On the security of keyed hashing based on an unkeyed block function

Jonathan Fuchs[1], Yann Rotella[2] and Joan Daemen[1]

[1] Radboud University, Nijmegen, The Netherlands
[2] Université Paris-Saclay, UVSQ, CNRS, Versailles, France
jonathan.fuchs@ru.nl,yann.rotella@uvsq.fr,joan.daemen@ru.nl

Abstract. In this paper we study the security of two constructions for variable-length universal hash functions by means of their universality. Both constructions make use of a fixed-length unkeyed function that we call a block function. One construction is serial and is an idealization of the compression phase of Pelican-MAC. The other construction is parallel and is an idealization of the compression phase of Farfalle. Both are instances of a class of functions we call *semi-group accumulators*.
We prove that the universality of these constructions is fully determined by the differential probability of block function differentials and, if not a permutation, the relative frequency of block function outputs. We show that both block function parallelization and serialization have equal security (against forgery) in the Wegman-Carter(-Shoup) construction. However, for the block functions we target, parallelization can provide significantly better security than serialization in the Protected Hash (PH) construction. Moreover, below a certain data limit, PH provides better security than WC(S) for the block function parallelization, despite the fact that it does not require a nonce. We show evidence of this effect by taking XOODOO[3] as the block function .

Keywords: Keyed hashing · Universality · Differential probability · Parallel · Serial · Permutation

1 Introduction

Message authentication codes take as input a variable-lenght message, optionally a fixed-length nonce, and return a tag. They strive for providing protection against forgery, where forgery is defined as follows. The adversary gets query access to a generation oracle that returns a tag for given message (and nonce) and a verification oracle, that returns for given message (and nonce) and tag whether the tag is valid. Forgery consists of a successful verification query where the message was not used in the tag generating query. Many message authentication code functions are composed of two phases, namely, a keyed compression phase and a scrambling phase. We identify two mainstream approaches to combine these steps.

The first is a nonce-based approach called Wegman-Carter(-Shoup) authenticators [21] [18]. Here the tag is the result of the addition of the digest to the output of an fixed-input-length cryptographic function applied to a nonce. In the case of Wegman-Carter (WC) this function is assumed to satisfy some level of pseudo-random function (PRF) security and in the case of Wegman-Carter-Shoup (WCS) it should satisfy some level of pseudo-random permutation (PRP) security.

The second approach does not have an established name and we will refer to it as *protected hash* (PH). Here the tag is the result of applying a fixed-input-length cryptographic function that satisfies some level of PRF or PRP security to the output of the compression phase and it does not require a nonce. A protected hash function realizes protection against forgery by achieving some level of variable-length-input PRF security. Both approaches are depicted in Figure 1.

The object of study in this paper is the keyed compression phase and we will refer to them as *keyed hash functions*.

In the forgery security level the *universality* of the keyed hash function plays an important role. Here we distinguish two metrics that were introduced by Stinson in [20]. The ε-universality is the maximum of the expected collision probability taken over all input pairs to the keyed hash function and the $\varepsilon\Delta$-universality is the maximum of the expected probability that the outputs have a particular output difference, taken over all input pairs and all output differences. We repeat the original definitions of these metrics in Section 2.1.

As we will discuss in subsection 2.4, in WCS the success probability of forgery is upper bound by q times the $\varepsilon\Delta$-universality of the keyed hash function plus the PRF/PRP distinguishing-bound of the scrambling function, with q the number of verification queries. As similarly discussed in subsection 2.4, in PH the success probability of forgery is upper bound by $\binom{q}{2}$ times the ε-universality of the keyed hash function plus the PRF/PRP distinguishing-bound of the scrambling function. Here q is the number of generation queries.

In practice we see that MAC functions are built using either strong cryptographic primitives on the one hand or make use of a keyed hash built using mathematically simple functions such as multiplication in a finite field. Notable examples of the former approach are block-cipher-based modes such as CBC-MAC [2] and PMAC [6] and hash-function-based constructions such as HMAC [1]. The best known examples of the latter approach are GMAC, used in the authentication encryption mode in GCM [16] and poly1305 [4]. The former approach has the disadvantage that the block cipher or hash function are over-engineered for their task as keyed hashing. The latter approach addresses this concern but field multiplication is still a reasonably heavy operation.

In this paper we will study lightweight keyed hashes that are built from an underlying unkeyed fixed-length function that we call a *block function*. We claim that even with block functions with low computational cost we can achieve high security strength against forgery. We study two constructions, one serial and one parallel. In our approach we do not assume the block function to be ideal, but express the security of our construction in terms of differential (and image) probabilities of the block function. In both cases, the variable-length message is (after padding) split into equal-length blocks. Then the key, that must be at least as long as the (padded) message, is added block-by-block to the message with a group operation. The resulting sequence of blocks are then processed by means of the block function. In the serial construction a block is added to the application of the block function to the previous block, analogous to CBC-MAC. In the parallel construction the block function is applied to all blocks and their outputs are added, analogous to PMAC. We study the universalities of both constructions and the properties of the block function that determine them. The idea is then to choose a block function that achieves good universality in these constructions with a much lower computation cost than the classical approaches. This approach can be seen in Pelican-MAC [13], Kravatte [5] and Xoofff [11].

1.1 Our Contributions

In this paper we focus on the upper bounds of the ε-universality and $\varepsilon\Delta$-universality of the constructions we present.

Our main contributions are:

- In section 3 we present a generalization of our two constructions called *semigroup accumulator*. We show that the collision probability (or more general the probability

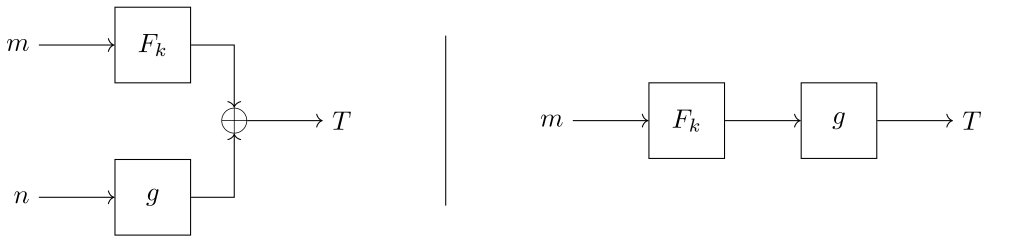

Figure 1: Depicted on the left is the CW(S) construction with g either a PRP or a PRF. Depicted on the right is the PH construction with g either a PRP or a PRF.

that they propagate to outputs with a given difference) of a pair of queries, depends on their difference only and is independent of the absolute values.

- In section 4 we present and study our serial construction. For the case of the block function being a permutation, we show that both ε and $\varepsilon\Delta$-universality are $\max_{a,b} \mathsf{DP}(a,b)$, with $\mathsf{DP}(a,b)$ the probability of differential (a,b) over the block function. For the non-permutation case we show that in general no such expression can be proven and we argue that it is better to restrict the serial construction to permutations.

- In section 5 we present and study our parallel construction. For the case of the block function being a permutation, we show that the $\varepsilon\Delta$-universality is $\max_{a,b} \mathsf{DP}(a,b)$, the same as for our serial construction. However, the ε-universality is given by $\max_{a\neq 0} \sum_t \mathsf{DP}^2(a,t)$. When the block function is an iterated permutation, ε-universality is usually smaller than $\varepsilon\Delta$-universality. We also determined both ε- and $\varepsilon\Delta$-universality for the case of non-invertible block functions. These are more complex expressions involving DP values but also *image probability* values of the block function.

- In section 6 we show the difference between $\max_{a,b} \mathsf{DP}(a,b)$ and $\max_{a\neq 0} \sum_t \mathsf{DP}^2(a,t)$ when the block function is taken to be an iterated permutation, namely, Xoodoo[3].

- In section 7 we conclude the paper by discussing the combined implications of Section 2.4 and Section 5. The parallel construction offers better or equal security to the serial construction in all cases. However, comparing forgery resistance of WCS and PH, both instantiated with our parallel construction, is interesting. WCS only has a linear increase in success probability while PH has a quadratic one. But in WCS the constant factor is the $\varepsilon\Delta$-universality while for PH it is the ε-universality, that is typically much smaller. The consequence is that there is an attack complexity, expressed in the number of queries, below which PH is the more secure of the two and above which WCS is the more secure, despite the fact that WCS requires a nonce and PH does not. This attack complexity is very close the ε-universality divided by the $\varepsilon\Delta$-universality.

2 Keyed hash functions

In this section we recall the notions of ε-universality and $\varepsilon\Delta$-universality from [19]. Then, we define keyed hash functions and recall the bkh model from [14].

2.1 Differential universality

We quote almost verbatim the following three definitions from [20]:

Definition 1 ($(N; n, m)$-hash family)**.** An $(N; n, m)$-*hash family* is a set of N functions F such that

$$f : X \to Y$$

for each $f \in F$, where $\#X = n$ and $\#Y = m$.

Definition 2 (ε-universal)**.** An $(N; n, m)$-hash family is ζ-universal provided that for any two distinct elements $x_1, x_2 \in X$, there exist at most εN functions $f \in F$ such that $f(x_1) = f(x_2)$.

Definition 3 ($\varepsilon\Delta$-universal)**.** Suppose that the functions in an $(N; n, m)$-hash family, F, have range $Y = G$, where G is an additive abelian group (of order m). F is called $\varepsilon\Delta$-universal provided that for any two distinct elements $x_1, x_2 \in X$ and for any element $y \in G$, there exists at most εN functions $f \in F$ such that $f(x_1) - f(x_2) = y$.

From now on, we denote the upper bound defined by ε-universality as ε and the upper bound defined by $\varepsilon\Delta$-universality as ζ.

2.2 Keyed hash functions

We introduce the following notation for keyed hash functions.

$$F : \mathcal{K} \times \mathcal{M} \to \mathcal{A}\,,$$

with:

- A key $k \in \mathcal{K}$ and a message $m \in \mathcal{M}$ as input and a digest $h \in \mathcal{A}$ as output.
- A finite key space $\mathcal{K}$.
- A finite message space $\mathcal{M}$.
- A space of digest values $\mathcal{A}$ that forms a finite group $(\mathcal{A}, +)$.

