other words, we compute  $y = P(x)$ , output  $\text{left}_a(y)$  and discard  $\text{right}_{n-a}(y)$ .

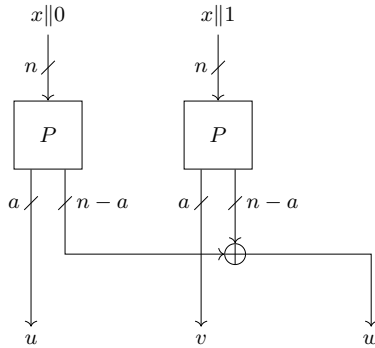


Fig. 1: Summation-Truncation Hybrid  $\text{STH}_a$  of (7).

We wish to make more economical use of  $\text{right}_{n-a}(y)$ . One way of doing so is to simply add the value with  $\text{left}_a(y)$ ; another way of doing so might be to split  $\text{right}_{n-a}(y)$  in two pieces, add the results, and append that value to the output of the truncation. It appears that, regardless of the adopted approach, one arrives at a generalized truncation function in the terminology of Mennink [50, 63]. His result describes that, whatever post-processing is applied to  $P$ , security of the scheme is tightly determined at  $2^{n-a'/2}$ , where  $a'$  is the output size of the generalized truncation function. (In the former example,  $a' = a$ , whereas in the latter example,  $a' = a + (n - a)/2 = (n + a)/2$ .) In other words, security of the construction does not increase and it might even decrease if the truncated data is attempted to be used more economically.

A next step is to look at *two* subsequent truncation calls, as appear, e.g., in the GCM-SIV key derivation (6). We present the Summation-Truncation Hybrid STH, that at a high level consists of two parallel evaluations of truncation, where the truncated parts are not discarded but rather summed together and appended to the output. In detail, if  $P$  is an  $n$ -bit permutation and  $a$  is a parameter satisfying  $0 \leq a \leq n$ , the Summation-Truncation Hybrid is a pseudorandom function that maps  $n - 1$  bits of input to  $n + a$  bits of output as follows:

$$\text{STH}_a^P(x) = \text{left}_a(P(x||0)) \parallel \text{left}_a(P(x||1)) \parallel \text{right}_{n-a}(P(x||0) \oplus P(x||1)) . \quad (7)$$

The function is depicted in Figure 1.

Clearly,  $\text{STH}_a$  is exactly as expensive as two evaluations of  $\text{Trunc}_a$ , but differs in that it outputs  $n - a$  bits *for free*. This may give a significant efficiency gain for repeated evaluations of truncation, for instance in the GCM-SIV key derivation in (6). Concretely, considering the case of GCM-SIV with 128-bit keys, it suffices to make three permutation calls instead of four, and for the case of 256-bit keys it suffices to make four permutation calls instead of six. We go into more detail for GCM-SIV in Section 7.

We also consider a variant of STH based on two permutations without domain separation. In detail, if  $P_1$  and  $P_2$  are  $n$ -bit permutations and  $a$  is a parameter

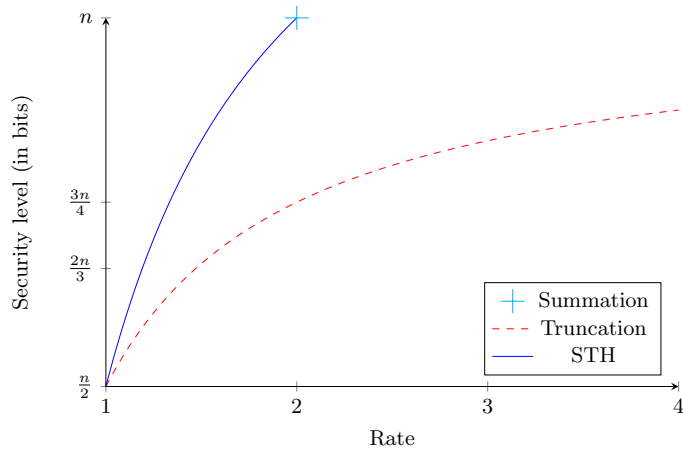


Fig. 2: Comparison between the efficiency and security of summation, truncation and the Summation-Truncation Hybrid. The rate denotes the average number of input bits needed for every output bit. Lower is more efficient.

satisfying  $0 \leq a \leq n$ , the Summation-Truncation Hybrid 2 is a pseudorandom function that maps  $n$  bits of input to  $n + a$  bits of output as follows:

$$\text{STH2}_a^{P_1, P_2}(x) = \text{left}_a(P_1(x)) \parallel \text{left}_a(P_2(x)) \parallel \text{right}_{n-a}(P_1(x) \oplus P_2(x)). \quad (8)$$

Its properties are very similar to the ones of the original STH.

#### 1.4 Security of Summation-Truncation Hybrid

In Section 3 we formally prove that the security of  $\text{STH}_a$  is determined by the security of truncation, i.e., that  $q$  evaluations of  $\text{STH}_a$  are approximately as secure as  $2q$  bits of truncation, *despite the  $q(n - a)$  bits of free random output*. A comparison between the efficiency and security of truncation, summation and STH is shown in Figure 2.

The core idea of the proof consists of separating the truncation and the summation. This is not directly possible, as both parts share some secret information: the random permutation. The separation, at a high level, is performed by getting rid of this shared secret so that one only has to reason based on public information.

In more detail, in the proof we first execute the truncation part of the construction, based on the secret permutation. Then, we select a *new* secret permutation for the summation. On the upside, this trick makes it possible to reason

about the truncation and summation parts *independently*. On the downside, replacing the secret permutation half-way gives rise to a different construction than the original one, of course. To remedy this, the new permutations is not selected from the set of all possible permutations, but rather from those that are *compatible* with the output generated by the truncation. This set is solely based on public information, i.e., information known by the adversary, as we do give the output of the truncation.

As a bonus, as truncation gives information about the outputs of the permutation directly, we see that this set of compatible permutations is easy to reason about. We demonstrate that choosing such a random permutation is the same as choosing a *family of permutations* with the indices equal to the outputs of the truncation. As we can now replace the truncation with a random function, relying on the extensive state of the art on truncation [5, 13, 26, 34, 62], these indices even become uniformly distributed, which makes them nice to handle.

This transition then brings us to the final part of the proof, that generalizes security of the sum of permutations with just a single permutation to the sum of permutations based on an arbitrary family of permutations. The analysis relies on the chi-squared method and generalizes the proof of Dai et al. [22], with the catch that we not only consider a family of permutations, but rather that the selection of permutations from this family is uniformly distributed as it depends on the outputs of the random function (that replaced the truncation).

## 2 Preliminaries

Let  $n, a, b \in \mathbb{N}$  with  $n \geq a, b$ . We denote by  $\{0, 1\}^n$  the set of bit strings of length  $n$ . If  $x \in \{0, 1\}^n$ ,  $\text{left}_a(x)$  returns the  $a$  leftmost bits and  $\text{right}_b(x)$  the  $b$  rightmost bits of  $x$ , in such a way that

$$x = \text{left}_a(x) \parallel \text{right}_{n-a}(x).$$

For  $n, m \in \mathbb{N}$ , we denote by  $\text{Perm}[n]$  the set of all permutations on  $\{0, 1\}^n$  with  $I_n \in \text{Perm}[n]$  the identity permutation, and by  $\text{Func}[n, m]$  the set of all functions from  $\{0, 1\}^n$  to  $\{0, 1\}^m$ .

If  $\mathcal{X}$  is a finite set,  $x \stackrel{\$}{\leftarrow} \mathcal{X}$  denotes the event of uniformly randomly drawing  $x$  from  $\mathcal{X}$ . For two distributions  $\mu, \nu$  over a finite event space  $\Omega$ , the statistical distance between  $\mu$  and  $\nu$  is defined as

$$\begin{aligned} \|\mu - \nu\| &= \sum_{x \in \Omega} \max(0, \mu(x) - \nu(x)) \\ &= \max_{A \subseteq \Omega} |\mu(A) - \nu(A)|. \end{aligned}$$

A randomized algorithm  $\mathcal{O}$  introduces its variables as random variables, so  $\mathbb{P}_{\mathcal{O}}[x = a]$  denotes the probability that the variable  $x$  in algorithm  $\mathcal{O}$  is equal to  $a$ . As a shorthand we denote  $\mathbb{P}_{\mathcal{O}}[a]$  and  $\mathbb{P}_{\mathcal{O}}[A]$ , with  $a$  a single value and  $A$  a set of values, for the probabilities that algorithm  $\mathcal{O}$  returns the value  $a$  or a

value in the set  $A$ , respectively. Just  $\mathbb{P}_{\mathcal{O}}$  denotes the distribution of the return value of algorithm  $\mathcal{O}$ . For a random variable  $X$  with distribution  $\mu$ , we denote its expectation by  $\mathbb{E}_{\mu}[X]$ . If the distribution is clear from the context, we write just  $\mathbb{E}[X]$ .

A distinguisher  $\mathcal{D}$  is an algorithm that is given access to an oracle  $\mathcal{O}$  to which it can make a certain amount of queries, and afterwards it outputs  $b \in \{0, 1\}$ .

We briefly state an elementary property of conditional expectation.

**Lemma 1.** *Suppose that  $\mathbb{E}[Y | X] = \mathbb{E}[Y]$ . Then  $\mathbb{E}[X \cdot Y] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$ .*

*Proof.* By the law of total expectation we have

$$\begin{aligned} \mathbb{E}[X \cdot Y] &= \mathbb{E}[\mathbb{E}[X \cdot Y | X]] \\ &= \mathbb{E}[X \cdot \mathbb{E}[Y | X]] \\ &= \mathbb{E}[X \cdot \mathbb{E}[Y]] \\ &= \mathbb{E}[X] \cdot \mathbb{E}[Y] . \end{aligned} \quad \square$$

## 2.1 Block Ciphers

Let  $\kappa, n \in \mathbb{N}$ . A block cipher  $E : \{0, 1\}^{\kappa} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$  is a permutation on  $n$ -bit strings for every fixed key  $k \in \{0, 1\}^{\kappa}$ . Denote by  $\text{Perm}[n]$  the set of all permutations on  $\{0, 1\}^n$ . The security of a block cipher is measured by the distance between  $E_k$  for secret key from a random permutation  $P \xleftarrow{\$} \text{Perm}[n]$ . The advantage of a distinguisher  $\mathcal{D}$  in breaking the PRP (pseudorandom permutation) security of  $E$  is defined as

$$\text{Adv}_E^{\text{prp}}(\mathcal{D}) = \left| \mathbb{P}[\mathcal{D}^{E_k} = 1] - \mathbb{P}[\mathcal{D}^P = 1] \right| , \quad (9)$$

where the probabilities are taken over the random drawing of  $k \xleftarrow{\$} \{0, 1\}^{\kappa}$ ,  $P \xleftarrow{\$} \text{Perm}[n]$ , and the randomness used by  $\mathcal{D}$ . Distinguisher  $\mathcal{D}$  is usually bound by a query complexity  $q$  and a time complexity  $t$ . The maximum over all such distinguishers is denoted by  $\text{Adv}_E^{\text{prp}}(q, t)$ .

## 2.2 Pseudorandom Functions

Let  $n, m \in \mathbb{N}$ . Let  $F^P \in \text{Func}[n, m]$  be a function from  $\{0, 1\}^n$  to  $\{0, 1\}^m$  that is instantiated with a permutation  $P \in \text{Perm}[n]$ . The security of  $F$  is measured by the distance between  $F^P$  for secret and uniformly randomly drawn  $P \xleftarrow{\$} \text{Perm}[n]$  from a random function  $R \xleftarrow{\$} \text{Func}[n, m]$ . The advantage of a distinguisher  $\mathcal{D}$  in breaking the PRF (pseudorandom function) security of  $F$  is defined as

$$\text{Adv}_F^{\text{prf}}(\mathcal{D}) = \left| \mathbb{P}[\mathcal{D}^{F^P} = 1] - \mathbb{P}[\mathcal{D}^R = 1] \right| . \quad (10)$$

For  $q \in \mathbb{N}$ , we define by  $\text{Adv}_F^{\text{prf}}(q)$  the maximum advantage over all distinguishers  $\mathcal{D}$  making  $q$  queries to the oracle.



