More Rounds, Less Security?

Ritam Bhaumik[1], Avijit Dutta[1], Jian Guo[2], Jérémy Jean[2], Nicky Mouha[3,4], and Ivica Nikolić[2]

[1] Indian Statistical Institute, Kolkata, India
[2] Nanyang Technological University, Singapore
[3] Dept. Electrical Engineering-ESAT/COSIC, KU Leuven, Leuven and iMinds, Ghent, Belgium
[4] Project-team SECRET, Inria, France
bhaumik.ritam@gmail.com, avirocks.dutta13@gmail.com, ntu.guo@gmail.com, JJean@ntu.edu.sg, Nicky.Mouha@esat.kuleuven.be, inikolic@ntu.edu.sg

Abstract. This paper focuses on a surprising class of cryptanalysis results for symmetric-key primitives: when the number of rounds of the primitive is increased, the complexity of the cryptanalysis result *decreases*. Our primary target will be primitives that consist of identical round functions, such as PBKDF1, the Unix password hashing algorithm, and the Chaskey MAC function. However, some of our results also apply to constructions with non-identical rounds, such as the PRIDE block cipher. First, we construct distinguishers for which the *data complexity* decreases when the number of rounds is increased. They are based on two well-known observations: iterating a random permutation increases the expected number of fixed points, and iterating a random function decreases the expected number of image points. We explain that these effects also apply to components of cryptographic primitives, such as a round of a block cipher. Second, we introduce a class of key-recovery and preimage-finding techniques that correspond to exhaustive search, however on a smaller part (e.g. one round) of the primitive. As the *time complexity* of a cryptanalysis result is usually measured by the number of full-round evaluations of the primitive, increasing the number of rounds will lower the time complexity. None of the observations in this paper result in more than a small speed-up over exhaustive search. Therefore, for lightweight applications, implementation advantages may outweigh the presence of these observations.

Keywords: Iterated cipher, fixed points, slide attack, PRIDE, Chaskey, PKCS, PBKDF1, Unix password hashing algorithm, Even-Mansour, FX-construction.

1 Introduction

How to determine the number of rounds of a symmetric-key primitive, for example of an iterated block cipher? O'Connor noted that "Most ciphers are secure after sufficiently many rounds," to which Massey replied that "Most ciphers are too slow after sufficiently many rounds." From this point of view, the challenge is to design a cipher that is both fast and secure.

But does increasing the number of rounds always make a primitive more secure? Although this is generally the case, it is not necessarily true. A common counterexample are primitives that are vulnerable to slide attacks [8,9]. As stated by Biruykov in [7], slide attacks "realize the dream of cryptanalysts: if the cipher is vulnerable to such an attack, the complexity of the attack is independent of the number of rounds of the cipher." In that context, this paper describes what certainly must be the nightmare of any cryptographer. We will discuss cryptanalysis results where the complexity decreases when the number of rounds is increased.

Before we continue, let us clarify how we measure complexity in terms of data complexity and time complexity. In symmetric-key cryptanalysis, the *data complexity* commonly refers to the number of input-output pairs of a certain keyed function, for example the number of plaintext-ciphertext pairs for a block cipher. The *time complexity* is usually defined as equivalent number of full-round evaluations of a cryptographic primitive. This is a very simplified model, but nevertheless useful for designers and cryptanalysts as the cost in practice of an attack is usually very difficult to estimate.

Our basic observations hold for primitives that have identical round functions, such as Unix's `crypt(3)` and PKCS #5's PBKDF1 password hashing algorithms, or the Chaskey MAC function [28]. But as we will show, our results also apply to primitives with non-identical round functions, such as the recently proposed PRIDE block cipher [1].

The very nature of the results in this paper seems to defy logic, however designers do not need to worry too much: their practical impact is very low, as all of them are very close to exhaustive search. Nevertheless, this paper should be interesting from an academic point of view, given the interest in recent years into "brute-force-like" cryptanalysis. These include, for example, the recently proposed biclique results for many ciphers, including the full AES [10]. However, it can be shown that the biclique attacks on AES are thwarted when the number of rounds is increased. This is an essential difference with the results that we will introduce in this paper, as they will have an even lower complexity when the number of rounds is increased.

In this light, this paper is particularly interesting for the domain of lightweight cryptography. Designers may want to explicitly allow small speed-ups of exhaustive search, if such design choices lead to a more efficient implementation. As such, we hope that this paper will eventually lead to a better theoretical understanding of the design of symmetric-key primitives. It is typically conjectured that a cipher becomes more secure when the number of rounds is increased. While a proof of this conjecture remains an open problem for many constructions, this paper will show that there exist constructions for which the conjecture does not hold.