When the key is fixed to a value k we denote the function as F_k. We can match the terminology used in Definition 1 to our own as following:

- $F_k : \mathcal{M} \to \mathcal{A}$ is the hash function denoted as $f : X \to Y$.
- $\#\mathcal{K}$ is the number of hash functions N in the set $\mathcal{F}$.
- $\#\mathcal{M}$ and $\#\mathcal{A}$ are equal to n and m respectively.

Therefore, the keyed hash function F defines a $(\#\mathcal{K}, \#\mathcal{M}, \#\mathcal{A})$-hash family.

2.3 Collision probability

We define the probability of generating a collision and relate it to the notions of universality.

Definition 4 (Collision probability)**.** Let $\{(m, \Delta), (m^*, \Delta^*)\} \subset \mathcal{M} \times \mathcal{A}$ be a pair of queries. The collision probability of p under F, denoted as $\mathsf{CP}_F(\{(m, \Delta), (m^*, \Delta^*)\}$, is defined as

$$\mathsf{CP}_F(\{(m, \Delta), (m^*, \Delta^*)\}) = \Pr[k \overset{\$}{\leftarrow} \mathcal{K} : F_k(m) + \Delta = F_k(m^*) + \Delta^*] \,.$$

We define the collision probability in such a way that it covers both ε- and $\varepsilon\Delta$-universality. In the case of ε-universality, Δ and Δ^* are always 0.

We can then define universality in terms of maxima over the collision probability.

Corollary 1. *The $\varepsilon\Delta$-universality of a keyed hash F is the maximum collision probability given by*

$$\zeta = \max_{\{(m,\Delta),(m^*,\Delta^*)\}} \mathsf{CP}_F(\{(m, \Delta), (m^*, \Delta^*)\}) \,.$$

Corollary 2. *The ε-universality of a keyed hash F is the maximum collision probability given by*

$$\varepsilon = \max_{\{(m,0),(m^*,0)\}} \mathsf{CP}_F(\{(m, 0), (m^*, 0)\}) \,.$$

2.4 Security against forgery

In this section we will remind the reader of the intuition behind the security against forgery for both the WCS and PH constructions. We assume that the keyed hash functions used in these analyses are susceptible to collision length extension. The setup is the following. The adversary gets query access to a generation oracle that returns a tag T for given message m and a verification oracle, that returns for given message m and tag T a binary value indicating whether the tag is valid (true) or not (false). In the case of WCS, both generation and verification queries include a nonce n and over generation queries the nonce must be unique. Note that an adversary that can make multiple generation queries with the same nonce can easily create forgeries.

Forgery consists of a verification query returning true on a query $(m, (n,)T)$, by an adversary that never sent a query $(m(, n)$ to the generation oracle.

For the following analyses we denote the scrambling phase as $g \colon \mathcal{A} \to \mathcal{A}'$ and assume that it is an ideal PRP/PRF. Furthermore, we assume that $\#\mathcal{A} \leq \#\mathcal{A}'$.

We start with the analysis on WCS. We now look at a forgery attempt with a single generation and a single verification query. From the generation query (m, n) the adversary receives T. Her goal is now to compute a different payload (m^*, n, T^*) that will be verified by the oracle. Her attack is successful if $F_k(m^*) + g(n) = T^*$, We then get the following derivation

$$\begin{aligned}
F_k(m^*) + g(n) &= T^*, \\
F_k(m^*) + g(n) &= T + \Delta', \\
F_k(m^*) + g(n) &= F_k(m) + g(n) + \Delta', \\
F_k(m^*) - F_k(m) &= \Delta' \,.
\end{aligned}$$

From Definition 3 it is clear to see that this forgery attempt has a success probability that is upper bound by the ζ of F. In case the adversary can make multiple verification queries, each attempt will have at most success probability ζ. Thanks to the nonce, this is even the case if the adversary can make multiple generation queries. Namely, as each generation query will have a different nonce, a verification query must address a certain generation query.

In the PH construction an attack with a single generation query and a single verification query goes as follows. Given (m, T) obtained from the generation query, the verification query leads to a successful forgery if the adversary can find a second message $m^* \neq m$ for which the generating oracle returns the same tag T. This happens only when $g(F_k(m)) = g(F_k(m^*))$. When g is a PRP, equality is achieved when $F_k(m) = F_k(m^*)$ and that is upper bound by ε of F. When g is a PRF, we need to add the distinguishing-bound of the scrambling phase, hence the probability of a successful forgery is $\varepsilon + \frac{1}{\#\mathcal{A}'}$. When the adversary can make multiple queries, she can send generating queries with different messages until a collision is found. If it is a collision in the output of the keyed hash function, for most keyed hash functions a collision can be generated. In case of keyed hash functions that accumulate all input into a fixed-length state this is the case. If m and m^* collide, so will $m\|x$ and $m^*\|x$. Once the attacker found a collision, she does a generation query with $m\|x$, get the tag and does a verification query with $m^*\|x$ and the tag. The success probability of generating a collision in the keyed hash is upper bound by $\binom{q}{2}\varepsilon$ with q the number of generation queries. We discuss the implications of this in Section 7.

3 Semigroup accumulators

We see that in the compression phase of Farfalle [5] first a key is added to the message and then the result is compressed by an unkeyed operation. We study its universality under the assumption that the key is at least as long as the message and is uniformly chosen. We denote a keyed hash function that operates by first adding the key with a semigroup operation and then compressing the result a *semigroup accumulator* and specify it formally in algorithm 1.

We denote an unkeyed function from a finite message space to a finite digest space by the term *compression function*. Naturally, the size of the digest space should be smaller than that of the message space.

Algorithm 1: Semigroup accumulator SA$[c]$

```
Parameters : A compression function $c\colon \mathcal{M} \to \mathcal{A}$
Inputs     : A message $m \in \mathcal{M}$ that forms a semigroup $(\mathcal{M}, +)$
             A key $k \in \mathcal{K}$ with $(\mathcal{K}, +)$ a subgroup of $(\mathcal{M}, +)$
Output     : A digest $h \in \mathcal{A}$
Processing :
$h \leftarrow c(k + m)$
return $h$
```

3.1 Offset-invariance

Semigroup accumulators have a symmetry property regarding their collision probability. This symmetric property comes from the offset of a query.

Definition 5 (offset of a query)**.** The offset of a query $(m, \Delta) \in \mathcal{M} \times \mathcal{A}$ by $(\mu, \delta) \in \mathcal{K} \times \mathcal{A}$, written $(m^*, \Delta^*) = (m, \Delta) + (\mu, \delta)$, is given by $m^* = \mu + m$ and $\Delta^* = \Delta + \delta$.

Definition 6 (offsetting of a query pair)**.** The offsetting of a query pair p by $(\mu, \delta) \in \mathcal{K} \times \mathcal{A}$, written $p^* = p + (\mu, \delta)$, is given by

$$p + (\mu, \delta) = \{(m^*, \Delta^*) \mid (m^*, \Delta^*) = (\mu + m, \delta + \Delta) \text{ and } (m, \Delta) \in p\}\,.$$

We can now prove that in semigroup accumulators the collision probability of any query pair is invariant to offsets.

Proposition 1 (Offset-invariance of $\mathsf{CP}_{\mathrm{SA}[c]}(p)$)**.** *The collision probability of any query pair p under a semigroup accumulator* $\mathrm{SA}[c]$ *is offset-invariant. For all pairs $p \subset \mathcal{M} \times \mathcal{A}$, and for all offsets $(\mu, \delta) \in \mathcal{K} \times \mathcal{A}$,*

$$\mathsf{CP}_{\mathrm{SA}[c]}(p + (\mu, \delta)) = \mathsf{CP}_{\mathrm{SA}[c]}(p) .$$

Proof. Since $\mathcal{M}$ forms a semigroup $(\mathcal{M}, +)$, $+$ is associative. This means that $\mathrm{SA}[c](k, (\mu + m)) = \mathrm{SA}[c]((k + \mu), m)$. Since $\mu \in \mathcal{K}$ and $\mathcal{K}$ is a group it holds that $\mathcal{K} + \mu = \mathcal{K}$. From Definition 4, $\mathsf{CP}_{\mathrm{SA}[c]}$ is defined over all keys, thus, the proposition holds. □

3.2 Block string spaces and block functions

In the constructions that we will present in the following sections, messages consist of strings of elements of an additive abelian group G that we call the *block space.* Typically, the block space is the additive group of a finite field $\mathbb{F}_{2^n}$ or $\mathbb{F}_p$ where n and p are chosen such that a security requirement is achieved.

We call the set of all strings of length 1 up to some limit length of block space elements a *block string space* and define an addition operation.

Definition 7 (Block string space)**.** Given a block space G, a corresponding block string space $\mathrm{BS}(G, \kappa)$ is the set of strings of elements of G with length 1 up to κ:

$$\mathrm{BS}(G, \kappa) = \bigcup_{i=1}^{\kappa} G^i .$$

We denote a block string by $\mathbf{m}$, its blocks by m_i where we start indexing from 1: $\mathbf{m} = m_1, m_2, \dots m_\ell$ and the number of blocks in $\mathbf{m}$ as $|\mathbf{m}|$.

Definition 8 (Addition in the block string space)**.** Let $\mathbf{m}, \mathbf{m}^* \in \mathrm{BS}(G, \kappa)$ be two elements of the block string space with $|\mathbf{m}| = \ell, |\mathbf{m}^*| = \ell^*$ and $\ell \leq \ell^*$. The addition of these two elements is given by:

$$\mathbf{m} + \mathbf{m}^* = m_1 + m_1^*, m_2 + m_2^*, \dots, m_\ell + m_\ell^* .$$

With this addition, a block string space $\mathrm{BS}(G, \kappa)$ forms an additive semigroup as the addition is associative and has a neutral element, namely 0^κ. The subset G^κ contains the neutral element, is closed under addition and it contains for each element in it also its inverse, hence it forms a subgroup.

It follows that a keyed hash function with $\mathcal{M} = \mathrm{BS}(G, \kappa)$ and $\mathcal{K} = G^\kappa$ that first adds the message and the key together and then applies an unkeyed compression function to the resulting block string, is a semigroup accumulator.