### 2.3 Truncation

Our security analysis will, in part, rely on the PRF security of truncation of (3). We copy the result of Stam [62], translated to cryptographic terminology [13, 50].

**Lemma 2 (Truncation [13, 50, 62]).** *Let  $n, a, q \in \mathbb{N}$  such that  $0 \leq a \leq n$ . We have:*

$$\mathbf{Adv}_{\text{Trunc}_a}^{\text{prf}}(q) \leq \left( \binom{q}{2} / 2^{2n-a} \right)^{1/2}. \quad (11)$$

### 3 Summation-Truncation Hybrid

Let  $n, a \in \mathbb{N}$  such that  $0 \leq a \leq n$ , and write  $b = n - a$ . Let  $P \in \text{Perm}[n]$  be a permutation. We define the Summation-Truncation Hybrid  $\text{STH}_a^P \in \text{Func}[n - 1, n + a]$  as follows:

$$\text{STH}_a^P(x) = \text{left}_a(P(x\|0)) \parallel \text{left}_a(P(x\|1)) \parallel \text{right}_b(P(x\|0) \oplus P(x\|1)). \quad (12)$$

The function is depicted in Figure 1. As expressed in this figure, we refer to the first  $a$  bits as  $u$ , the second  $a$  bits as  $v$ , and the final  $b$  bits as  $w$ , and write  $y = u\|v\|w$ .

Clearly,  $\text{STH}_a$  is equivalent to SoP for  $a = 0$ . If  $a = n$ ,  $\text{STH}_a$  consists of a concatenation of two block cipher evaluations. For general  $a$ , one evaluation of  $\text{STH}_a$  with the last  $b$  bits discarded is equivalent to a double evaluation of  $\text{Trunc}_a$ . As we will show, however, there is no reason to discard these  $b$  bits. Stated differently,  $q$  evaluations of  $\text{STH}_a$  are roughly as secure as  $2q$  evaluations of  $\text{Trunc}_a$ , although the former outputs significantly more random data.

**Theorem 1.** *Let  $n, a, q \in \mathbb{N}$  such that  $0 \leq a \leq n$ , and write  $b = n - a$ . Assume that  $b \geq \max(n/12, 10)$ . We have:*

$$\mathbf{Adv}_{\text{STH}_a}^{\text{prf}}(q) \leq 3 \left( \frac{q}{2^{n-a/3}} \right)^{3/2} + \frac{1}{\sqrt{2\pi}} \left( \frac{q}{2^{n-5}} \right)^{2^{b-2}} + \frac{q}{2^n} + \mathbf{Adv}_{\text{Trunc}_a}^{\text{prf}}(2q). \quad (13)$$

The dominating bound of Theorem 1 is, in fact,  $\mathbf{Adv}_{\text{Trunc}_a}^{\text{prf}}(2q)$ , and therefore, security of  $\text{STH}_a$  is only marginally worse than that of  $\text{Trunc}_a$ , even though it is much more efficient. We prove Theorem 1 in Section 4.

*Remark 1.* In Theorem 1, as well as in Lemma 2, we focus on PRF security in the information-theoretic setting, where the underlying primitive is a secret random permutation. One can easily transfer these results to a complexity-theoretic setting where  $P$  is defined as a block cipher instance  $E_k$  for secret key. More detailed, the bound of  $\text{Trunc}_a$  (Lemma 2) carries over with an additional loss of  $\mathbf{Adv}_E^{\text{ppf}}(q, t)$ , and the bound of  $\text{STH}_a$  carries over with an additional loss of  $\mathbf{Adv}_E^{\text{ppf}}(2q, t)$ , where the time complexity is bounded by  $t$ .

We also consider a variant of STH based on two independent random permutations without domain separation. Let  $n, a \in \mathbb{N}$  such that  $0 \leq a \leq n$ , and write  $b = n - a$ . Let  $P_1, P_2 \in \text{Perm}[n]$  be two permutations. We define  $\text{STH}_a^{P_1, P_2} \in \text{Func}[n, n + a]$  as follows:

$$\text{STH}_a^{P_1, P_2}(x) = \text{left}_a(P_1(x)) \parallel \text{left}_a(P_2(x)) \parallel \text{right}_b(P_1(x) \oplus P_2(x)). \quad (14)$$

We get a similar bound as for the original STH.

**Theorem 2.** *Let  $n, a, q \in \mathbb{N}$  such that  $0 \leq a \leq n$ , and write  $b = n - a$ . Assume that  $b \geq \max(n/12, 10)$ . We have:*

$$\text{Adv}_{\text{STH}_a}^{\text{prf}}(q) \leq 3 \left( \frac{q}{2^{n-a/3}} \right)^{3/2} + \frac{1}{\sqrt{2\pi}} \left( \frac{q}{2^{n-5}} \right)^{2^{b-2}} + 2\text{Adv}_{\text{Trunc}_a}^{\text{prf}}(q). \quad (15)$$

We prove Theorem 2 in Section 6.

One might also be interested in creating a larger instance of the hybrid. One approach would be to consider applying other functions than summation on the discarded parts. For example, one could apply the generalized CENC construction [37] on top of them. This could lead to other interesting results and might improve the efficiency as well, but this is left for potential future work.

## 4 Proof of Theorem 1

Let  $n, a, q \in \mathbb{N}$  such that  $0 \leq a \leq n$ , and write  $b = n - a$ . As of now, we will drop subscript  $a$  to STH for brevity. Consider any distinguisher  $\mathcal{D}$  making  $q$  queries to its oracle. Without loss of generality,  $\mathcal{D}$  is deterministic and does not make pointless queries, i.e.,  $x_i \neq x_j$  for all  $i \neq j$ . Our goal is to bound the distance between  $\text{STH}^P$  for a random permutation  $P \xleftarrow{\$} \text{Perm}[n]$  on the one hand and a random function  $R \xleftarrow{\$} \text{Func}[n - 1, n + a]$  on the other hand:

$$\text{Adv}_{\text{STH}}^{\text{prf}}(\mathcal{D}) = \left\| \mathbb{P} \left[ \mathcal{D}^{\text{STH}^P} = 1 \right] - \mathbb{P} \left[ \mathcal{D}^R = 1 \right] \right\|. \quad (16)$$

We will bound (16) in multiple steps. The first step (Section 4.1) will be to show that, without loss of generality, we can move to a non-adaptive setting and argue based on probabilities of transcripts to occur. Then, the second step (Section 4.2) consists of transforming the real world  $\text{STH}^P$  into a world that separates the Trunc and SoP parts within STH. Then, the third step (Section 4.3) replaces that Trunc part by a random function, at the cost of  $\text{Adv}_{\text{Trunc}}^{\text{prf}}(2q)$ . Then, the fourth step (Section 4.4) operates on the ideal world  $R$ : it transforms it into a world that does not output strings of the form  $u \parallel u \parallel 0^b$  for  $u \in \{0, 1\}^a$ , noting that these never occur in the real world we were left with in the third step. Finally, the fifth step (Section 4.5) bounds the remaining two worlds using the chi-squared method.

## 4.1 Moving Towards Transcripts

As a first step, we note that the input adaptivity does not help and it suffices to simply consider the probability of transcripts to occur in the real world and in the ideal world. Let  $\mathcal{O}_1$  be an oracle that generates transcripts as lists of random strings, Algorithm 1, and  $\mathcal{O}_2 = \text{NSTH}$  (Non-adaptive STH) be an oracle that generates transcripts as results of  $\text{STH}^P$  with random  $P \xleftarrow{\$} \text{Perm}[n]$  and fixed inputs  $x_i = i$ , Algorithm 2.

We will show that the advantage of attacker  $\mathcal{D}$  in distinguishing the two worlds of (16) is at most the statistical distance between world  $\mathcal{O}_1$  and world  $\mathcal{O}_2$ . Suppose we get a transcript  $\tau$  from any of the two oracles  $\mathcal{O}_1$  or  $\mathcal{O}_2$ , we will use it to simulate  $\mathcal{D}$ 's oracles as follows. If  $\mathcal{D}$  makes query  $x_i$ , we respond with  $u_i \| v_i \| w_i$  from the transcript. Denote by  $A$  the set of all transcripts  $\tau$  for which  $\mathcal{D}$  returns 1. Note that we can cleanly define this, as  $\mathcal{D}$  is deterministic and its decision only depends on  $\tau$ ; moreover, the fact that world  $\mathcal{O}_2$  uses fixed inputs for  $P$  does not matter as it is a random permutation. Then, the advantage of  $\mathcal{D}$  is at most

$$(16) \leq |\mathbb{P}_{\mathcal{O}_1}[A] - \mathbb{P}_{\mathcal{O}_2}[A]| \leq \|\mathbb{P}_{\mathcal{O}_1} - \mathbb{P}_{\mathcal{O}_2}\|. \quad (17)$$

Henceforth, it is sufficient to restrict our focus to the statistical distance between  $\mathcal{O}_1$  and  $\mathcal{O}_2$ .

---

### Algorithm 1 $\mathcal{O}_1 = R_1$

---

```

1: function  $R_1$ 
2:   for  $i \leftarrow 1$  to  $q$  do
3:      $u_i \xleftarrow{\$} \{0, 1\}^a$ 
4:      $v_i \xleftarrow{\$} \{0, 1\}^a$ 
5:      $w_i \xleftarrow{\$} \{0, 1\}^b$ 
6:    $\mathbf{u} = (u_1, \dots, u_q)$ 
7:    $\mathbf{v} = (v_1, \dots, v_q)$ 
8:    $\mathbf{w} = (w_1, \dots, w_q)$ 
9:   return  $\tau = (\mathbf{u}, \mathbf{v}, \mathbf{w})$ 

```

---



---

### Algorithm 2 $\mathcal{O}_2 = \text{NSTH}$

---

```

1: function NSTH
2:    $P \xleftarrow{\$} \text{Perm}[n]$ 
3:   for  $i \leftarrow 1$  to  $q$  do
4:      $u_i \leftarrow \text{left}_a(P(i \| 0))$ 
5:      $v_i \leftarrow \text{left}_a(P(i \| 1))$ 
6:      $U_i \leftarrow \text{right}_b(P(i \| 0))$ 
7:      $V_i \leftarrow \text{right}_b(P(i \| 1))$ 
8:      $w_i = U_i \oplus V_i$ 
9:    $\mathbf{u} = (u_1, \dots, u_q)$ 
10:   $\mathbf{v} = (v_1, \dots, v_q)$ 
11:   $\mathbf{w} = (w_1, \dots, w_q)$ 
12:  return  $\tau = (\mathbf{u}, \mathbf{v}, \mathbf{w})$ 

```

---

## 4.2 Permutation-Separated STH

We define a variant of  $\mathcal{O}_2$ , namely  $\mathcal{O}_3 = \text{PSTH}$  (Permutation-separated STH), in Algorithm 3. This oracle “separates” the Trunc part and the SoP part within NSTH. In more detail, it first calls internal procedure PTrunc that draws a random permutation  $P \xleftarrow{\$} \text{Perm}[n]$  and outputs the lists  $\mathbf{u}$  and  $\mathbf{v}$ . Then, it calls internal procedure PSOP that takes the two lists  $(\mathbf{u}, \mathbf{v})$  and returns a list  $\mathbf{w}$  using a random permutation  $P'$ . This permutation is randomly drawn from a set

---

**Algorithm 3**  $\mathcal{O}_3 = \text{PSTH}$ 

---

```
1: function PSTH
2:    $(\mathbf{u}, \mathbf{v}) \leftarrow \text{PTrunc}$ 
3:    $\mathbf{w} \leftarrow \text{PSoP}(\mathbf{u}, \mathbf{v})$ 
4:   return  $\boldsymbol{\tau} = (\mathbf{u}, \mathbf{v}, \mathbf{w})$ 

1: function PTrunc
2:    $P \xleftarrow{\$} \text{Perm}[n]$ 
3:   for  $i \leftarrow 1$  to  $q$  do
4:      $u_i \leftarrow \text{left}_a(P(i||0))$ 
5:      $v_i \leftarrow \text{left}_a(P(i||1))$ 
6:    $\mathbf{u} = (u_1, \dots, u_q)$ 
7:    $\mathbf{v} = (v_1, \dots, v_q)$ 
8:   return  $(\mathbf{u}, \mathbf{v})$ 

1: function PSoP( $\mathbf{u}, \mathbf{v}$ )
2:   if  $\text{Perm}_{\text{comp}}(\mathbf{u}, \mathbf{v}) \neq \emptyset$  then
3:      $P' \xleftarrow{\$} \text{Perm}_{\text{comp}}(\mathbf{u}, \mathbf{v})$ 
4:   else
5:      $P' \leftarrow I_n$ 
6:   for  $i \leftarrow 1$  to  $q$  do
7:      $U_i \leftarrow \text{right}_b(P'(i||0))$ 
8:      $V_i \leftarrow \text{right}_b(P'(i||1))$ 
9:      $w_i = U_i \oplus V_i$ 
10:   $\mathbf{w} = (w_1, \dots, w_q)$ 
11:  return  $\mathbf{w}$ 
```

---

$\text{Perm}_{\text{comp}}(\mathbf{u}, \mathbf{v}) \subseteq \text{Perm}[n]$  defined as the set of all permutations from  $\text{Perm}[n]$  for which PTrunc would return  $(\mathbf{u}, \mathbf{v})$ . Note that in our analysis this set will never be empty, so the ‘else’ branch will never be taken and is included solely to complete the algorithm.