None of the techniques in this paper will result in a speed-up over exhaustive search by more than the number of rounds. A lightweight design with an n-bit block size that is vulnerable against such attacks, may therefore suggest that it

is secure up to 2^n round function evaluations, instead of 2^n evaluations of the entire primitive.

Our Contributions. In the domain of provable security, it is well-known that iterating an ideal primitive will result in a loss of security. This is already evident from the very first results in provable security, so we certainly do not claim a new result in that area. However, after performing a thorough literature study on the cryptanalysis of non-ideal primitives, it appears to be a little-known fact that for certain constructions, the complexity of a cryptanalysis result can *decrease* when the number of rounds of the primitive is *increased*. The goal of this paper is to perform the first comprehensive study on this subject. We give an overview of the existing literature, which typically involves ideal primitives. Then, we apply these insights to non-ideal components (e.g. one round of a block cipher) that arise in the field of cryptanalysis. We obtain new cryptanalysis results on ciphers built on the Even-Mansour block cipher and the FX-construction, which we then apply respectively to Chaskey and PRIDE.

Outline. After discussing related work in Sect. 2, we describe present security bounds in Sect. 3. We show that these security bounds are tight by providing matching attacks. In Sect. 4, we argue that these distinguishers also apply when non-ideal building blocks are iterated. Speed-ups of exhaustive search on iterated primitives are discussed in Sect. 5. We apply these findings to a variety of primitives, and present new results for Chaskey MAC function, the PRIDE block cipher and a variant of PBKDF1 without a salt. We conclude the paper in Sect. 6, where we also provide suggestions for future work.

2 Related Work

Wagner and Goldberg [32] performed an analysis of the Unix password hashing algorithm, which consists of 25 applications of DES on an all-zero plaintext. The password of the user is used as the key of the DES algorithm. Note that this description omits the salt value, an important feature of the password hashing algorithm. However, Wagner and Goldberg do not analyze the effect of the salt in their paper. As the salt value is used to create different variants of the DES algorithm, their analysis therefore effectively considers the Unix password hashing algorithm when only one salt value is used.

They analyzed this construction by observing its close relation to the security the CBC-MAC algorithm [3, 4]. Their security bound shows that Unix password hashes may not be uniformly random. However, as their analysis assumes that the adversary obtains only one password hash, this effect becomes negligible. The implications of changing the number of iterations is not considered in their paper.

The same effect was analyzed by Bard et al. [2], when they calculated the expected number of fixed points of a random permutation that is iterated $k \leq 2^n$

times. They found that it is equal to $\tau(k)$, where τ is the number-of-divisors function. For example, when this result is applied to Unix password hashing where $k = 25$, they calculated that the expected number of fixed points is $\tau(25) = 3$, as 25 has three divisors: 1, 5 and 25. Thus, by counting the number of fixed points, we can distinguish this construction from random. Note that for the particular case of Unix password hashing, this distinguisher will have a low, but nevertheless non-negligible success probability. The reason for the low success probability is that the length of the password (56 bits) is smaller than the length of the hash value (64 bits).

Bard et al. did not state that their cryptanalysis result becomes better when the number of rounds is increased. In fact, strictly speaking, such a statement would not be correct. There are infinitely many prime numbers, and if the number of rounds is a prime number, the expected number of fixed points is always two. However, the limit superior of $\tau(k)$ goes to infinity, and $\tau(k)$ is strictly increasing for commonly used subsets of k, such as powers of two: $\tau(2^\ell) = \ell + 1$, or powers of ten: $\tau(10^\ell) = (\ell + 1)^2$. Under this more narrow interpretation, we argue that this can be seen as a distinguisher for which the complexity decreases when the number of rounds is increased.

In subsequent work, Yao and Yin [34, 35] analyzed the two standardized password-based key derivation functions of PKCS #5: PBKDF1 and PBKDF2. PBKDF1 concatenates the password and salt, and then iteratively applies a hash function k times to the result. Note that k is not a fixed parameter, but depends on the implementation. They argue that PBKDF1 becomes more secure when k is increased, which they refer to as the effect of "key-stretching."

However, the numerical examples from which Yao and Yin derive this statement, assume that the adversary performs only a small number of hash function evaluations. When we consider adversaries that make a very large number of hash function evaluations, the security bound on PBKDF1 becomes weaker when the iteration count k is increased. We should note that in practice, passwords often have a very low entropy, in which case adversaries only need to make a small amount of queries. Nevertheless, it should be interesting to understand the generic security of iterated constructions for any number of queries.