In our functions we build the unkeyed variable-input-length compression function from a fixed-input-length *block function.*

We introduce the definitions of reduced form query pairs and differences in block string space that will allow us to show that query pairs of equal query differences have equal collision probability.

Definition 9 (Reduced form of a query pair of different length)**.** Let $p = \{(\mathbf{m}, \Delta), (\mathbf{m}^*, \Delta^*)\} \subset \mathrm{BS}(G, \kappa) \times \mathcal{A}$ be a query pair where $|\mathbf{m}| = \ell, |\mathbf{m}^*| = \ell^*$ and $\ell < \ell^*$. Let $\mathbf{m}^0 = m_1^*, m_2^*, \dots, m_\ell^*, 0, \dots, 0 \in \mathcal{K}$. The reduced pair $\hat{p}$ is defined as

$$\hat{p} = p - (\mathbf{m}^0, \Delta^*) .$$

Definition 10 (Reduced forms of a query pair of equal length)**.** Let $p = \{(\mathbf{m}, \Delta), (\mathbf{m}^*, \Delta^*)\} \subset \mathrm{BS}(G, \kappa) \times \mathcal{A}$ be a query pair where $|\mathbf{m}| = \ell, |\mathbf{m}^*| = \ell^*$ and $\ell = \ell^*$. Let $\mathbf{m}_1^0 = m_1^*, m_2^*, \dots, m_\ell^*, 0, \dots, 0 \in \mathcal{K}$ and $\mathbf{m}_2^0 = m_1, m_2, \dots, m_\ell, 0, \dots, 0 \in \mathcal{K}$. The reduced forms of p form a pair of queries $\{\hat{p}, \hat{p}'\}$ and is defined as

$$\{\hat{p}, \hat{p}'\} = \{p - (\mathbf{m}_1^0, \Delta^*), p - (\mathbf{m}_2^0, \Delta)\} .$$

When studying the CP of F, it does not matter whether we choose $\hat{p}$ or $\hat{p}'$ as the reduced form, due to offset-invariance property of semigroup accumulators (Proposition 1).

In the reduced form of a query pair of, one of the queries is of the form $(0^{\ell^*}, 0)$.

Definition 11 (Query pair difference)**.** Let $\hat{p} = \{(\mathbf{a}, b), (0^{\ell^*}, 0)\} \subset \mathrm{BS}(G, \kappa) \times \mathcal{A}$ be a reduced form pair where $|\mathbf{a}| = \ell$ and $1 \leq \ell \leq \ell^*$. The difference $(\mathbf{a}, b) - (0^{\ell^*}, 0) = (\mathbf{a}, \lambda, b)$ with $\lambda \in \mathbb{Z}_{\geq 0}$ is the non-zero query of the reduced form, augmented with difference in length $\lambda = \ell^* - \ell$.

For messages of different length $\ell \neq \ell^*$, Definitions 10 and 11 are non-ambiguous. However, in the case of $\ell = \ell^*$, it allows for two possible differences, namely, $(\mathbf{m}, \Delta) - (\mathbf{m}^*, \Delta^*)$ and $(\mathbf{m}^*, \Delta^*) - (\mathbf{m}, \Delta)$.

We say that a difference between messages of equal length is *symmetric*, i.e., if $\lambda = 0$ and denote is as $(\mathbf{a}, b)$, otherwise we call it *asymmetric*.

We now prove using offset invariance that the collision probability of a query pair is fully determined by the query pair difference.

Proposition 2. *All query pairs $p \subset \mathcal{M} \times \mathcal{A}$ with equal query difference have equal collision probability.*

Proof. From Definitions 10 and 11, it follows that query pairs that map to the same reduced form share the same difference. We know from Proposition 1 that they have equal CP. □

We can then define the collision probability as a function of query difference:

Definition 12 (Collision probability of a query difference)**.** The collision probability for all query pairs p with difference $(\mathbf{a}, \lambda, b)$ under $\mathrm{SA}[c]$, denoted as $\mathsf{CP}'_{\mathrm{SA}[c]}(\mathbf{a}, \lambda, b)$, is given by

$$\mathsf{CP}'_{\mathrm{SA}[c]}(\mathbf{a}, \lambda, b) = \mathsf{CP}_{\mathrm{SA}[c]}(p)\,.$$

3.3 Differential properties of a block function

A block function is a fixed-input-length fixed-output-length unkeyed function. Examples are cryptographic permutations but it does not have to be restricted to those.

Definition 13 (Block function)**.** A block function $f\colon G \to \mathcal{A}$ is a function that maps elements of the block space G to the digest space $\mathcal{A}$.

The CP' of any difference can be expressed in terms of the following properties and we dedicate the next two sections to prove this:

Definition 14 (Differential probability)**.** Let $f\colon G \to \mathcal{A}$ be a block function and $a \in G$, $b \in \mathcal{A}$ be elements of their respective groups. The differential probability of a differential (a, b), denoted as $\mathsf{DP}(a, b)$, is:

$$\mathsf{DP}(a, b) = \frac{\#\{x \in G \mid f(x + a) - f(x) = b\}}{\#G}$$

We say that the input difference a propagates to the output difference b with probability $\mathsf{DP}(a, b)$.

Definition 15 (Image probability)**.** Let $f\colon G \to \mathcal{A}$ be a block function, the image probability of an output g of f, denoted as $\mathsf{IP}(g)$, is the number of inputs that f maps to g divided by the total number of possible outputs, namely,

$$\mathsf{IP}(g) = \frac{\#\{x \in G \mid f(x) = g\}}{\#G}\,.$$

For the case of f a permutation, $\forall g \in G : \mathsf{IP}(g) = 1/\#\mathcal{A}$.

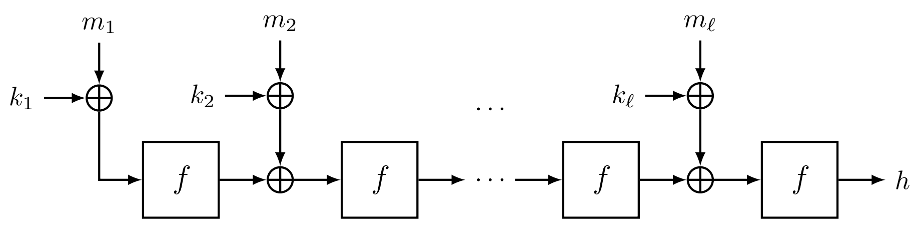

Figure 2: The serialization Serial $[f, \kappa]$

4 The serialization of a block function

In this section we define a construction for building keyed hash functions from a block function f in a serial way and express its ε-universality and $\varepsilon\Delta$-universality in terms of the differential and image probabilities of f. We divide the study into two cases, when the query difference is asymmetric and when it is symmetric. For the asymmetric case we are able to prove that the collision probability is fully determined by the image probability of the block function and the output difference and is independent of the length of the messages. For the symmetric case, we define the effective length of a difference and define the concept of universality restricted to symmetric differences with effective length ℓ.

4.1 The serialization construction

In the serialization construction the block space is the same as the digest space: $G = \mathcal{A}$ and hence the block function is a transformation of $\mathcal{A}$.

We specify the serialization of a block function in Algorithm 2 and depict it in Figure 2.

Algorithm 2: The serialization Serial $[f, \kappa]$

```
Parameters : A block function $f\colon \mathcal{A} \to \mathcal{A}$ and a key length $\kappa$
Inputs     : A key $\mathbf{k} \in \mathcal{A}^\kappa$ and a message $\mathbf{m} \in \mathrm{BS}(\mathcal{A}, \kappa)$
Output     : A digest $h \in \mathcal{A}$
Processing :
$\mathbf{x} \leftarrow \mathbf{m} + \mathbf{k}$
$h \leftarrow 0$
for $i \leftarrow 1$ to $|\mathbf{m}|$ do
    $h \leftarrow f(x_i + h)$
end
return $h$
```

We will assume κ is known from the context and we write Serial$[f]$. Clearly, the serialization operates by first adding the key to the message and then compresses the resulting string by serial chaining as in the CBC-MAC [15] mode.

Proposition 3. *The serialization of a block function is a semigroup accumulator.*

4.2 Collision probability of asymmetric differences

We now look at the collision probability of a pair of queries with messages of different length.

Proposition 4. *Let* $(\mathbf{a}, \lambda, b)$ *be an asymmetric difference. The collision probability of* $(\mathbf{a}, \lambda, b)$ *is given by:*

$$\mathsf{CP}'(\mathbf{a}, \lambda, b) = \sum_{x \in \mathcal{A}} \mathsf{IP}(x)\mathsf{IP}(x-b)\,.$$

Proof. Let $\hat{p} = \{(\mathbf{a}, \Delta), (0^{\ell^*}, 0)\}$ be a reduced form query pair with difference $(\mathbf{a}, \lambda, b)$ and $\lambda > 0$. The probability that $\mathrm{Serial}[f](\mathbf{a} + \mathbf{k}) = x$ is given by $\mathsf{IP}(x)$ since it is the result of $f(cv + k_\ell + a_\ell)$, where cv is the chaining value of the first $\ell - 1$ blocks. The presence of k_ℓ ensures that the input to f is uniform, hence we can apply Definition 15. Similarly, the probability that $\mathrm{Serial}[f](0^{\ell^*} + \mathbf{k}) = y$ is given by $\mathsf{IP}(y)$ since it is the result of $f(cv^* + k_{\ell^*})$, where cv^* is the accumulator of the first $\ell^* - 1$ blocks. $\mathsf{IP}(x)$ and $\mathsf{IP}(y)$ are independent of each other since they result from f being computed under the addition of k_ℓ and k_{ℓ^*} respectively and k_ℓ, k_{ℓ^*} are two secret key blocks chosen independently of each other. A collision occurs if $y = x - b$. Hence, we can partition the probability space into the conditional partitions of $\mathrm{Serial}[f](\mathbf{a} + \mathbf{k}) = x$ given the event $\mathrm{Serial}[f](0^{\ell^*} + \mathbf{k}) = x - b$. This occurs with probability $\mathsf{IP}(x)\mathsf{IP}(x - b)$. By applying the law of total probability we derive the expression in Proposition 4.