We will prove that any transcript  $\boldsymbol{\tau}$  is equally likely in  $\mathcal{O}_2$  and  $\mathcal{O}_3$ . Consider any valid transcript  $\boldsymbol{\tau}$ , and define by  $\text{Perm}_{\text{result}}(\boldsymbol{\tau}) \subseteq \text{Perm}[n]$  the set of all permutations that give result  $\boldsymbol{\tau} = (\mathbf{u}, \mathbf{v}, \mathbf{w})$  when used in  $\mathcal{O}_2 = \text{NSTH}$ . Then,

$$\mathbb{P}_{\mathcal{O}_2}[\boldsymbol{\tau}] = \mathbb{P}_{\text{NSTH}}[\boldsymbol{\tau}] = \frac{|\text{Perm}_{\text{result}}(\boldsymbol{\tau})|}{|\text{Perm}[n]|}.$$

On the other hand, for  $\mathcal{O}_3 = \text{PSTH}$ , we first have to get the right  $(\mathbf{u}, \mathbf{v})$ :

$$\mathbb{P}_{\text{PTrunc}}[(\mathbf{u}, \mathbf{v})] = \frac{|\text{Perm}_{\text{comp}}(\mathbf{u}, \mathbf{v})|}{|\text{Perm}[n]|}.$$

Next, we have to get the right  $\mathbf{w}$ . As  $\text{Perm}_{\text{result}}(\boldsymbol{\tau}) \subseteq \text{Perm}_{\text{comp}}(\mathbf{u}, \mathbf{v})$ , this probability is equal to:

$$\mathbb{P}_{\text{PSoP}(\mathbf{u}, \mathbf{v})}[\mathbf{w}] = \frac{|\text{Perm}_{\text{result}}(\boldsymbol{\tau})|}{|\text{Perm}_{\text{comp}}(\mathbf{u}, \mathbf{v})|}.$$

The randomnesses in PTrunc and PSoP are independent, hence the two probabilities are independent as well. This means that the probability of getting  $\tau$  in  $\mathcal{O}_3 = \text{PSTH}$  is equal to their product. In other words:

$$\begin{aligned}
\mathbb{P}_{\mathcal{O}_3}[\tau] &= \mathbb{P}_{\text{PSTH}}[\tau] = \mathbb{P}_{\text{PSTH}}[\tau \mid (\mathbf{u}, \mathbf{v}) \leftarrow \text{PTrunc}] \cdot \mathbb{P}_{\text{PTrunc}}[(\mathbf{u}, \mathbf{v})] \\
&\quad + \mathbb{P}_{\text{PSTH}}[\tau \mid (\mathbf{u}, \mathbf{v}) \not\leftarrow \text{PTrunc}] \cdot (1 - \mathbb{P}_{\text{PTrunc}}[(\mathbf{u}, \mathbf{v})]) \\
&= \mathbb{P}_{\text{PSoP}(\mathbf{u}, \mathbf{v})}[\mathbf{w}] \cdot \mathbb{P}_{\text{PTrunc}}[(\mathbf{u}, \mathbf{v})] + 0 \\
&= \frac{|\text{Perm}_{\text{result}}(\tau)|}{|\text{Perm}_{\text{comp}}(\mathbf{u}, \mathbf{v})|} \cdot \frac{|\text{Perm}_{\text{comp}}(\mathbf{u}, \mathbf{v})|}{|\text{Perm}[n]|} \\
&= \frac{|\text{Perm}_{\text{result}}(\tau)|}{|\text{Perm}[n]|} \\
&= \mathbb{P}_{\text{NSTH}}[\tau] = \mathbb{P}_{\mathcal{O}_2}[\tau].
\end{aligned}$$

We have henceforth obtained that

$$\|\mathbb{P}_{\mathcal{O}_2} - \mathbb{P}_{\mathcal{O}_3}\| = 0. \quad (18)$$

### 4.3 Isolating Truncation Advantage

Next, we define  $\mathcal{O}_4 = \text{RSTH}$  (Random function-based STH) in Algorithm 4. The algorithm is identical to  $\mathcal{O}_3 = \text{PSTH}$ , but with the function PTrunc replaced by a random function  $S$ . Note that  $S$  is written as separate procedure; this is done to suit further analysis in Section 4.5.

---

#### Algorithm 4 $\mathcal{O}_4 = \text{RSTH}$

---

```

1: function RSTH
2:    $(\mathbf{u}, \mathbf{v}) \leftarrow S$ 
3:    $\mathbf{w} \leftarrow \text{PSoP}(\mathbf{u}, \mathbf{v})$  ▷ See Algorithm 3
4:   return  $\tau = (\mathbf{u}, \mathbf{v}, \mathbf{w})$ 

1: function  $S$ 
2:   for  $i \leftarrow 1$  to  $q$  do
3:      $u_i \xleftarrow{\$} \{0, 1\}^a$ 
4:      $v_i \xleftarrow{\$} \{0, 1\}^a$ 
5:    $\mathbf{u} = (u_1, \dots, u_q)$ 
6:    $\mathbf{v} = (v_1, \dots, v_q)$ 
7:   return  $(\mathbf{u}, \mathbf{v})$ 

```

---

The only difference between  $\mathcal{O}_3 = \text{PSTH}$  and  $\mathcal{O}_4 = \text{RSTH}$  is in the generation of  $(\mathbf{u}, \mathbf{v})$ : in the former, they are generated as a truncated permutation, whereas in the latter they are generated as a random function. Therefore, we immediately have:

$$\|\mathbb{P}_{\mathcal{O}_3} - \mathbb{P}_{\mathcal{O}_4}\| \leq \|\mathbb{P}_{\text{PTrunc}} - \mathbb{P}_S\| = \mathbf{Adv}_{\text{Trunc}}^{\text{prf}}(2q). \quad (19)$$

#### 4.4 Discarding the Zero

We proceed on the other end of (17). We turn  $\mathcal{O}_1 = R_1$  into  $\mathcal{O}_0 = R_0$  that operates identically except that it never returns  $w_i = 0^b$  when  $u_i = v_i$ . The oracle is given in Algorithm 5. As before, we write  $T$  as a separate procedure to suit further analysis in Section 4.5.

---

#### Algorithm 5 $\mathcal{O}_0 = R_0$

---

```

1: function  $R_0$ 
2:    $(\mathbf{u}, \mathbf{v}) \leftarrow S$ 
3:    $\mathbf{w} \leftarrow T(\mathbf{u}, \mathbf{v})$ 
4:   return  $\boldsymbol{\tau} = (\mathbf{u}, \mathbf{v}, \mathbf{w})$ 

1: function  $T(\mathbf{u}, \mathbf{v})$ 
2:   for  $i \leftarrow 1$  to  $q$  do
3:     if  $u_i \neq v_i$  then
4:        $w_i \xleftarrow{\$} \{0, 1\}^b$ 
5:     else
6:        $w_i \xleftarrow{\$} \{0, 1\}^b \setminus \{0^b\}$ 
7:   return  $\mathbf{w} = (w_1, \dots, w_q)$ 

```

---

We look at the statistical distance between  $\mathbb{P}_{\mathcal{O}_1}$  and  $\mathbb{P}_{\mathcal{O}_0}$ . Let  $\text{bad}_1$  be the set of transcripts  $\boldsymbol{\tau} = (\mathbf{u}, \mathbf{v}, \mathbf{w})$  such that  $u_i = v_i$  and  $w_i = 0^b$  for some  $i$ . As  $\mathbb{P}_{\mathcal{O}_0}[\boldsymbol{\tau}] = 0$  for  $\boldsymbol{\tau} \in \text{bad}_1$  and  $\mathbb{P}_{\mathcal{O}_1}[\boldsymbol{\tau}] \leq \mathbb{P}_{\mathcal{O}_0}[\boldsymbol{\tau}]$  for  $\boldsymbol{\tau} \notin \text{bad}_1$  we see, where  $A$  can be any set of transcripts, that

$$\begin{aligned}
\|\mathbb{P}_{\mathcal{O}_1} - \mathbb{P}_{\mathcal{O}_0}\| &= \max_A |\mathbb{P}_{\mathcal{O}_1}[A] - \mathbb{P}_{\mathcal{O}_0}[A]| \\
&= \mathbb{P}_{\mathcal{O}_1}[\text{bad}_1] \\
&\leq \sum_{i=1}^q \mathbb{P}_{\mathcal{O}_1}[u_i = v_i, w_i = 0^b] \\
&= \frac{q}{2^n}.
\end{aligned} \tag{20}$$

#### 4.5 Final Step

Looking back, equations (17), (18), (19), and (20) have transformed our original goal (16) into

$$\text{Adv}_{\text{STH}}^{\text{prf}}(\mathcal{D}) \leq \|\mathbb{P}_{\mathcal{O}_0} - \mathbb{P}_{\mathcal{O}_4}\| + \frac{q}{2^n} + \text{Adv}_{\text{Trunc}}^{\text{prf}}(2q). \tag{21}$$

We now look at the worlds  $\mathcal{O}_0$  and  $\mathcal{O}_4$ . Noting that in both worlds  $\mathbf{u}$  and  $\mathbf{v}$  are generated identically, we can parameterize these worlds. We define  $\mathcal{O}_0^{\mathbf{u}, \mathbf{v}} =$

$T(\mathbf{u}, \mathbf{v})$  and  $\mathcal{O}_4^{\mathbf{u}, \mathbf{v}} = \text{PSoP}(\mathbf{u}, \mathbf{v})$ , so that in both cases  $\mathbf{u}$  and  $\mathbf{v}$  are generated by  $S$  and  $\mathbf{w}$  by  $\mathcal{O}_b^{\mathbf{u}, \mathbf{v}}$  for  $b \in \{0, 4\}$ . This means that

$$\begin{aligned} \|\mathbb{P}_{\mathcal{O}_0} - \mathbb{P}_{\mathcal{O}_4}\| &= \sum_{\mathbf{u}, \mathbf{v}} \sum_{\mathbf{w}} \max(0, \mathbb{P}_{\mathcal{O}_0}[(\mathbf{u}, \mathbf{v}, \mathbf{w})] - \mathbb{P}_{\mathcal{O}_4}[(\mathbf{u}, \mathbf{v}, \mathbf{w})]) \\ &= \sum_{\mathbf{u}, \mathbf{v}} \sum_{\mathbf{w}} \max(0, \mathbb{P}_{\mathcal{O}_0^{\mathbf{u}, \mathbf{v}}}[\mathbf{w}] - \mathbb{P}_{\mathcal{O}_4^{\mathbf{u}, \mathbf{v}}}[\mathbf{w}]) \cdot \mathbb{P}_S[(\mathbf{u}, \mathbf{v})] \\ &= \sum_{\mathbf{u}, \mathbf{v}} \left\| \mathbb{P}_{\mathcal{O}_0^{\mathbf{u}, \mathbf{v}}} - \mathbb{P}_{\mathcal{O}_4^{\mathbf{u}, \mathbf{v}}} \right\| \cdot \mathbb{P}_S[(\mathbf{u}, \mathbf{v})] \\ &= \mathbb{E}_{\mathbf{u}, \mathbf{v}} \left[ \left\| \mathbb{P}_{\mathcal{O}_0^{\mathbf{u}, \mathbf{v}}} - \mathbb{P}_{\mathcal{O}_4^{\mathbf{u}, \mathbf{v}}} \right\| \right], \end{aligned}$$

with  $\mathbf{u}$  and  $\mathbf{v}$  drawn uniformly. The remaining task boils down to bounding the distance between PSOP of Algorithm 3 and random function  $T$  of Algorithm 5.