□

Corollary 3. *If the block function* f *is a permutation, the collision probability of asymmetric differences* $(\mathbf{a}, \lambda, b)$ *under the serialization of* f *is*

$$\mathsf{CP}_{Serial[f]}(p) = \frac{1}{\#\mathcal{A}}\,.$$

Proof. This is a direct corollary of Proposition 4.

$$\mathsf{CP}'(\mathbf{a}, \lambda, b) = \sum_{x \in \mathcal{A}} \mathsf{IP}(x) * \mathsf{IP}(x - b) = \sum_{a \in \mathcal{A}} \frac{1}{\#\mathcal{A}^2} = \frac{1}{\#\mathcal{A}}\,.$$

□

Proposition 5. *Asymmetric differences with* $b = 0$ *are the ones with the highest* CP *and it is given by the expression* $\sum_{x \in \mathcal{A}} \mathsf{IP}(x)^2$.

Proof. $\sum_{x \in \mathcal{A}} \mathsf{IP}(x) * \mathsf{IP}(x - b)$ can be seen as a scalar product of vectors with components indexed by x. The scalar product of two vectors is upper bound by the maximum of the norm of the two vectors where the norm of a vector is the square of its length:

$$\max_b \sum_{x \in \mathcal{A}} \mathsf{IP}(x) * \mathsf{IP}(x - b) \leq \max_b \left\{ \sum_x \mathsf{IP}(x)^2, \sum_x \mathsf{IP}(x - b)^2 \right\} = \sum_x \mathsf{IP}(x)^2\,.$$

Equality is realized by taking $b = 0$ and hence the maximum is $\sum_x \mathsf{IP}(x)^2$. □

4.3 Collision probability of symmetric differences

Lemma 1 (Extra block CP)**.** *Let* $(\mathbf{a}, b)$ *be a symmetric difference and let* $\mathbf{a}, a_{\ell+1}$ *be the concatenation of* $\mathbf{a}$ *with a single-block difference* $a_{\ell+1}$*. Then, the collision probability of* $(\mathbf{a}, a_{\ell+1}, b)$ *under Serial*$[f, \kappa]$ *is*

$$\mathsf{CP}'_{Serial[f]}(\mathbf{a} \| a_{\ell+1}, b) = \sum_{t \in \mathcal{A}} \mathsf{CP}'_{Serial[f]}(\mathbf{a}, t)\mathsf{DP}(a_{\ell+1} - t, b)\,.$$

Proof. We prove this using the law of total probability. We start by partitioning the probability space with the conditional event that $\mathbf{a}$ collides with t for all t under Serial $[f, \kappa]$. This happens with probability $\mathsf{CP}'_{\mathrm{Serial}[f]}(\mathbf{a}, t)$. From Algorithm 2, $\mathbf{a}, a_{\ell+1}$ collides with b, given that $\mathbf{a}$ collides with t, with probability $\mathsf{DP}(a_{\ell+1} - t, b)$. Thus, the probability of each partition is given by $\mathsf{CP}'_{\mathrm{Serial}[f]}(\mathbf{a}, t)\mathsf{DP}(a_{\ell+1} - t, b)$. By applying the law of total probability we get the expression in Lemma 1. $\square$

Theorem 1 (Collision probability of symmetric differences)**.** *The collision probability of a symmetric difference* $(\mathbf{a}, b)$ *is given by*

$$\mathsf{CP}'_{Serial[f]}(\mathbf{a}, b) = \sum_{(t_0=0,t_1,t_2,\ldots,t_{\ell-1},t_\ell=-b)\in T} \left(\prod_{i=1}^{\ell} \mathsf{DP}(a_i + t_{i-1}, t_i) \right) ,$$

where ℓ *is the number of blocks in a and* $T \subset \mathcal{A}^{\ell+1}$ *is the subset of* $\ell + 1$ *tuples whose first and last elements are fixed to* 0 *and* $-b$ *respectively. For a single-block input difference* (a_1, b) *we get*

$$\mathsf{CP}'_{Serial[f]}(a_1, b) = \sum_{(t_0=0,t_1=-b)} \mathsf{DP}(a_1 + t_0, t_1) = \mathsf{DP}(a_1, -b) .$$

Proof. We start by proving the base case $\ell = 1$: From Algorithm 2, the collision probability of a query pair with difference (a_0, b) is equal to $\mathsf{DP}(a_0, -b)$. By applying Theorem 1 to the input difference (a_0, b) we get equivalence. Thus, we have proven that the theorem holds for the base case. By applying Lemma 1 as the induction step we prove it for the generic case $\ell \geq 1$. $\square$

4.4 Effective length of message differences with f a permutation

In this section, we define the *effective length* of differences. Doing so will allow us to define a universality restricted to differences of a given effective length and reason about the hierarchy of the universality restricted to different effective fixed lengths.

We start by motivating the need for an effective length:

Corollary 4. *Let* $n \in \mathbb{Z}_{\geq 0}$ *and let* $0 \in G$ *be the neutral element of the group* G*. Let* $(0^n, \mathbf{a}, b)$ *be a symmetric difference, then the following holds*

$$\mathsf{CP}'_{Serial[f]}(0^n, \mathbf{a}, b) = \mathsf{CP}'_{Serial[f]}(\mathbf{a}, b) .$$

Proof. From Theorem 1 we the following expression

$$\begin{aligned}\mathsf{CP}'_{\mathrm{Serial}[f]}(\mathbf{a}, b) &= \sum_{(t_0=0,t_1,t_2,\ldots,t_{\ell-1},t_\ell=-b)\in T} \left(\prod_{i=1}^{\ell} \mathsf{DP}(a_i + t_{i-1}, t_i) \right) \\ &= \sum_{(t_0=0,t_1,t_2,\ldots,t_{\ell-1},t_\ell=-b)\in T} \left(\prod_{i=1}^{n} \mathsf{DP}(a_i + t_{i-1}, t_i) \prod_{j=n+1}^{\ell} \mathsf{DP}(a_j + t_{j-1}, t_j) \right) .\end{aligned}$$

Since $t_0 = 0$ and $a_i = 0$ for $1 \leq i \leq n$ we get

$$\begin{aligned}&= \sum_{(t_0=0,t_1,t_2,\ldots,t_{\ell-1},t_\ell=-b)\in T} \left(\prod_{i=1}^{n} \mathsf{DP}(0,0) \prod_{j=n+1}^{\ell} \mathsf{DP}(a_j + t_{j-1}, t_j) \right) \\ &= \sum_{(t_0=0,t_1,t_2,\ldots,t_{\ell-1},t_\ell=-b)\in T} \left(\prod_{j=n+1}^{\ell} \mathsf{DP}(a_j + t_{j-1}, t_j) \right) \\ &= \mathsf{CP}'_{\mathrm{Serial}[f]}(\mathbf{a}, b) .\end{aligned}$$

$\square$

Proposition 6. *Let $(\mathbf{a}, b), (\mathbf{a}^*, 0)$ be two symmetric differences where $\mathbf{a}^* = \mathbf{a}, b$ is the concatenation of $\mathbf{a}$ with the single-block difference b. If f is a permutation then*

$$\mathsf{CP}'_{Serial[f]}(\mathbf{a}, b) = \mathsf{CP}'_{Serial[f]}(\mathbf{a}^*, 0)\,.$$

Proof. From Algorithm 2, a collision in p for $(\mathbf{a}, b)$ occurs when the following equation is satisfied:

$$\mathrm{Serial}[f](\mathbf{k}, \mathbf{m}) + \Delta = \mathrm{Serial}[f](\mathbf{k}, \mathbf{m}^*) + \Delta^*\,.$$

A collision in p' for $(\mathbf{a}^*, 0)$ occurs when

$$f(\mathrm{Serial}[f](\mathbf{k}, \mathbf{m}) + \Delta + k_{\ell+1}) = f(\mathrm{Serial}[f](\mathbf{k}, \mathbf{m}^*) + \Delta^* + k_{\ell+1})\,.$$

Since f is a permutation, this happens if and only if

$$\mathrm{Serial}[f](\mathbf{k}, \mathbf{m}) + \Delta + k_{\ell+1} = \mathrm{Serial}[f](\mathbf{k}, \mathbf{m}^*) + \Delta^* + k_{\ell+1}$$

□

Corollary 5. *Let $n \in \mathbb{Z}_{\geq 0}$. Let $(\mathbf{a}, 0)$ be the query difference of a query pair with messages of length ℓ. If f is a permutation and $\ell + n \leq \kappa$ then the following holds*

$$\mathsf{CP}'_{Serial[f]}(\mathbf{a}, 0) = \mathsf{CP}'_{Serial[f]}(\mathbf{a}^*, 0)\,,$$

where $\mathbf{a}^* = \mathbf{a}, 0^n$ is the concatenation of $\mathbf{a}$ with $0^n \in G^n$, the neutral element of the group G^n.

Proof. By applying Proposition 6 n times to $(\mathbf{a}, 0)$ we show that this holds. □

This allows us to remove the trailing 0 difference blocks in the case that $b = 0$.

These results lead us to the following definition of effective length when f is a permutation.

Definition 16 (Effective length of symmetric differences with f a permutation)**.** Let f be a permutation. The effective length of symmetric differences of length $\ell \geq 1$ with $b \neq 0$ under Serial $[f, \kappa]$, denoted as $\mathcal{L}$, is given by

$$\mathcal{L} = \ell - \ell_0\,,$$

where ℓ_0 is the total number of leading 0 blocks of the message difference.

In the case of $b = 0$, we must first remove all trailing 0 blocks from the difference by applying apply Corollary 5 before we are able to compute the effective length.

4.5 Length-specific universality with f a permutation

In this subsection, we define universality per effective length for when f is a permutation and establish relations between them. When f is a non-permutation, the relations between universalities per effective length behave quite differently hence we have a dedicated analysis in the next section.