For this, we first define  $C_{\mathbf{u}, \mathbf{v}}(i)$  and  $C'_{\mathbf{u}, \mathbf{v}}(i)$  as the number of previous elements in  $\mathbf{u}, \mathbf{v}$  equal to  $u_i$  and  $v_i$ , respectively, as follows:

$$\begin{aligned} C_{\mathbf{u}, \mathbf{v}}(i) &= |\{j \mid j < i, u_j = u_i\}| + |\{j \mid j < i, v_j = u_i\}|, \\ C'_{\mathbf{u}, \mathbf{v}}(i) &= |\{j \mid j < i, u_j = v_i\}| + |\{j \mid j < i, v_j = v_i\}|. \end{aligned}$$

In our derivation we want to assume that these values stay below  $2^{b-2}$ . We define  $\mathbf{bad}_2$  as the set of all  $(\mathbf{u}, \mathbf{v})$  such that  $C_{\mathbf{u}, \mathbf{v}}(i) \geq 2^{b-2}$  or  $C'_{\mathbf{u}, \mathbf{v}}(i) \geq 2^{b-2}$  for some  $i$ . We want to discard the bad cases while still reasoning about  $\mathbf{u}$  and  $\mathbf{v}$  as uniformly random values. For this, we use the following lemma.

**Lemma 3.** *Let  $f$  be a non-negative function such that  $\left\| \mathbb{P}_{\mathcal{O}_0^{\mathbf{u}, \mathbf{v}}} - \mathbb{P}_{\mathcal{O}_4^{\mathbf{u}, \mathbf{v}}} \right\| \leq f(\mathbf{u}, \mathbf{v})$  for  $(\mathbf{u}, \mathbf{v}) \notin \mathbf{bad}_2$ . Then*

$$\mathbb{E}_{\mathbf{u}, \mathbf{v}} \left[ \left\| \mathbb{P}_{\mathcal{O}_0^{\mathbf{u}, \mathbf{v}}} - \mathbb{P}_{\mathcal{O}_4^{\mathbf{u}, \mathbf{v}}} \right\| \right] \leq \mathbb{E}_{\mathbf{u}, \mathbf{v}} [f(\mathbf{u}, \mathbf{v})] \quad (22)$$

$$+ \mathbb{P}_{\mathbf{u}, \mathbf{v}} [(\mathbf{u}, \mathbf{v}) \in \mathbf{bad}_2]. \quad (23)$$

Note that in  $\mathbb{E}_{\mathbf{u}, \mathbf{v}} [f(\mathbf{u}, \mathbf{v})]$  the values  $\mathbf{u}, \mathbf{v}$  are still drawn uniformly.

*Proof.* For  $(\mathbf{u}, \mathbf{v}) \notin \mathbf{bad}_2$  we have that  $\left\| \mathbb{P}_{\mathcal{O}_0^{\mathbf{u}, \mathbf{v}}} - \mathbb{P}_{\mathcal{O}_4^{\mathbf{u}, \mathbf{v}}} \right\| \leq f(\mathbf{u}, \mathbf{v})$ . On the other hand, for  $(\mathbf{u}, \mathbf{v}) \in \mathbf{bad}_2$  we get  $\left\| \mathbb{P}_{\mathcal{O}_0^{\mathbf{u}, \mathbf{v}}} - \mathbb{P}_{\mathcal{O}_4^{\mathbf{u}, \mathbf{v}}} \right\| \leq 1 \leq f(\mathbf{u}, \mathbf{v}) + 1$ . Together, this means that  $\left\| \mathbb{P}_{\mathcal{O}_0^{\mathbf{u}, \mathbf{v}}} - \mathbb{P}_{\mathcal{O}_4^{\mathbf{u}, \mathbf{v}}} \right\| \leq f(\mathbf{u}, \mathbf{v}) + \mathbf{1}_{\mathbf{bad}_2}(\mathbf{u}, \mathbf{v})$ , where  $\mathbf{1}_{\mathbf{bad}_2}$  is the indicator function of  $\mathbf{bad}_2$ , which is 1 for  $(\mathbf{u}, \mathbf{v}) \in \mathbf{bad}_2$  and 0 otherwise. By taking the expectation on both sides this results in

$$\begin{aligned} \mathbb{E}_{\mathbf{u}, \mathbf{v}} \left[ \left\| \mathbb{P}_{\mathcal{O}_0^{\mathbf{u}, \mathbf{v}}} - \mathbb{P}_{\mathcal{O}_4^{\mathbf{u}, \mathbf{v}}} \right\| \right] &\leq \mathbb{E}_{\mathbf{u}, \mathbf{v}} [f(\mathbf{u}, \mathbf{v}) + \mathbf{1}_{\mathbf{bad}_2}(\mathbf{u}, \mathbf{v})] \\ &= \mathbb{E}_{\mathbf{u}, \mathbf{v}} [f(\mathbf{u}, \mathbf{v})] + \mathbb{P}_{\mathbf{u}, \mathbf{v}} [(\mathbf{u}, \mathbf{v}) \in \mathbf{bad}_2]. \quad \square \end{aligned}$$

We derive bounds for (22) with suitable  $f$  and (23) separately.

**4.5.1 Bounding (22).** As a first step we have to find a non-negative function  $f$  such that  $\|\mathbb{P}_{\mathcal{O}_0^{\mathbf{u},\mathbf{v}}} - \mathbb{P}_{\mathcal{O}_4^{\mathbf{u},\mathbf{v}}}\| \leq f(\mathbf{u}, \mathbf{v})$  for  $(\mathbf{u}, \mathbf{v}) \notin \text{bad}_2$ . The following theorem gives such function.

**Theorem 3.** *Let  $a, b, q \in \mathbb{N}$  and let  $\mathbf{u} = (u_1, \dots, u_q)$  and  $\mathbf{v} = (v_1, \dots, v_q)$  be vectors of length  $q$  with elements in  $\{0, 1\}^a$  such that  $C_{\mathbf{u},\mathbf{v}}(i), C'_{\mathbf{u},\mathbf{v}}(i) < 2^{b-2}$  for all  $i$ . Let  $\mathcal{O}$  and  $\mathcal{R}$  be as in Algorithm 6. Then*

$$\|\mathbb{P}_{\mathcal{O}} - \mathbb{P}_{\mathcal{R}}\| \leq \sqrt{\frac{4}{2^{3b}} \sum_{i=1}^q C_{\mathbf{u},\mathbf{v}}(i) \cdot C'_{\mathbf{u},\mathbf{v}}(i)}.$$

---

**Algorithm 6  $\mathcal{O}$**

---

```

1: function  $\mathcal{O}$ 
2:   for  $k \in \{0, 1\}^a$  do
3:      $P_k \xleftarrow{\$} \text{Perm}[b]$ 
4:   for  $i \leftarrow 1$  to  $q$  do
5:      $U_i \leftarrow P_{u_i}(\langle C_{\mathbf{u},\mathbf{v}}(i) \rangle_{b-1} \| 0)$ 
6:      $V_i \leftarrow P_{v_i}(\langle C'_{\mathbf{u},\mathbf{v}}(i) \rangle_{b-1} \| 1)$ 
7:      $w_i \leftarrow U_i \oplus V_i$ 
8:   return  $\mathbf{w} = (w_1, \dots, w_q)$ 

```

---



---

**Algorithm 7  $\mathcal{R}$**

---

```

1: function  $\mathcal{R}$ 
2:   for  $i \leftarrow 1$  to  $q$  do
3:     if  $u_i \neq v_i$  then
4:        $w_i \xleftarrow{\$} \{0, 1\}^b$ 
5:     else
6:        $w_i \xleftarrow{\$} \{0, 1\}^b \setminus \{0^b\}$ 
7:   return  $\mathbf{w} = (w_1, \dots, w_q)$ 

```

---

Here  $\langle x \rangle_n$  is the encoding of  $x$  as a  $n$ -bit string.

The proof of Theorem 3 will be given in Section 5.

It is obvious that  $\mathcal{R}$  equals  $\mathcal{O}_0^{\mathbf{u},\mathbf{v}}$ . We will next show that  $\mathcal{O}$  generates the same distribution as  $\mathcal{O}_4^{\mathbf{u},\mathbf{v}}$ , by looking at the distribution of  $U_i$  given all previous values (the analysis is symmetrical for the values  $V_i$ ).

In world  $\mathcal{O}$ , the value is generated by the permutation  $P_{u_i}$  with the input  $\langle C_{\mathbf{u},\mathbf{v}}(i) \rangle_{b-1} \| 0$ . Note that we can encode  $C_{\mathbf{u},\mathbf{v}}(i)$  as a  $b-1$ -bit string, as we assume that  $C_{\mathbf{u},\mathbf{v}}(i) < 2^{b-2} < 2^{b-1}$ . The output value of  $P_{u_i}$  will be distributed uniformly from  $\{0, 1\}^b$  minus its previously generated values. These values, in turn, are  $U_j$  and  $V_j$  such that  $u_j = u_i$  or  $v_j = u_i$ , respectively, for  $j < i$ . Note that we do get a new value, as  $\langle C_{\mathbf{u},\mathbf{v}}(i) \rangle_{b-1} \| 0$  is different from  $\langle C_{\mathbf{u},\mathbf{v}}(j) \rangle_{b-1} \| 0$  or  $\langle C'_{\mathbf{u},\mathbf{v}}(j) \rangle_{b-1} \| 1$  for such  $j$ .

In world  $\mathcal{O}_4^{\mathbf{u},\mathbf{v}}$ , the value is generated by the single permutation  $P'$  selected from the set  $\text{Perm}_{\text{comp}}(\mathbf{u}, \mathbf{v})$  with the new input  $i \| 0$ . Note that  $\text{Perm}_{\text{comp}}(\mathbf{u}, \mathbf{v})$  is never empty as we assume that  $C_{\mathbf{u},\mathbf{v}}(i), C'_{\mathbf{u},\mathbf{v}}(i) < 2^{b-2} < 2^b$  for all  $i$ , hence there always exists a permutation that would generate  $\mathbf{u}$  and  $\mathbf{v}$ . We know that the first  $a$  bits of the output of  $P'$  have to be equal to  $u_i$ . This means that previously generated values of  $P'$  do not matter as long as their first  $a$  bits are different. Again, for the last  $b$  bits we know that they cannot be equal to  $U_j$  or  $V_j$  with  $u_j = u_i$  or  $v_j = u_i$ , respectively, for  $j < i$ . Furthermore, the value is uniformly chosen from the remaining elements in the set  $\{0, 1\}^b$ , as  $P'$  is selected uniformly from  $\text{Perm}_{\text{comp}}(\mathbf{u}, \mathbf{v})$ .



This means that the distribution of all  $U_i$ 's is the same in both worlds. As the analysis of all  $V_i$ 's is similar, both  $\mathcal{O}$  and  $\mathcal{O}_4^{\mathbf{u}, \mathbf{v}}$  have the same distribution.

We will now use Theorem 3 to bound (22). As the property  $\mathbb{E}[X]^2 \leq \mathbb{E}[X^2]$  implies that  $\mathbb{E}[\sqrt{X}] \leq \sqrt{\mathbb{E}[X]}$ , we get

$$\begin{aligned} \mathbb{E}_{\mathbf{u}, \mathbf{v}} [f(\mathbf{u}, \mathbf{v})] &\leq \sqrt{\mathbb{E}_{\mathbf{u}, \mathbf{v}} \left[ \frac{4}{2^{3b}} \sum_{i=1}^q C_{\mathbf{u}, \mathbf{v}}(i) \cdot C'_{\mathbf{u}, \mathbf{v}}(i) \right]} \\ &= \sqrt{\frac{4}{2^{3b}} \sum_{i=1}^q \mathbb{E}_{\mathbf{u}, \mathbf{v}} [C_{\mathbf{u}, \mathbf{v}}(i) \cdot C'_{\mathbf{u}, \mathbf{v}}(i)]}. \end{aligned} \quad (24)$$

Although  $C_{\mathbf{u}, \mathbf{v}}(i)$  and  $C'_{\mathbf{u}, \mathbf{v}}(i)$  are not independent, we will show that their *expectations* are independent, i.e. that  $\mathbb{E}_{\mathbf{u}, \mathbf{v}} [C'_{\mathbf{u}, \mathbf{v}}(i) \mid C_{\mathbf{u}, \mathbf{v}}(i)] = \mathbb{E}_{\mathbf{u}, \mathbf{v}} [C'_{\mathbf{u}, \mathbf{v}}(i)]$ . First of all, as  $\mathbf{u}$  and  $\mathbf{v}$  are distributed uniform, every  $u_j$  and  $v_j$  has a probability of  $1/2^a$  of being equal to  $u_i$  or  $v_i$  for  $j < i$ , hence

$$\mathbb{E}_{\mathbf{u}, \mathbf{v}} [C_{\mathbf{u}, \mathbf{v}}(i)] = \mathbb{E}_{\mathbf{u}, \mathbf{v}} [C'_{\mathbf{u}, \mathbf{v}}(i)] = \frac{2(i-1)}{2^a}.$$

Next, we have to compute  $\mathbb{E}_{\mathbf{u}, \mathbf{v}} [C'_{\mathbf{u}, \mathbf{v}}(i) \mid C_{\mathbf{u}, \mathbf{v}}(i)]$ . In this case we condition over the event that  $u_i = v_i$ . If this is the case, we know the value of  $C'_{\mathbf{u}, \mathbf{v}}(i)$  exactly, as it is equal to  $C_{\mathbf{u}, \mathbf{v}}(i)$ . On the other hand, if  $u_i \neq v_i$ , we know that there are  $C_{\mathbf{u}, \mathbf{v}}(i)$  less candidates, but also that every candidate has a higher probability  $1/(2^a - 1)$  of being equal to  $v_i$ . This gives the following result:

$$\begin{aligned} \mathbb{E}_{\mathbf{u}, \mathbf{v}} [C'_{\mathbf{u}, \mathbf{v}}(i) \mid C_{\mathbf{u}, \mathbf{v}}(i)] &= \mathbb{P}_{\mathbf{u}, \mathbf{v}} [u_i = v_i] \cdot \mathbb{E}_{\mathbf{u}, \mathbf{v}} [C'_{\mathbf{u}, \mathbf{v}}(i) \mid C_{\mathbf{u}, \mathbf{v}}(i), u_i = v_i] \\ &\quad + \mathbb{P}_{\mathbf{u}, \mathbf{v}} [u_i \neq v_i] \cdot \mathbb{E}_{\mathbf{u}, \mathbf{v}} [C'_{\mathbf{u}, \mathbf{v}}(i) \mid C_{\mathbf{u}, \mathbf{v}}(i), u_i \neq v_i] \\ &= \frac{1}{2^a} \cdot C_{\mathbf{u}, \mathbf{v}}(i) + \left(1 - \frac{1}{2^a}\right) \cdot \frac{2(i-1) - C_{\mathbf{u}, \mathbf{v}}(i)}{2^a - 1} \\ &= \frac{1}{2^a} \cdot C_{\mathbf{u}, \mathbf{v}}(i) + \frac{2^a - 1}{2^a} \cdot \frac{2(i-1) - C_{\mathbf{u}, \mathbf{v}}(i)}{2^a - 1} \\ &= \frac{1}{2^a} \cdot C_{\mathbf{u}, \mathbf{v}}(i) + \frac{1}{2^a} \cdot (2(i-1) - C_{\mathbf{u}, \mathbf{v}}(i)) \\ &= \frac{2(i-1)}{2^a} \\ &= \mathbb{E}_{\mathbf{u}, \mathbf{v}} [C'_{\mathbf{u}, \mathbf{v}}(i)]. \end{aligned}$$

By Lemma 1 this means that we have  $\mathbb{E}_{\mathbf{u}, \mathbf{v}} [C_{\mathbf{u}, \mathbf{v}}(i) \cdot C'_{\mathbf{u}, \mathbf{v}}(i)] = \mathbb{E}_{\mathbf{u}, \mathbf{v}} [C_{\mathbf{u}, \mathbf{v}}(i)] \cdot \mathbb{E}_{\mathbf{u}, \mathbf{v}} [C'_{\mathbf{u}, \mathbf{v}}(i)]$ , so

$$\begin{aligned}
(24) &= \sqrt{\frac{4}{2^{3b}} \sum_{i=1}^q \mathbb{E}_{\mathbf{u}, \mathbf{v}} [C_{\mathbf{u}, \mathbf{v}}(i)] \cdot \mathbb{E}_{\mathbf{u}, \mathbf{v}} [C'_{\mathbf{u}, \mathbf{v}}(i)]} \\
&= \sqrt{\frac{4}{2^{3b}} \sum_{i=1}^q \left(\frac{2(i-1)}{2^a}\right)^2} \\
&\leq \sqrt{\frac{4}{2^{3b}} \frac{4 \cdot q^3}{3 \cdot 2^{2a}}} \\
&\leq \sqrt{6 \cdot \frac{q^3}{2^{3n-a}}} \\
&\leq 3 \left(\frac{q}{2^{n-a/3}}\right)^{3/2}.
\end{aligned}$$

This finishes the first part of the bound.

**4.5.2 Bounding (23).** We now look at (23). The event  $(\mathbf{u}, \mathbf{v}) \in \text{bad}_2$  occurs when a  $2^{b-2}$ -collision occurs inside  $(\mathbf{u}, \mathbf{v})$ . As  $\mathbf{u}$  and  $\mathbf{v}$  are chosen uniformly, the probability of getting a  $t$ -collision is bounded by

$$\frac{(2q)^t}{2^{a(t-1)} \cdot t!},$$

where we later substitute  $t = 2^{b-2}$ . By Stirling's approximation, which says that

$$t! \geq \sqrt{2\pi t} \left(\frac{t}{e}\right)^t \geq \sqrt{2\pi} \left(2^{-3/2} \cdot t\right)^t,$$

we get that

$$\begin{aligned}
(23) &\leq \frac{1}{\sqrt{2\pi}} \cdot \frac{(2q)^t}{2^{a(t-1)}} \cdot \left(\frac{1}{2^{-3/2} \cdot t}\right)^t \\
&= \frac{2^a}{\sqrt{2\pi}} \cdot \left(\frac{2q}{2^a}\right)^t \cdot \left(\frac{1}{2^{-3/2} \cdot t}\right)^t \\
&= \frac{2^a}{\sqrt{2\pi}} \cdot \left(\frac{2q}{2^{a-3/2} \cdot t}\right)^t.
\end{aligned} \tag{25}$$

From the assumption that  $b \geq n/12$  and  $b \geq 10$  (hence  $b \leq 2^b/96$ ), we get that  $a \leq n \leq 12b \leq 2^b/8 = t/2$ , so

$$(25) \leq \frac{1}{\sqrt{2\pi}} \left(\frac{2q}{2^{a-2} \cdot t}\right)^t. \tag{26}$$

Finally, by substituting  $t = 2^{b-2}$  we get

$$\begin{aligned} (26) &= \frac{1}{\sqrt{2\pi}} \left( \frac{2q}{2^{a-2} \cdot 2^{b-2}} \right)^{2^{b-2}} \\ &= \frac{1}{\sqrt{2\pi}} \left( \frac{q}{2^{n-5}} \right)^{2^{b-2}}. \end{aligned}$$

This finishes the second part of the bound.

## 5 Proof of Theorem 3

Let  $a, b, q \in \mathbb{N}$  and let  $\mathbf{u} = (u_1, \dots, u_q)$  and  $\mathbf{v} = (v_1, \dots, v_q)$  be vectors of length  $q$  with elements in  $\{0, 1\}^a$  such that  $C_{\mathbf{u}, \mathbf{v}}(i), C'_{\mathbf{u}, \mathbf{v}}(i) < 2^{b-2}$  for all  $i$ . Let  $\mathcal{O}$  and  $\mathcal{R}$  be as in Algorithm 6. We denote their outputs by  $\mathbf{w} = (w_1, \dots, w_q)$ . Further, for  $i \in \{0, \dots, q\}$  denote  $\mathbf{w}_i = (w_1, \dots, w_i)$ .

We will rely on the chi-squared method by Dai et al. [22]. For each  $i = 1, \dots, q$  and each  $\mathbf{w}_{i-1}$ , define

$$\chi^2(\mathbf{w}_{i-1}) = \sum_w \frac{(\mathbb{P}_{\mathcal{O}}[w_i = w \mid \mathbf{w}_{i-1}] - \mathbb{P}_{\mathcal{R}}[w_i = w \mid \mathbf{w}_{i-1}])^2}{\mathbb{P}_{\mathcal{R}}[w_i = w \mid \mathbf{w}_{i-1}]}. \quad (27)$$

The chi-squared method gives the following bound [22]:

**Lemma 4 (Chi-Squared Method).** *Consider two systems  $\mathcal{O}, \mathcal{R}$ . Suppose that for any vector  $\mathbf{w}_i$ ,  $\mathbb{P}_{\mathcal{R}}[\mathbf{w}_i] > 0$  whenever  $\mathbb{P}_{\mathcal{O}}[\mathbf{w}_i] > 0$ . Then,*

$$\|\mathbb{P}_{\mathcal{O}} - \mathbb{P}_{\mathcal{R}}\| \leq \left( \frac{1}{2} \sum_{i=1}^q \mathbb{E}_{\mathcal{O}}[\chi^2(\mathbf{w}_{i-1})] \right)^{1/2}.$$

Our proof of Theorem 3 is related to that of Dai et al. [22], where they look at both the SoSP construction of (2) for a single permutation and the SoP construction of (1) based on two different permutations. In our terminology, these correspond to the case of  $\mathbf{u} = \mathbf{v} = (0^a, \dots, 0^a)$  and  $\mathbf{u} = (0^a, \dots, 0^a)$ ,  $\mathbf{v} = (1^a, \dots, 1^a)$ , respectively. Our analysis, thus, carefully combines and generalizes these approaches. An additional difficulty arises from the fact that the different cases *depend on* the values of  $\mathbf{u}$  and  $\mathbf{v}$ , that may be arbitrary.

In the chi-squared method we have to reason over  $\mathbb{P}_{\mathcal{O}}[w_i = w \mid \mathbf{w}_{i-1}]$ . However, in our case it is difficult to do this directly, as the conditional probability does not give information about the intermediate values  $U_j$  and  $V_j$  for  $j < i$ , but only about their sum  $w_j = U_j \oplus V_j$ . The following lemma shows that we can assume this extra information without increasing the bound. Intuitively, this is similar to the fact that giving an adversary more information does not lower its advantage.