Definition 17. The $\varepsilon\Delta$-universality restricted to effective length ℓ, denoted as ζ_ℓ, is the maximum collision probability, under Serial $[f, \kappa]$ with f a permutation, of differences with effective length $\mathcal{L} = \ell$:

$$\zeta_\ell = \max_{(\mathbf{a}, b) \in G^\ell \times \mathcal{A}} \mathsf{CP}(\mathbf{a}, b)\,.$$

Definition 18. The ε-universality restricted to effective length ℓ, denoted as ε_ℓ, is the maximum collision probability, under Serial $[f, \kappa]$ with f a permutation, of differences with effective length $\mathcal{L} = \ell$:

$$\varepsilon_\ell = \max_{(\mathbf{a},0)\in G^\ell \times \mathcal{A}} \mathsf{CP}(\mathbf{a}, 0)\,.$$

Corollary 6. *The $\varepsilon\Delta$-universality restricted to effective length ℓ is greater or equal to the ε-universality restricted to effective length ℓ under Serial $[f, \kappa]$ with f a permutation*

$$\zeta_\ell \geq \varepsilon_\ell\,.$$

Proof. This is a direct corollary of Definitions 17 and 18. □

Proposition 7. *If f is a permutation, the $\varepsilon\Delta$-universality for effective length $\ell + 1$ is upper bound by the $\varepsilon\Delta$-universality for effective length ℓ:*

$$\zeta_{\ell+1} \leq \zeta_\ell\,.$$

Proof. From Definition 18 and Lemma 1 we know that the following holds:

$$\begin{aligned}
\zeta_{\ell+1} &= \max_{(\mathbf{a},a_{\ell+1},b)\in G^{\ell+1}\times\mathcal{A}} \left(\sum_{t\in\mathcal{A}} \mathsf{CP}'_{\mathrm{Serial}[f]}(\mathbf{a}, -t)\mathsf{DP}(a_{\ell+1} + t, -b) \right) \\
&\leq \max_{(\mathbf{a},a_{\ell+1},b)\in G^{\ell+1}\times\mathcal{A}} \left(\sum_{t\in\mathcal{A}} \zeta_\ell \mathsf{DP}(a_{\ell+1} + t, -b) \right) \\
&= \zeta_\ell \max_{(\mathbf{a},a_{\ell+1},b)\in G^{\ell+1}\times\mathcal{A}} \left(\sum_{t\in\mathcal{A}} \mathsf{DP}(a_{\ell+1} + t, -b) \right) \\
&= \zeta_\ell\,.
\end{aligned}$$

□

Through Proposition 7 we show that there exists an hierarchy of maximum collision probability based on the effective length of the difference when f is a permutation. We now show the relation between ζ_ℓ and $\varepsilon_{\ell+1}$ and prove a relation between ε_ℓ and $\varepsilon_{\ell+1}$.

Proposition 8. *If f is a permutation, ζ_ℓ is equivalent to $\varepsilon_{\ell+1}$ for $\ell \geq 1$.*

Proof. From Definition 18 and Lemma 1 we get the following derivation:

$$\begin{aligned}
\varepsilon_{\ell+1} &= \max_{\mathbf{a},a_{\ell+1}\in G^{\ell+1}} \mathsf{CP}'(\mathbf{a}, a_{\ell+1}, 0) \\
&= \max_{\mathbf{a},a_{\ell+1}\in G^{\ell+1}} \sum_{t\in G} \mathsf{CP}'(\mathbf{a}, t)\mathsf{DP}(a_{\ell+1} - t, 0)\,.
\end{aligned}$$

Since f is a permutation, $\mathsf{DP}(0, 0) = 1$ and $\mathsf{DP}(x, 0) = 0$ if $x \neq 0$, thus we get

$$\begin{aligned}
\max_{\mathbf{a},a_{\ell+1}\in G^{\ell+1}} \sum_{t\in G} \mathsf{CP}'(\mathbf{a}, t)\mathsf{DP}(a_{\ell+1} - t, 0) &= \max_{(\mathbf{a},a_{\ell+1})\in G^\ell\times\mathcal{A}} \mathsf{CP}'(\mathbf{a}, a_{\ell+1})\mathsf{DP}(0, 0) \\
&= \zeta_\ell\,.
\end{aligned}$$

□

Corollary 7. *If f is a permutation, the ε-universality for effective length $\ell + 1$ is upper bound by the ε-universality for effective length ℓ:*

$$\varepsilon_{\ell+1} \leq \varepsilon_\ell\,.$$

Proof. This is a direct corollary from Propositions 7 and 8. □

4.6 Effective length of message differences with f a non-permutation

Since Corollary 4 makes no assumption on f, it holds when it is a non-permutation. Thus, we are able to define the effective length in this scenario as following.

Definition 19 (Effective length of symmetric differences with f a non-permutation)**.** Let f be a non-permutation. The effective length of equal-length symmetric differences of length $\ell \geq 1$ under Serial $[f, \kappa]$, denoted as $\mathcal{L}$, is given by

$$\mathcal{L} = \ell - \ell_0 ,$$

where ℓ_0 is the total number of leading 0 blocks of the message difference.

4.7 Length-specific universality f a non-permutation

When f is a non-permutation, we are able to prove that the universality per effective length increases as the length increase.

Proposition 9. *If f is a non-permutation, the $\varepsilon\Delta$-universality for effective length $\ell + 1$ is greater than the $\varepsilon\Delta$-universality for effective length ℓ:*

$$\zeta_{\ell+1} > \zeta_\ell .$$

Proof. We can derive it from Definition 17 using Lemma 1:

$$\begin{aligned}
\zeta_{\ell+1} &= \max_{(\mathbf{a}, a_{\ell+1}, b) \in G^{\ell+1} \times \mathcal{A}} \mathsf{CP}'(\mathbf{a}, a_{\ell+1}, b) = \max_{(\mathbf{a}, a_{\ell+1}, b) \in G^{\ell+1} \times \mathcal{A}} \sum_{t \in G} \mathsf{CP}'(\mathbf{a}, t)\mathsf{DP}(a_{\ell+1} - t, b) \\
&= \max_{(\mathbf{a}, a_{\ell+1}, b) \in G^{\ell+1} \times \mathcal{A}} \left\{ \mathsf{CP}'(\mathbf{a}, a_{\ell+1})\mathsf{DP}(0, b) + \sum_{t \neq -a_{\ell+1}} \mathsf{CP}'(\mathbf{a}, t)\mathsf{DP}(a_{\ell+1} - t, b) \right\}
\end{aligned}$$

By taking $b = 0$ we prove the following lower bound on $\zeta_{\ell+1}$:

$$\begin{aligned}
&= \max_{\mathbf{a}, a_{\ell+1} \in G^{\ell+1}} \left\{ \mathsf{CP}'(\mathbf{a}, a_{\ell+1})\mathsf{DP}(0, 0) + \sum_{t \neq -a_{\ell+1}} \mathsf{CP}'(\mathbf{a}, t)\mathsf{DP}(a_{\ell+1} - t, 0) \right\} \\
&= \zeta_\ell + \max_{\mathbf{a}, a_{\ell+1} \in G^{\ell+1}} \left\{ \sum_{t \neq -a_{\ell+1}} \mathsf{CP}'(\mathbf{a}, t)\mathsf{DP}(a_{\ell+1} - t, 0) \right\}
\end{aligned}$$

Since $\sum_{t \neq -a_{\ell+1}} \mathsf{CP}'(\mathbf{a}, t)\mathsf{DP}(a_{\ell+1} - t, 0)$ is positive when f is a non-permutation, we have proven the relation in Proposition 10. □

Proposition 10. *If f is a non-permutation, the ε-universality for effective length $\ell + 1$ is greater than the ε-universality for effective length ℓ:*

$$\varepsilon_{\ell+1} > \varepsilon_\ell .$$

Proof. The same proof as Proposition 9 applies here. By choosing $a_{\ell+1} = 0, b = 0$ we get

$$\begin{aligned}
\varepsilon_{\ell+1} &= \max_{\mathbf{a} \in G^\ell} \left\{ \mathsf{CP}'(\mathbf{a}, 0)\mathsf{DP}(0, 0) + \sum_{t \neq 0} \mathsf{CP}'(\mathbf{a}, t)\mathsf{DP}(-t, 0) \right\} \\
&= \varepsilon_\ell + \max_{\mathbf{a} \in G^\ell} \left\{ \sum_{t \neq 0} \mathsf{CP}'(\mathbf{a}, t)\mathsf{DP}(-t, 0) \right\}
\end{aligned}$$

□

The results of these sections are summarized in Table 1.

Table 1: The universality of the serialization of a block function

	f Permutation	f Non-Permutation
ζ	$\zeta_1 = \max_{a,b \in G \times \mathcal{A}} \mathsf{DP}(a,b)$	$\max\{\zeta_\kappa, \sum_{x \in G} \mathsf{IP}(x)^2\}$
ε	$\varepsilon_2 = \zeta_1$	$\max\{\varepsilon_\kappa, \sum_{x \in G} \mathsf{IP}(x)^2\}$

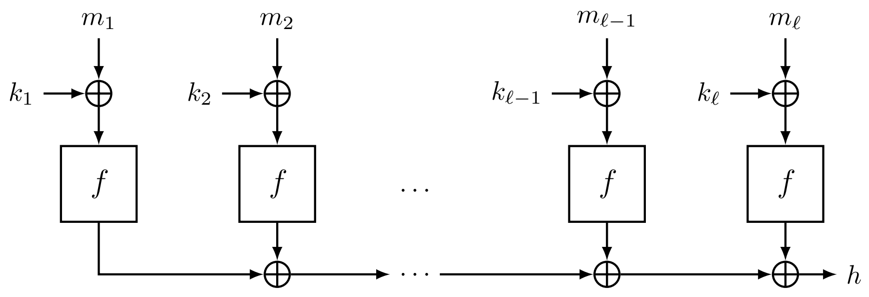

Figure 3: The parallelization Parallel $[f, \kappa]$

5 The parallelization of a block function

Similarly to the previous section, we define and study the universality of a construction for building keyed hash functions from a block function f. The construction in this section is a parallel one. We first specify the construction and then show that the CP' of a query difference can be expressed as a sum of stochastic variables whose distribution are given by the DP and the IP of the block function. We then prove that the CP' of a query difference depends only on the non-zero difference blocks and the length difference λ. Similar to the serialization, we define the concept of universality restricted to a fixed effective length and length difference.

5.1 The parallelization construction

Messages consist of strings of elements of an additive group G called the *block space* and the digest space $\mathcal{A}$ is also an abelian group.

We specify the parallelization of a block function in Algorithm 3 and illustrate it in Figure 3.

Algorithm 3: The parallelization Parallel $[f, \kappa]$

```
Parameters : A block function $f\colon G \to \mathcal{A}$ and a key length $\kappa$
Inputs     : A key $\mathbf{k} \in G^\kappa$ and a message $\mathbf{m} \in \mathrm{BS}(G, \kappa)$
Output     : A digest $h \in \mathcal{A}$
$\mathbf{x} \leftarrow \mathbf{m} + \mathbf{k}$
$h \leftarrow 0$
for $i \leftarrow 1$ to $|\mathbf{m}|$ do
    $h \leftarrow h + f(x_i)$
end
return $h$
```

The parallelization of a block function idealizes the compression part of Farfalle [5]. In Farfalle, the keys are not independent and the study of its collision security requires dedicated analysis. It is also similar to the protected counter sums [3] and PMAC [6]. However, unlike protected counter sums or PMAC, we do not model the underlying primitive as a pseudorandom permutation or function for deriving security but base ourselves on the differential propagation properties and the image probability of an (unkeyed) block function f.

Unlike in the serialization, the block group and the digest group can be different and may even have different sizes.

Proposition 11. *The parallelization of the a block function is a semigroup accumulator.*

5.2 Probability mass functions and stochastic variables

We remind the reader of the following probability theory definitions and lemma:

Definition 20 (Probability mass function over a finite group)**.** Let $g\colon \mathcal{A} \to [0,1]$ be a mapping. g is said to be a probability mass function (PMF) if the following two conditions hold:

$$\sum_{x\in\mathcal{A}} g(x) = 1,$$

From this definition, it is clear that $\mathsf{DP}(a,b)$ with a fixed is a PMF over the digest space. IP is also a PMF over the digest space.

Definition 21 (Convolution of PMFs)**.** The convolution of two PMFs g, g', denoted as $g'' = g * g'$ is a PMF g'' given by

$$g''(x) = \sum_t g(t)g'(x-t)\,.$$

The set of all PMFs over a finite group and the convolution form a commutative monoid whose neutral element is the Kronecker Delta, denoted as δ, which is the PMF $\delta(0) = 1$ and $\forall x \neq 0\, \delta(x) = 0$. This implies that the convolution of PMFs over a finite group is commutative, associative and has a neutral element.

For our purposes, we introduce a new definition of the convolutional power as following

Definition 22. Let g be a PMF. Let $n \geq 2 \in \mathbb{Z}_{\geq 0}$. g to the convolutional power n, denoted as g^n, is given by g convoluted with itself n times

$$g^n = g^{n-1} * g\,,$$

with $g^1 = g$.

Lemma 2 (Sum of two independent stochastic variables)**.** *Let the stochastic variable $X = X_1 + X_2$ be the sum of two independent stochastic variables X_1, X_2 with PMFs g and g' respectively. The PMF of X is given by the convolution of g and g':*

$$\Pr(X = x) = (g * g')(x) = \sum_t g(t)g'(x-t)\,.$$

This is a well known result, we provide a simple proof:

Proof. We prove this using the law of total probability. We partition the probability space with the conditional probability that $X = x$ given that $X_1 = t$. Since X_1 and X_2 are independent, this happens with probability $f(t)g(x-t)$. By applying the law of total probability we derive the expression in Lemma 2. □

Lemma 3. *Let $g'' = g * g'$ be a PMF, then*

$$\max_x g''(x) \leq \min\{\max_x g'(x), \max_x g(x)\}$$

Proof. Let us assume without loss of generality that $\max_x g(x) < \max_x g'(x)$. The following relation holds:

$$\begin{aligned}\max_x g'' &= \max_x \sum_t g(t)g'(x-t)\\ &\leq \max_x \sum_t (\max_y g(y))(g'(x-t))\\ &= \max_x \max_y g(y) \sum_t g'(x-t) = \max_y g(y)\,.\end{aligned}$$

□

5.3 Collision probability as a function of query difference

Unlike the serialization construction, the collision probability in the parallelization of a block function of any type of query difference is covered by a single equation that we introduce and prove correct in this section.

A query difference specifies a PMF as following:

Definition 23. Let DP_a denote the PMF $\mathsf{DP}(a, b)$ with a fixed. Let $(\mathbf{a}, \lambda, b)$ be a query difference. The PMF associated with $(\mathbf{a}, \lambda, b)$, denoted as $\mathsf{DP}_{\mathbf{a},\lambda}$ is given by

$$\mathsf{DP}_{\mathbf{a},\lambda} = \mathsf{DP}_{a_1} * \mathsf{DP}_{a_2} \ldots * \mathsf{DP}_{a_\ell} * \mathsf{IP}^{\lambda}\,.$$

Theorem 2. *The collision probability of a query difference* $(\mathbf{a}, \lambda, b)$ *is given by*

$$\mathsf{CP}'(\mathbf{a}, \lambda, b) = \mathsf{DP}_{\mathbf{a},\lambda}(-b)\,.$$

Proof. We define the following two stochastic variables. The image random variable $Y\colon \mathcal{K} \to \mathcal{A}$ is given by

$$P(Y = v) = \mathsf{IP}(v)\,.$$

The propagation stochastic variable $X_x\colon \mathcal{K} \to \mathcal{A}$ is given by

$$P(X_x = v) = \mathsf{DP}(x, v)\,.$$

From Algorithm 3, a collision event occurs in a query pair with difference (a, λ, b) if the following holds:

$$\sum_{i=1}^{\ell} (f(m_i + k_i) - f(m_i^* + k_i)) + \sum_{j=\ell+1}^{\ell+\lambda} f(m_j + k_j) = \Delta^* - \Delta\,. \tag{1}$$

For message blocks in indices in $1 \leq i \leq \ell$ $f(m_i + k_i) - f(m_i^* + k_i)$ can be seen as the output of the propagation random variable X_{a_i} with $a_i = m_i - m_i^*$. Similarly, for message blocks in indices $\ell + 1 \leq j \leq \ell + \lambda$, $f(m_j + k_j)$ can be seen as the output of the image random variable Y due to the key blocks being uniformly random and independent. Hence, the PMF of the difference at the output of the keyed hash function is the convolution of the PMFs of the corresponding stochastic variables. By evaluating said PMF at $-b = \Delta^* - \Delta$ we get the collision probability. □

5.4 Effective length of message differences

In this section, we define the effective length of a difference as a function of the number of non-zero differences.

Corollary 8. *The collision probability of a query pair with difference* $(\mathbf{a}, \lambda, b)$ *is fully determined by the number of non-zero differences and length difference* λ*.*

Proof. A 0 input difference block results in a stochastic variable with PMF $\delta(x)$. Since $\delta(x)$ is the neutral element of the convolution, the resulting PMF depends only on the non-zero difference blocks and the length difference. □

Definition 24 (Effective Length of a query difference)**.** The effective length of a query difference $(\mathbf{a}, \lambda, b)$ is defined as the number of non-zero differences in $\mathbf{a}$,

$$\mathcal{L}(\mathbf{a}) = \sum_{a_i} (1 - \delta(a_i)) \, .$$

5.5 Length-specific $\varepsilon\Delta$-universality

In this section we define $\varepsilon\Delta$-universality per effective length and length difference and establish relations between them, similar to what we did for the serial construction.

Definition 25. The $\varepsilon\Delta$-universality restricted to effective length ℓ and length difference λ, denoted as $\zeta_{\ell,\lambda}$, is the maximum collision probability, under Parallel $[f, \kappa]$, of differences $(\mathbf{a}, \lambda, b)$ with effective length $\mathcal{L} = \ell$:

$$\zeta_{\ell,\lambda} = \max_{(\mathbf{a},\lambda,b) \text{ with } \mathcal{L}(\mathbf{a})=\ell} \mathsf{CP}'(\mathbf{a}, \lambda, b)$$

Lemma 4. *If f is a block function, the $\varepsilon\Delta$-universality for effective length ℓ and length difference λ is upper bounding the $\varepsilon\Delta$-universality for effective length $\ell + 1$ and λ.*

$$\zeta_{\ell+1,\lambda} \leq \zeta_{\ell,\lambda} \, .$$

Proof. From Theorem 2 we know that adding an extra non-zero difference or an extra length difference results in one extra convolution with the corresponding PMF. Hence from Lemma 3 this holds. □

Lemma 5. *If f is a block function, the $\varepsilon\Delta$-universality for effective length ℓ and length difference λ is upper bounding the $\varepsilon\Delta$-universality for effective length ℓ and $\lambda + 1$.*

$$\zeta_{\ell,\lambda+1} \leq \zeta_{\ell,\lambda} \, .$$

Proof. Same as Lemma 4. □

Theorem 3. *The $\varepsilon\Delta$-universality of the parallelization of a block function f, denoted as ζ, is given by*

$$\zeta = \max \{\zeta_{1,0}, \zeta_{0,1}\} = \max \left\{ \max_{a,b} \mathsf{DP}(a, b), \max_{a} \mathsf{IP}(a) \right\} \, .$$

Proof. From Lemmas 4 and 5 we know that ζ is achieved either by $\zeta_{1,0}$ or by $\zeta_{0,1}$. By applying Theorem 2, we derive the expression of Theorem 3. □

5.6 Length-specific ε-universality

Similar to the previous section, we define ε-universality per effective length and establish relations between them.