**Lemma 5.** Let  $Z_{i-1}$  be a random variable in world  $\mathcal{O}$  (but not necessarily in world  $\mathcal{R}$ ). Then,

$$\mathbb{E}_{\mathcal{O}} [\chi^2(\mathbf{w}_{i-1})] \leq \sum_w \mathbb{E}_{\mathcal{O}} \left[ \frac{(\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{w}_{i-1}, Z_{i-1}] - \mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}])^2}{\mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}]} \right].$$

*Proof.* Recall that

$$\chi^2(\mathbf{w}_{i-1}) = \sum_w \frac{(\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{w}_{i-1}] - \mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}])^2}{\mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}]}.$$

Let  $\mathbf{w}_{i-1}$  and  $w$  be fixed and write  $p = \mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}]$ . Then

$$\begin{aligned} & \frac{1}{p} (\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{w}_{i-1}] - p)^2 \\ &= \frac{1}{p} \left( \sum_z \mathbb{P}_{\mathcal{O}} [Z_{i-1} = z \mid \mathbf{w}_{i-1}] \cdot \mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{w}_{i-1}, Z_{i-1} = z] - p \right)^2 \\ &= \frac{1}{p} \left( \mathbb{E}_{\mathcal{O}} \left[ \mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{w}_{i-1}, Z_{i-1}] \mid \mathbf{w}_{i-1} \right] - p \right)^2 \\ &= \frac{1}{p} \mathbb{E}_{\mathcal{O}} \left[ \mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{w}_{i-1}, Z_{i-1}] - p \mid \mathbf{w}_{i-1} \right]^2 \\ &\leq \frac{1}{p} \mathbb{E}_{\mathcal{O}} \left[ (\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{w}_{i-1}, Z_{i-1}] - p)^2 \mid \mathbf{w}_{i-1} \right] \\ &= \mathbb{E}_{\mathcal{O}} \left[ \frac{1}{p} (\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{w}_{i-1}, Z_{i-1}] - p)^2 \mid \mathbf{w}_{i-1} \right]. \end{aligned}$$

Furthermore, by taking the expectation on both sides we get

$$\begin{aligned} & \mathbb{E}_{\mathcal{O}} \left[ \frac{(\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{w}_{i-1}] - \mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}])^2}{\mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}]} \right] \\ &\leq \mathbb{E}_{\mathcal{O}} \left[ \mathbb{E}_{\mathcal{O}} \left[ \frac{(\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{w}_{i-1}, Z_{i-1}] - \mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}])^2}{\mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}]} \mid \mathbf{w}_{i-1} \right] \right] \\ &= \mathbb{E}_{\mathcal{O}} \left[ \frac{(\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{w}_{i-1}, Z_{i-1}] - \mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}])^2}{\mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}]} \right]. \end{aligned}$$

The proof is completed by combining both equations.  $\square$

In our case we take  $Z_i = (\mathbf{U}_i, \mathbf{V}_i)$  with  $\mathbf{U}_i = (U_1, \dots, U_i)$  and  $\mathbf{V}_i = (V_1, \dots, V_i)$ . Note that in this case we can ignore  $\mathbf{w}_i$ , as its value is fixed given  $Z_i$ .

We now reformulate  $\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{U}_{i-1}, \mathbf{V}_{i-1}]$ . Given  $\mathbf{U}_{i-1}$  and  $\mathbf{V}_{i-1}$ , we look at the probability that  $U_i \oplus V_i = w$  for an arbitrary  $w$ . For this, we define:

$$\begin{aligned} S_i &= \{U_j \mid j < i, u_j = u_i\} \cup \{V_j \mid j < i, v_j = u_i\}, \\ S'_i &= \{U_j \mid j < i, u_j = v_i\} \cup \{V_j \mid j < i, v_j = v_i\}. \end{aligned}$$

We write  $s_i = |S_i|$ ,  $s'_i = |S'_i|$ , and  $D_{i,w} = |S_i \cap (S'_i \oplus w)|$ .

In order for  $U_i \oplus V_i$  to be equal to  $w$ , the variable  $U_i$  must take a value from  $\{0, 1\}^b \setminus (S_i \cup (S'_i \oplus w))$ . The number of choices for this is exactly

$$\begin{aligned} 2^b - |S_i \cup (S'_i \oplus w)| &= 2^b - |S_i| - |S'_i \oplus w| + |S_i \cap (S'_i \oplus w)| \\ &= 2^b - s_i - s'_i + D_{i,w}. \end{aligned} \quad (28)$$

Moreover, the choice of  $V_i$  is fixed to  $U_i \oplus w$ .

We claim that, regardless of whether  $u_i$  and  $v_i$  are equal or distinct,

$$\mathbb{E}_{\mathcal{O}} \left[ \frac{(\mathbb{P}_{\mathcal{O}}[w_i = w \mid \mathbf{U}_{i-1}, \mathbf{V}_{i-1}] - \mathbb{P}_{\mathcal{R}}[w_i = w \mid \mathbf{w}_{i-1}])^2}{\mathbb{P}_{\mathcal{R}}[w_i = w \mid \mathbf{w}_{i-1}]} \right] \leq \frac{8s_i s'_i}{2^{4b}}. \quad (29)$$

The proof of (29) will be given in Section 5.2 (for the case where  $u_i = v_i$ ) and in Section 5.3 (for the case where  $u_i \neq v_i$ ). The two proofs will rely on some probabilistic analysis of  $D_{i,w}$ , given in Section 5.1.

Before getting there, however, we first complete the proof under the hypothesis that (29) holds. Note that  $s_i$  and  $s'_i$  do not depend on the specific values of  $U_j$  or  $V_j$ , they only depend on the value of  $\mathbf{u}$  and  $\mathbf{v}$ . In fact  $s_i = C_{\mathbf{u},\mathbf{v}}(i)$  and  $s'_i = C'_{\mathbf{u},\mathbf{v}}(i)$ , which means that

$$\begin{aligned} &\|\mathbb{P}_{\mathcal{O}} - \mathbb{P}_{\mathcal{R}}\|^2 \\ &\leq \frac{1}{2} \sum_{i=1}^q \mathbb{E}_{\mathcal{O}} [\chi^2(\mathbf{w}_{i-1})] \\ &\leq \frac{1}{2} \sum_{i=1}^q \sum_w \mathbb{E}_{\mathcal{O}} \left[ \frac{(\mathbb{P}_{\mathcal{O}}[w_i = w \mid \mathbf{U}_{i-1}, \mathbf{V}_{i-1}] - \mathbb{P}_{\mathcal{R}}[w_i = w \mid \mathbf{w}_{i-1}])^2}{\mathbb{P}_{\mathcal{R}}[w_i = w \mid \mathbf{w}_{i-1}]} \right] \\ &\leq \frac{4}{2^{4b}} \sum_{i=1}^q \sum_w s_i s'_i \\ &\leq \frac{4}{2^{3b}} \sum_{i=1}^q C_{\mathbf{u},\mathbf{v}}(i) \cdot C'_{\mathbf{u},\mathbf{v}}(i). \end{aligned}$$

### 5.1 Expectation and Variance of $D_{i,w}$

The value  $D_{i,w}$  counts the number of elements  $g \in \{0, 1\}^b$  such that  $g \in S_i$  and  $g \oplus w \in S'_i$ . Our goal is to derive two bounds, one on its expected value  $\mathbb{E}[D_{i,w}]$  and one on its variance  $\mathbf{Var}[D_{i,w}]$ , with the randomness chosen over the sets  $S_i$  and  $S'_i$  which are chosen uniform from  $\{0, 1\}^b$  without replacement. Note that this corresponds with world  $\mathcal{O}$ . We will, again, do so for the two different cases: equal permutations (for which  $S_i$  and  $S'_i$  are identical) in Section 5.1.2 and different permutations (for which  $S_i$  and  $S'_i$  are independent) in Section 5.1.3. The proofs share common analysis, which is first given in Section 5.1.1.

The proof is based on Lemma 4 of Bhattacharya and Nandi [13], that considers a variant of SoP where a single output is summed with multiple other

outputs, but where all outputs are still from the same permutation. We look at the special case where it is summed with just one value, but extend the analysis to the case of different independent permutations.

**5.1.1 General Analysis.** Let  $I_g$  be the random variable that is 1 if  $g \in S_i$  and  $g \oplus w \in S'_i$ , and 0 otherwise. Note that  $D_{i,w} = \sum_{g \in \{0,1\}^b} I_g$ . For the expectation we have that

$$\begin{aligned} \mathbb{E}[I_g] &= \mathbb{P}[g \in S_i, g \oplus w \in S'_i] \\ &= \mathbb{P}[g \in S_i] \mathbb{P}[g \oplus w \in S'_i \mid g \in S_i], \end{aligned}$$

where we have to compute this value separately for equal and different permutations. For the expectation of  $D_{i,w}$  we simply find

$$\mathbb{E}[D_{i,w}] = \sum_g \mathbb{E}[I_g].$$

We now look at the variance of  $D_{i,w}$ . We use the following property:

$$\begin{aligned} \mathbf{Var}[D_{i,w}] &= \mathbf{Var}\left[\sum_g I_g\right] \\ &= \sum_g \mathbf{Var}[I_g] + \sum_{g \neq g'} \mathbf{Cov}(I_g, I_{g'}), \end{aligned} \quad (30)$$

where

$$\begin{aligned} \mathbf{Cov}(I_g, I_{g'}) &= \mathbb{E}[I_g I_{g'}] - \mathbb{E}[I_g] \mathbb{E}[I_{g'}] \\ &= \mathbb{E}[I_g] \mathbb{P}[I_{g'} = 1 \mid I_g = 1] - \mathbb{E}[I_g] \mathbb{E}[I_{g'}]. \end{aligned}$$

First, we will argue that  $\mathbf{Cov}(I_g, I_{g'}) \leq 0$  whenever  $g' \neq g \oplus w$ . Indeed, if this condition is satisfied, we have that  $g', g' \oplus w, g$  and  $g \oplus w$  are mutually distinct, and thus that

$$\begin{aligned} \mathbb{P}[I_{g'} = 1 \mid I_g = 1] &= \mathbb{P}[g' \in S_i, g' \oplus w \in S'_i \mid g \in S_i, g \oplus w \in S'_i] \\ &\leq \mathbb{P}[g' \in S_i, g' \oplus w \in S'_i] \\ &= \mathbb{E}[I_{g'}]. \end{aligned}$$

For the derivation of the inequality, we have used the following observation. On the one hand, for equal permutations,  $S_i$  and  $S'_i$  are identical, so the inequality is satisfied as the probability of having two specific elements in a set of fixed size decreases when it is known that two other elements are already in it. On the other hand, for different permutations,  $S_i$  and  $S'_i$  are independent, so the inequality boils down to two independent cases with one element instead of two. Henceforth, we obtained that  $\mathbf{Cov}(I_g, I_{g'}) \leq 0$  whenever  $g' \neq g \oplus w$ .

Having eliminated the case of  $\mathbf{Cov}(I_g, I_{g'})$  for  $g' \neq g \oplus w$ , we can proceed as follows for the second term of (30):

$$\begin{aligned}
\sum_{g \neq g'} \mathbf{Cov}(I_g, I_{g'}) &\leq \sum_g \mathbf{Cov}(I_g, I_{g \oplus w}) \\
&= \sum_g \mathbb{E}[I_g] \mathbb{P}[I_{g \oplus w} = 1 \mid I_g = 1] - \mathbb{E}[I_g] \mathbb{E}[I_{g \oplus w}] \\
&\leq \sum_g \mathbb{E}[I_g] - \mathbb{E}[I_g] \mathbb{E}[I_{g \oplus w}] \\
&= \sum_g \mathbb{E}[I_g^2] - \mathbb{E}[I_g]^2 \\
&= \sum_g \mathbf{Var}[I_g].
\end{aligned}$$

Concluding,

$$\begin{aligned}
\mathbf{Var}[D_{i,w}] &\leq 2 \sum_g \mathbf{Var}[I_g] \\
&= 2 \sum_g \mathbb{E}[I_g] (1 - \mathbb{E}[I_g]) \\
&\leq 2 \sum_g \mathbb{E}[I_g] \\
&= 2 \cdot \mathbb{E}[D_{i,w}].
\end{aligned}$$

**5.1.2 Equal Permutations.** In this case we have that  $S_i$  and  $S'_i$  are identical. This means that for  $w \neq 0^b$

$$\mathbb{P}[g \in S_i] \mathbb{P}[g \oplus w \in S'_i \mid g \in S_i] = \frac{s_i(s_i - 1)}{2^b(2^b - 1)}.$$

Hence, we have obtained:

$$\mathbb{E}_{\mathcal{O}}[D_{i,w}] = \frac{s_i(s_i - 1)}{2^b - 1}, \quad (31)$$

$$\mathbf{Var}_{\mathcal{O}}[D_{i,w}] \leq \frac{2s_i(s_i - 1)}{2^b - 1} \leq \frac{2s_i s'_i}{2^b}. \quad (32)$$