Definition 26. The ε-universality restricted to effective length ℓ and length difference λ, denoted as $\varepsilon_{\ell,\lambda}$, is the maximum collision probability, under Parallel $[f, \kappa]$, of differences $(\mathbf{a}, \lambda, 0)$ with effective length $\mathcal{L} = \ell$:

$$\varepsilon_{\ell,\lambda} = \max_{(a,\lambda,0) \text{ with } \mathcal{L}=\ell} \mathsf{CP}'(\mathbf{a}, \lambda, 0)$$

Corollary 9. *The $\varepsilon\Delta$-universality restricted to effective length ℓ and length difference λ is greater or equal to the the ε-universality restricted to effective length ℓ and length difference λ,*

$$\varepsilon_{\ell,\lambda} \leq \zeta_{\ell,\lambda} \,.$$

Proof. This is a direct corollary of Definitions 25 and 26. □

Proposition 12 ($\varepsilon_{2,0}$ bound)**.** *The $\varepsilon\Delta$-universality restricted to symmetric differences of effective length 2 is equivalent to the ε-universality restricted to symmetric differences of effective length 2,*

$$\zeta_{2,0} = \zeta^0_{2,0} = \max_{a \neq 0} \sum_{t \in \mathcal{A}} \mathsf{DP}^2(a, t) \,.$$

Proof. We know from Theorem 2 that given a difference $((a_1, a_2), b)$ it holds that:

$$\mathsf{CP}'((a_1, a_2), b) = \sum_{t \in \mathcal{A}} \mathsf{DP}(a_2, -t - b)\mathsf{DP}(a_1, t) \,. \tag{2}$$

Similarly to the proof of Proposition 5, the right-hand part of (2) can be seen as a scalar product of vectors with components indexed by t. Hence, we have the following upper-bound.

$$\begin{aligned} \mathsf{CP}'((a_1, a_2), -b) &\leq \max \left\{ \sum_{t \in \mathcal{A}} \mathsf{DP}^2(a_1, t), \sum_{t \in \mathcal{A}} \mathsf{DP}^2(a_2, -t - b) \right\} \\ &= \max \left\{ \sum_{t \in \mathcal{A}} \mathsf{DP}^2(a_1, t), \sum_{t \in \mathcal{A}} \mathsf{DP}^2(a_2, t) \right\} \,. \end{aligned}$$

This is true for any a_1 or a_2, hence we have:

$$\zeta_{2,0} \leq \max_a \sum_t \mathsf{DP}^2(a, t) \,.$$

This bound is realized by taking a query pair with difference $((a, -a), 0)$:

$$\mathsf{CP}((a, -a), 0)) = \sum_{t \in \mathcal{A}} \mathsf{DP}(a, t)\mathsf{DP}(-a, -t) = \sum_{t \in \mathcal{A}} \mathsf{DP}^2(a, t) \,,$$

as for all a, t, $\mathsf{DP}(-a, -t) = \mathsf{DP}(a, t)$. □

From Corollary 9 we know that ζ upper bounds ε, however, in cases where this trivial bound does not provide enough security, it is possible to identify specific candidates for ε for which we may derive better, non-trivial bounds. We provide an example of such instance in the following proposition.

Table 2: The universality of the parallelization of a block function.

	f Permutation	f Non-Permutation
ζ	$\zeta_{1,0}$	$\max\{\zeta_{1,0}, \zeta_{0,1}\}$
ε	$\varepsilon_{2,0}$	$\max\{\varepsilon_{0,1}, \varepsilon_{1,0}, \zeta_{1,1}, \varepsilon_{2,0}, \zeta_{0,2}\}$

Proposition 13. *The ε-universality of the parallelization of a block function f, denoted as ε, is given by*

$$\varepsilon = \max\{\varepsilon_{0,1}, \varepsilon_{1,0}, \zeta_{1,1}, \varepsilon_{2,0}, \zeta_{0,2}\} \ ,$$

where

$$\begin{aligned}
\varepsilon_{0,1} &= \mathsf{IP}(0)\,,\\
\varepsilon_{1,0} &= \max_{a\in G\setminus\{0\}} \mathsf{DP}(a,0)\,,\\
\zeta_{1,1} &= \max_{a\in G\setminus\{0\}} \sum_{t\in\mathcal{A}} \mathsf{DP}(a,t)\mathsf{IP}(-t)\,,\\
\varepsilon_{2,0} &= \max_{a\in G\setminus\{0\}} \sum_{t\in\mathcal{A}} \mathsf{DP}^2(a,t)\,,\\
\zeta_{0,2} &= \max_{b} \sum_{x\in\mathcal{A}} \mathsf{IP}(x)\mathsf{IP}(-b-x)\,.
\end{aligned}$$

Proof. By combining Lemmas 4,5 and Corollary 9, it is clear that $\varepsilon_{\ell,\lambda}$ for $\ell+\lambda \geq 2$ is upper bound by one of these three candidates: $\zeta_{1,1}, \zeta_{2,0}, \zeta_{0,2}$. Furthermore, from Proposition 12 we know that $\zeta_{2,0} = \varepsilon_{2,0}$. For the case of $\ell + \lambda = 1$, they are upper bound by $\varepsilon_{0,1}, \varepsilon_{1,0}$ from Definition 26. By applying Theorem 2 to these bounds we derive the expressions in Proposition 13. □

The results of this section are summarized in Table 2.

6 Illustration of difference between ε and ζ with Xoodoo

In this section we illustrate the difference between ε and ζ for the parallellization of Xoodoo with 3 rounds.

Xoodoo is a family of 384-bit permutations with a classical iterated structure: it iteratively applies a round function to a state. It is parameterized by its number of rounds: we denote Xoodoo with n_r rounds Xoodoo$[n_\mathrm{r}]$. The round function consist of a linear layer that we call λ and a non-linear layer called χ that has algebraic degree two. The non-linear layer χ operates on 128 3-bit *columns* in parallel by means of what can be seen as a 3-bit S-box. The linear layer λ has the property that a single bit difference at its input propagates to 7 bits at its output, in 7 different columns. However, a two-bit difference at its input may propagate to a two-bit difference at its output, provided that the input bits have a specific relative position. This is called an *orbital.* A difference that consists of only orbitals is said to be *in the kernel.*

6.1 Differential trails and trail cores

Our analysis is based in differential trails and trail cores.

Definition 27 (Differential trail)**.** A n-round differential trail, denoted as Q, is a sequence of $n+1$ differences: the input difference to the first round, $n-1$ intermediate differences and the output difference of the n-th round, where the round differentials have non-zero DP, namely,

$$Q = (q_0, q_1, q_2, \ldots, q_{n-1}, q_n) \text{ with } \mathsf{DP}(q_{i-1}, q_i) > 0 \text{ for all } i\,,$$

and $\mathsf{DP}(q_{i-1}, q_i)$ the differential probability of differential (q_{i-1}, q_i) over the round function.

The differential probability of a trail, denoted $\mathsf{DP}(Q)$, is the probability that a random pair with input difference q_0 propagates via intermediate differences $q_1, q_2, \ldots$ to output difference q_n.

A trail core is a set of trails with common intermediate differences.

Definition 28 (Differential trail core [9])**.** A n-round differential trail core, denoted as T, is a set of differential trails over n rounds with a shared core of differences $(q_1, q_2, \ldots, q_{n-1})$ with non-zero DP.

In [11], the 3-round trail cores with the highest expected DP values were published. More exactly, it contains the trails with the lowest *restriction weight.*

Definition 29 (Restriction weight [17])**.** The restriction weight of a differential $\mathsf{DP}(a, b) > 0$ is defined as $w(a, b) = -\log_2(\mathsf{DP}(a, b))$.

The restriction weight of a round differential is determined by the non-linear layer: it is actually 2 times the number of non-zero columns (called active) in the difference,

Definition 30 (Restriction weight of a differential trail [17])**.** The restriction weight of a differential trail $Q = (q_0, q_1, \ldots, q_{n-1})$ is defined as $w(Q) = \sum_i w(q_{i-1}, q_i)$.

In the following we will omit the qualification “restriction” and simply speak of weight.

As the non-linear layer in the Xoodoo round function has algebraic degree 2 and its inverse too, we know from [8] that all trails in a trail core have the same weight. Therefore, it makes sense to speak about the weight of a trail core: $w(T)$. We use the term *weight profile* of a trail core as the sequence of weights of its round differentials.

If the round differentials of a trail are independent, we have $\mathsf{DP}(Q) = 2^{-w(Q)}$. Moreover, if for a trail $Q = (q_0, q_1, \ldots, q_{n-1})$ there exists no other trail with initial difference q_0 and final difference q_1, then the DP of the differential (q_0, q_{n-1}) over n rounds is that of the trail Q. If there are multiple trails between an initial difference and a final difference, we speak of *trail clustering.* In the absence of trail clustering and if the round differentials are independent, we have $\mathsf{DP}(q_0, q_{n-1}) = 2^{-w(T)}$ for any trail Q in the trail core T.

6.2 Block function is Xoodoo[3]

In [11] the 3-round trail cores of Xoodoo[3] with lowest weight are described. There are 4 essentially different ones and thanks to the shift-invariance of the Xoodoo round function each of them has 127 variants obtained by shifting the patterns horizontally. We will only talk about a single trail core in such an equivalence class and call that *canonical.* Furthermore, in [7] it is shown that for all 3-round trails with weight below 50 there is no clustering and the round differentials are independent. Hence it is very likely that the differentials with highest DP over Xoodoo[3] have $\mathsf{DP} = 2^{-36}$, implying $\zeta = 2^{-36}$.

For estimating ε we have to look at the weight profiles of trail cores. For the trail cores with weight 36 we see two different weight profiles: $(12, 12, 12)$ for the so-called *vortex* and $(4, 4, 28)$ for the three so-called *single-orbital fans.* We are now interested in $\sum_{q_3} \mathsf{DP}^2(q_0, q_3)$ given such a trail core. If we choose an input difference q_0, the number of trails in a given trail core is 2^{w_3}, where w_i is the weight of the i-th round differential. Using the knowledge that no clustering takes place for these trails and that round differentials are independent, we have $\sum_{q_3} \mathsf{DP}^2(q_0, q_3) = 2^{-w(T)}2^{w_3} = 2^{-(2w_1+2w_2+w_3)}$, with the sum taken over all output differences compatible with the trail core.

This sum yields is $2^{-(24+24+12)} = 2^{-60}$ for the trail cores with weight profile $(12, 12, 12)$ and $2^{-(8+8+28)} = 2^{-44}$ for those with weight profile $(4, 4, 28)$. We checked for all 3-round trail cores with weight up to 50 and the lowest sum is given by the weight profile $(4, 4, 28)$.

But to compute ε we have to sum over all output differences and not only those that are compatible with a given trail core. In other words, there are other trail cores compatible with q_0 that lead to different output differences. Let us take a look at trails that follow the trail core up to the second round: they have intermediate difference after one round of q_1 but after two rounds $q'_2 \neq q_2$. The difference at the input of χ of the 2nd round are two individual bits in two separate active S-boxes. In the single-orbital fan trail core, these propagate through χ to the same two bit positions. These two bits do not form an orbital in the second round and they each propagate through λ to 7 different columns. Each active column in the difference at the input to χ contributes 2 to the weight of a round differential, so that is why $w_3 = 28$.

The trails with q'_2 compatible with q_1 with next highest DP are the ones that have 3 active bits at the output of χ in the 2nd round rather than 2. This additional active bit will typically spread to 7 different S-boxes in the 3rd round and add 7 to the weight, so would give a weight profile $(4, 4, 42)$. In [10] there are all the 3-round trail cores for XOODOO up to weight 50 and these trails are indeed there. There are 3 different single-orbital fans and for each them there are 4 trail cores that have equal input differences in the 2nd round and output differences in the 2nd round with an additional bit. For each single-orbital fan two of these trail cores have weight profile $(4, 4, 38)$, so they do not go to all different columns, leading to total weight 46 and for two they do go to different columns and have $(4, 4, 42)$, so total weight 50. Differentials in the former will contribute $2^{38}2^{-92} = 2^{-54}$ and in the latter $2^{42}2^{-100} = 2^{-56}$, hence their contribution in comparison with 2^{-44} is quite small.

There are 6 output differences of the 2nd round χ with 2 additional active bits, 4 with with 3 additional active bits and one with 4 additional active bits. These additional bits lead to even higher values in w_3 and hence they contribute even less to ε. Trails starting from q_0 can already start diverging from q_1. However, there is only a single difference q_1 compatible with q_0 that is in the kernel. So all these alternative output differences make w_2 explode from 4 to 18 and give even higher w_3.

So we conclude that $\varepsilon \approx 2^{-44}$ and that this is obtained by just taking the trails in the dominant trail core. We see that for XOODOO[3] ζ is a factor 2^8 larger than ε.

For higher number of rounds we expect this factor to be much larger. In XOODOO[4] the trail cores with the lowest weight have weight 80 and they are described in [12]. Assuming trail clustering is negligible for these trails and independence of round differentials yields $\zeta \approx 2^{-80}$.

For ε we need again to look at the weight profiles. There are in fact only two canonical 4-round trail cores with weight 80 and their weight profiles are respectively $(8, 16, 24, 32)$ and $(32, 24, 16, 8)$. These trails remain in the kernel in the three λ steps in between the non-linear layers.

If we approximate ε by the contribution of all trails in the core, the former would give $\varepsilon \approx 2^{-160}2^{32} = 2^{-128}$ and the latter $\varepsilon \approx 2^{-160}2^{8} = 2^{-152}$. Of course it may be that for more rounds the contribution of trails outside the trail core has relatively more importance, but the difference to bridge is much higher as here with this estimation ζ is a factor 2^{48} larger than ε.

7 Conclusion

The serialization and the parallelization of a block functions are two new ways to create a keyed hash based on an unkeyed premitive. It does not require the premitive to be

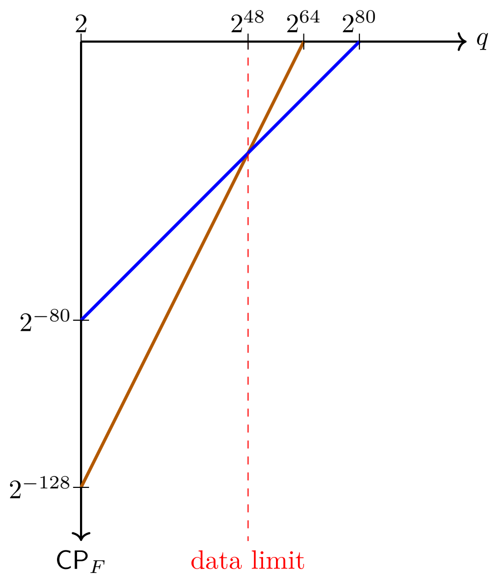

Figure 4: Multi-query attack on the parallelization of Xoodoo[4] for the estimated ε and ζ. Attacks on WC(S) start with a probability $\zeta = 2^{-80}$ and grow linearly in the number of queries q while attacks on PH start with probability $\varepsilon = 2^{-128}$ but grow quadratically.

ideal nor does it require a nonce. The collision probability of these constructions are directly defined by the DP and IP of the block function and nothing else. The bounds of the parallelization brings attention to the computation of the $\sum \mathsf{DP}^2$ of a cryptographic primitive, something that was not common before. Furthermore, the collision probability of the serialization lower bounds the security of the parallelization due to different terms used for the computation of ζ and ε of the parallelization. This difference also comes to play when we compare the security against forgery when using the parallelization of an iterated permutation with WC(S) and PH. In the case of an iterated permutation, ε is usually smaller than ζ. This means that in the case of a multi-query attack, WC(S) and PH provide different levels security and one approach is better than the other depending on data complexity limit. Let q be the number of queries of the attacker, WC(S) and PH achieve the same security when $q \approx \frac{2\varepsilon}{\zeta}$. If the data complexity limit is below this threshold then PH with the parallelization of a cryptographic permutation is a better approach than WC(S) with the parallelization of a cryptographic permutation and vice-versa. We have visualized in Figure 4.

Acknowledgments

Joan Daemen and Jonathan Fuchs are supported by the European Research Council under the ERC advanced grant agreement under grant ERC-2017-ADG Nr. 788980 ESCADA.

References

[1] Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Koblitz, N. (ed.) Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1109, pp. 1–15. Springer (1996), https://doi.org/10.1007/3-540-68697-5_1

[2] Bellare, M., Kilian, J., Rogaway, P.: The security of cipher block chaining. In: Desmedt, Y. (ed.) Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings. Lecture Notes in Computer Science, vol. 839, pp. 341–358. Springer (1994), https://doi.org/10.1007/3-540-48658-5_32

[3] Bernstein, D.J.: How to stretch random functions: The security of protected counter sums. J. Cryptology 12(3), 185–192 (1999), http://dx.doi.org/10.1007/s001459900051, https://cr.yp.to/papers.html#stretch

[4] Bernstein, D.J.: The poly1305-aes message-authentication code. In: Gilbert, H., Handschuh, H. (eds.) Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers. Lecture Notes in Computer Science, vol. 3557, pp. 32–49. Springer (2005), https://doi.org/10.1007/11502760_3

[5] Bertoni, G., Daemen, J., Hoffert, S., Peeters, M., Assche, G.V., Keer, R.V.: Farfalle: parallel permutation-based cryptography. IACR Trans. Symmetric Cryptol. 2017(4), 1–38 (2017), https://tosc.iacr.org/index.php/ToSC/article/view/801

[6] Black, J., Rogaway, P.: A block-cipher mode of operation for parallelizable message authentication. In: Knudsen, L.R. (ed.) Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2332, pp. 384–397. Springer (2002), https://doi.org/10.1007/3-540-46035-7_25

[7] Bordes, N., Daemen, J., Kuijsters, D., Assche, G.V.: Thinking outside the superbox. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12827, pp. 337–367. Springer (2021), https://doi.org/10.1007/978-3-030-84252-9_12

[8] Daemen, J.: Cipher and hash function design, strategies based on linear and differential cryptanalysis, PhD Thesis. K.U.Leuven (1995), http://jda.noekeon.org/

[9] Daemen, J., Assche, G.V.: Differential propagation analysis of keccak. In: Canteaut, A. (ed.) Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers. Lecture Notes in Computer Science, vol. 7549, pp. 422–441. Springer (2012), https://doi.org/10.1007/978-3-642-34047-5_24

[10] Daemen, J., Hoffert, S., Assche, G.V., Keer, R.V.: DC-Xoodoo-3r.txt. https://github.com/KeccakTeam/Xoodoo/blob/master/XooTools/Trails/DC-Xoodoo-3r.txt/ (2018)

[11] Daemen, J., Hoffert, S., Assche, G.V., Keer, R.V.: The design of xoodoo and xoofff. IACR Trans. Symmetric Cryptol. 2018(4), 1–38 (2018), https://doi.org/10.13154/tosc.v2018.i4.1-38

[12] Daemen, J., Mella, S., Assche, G.V.: Tighter trail bounds for xoodoo. Cryptology ePrint Archive, Paper 2022/1088 (2022), https://eprint.iacr.org/2022/1088, https://eprint.iacr.org/2022/1088

[13] Daemen, J., Rijmen, V.: The pelican MAC function. IACR Cryptol. ePrint Arch. 2005, 88 (2005), http://eprint.iacr.org/2005/088

[14] Gunsing, A., Daemen, J., Mennink, B.: Deck-based wide block cipher modes and an exposition of the blinded keyed hashing model. IACR Trans. Symmetric Cryptol. 2019(4), 1–22 (2019), https://doi.org/10.13154/tosc.v2019.i4.1-22

[15] ISO/IEC: Information technology – security techniques – message authentication codes (macs) – part 1: Mechanisms using a block cipher (1999)

[16] McGrew, D.A., Viega, J.: The use of galois message authentication code (GMAC) in ipsec ESP and AH. RFC 4543, 1–14 (2006), https://doi.org/10.17487/RFC4543

[17] Mella, S., Daemen, J., Assche, G.V.: New techniques for trail bounds and application to differential trails in keccak. IACR Trans. Symmetric Cryptol. 2017(1), 329–357 (2017), https://doi.org/10.13154/tosc.v2017.i1.329-357

[18] Shoup, V.: On fast and provably secure message authentication based on universal hashing. In: Koblitz, N. (ed.) Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1109, pp. 313–328. Springer (1996), https://doi.org/10.1007/3-540-68697-5_24

[19] Stinson, D.R.: Universal hashing and authentication codes. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO '91. Lecture Notes in Computer Science, vol. 576, pp. 74–85. Springer (1991), https://doi.org/10.1007/3-540-46766-1_5

[20] Stinson, D.R.: On the connections between universal hashing, combinatorial designs and error-correcting codes. Electron. Colloquium Comput. Complex. 2(52) (1995), http://eccc.hpi-web.de/eccc-reports/1995/TR95-052/index.html

[21] 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), https://doi.org/10.1016/0022-0000(81)90033-7