**5.1.3 Different Permutations.** Now  $S_i$  and  $S'_i$  are independent, and hence

$$\mathbb{P}[g \in S_i] \mathbb{P}[g \oplus w \in S'_i \mid g \in S_i] = \frac{s_i s'_i}{2^{2b}}.$$

Hence, we have obtained:

$$\mathbb{E}_{\mathcal{O}}[D_{i,w}] = \frac{s_i s'_i}{2^b}, \quad (33)$$

$$\mathbf{Var}_{\mathcal{O}}[D_{i,w}] \leq \frac{2s_i s'_i}{2^b}. \quad (34)$$

## 5.2 (29) for Equal Permutations

From (28) the number of valid choices for  $U_i$  and  $V_i$  is equal to  $2^b - 2s_i + D_{i,w}$ , as  $s_i = s'_i$  for equal permutations. Furthermore, the total number of possible choices is  $2^b - s_i$  for  $U_i$  and  $2^b - s_i - 1$  for  $V_i$ . This means that

$$\begin{aligned}\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{U}_{i-1}, \mathbf{V}_{i-1}] &= \frac{2^b - 2s_i + D_{i,w}}{(2^b - s_i)(2^b - s_i - 1)} \\ &= \frac{(2^b - 1) - s_i - (s_i - 1) + D_{i,w}}{((2^b - 1) - (s_i - 1))(2^b - 1 - s_i)}.\end{aligned}$$

As  $0^b$  is not possible in our modified ideal world, we have that  $\mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}] = 1/(2^b - 1)$ , which results in

$$\begin{aligned}(\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{U}_{i-1}, \mathbf{V}_{i-1}] - \mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}])^2 &= \left( \frac{(2^b - 1) - s_i - (s_i - 1) + D_{i,w}}{((2^b - 1) - (s_i - 1))(2^b - 1 - s_i)} - \frac{1}{2^b - 1} \right)^2 \\ &= \left( \frac{D_{i,w} - s_i(s_i - 1)/(2^b - 1)}{(2^b - s_i)(2^b - s_i - 1)} \right)^2 \\ &\leq \frac{4(D_{i,w} - s_i(s_i - 1)/(2^b - 1))^2}{2^{4b}},\end{aligned}$$

using that  $s_i < 2^{b-2}$ . We know from (31-32) that  $\mathbb{E}_{\mathcal{O}} [D_{i,w}] = s_i(s_i - 1)/(2^b - 1)$  and  $\mathbf{Var}_{\mathcal{O}} [D_{i,w}] \leq 2s_i^2/2^b$  for any  $w \in \{0, 1\}^b \setminus \{0^b\}$ , hence

$$\begin{aligned}\mathbb{E}_{\mathcal{O}} \left[ \frac{(\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{U}_{i-1}, \mathbf{V}_{i-1}] - \mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}])^2}{\mathbb{P}_{\mathcal{R}} [w_i = w \mid \mathbf{w}_{i-1}]} \right] &\leq \frac{4}{2^{3b}} \cdot \mathbb{E}_{\mathcal{O}} \left[ D_{i,w} - \frac{s_i(s_i - 1)}{2^b - 1} \right] \\ &= \frac{4}{2^{3b}} \cdot \mathbf{Var}_{\mathcal{O}} [D_{i,w}] \\ &\leq \frac{8s_i s'_i}{2^{4b}}.\end{aligned}$$

## 5.3 (29) for Different Permutations

From (28) the number of valid choices for  $U_i$  and  $V_i$  is equal to  $2^b - s_i - s'_i + D_{i,w}$ . Furthermore, the total number of possible choices is  $2^b - s_i$  for  $U_i$  and  $2^b - s'_i$  for  $V_i$ . This means that

$$\mathbb{P}_{\mathcal{O}} [w_i = w \mid \mathbf{U}_{i-1}, \mathbf{V}_{i-1}] = \frac{2^b - s_i - s'_i + D_{i,w}}{(2^b - s_i)(2^b - s'_i)}.$$



As  $u_i \neq v_i$ , all values in  $\{0, 1\}^b$  are possible in the ideal world, hence  $\mathbb{P}_{\mathcal{R}}[w_i = w \mid \mathbf{w}_{i-1}] = 1/2^b$ . This results in

$$\begin{aligned} & (\mathbb{P}_{\mathcal{O}}[w_i = w \mid \mathbf{U}_{i-1}, \mathbf{V}_{i-1}] - \mathbb{P}_{\mathcal{R}}[w_i = w \mid \mathbf{w}_{i-1}])^2 \\ &= \left( \frac{2^b - s_i - s'_i + D_{i,w}}{(2^b - s_i)(2^b - s'_i)} - \frac{1}{2^b} \right)^2 \\ &= \left( \frac{D_{i,w} - s_i s'_i}{(2^b - s_i)(2^b - s'_i)} \right)^2 \\ &\leq \frac{4(D_{i,w} - s_i s'_i/2^b)^2}{2^{4b}}, \end{aligned}$$

using that  $s_i, s'_i < 2^{b-2}$ . We know from (33-34) that  $\mathbb{E}_{\mathcal{O}}[D_{i,w}] = s_i s'_i / 2^b$  and  $\mathbf{Var}_{\mathcal{O}}[D_{i,w}] \leq 2s_i s'_i / 2^b$  for any  $w \in \{0, 1\}^b$ , hence

$$\begin{aligned} & \mathbb{E}_{\mathcal{O}} \left[ \frac{(\mathbb{P}_{\mathcal{O}}[w_i = w \mid \mathbf{U}_{i-1}, \mathbf{V}_{i-1}] - \mathbb{P}_{\mathcal{R}}[w_i = w \mid \mathbf{w}_{i-1}])^2}{\mathbb{P}_{\mathcal{R}}[w_i = w \mid \mathbf{w}_{i-1}]} \right] \\ &\leq \frac{4}{2^{3b}} \cdot \mathbb{E}_{\mathcal{O}} \left[ \left( D_{i,w} - \frac{s_i s'_i}{2^b} \right)^2 \right] \\ &= \frac{4}{2^{3b}} \cdot \mathbf{Var}_{\mathcal{O}}[D_{i,w}] \\ &\leq \frac{8s_i s'_i}{2^{4b}}. \end{aligned}$$

## 6 Proof of Theorem 2

The proof of Theorem 2 is very similar to the proof of Theorem 1, but with a few minor differences. First of all, the steps 4.1 and 4.2 remain basically the same and can be modified in a straightforward way. The step 4.3 is slightly different, as truncation is applied to two different permutations. This leads to the term  $2\mathbf{Adv}_{\text{Trunc}}^{\text{prf}}(q)$  instead of the old  $\mathbf{Adv}_{\text{Trunc}}^{\text{prf}}(2q)$ . Furthermore, the step 4.4 becomes obsolete as we do not have to limit the range in the case of two independent permutations. This means that the term  $q/2^n$  vanishes. Finally, the final step 4.5 remains roughly the same. In fact, as there are two independent permutations,  $\mathbf{u}$  and  $\mathbf{v}$  can be viewed separately. We might be able to use this information to improve some constants, but the gain is limited to those. We do not go into such detail and just reuse the old ones.

## 7 Application to GCM-SIV

GCM-SIV is a nonce misuse resistant authenticated encryption scheme of Gueron and Lindell, for which various versions exist [29, 32, 33, 39]. We consider the most recent one, that is also specified in internet draft IETF RFC [44]. It is built on top

of a block cipher  $E : \{0, 1\}^\kappa \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ , and the internet draft considers an instantiation with AES-128 (where  $\kappa = n = 128$ ) or AES-256 (where  $\kappa = 256$  and  $n = 128$ ).

If  $E$  is instantiated with AES-128, the first step of an evaluation of GCM-SIV is to derive two 128-bit subkeys  $k_1 \parallel k_2 \in \{0, 1\}^{256}$  based on key  $k$  and nonce  $\nu$  as in (6):

$$\begin{aligned} k_1 &= \text{left}_{n/2}(E_k(\nu\|0)) \parallel \text{left}_{n/2}(E_k(\nu\|1)), \\ k_2 &= \text{left}_{n/2}(E_k(\nu\|2)) \parallel \text{left}_{n/2}(E_k(\nu\|3)). \end{aligned} \quad (35)$$

Then, the associated data, message, and nonce are properly fed to the GHASH universal hash function (keyed with  $k_1$ ), its outcome is encrypted using  $E_{k_2}$ , and the resulting value is set as tag. This tag is, subsequently, set as input to counter mode based on  $E_{k_2}$  to obtain a keystream that is added to the plaintext to obtain the ciphertext. If  $E$ , on the other hand, is instantiated with AES-256, the procedure is the same but with a 128-bit and a 256-bit subkey  $k_1 \parallel k_2 \in \{0, 1\}^{384}$  as derived in (6):

$$\begin{aligned} k_1 &= \text{left}_{n/2}(E_k(\nu\|0)) \parallel \text{left}_{n/2}(E_k(\nu\|1)), \\ k_2 &= \text{left}_{n/2}(E_k(\nu\|2)) \parallel \dots \parallel \text{left}_{n/2}(E_k(\nu\|5)). \end{aligned} \quad (36)$$

We refer to [52, Figure 3] for a clean picture of this algorithm.

The isolated character of the key derivation function in GCM-SIV is also well-reflected in the security bound of GCM-SIV. The security bound of GCM-SIV as outlined by Mennink and Neves [52, Theorem 3], which is in turn taken from Iwata and Seurin [41], consists of two separated terms:

- A term upper bounding the PRF security of the key derivation function, namely

$$\mathbf{Adv}_{\text{Trunc}_{n/2}}^{\text{PRF}}(c \cdot q) + \mathbf{Adv}_E^{\text{PRP}}(c \cdot q, t), \quad (37)$$

where  $q$  is the number of invocations of the key derivation function, and where  $c = 4$  for the 128-bit keyed variant and  $c = 6$  for the 256-bit keyed variant;

- A term that describes the security of GCM-SIV as an authenticated encryption scheme once  $k_1$  and  $k_2$  are uniformly random. This term is irrelevant for current discussion.

Now, if we would replace the truncation in the key derivation of GCM (equations (35) and (36)) by STH, we would get

$$\begin{aligned} k_1 &= \text{left}_{n/2}(E_k(\nu\|0)) \parallel \text{left}_{n/2}(E_k(\nu\|1)), \\ k_2 &= \text{right}_{n/2}(E_k(\nu\|0) \oplus E_k(\nu\|1)) \parallel \text{left}_{n/2}(E_k(\nu\|2)) \end{aligned}$$

for the 128-bit keyed variant, and

$$\begin{aligned} k_1 &= \text{left}_{n/2}(E_k(\nu\|0)) \parallel \text{left}_{n/2}(E_k(\nu\|1)), \\ k_2 &= \text{right}_{n/2}(E_k(\nu\|0) \oplus E_k(\nu\|1)) \parallel \\ &\quad \text{left}_{n/2}(E_k(\nu\|2)) \parallel \text{left}_{n/2}(E_k(\nu\|3)) \parallel \text{right}_{n/2}(E_k(\nu\|2) \oplus E_k(\nu\|3)) \end{aligned}$$

for the 256-bit keyed variant. When we use **STH**, we see that for the derivation of a 256-bit subkey the underlying block cipher  $E$  is called three times instead of four times, and for the derivation of a 384-bit subkey it is called four times instead of six times. As for security, the original bound of Iwata and Seurin [41] (see also [52, Theorem 3]) carries over with (37) replaced by

$$\mathbf{Adv}_{\text{STH}_{n/2}}^{\text{prf}}(2 \cdot q) + \mathbf{Adv}_E^{\text{prp}}(c \cdot q, t),$$

where  $c = 3$  for the 128-bit keyed variant and  $c = 4$  for the 256-bit keyed variant. As the PRF security of  $\text{STH}_{n/2}$  (Theorem 1) is similar to the PRF security of truncation (Lemma 2), there is no significant loss in security. In particular, when we allow for a maximum advantage of  $2^{-32}$ , we are able to derive approximately  $2^{64}$  different keys for both instantiations, even when  $t \gg 2^{64}$ . Hence the security does not reduce when using the more efficient **STH** version.

We conclude by noting that this only discusses the key derivation *in isolation*. As bijectivity in the key derivation is not an issue in the bigger picture of GCM-SIV, one can get away by simply taking untruncated block ciphers [17]. However, there are many more applications where replacing block cipher evaluations by **STH** truly lead to security gains, most notably Wegman-Carter and counter mode encryption, as also outlined in Section 1.

**ACKNOWLEDGMENTS.** The authors would like to thank the anonymous reviewers of CRYPTO 2020 for their valuable feedback. Aldo Gensing is supported by the Netherlands Organisation for Scientific Research (NWO) under TOP grant TOP1.18.002 SCALAR.

## References

1. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404 (2013)
2. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 123–153. Springer (2016)
3. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A Concrete Security Treatment of Symmetric Encryption. In: FOCS '97. pp. 394–403. IEEE Computer Society (1997)
4. Bellare, M., Guérin, R., Rogaway, P.: XOR MACs: New Methods for Message Authentication Using Finite Pseudorandom Functions. In: Coppersmith, D. (ed.) CRYPTO '95. LNCS, vol. 963, pp. 15–28. Springer (1995)
5. Bellare, M., Impagliazzo, R.: A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP to PRF conversion. Cryptology ePrint Archive, Report 1999/024 (1999)
6. Bellare, M., Kilian, J., Rogaway, P.: The Security of Cipher Block Chaining. In: Desmedt, Y. (ed.) CRYPTO '94. LNCS, vol. 839, pp. 341–358. Springer (1994)
7. Bellare, M., Krovetz, T., Rogaway, P.: Luby-Rackoff Backwards: Increasing Security by Making Block Ciphers Non-invertible. In: Nyberg, K. (ed.) EUROCRYPT '98. LNCS, vol. 1403, pp. 266–280. Springer (1998)

8. Bellare, M., Rogaway, P.: The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In: Vaudenay [64], pp. 409–426
9. Bernstein, D.J.: SURF: Simple Unpredictable Random Function. <https://cr.yp.to/papers.html#surf> (1997)
10. Bernstein, D.J.: How to Stretch Random Functions: The Security of Protected Counter Sums. *J. Cryptology* 12(3), 185–192 (1999)
11. Bernstein, D.J.: Stronger Security Bounds for Wegman-Carter-Shoup Authenticators. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 164–180. Springer (2005)
12. Bhargavan, K., Leurent, G.: On the Practical (In-)Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 2016. pp. 456–467. ACM (2016)
13. Bhattacharya, S., Nandi, M.: A note on the chi-square method: A tool for proving cryptographic security. *Cryptography and Communications* 10(5), 935–957 (2018)
14. Bhattacharya, S., Nandi, M.: Revisiting Variable Output Length XOR Pseudorandom Function. *IACR Trans. Symmetric Cryptol.* 2018(1), 314–335 (2018)
15. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer (2007)
16. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçın, T.: PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer (2012)
17. Bose, P., Hoang, V.T., Tessaro, S.: Revisiting AES-GCM-SIV: Multi-user Security, Faster Key Derivation, and Better Bounds. In: Nielsen and Rijmen [53], pp. 468–499
18. Brassard, G.: On Computationally Secure Authentication Tags Requiring Short Secret Shared Keys. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) CRYPTO '82. pp. 79–86. Plenum Press, New York (1982)
19. Chang, D., Nandi, M.: A Short Proof of the PRP/PRF Switching Lemma. *Cryptology ePrint Archive*, Report 2008/078 (2008)
20. Cogliati, B., Seurin, Y.: EWCDM: An Efficient, Beyond-Birthday Secure, Nonce-Misuse Resistant MAC. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 121–149. Springer (2016)
21. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. *Information Security and Cryptography*, Springer (2002)
22. Dai, W., Hoang, V.T., Tessaro, S.: Information-Theoretic Indistinguishability via the Chi-Squared Method. In: Katz and Shacham [42], pp. 497–523
23. De Cannière, C., Dunkelman, O., Knezevic, M.: KATAN and KTANTAN - A Family of Small and Efficient Hardware-Oriented Block Ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer (2009)
24. Dworkin, M.: NIST SP 800-38A: Recommendation for Block Cipher Modes of Operation: Methods and Techniques (2001)
25. Freedman, D.: A Remark on the Difference between Sampling with and without Replacement. *Journal of the American Statistical Association* 72(359), 681–681 (1977)
26. Gilboa, S., Gueron, S.: The Advantage of Truncated Permutations. *CoRR* abs/1610.02518 (2016)

27. Gilboa, S., Gueron, S., Morris, B.: How Many Queries are Needed to Distinguish a Truncated Random Permutation from a Random Function? *J. Cryptology* 31(1), 162–171 (2018)
28. Gong, Z., Nikova, S., Law, Y.W.: KLEIN: A New Family of Lightweight Block Ciphers. In: Juels, A., Paar, C. (eds.) *RFIDSec 2011*. LNCS, vol. 7055, pp. 1–18. Springer (2011)
29. Gueron, S., Langley, A., Lindell, Y.: AES-GCM-SIV: Specification and Analysis. *Cryptology ePrint Archive*, Report 2017/168 (2017)
30. Gueron, S., Langley, A., Lindell, Y.: AES-GCM-SIV: Nonce Misuse-Resistant Authenticated Encryption. Internet-Draft draft-irtf-cfrg-gcmsiv-09, Internet Engineering Task Force (Nov 2018), work in Progress
31. Gueron, S., Langley, A., Lindell, Y.: AES-GCM-SIV: Nonce Misuse-Resistant Authenticated Encryption. Request for Comments (RFC) 8452 (April 2019), <http://tools.ietf.org/html/rfc8452>
32. Gueron, S., Lindell, Y.: GCM-SIV: Full Nonce Misuse-Resistant Authenticated Encryption at Under One Cycle per Byte. In: Ray, I., Li, N., Kruegel, C. (eds.) *ACM CCS 2015*. pp. 109–119. ACM (2015)
33. Gueron, S., Lindell, Y.: Better Bounds for Block Cipher Modes of Operation via Nonce-Based Key Derivation. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) *ACM CCS 2017*. pp. 1019–1036. ACM (2017)
34. Hall, C., Wagner, D.A., Kelsey, J., Schneier, B.: Building PRFs from PRPs. In: Krawczyk, H. (ed.) *CRYPTO '98*. LNCS, vol. 1462, pp. 370–389. Springer (1998)
35. Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., Chee, S.: HIGHT: A New Block Cipher Suitable for Low-Resource Device. In: Goubin, L., Matsui, M. (eds.) *CHES 2006*. LNCS, vol. 4249, pp. 46–59. Springer (2006)
36. Impagliazzo, R., Rudich, S.: Limits on the Provable Consequences of One-way Permutations. In: Goldwasser, S. (ed.) *CRYPTO '88*. LNCS, vol. 403, pp. 8–26. Springer (1988)
37. Iwata, T.: New Blockcipher Modes of Operation with Beyond the Birthday Bound Security. In: Robshaw, M.J.B. (ed.) *FSE 2006*. LNCS, vol. 4047, pp. 310–327. Springer (2006)
38. Iwata, T., Mennink, B., Vizár, D.: CENC is Optimally Secure. *Cryptology ePrint Archive*, Report 2016/1087 (2016)
39. Iwata, T., Minematsu, K.: Stronger Security Variants of GCM-SIV. *IACR Trans. Symmetric Cryptol.* 2016(1), 134–157 (2016)
40. Iwata, T., Ohashi, K., Minematsu, K.: Breaking and Repairing GCM Security Proofs. In: Safavi-Naini, R., Canetti, R. (eds.) *CRYPTO 2012*. LNCS, vol. 7417, pp. 31–49. Springer (2012)
41. Iwata, T., Seurin, Y.: Reconsidering the Security Bound of AES-GCM-SIV. *IACR Trans. Symmetric Cryptol.* 2017(4), 240–267 (2017)
42. Katz, J., Shacham, H. (eds.): *CRYPTO 2017, Part III*, LNCS, vol. 10403. Springer (2017)
43. Lim, C.H., Korkishko, T.: mCrypton - A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In: Song, J., Kwon, T., Yung, M. (eds.) *WISA 2005*. LNCS, vol. 3786, pp. 243–258. Springer (2005)
44. Lindell, Y., Langley, A., Gueron, S.: AES-GCM-SIV: Nonce Misuse-Resistant Authenticated Encryption. Internet-Draft draft-irtf-cfrg-gcmsiv-05, Internet Engineering Task Force (May 2017), work in Progress
45. Luby, M., Rackoff, C.: How to Construct Pseudorandom Permutations from Pseudorandom Functions. *SIAM J. Comput.* 17(2), 373–386 (1988)

46. Lucks, S.: The Sum of PRPs Is a Secure PRF. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 470–484. Springer (2000)
47. Luykx, A., Preneel, B.: Optimal Forgeries Against Polynomial-Based MACs and GCM. In: Nielsen and Rijmen [53], pp. 445–467
48. McGrew, D.: Impossible plaintext cryptanalysis and probable-plaintext collision attacks of 64-bit block cipher modes. Cryptology ePrint Archive, Report 2012/623 (2012)
49. McGrew, D.A., Viega, J.: The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 343–355. Springer (2004)
50. Mennink, B.: Linking Stam’s Bounds with Generalized Truncation. In: Matsui, M. (ed.) CT-RSA 2019. LNCS, vol. 11405, pp. 313–329. Springer (2019)
51. Mennink, B., Neves, S.: Encrypted Davies-Meyer and Its Dual: Towards Optimal Security Using Mirror Theory. In: Katz and Shacham [42], pp. 556–583
52. Mennink, B., Neves, S.: Optimal PRFs from Blockcipher Designs. IACR Trans. Symmetric Cryptol. 2017(3), 228–252 (2017)
53. Nielsen, J.B., Rijmen, V. (eds.): EUROCRYPT 2018, Part I, LNCS, vol. 10820. Springer (2018)
54. NIST: Block Cipher Techniques – Modes Development. <https://csrc.nist.gov/projects/block-cipher-techniques/bcm/modes-develoment> (2020)
55. NSA IA: Key Recovery Attacks on AES-GCM-SIV. <https://mailarchive.ietf.org/arch/msg/cfrg/k2mpWgod4mbd0xsvN6EtXHb0BAg> (2017)
56. Patarin, J.: A Proof of Security in  $O(2^n)$  for the Xor of Two Random Permutations. In: Safavi-Naini, R. (ed.) ICITS 2008. LNCS, vol. 5155, pp. 232–248. Springer (2008)
57. Patarin, J.: Introduction to Mirror Theory: Analysis of Systems of Linear Equalities and Linear Non Equalities for Cryptography. Cryptology ePrint Archive, Report 2010/287 (2010)
58. Patarin, J.: Security in  $O(2^n)$  for the Xor of Two Random Permutations – Proof with the standard  $H$  technique–. Cryptology ePrint Archive, Report 2013/368 (2013)
59. Rogaway, P., Shrimpton, T.: A Provable-Security Treatment of the Key-Wrap Problem. In: Vaudenay [64], pp. 373–390
60. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: Piccolo: An Ultra-Lightweight Blockcipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 342–357. Springer (2011)
61. Shoup, V.: On Fast and Provably Secure Message Authentication Based on Universal Hashing. In: Koblitz, N. (ed.) CRYPTO ’96. LNCS, vol. 1109, pp. 313–328. Springer (1996)
62. Stam, A.J.: Distance between sampling with and without replacement. Statistica Neerlandica 32(2), 81–91 (1978)
63. Stam, A.J.: A note on sampling with and without replacement. Statistica Neerlandica 40(1), 35–38 (1986)
64. Vaudenay, S. (ed.): EUROCRYPT 2006, LNCS, vol. 4004. Springer (2006)
65. Wegman, M.N., Carter, L.: New Hash Functions and Their Use in Authentication and Set Equality. J. Comput. Syst. Sci. 22(3), 265–279 (1981)
66. Wu, W., Zhang, L.: LBlock: A Lightweight Block Cipher. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 327–344 (2011)