Gligoroski and Klima [19] showed how this weaker security bound also corresponds to an observation on iterated random functions such as PBKDF1. They recalled a theorem by Flajolet and Odlyzko [18]: if an n-bit random function is iterated k times, the expected number of image points is $(1 - t_k) \cdot 2^n$ (for large n), where t_k satisfies the recurrence relation $t_0 = 0$ and $t_{k+1} = \exp(-1 + t_k)$. From this, it can be shown that the expected number of image points is 2^{n-i+1} when a random function is iterated $k = 2^i$ times. This approximation is valid for $i \geq 5$, and becomes more accurate when i increases. Note that the observations of Gligoroski and Kilma also hold if k independent random functions are used, instead of one random function that is iterated k times.

Using this theorem, Gligoroski and Klima constructed distinguishers for several iterated constructions, including PBKDF1 and several narrow-pipe hash functions. From their observations, they argued for wide-pipe instead of narrow-

pipe hash functions. They did not, however, point out that increasing the number of rounds of a primitive makes their distinguishers more successful. This is of course evident from the formulas, but it was not part of their story line.

The same "entropy loss" was evaluated in the context of stream ciphers, for example by Hong and Kim [20] and by Röck [31] for the MICKEY stream cipher. However, the context is not the same as the one that we consider in this paper. Their analysis concludes that security decreases when more output bits are known, whereas we do not increase the number of output bits, but only the number of rounds of the primitive.

The security of iterated random permutations was studied very recently by Minaud and Seurin [27] at CRYPTO 2015. When the number of iterations k is constant, they provide a security bound as well as a simple attack that matches their bound up to a constant factor. An alternative simple proof was later provided by Nandi [30]. We will recall their result in this paper, and analyze the security of an iterated random function in the same setting. We argue that resulting distinguishers also apply when a non-ideal permutation is iterated, and support this claim by experiments.

At the rump session of the ECRYPT II Hash Function Retreat in 2009 [14], Dunkelman showed how to speed up preimage search for the ESSENCE hash function [26] by the number of rounds. Unfortunately, this result was never published, and appears to be unknown to most of the research community. In the context of the 3D Cipher, the technique of Dunkelman was independently rediscovered by Wang et al. [33], who point out that the complexity decreases when the number of rounds is increased. In Sect. 5, we will recall this observation and extend it to other constructions.

It should be pointed out that the complexity of typical cryptanalysis attacks does not always increase monotonically. Biham and Chen found that 82 rounds of SHA-0 are easier to attack than 80 rounds [6]. For the Whirlpool hash function, Sasaki found that the 6.5 rounds are weaker against a preimage attack than 6 rounds. Similarly, Ma et al. [25] constructed preimage attacks on 5 rounds and 6.5 rounds of the GOST-256 hash function, but could not attack 6 rounds. In this paper, we will not consider these attacks, as we will only focus on results where the complexity decreases (monotonically or non-monotonically) as the number of rounds is increased.

The indifferentiability setting will not be studied in this paper. Of particular interest in this area, however, is a result by Dodis et al. [13] that show that the second iterate $H^2(M) = H(H(M))$ of a random oracle H has poor concrete security in the sense of indifferentiability from a random oracle.

3 Distinguishers on Ideal Iterated Primitives

In this section, we will study the security of iterated primitives. Firstly, we will look at constructions where an ideal primitive is iterated, which can be a random function or a random permutation. Then, we state security bounds for the resulting constructions. For a fixed number of iterations k, we will explain that

these security bounds are tight. We do this by recalling the cycle distinguisher of Minaud and Seurin [27] for an iterated random permutation, and by introducing a collision distinguisher for an iterated random function.

We then recall two other distinguishers on these constructions: entropy-loss distinguishers and fixed-point distinguishers. As mentioned in Sect. 2, these distinguishers have been described in existing literature when a random function or a random permutation is used. We will explain how they also apply to constructions with non-ideal primitives, by presenting a fixed-point distinguisher for the Chaskey MAC function. Interestingly, doubling the number of rounds will *decrease* the complexity of this fixed-point distinguisher.

3.1 Security Bounds on Iterated Primitives

Let us assume that adversaries are computationally unbounded, and are only limited by the number of queries that they make. Without loss of generality, we assume that all queries made by an adversary are distinct. Let $\mathsf{perm}(n)$ denote the set of all permutations on n bits, and let $\mathsf{func}(n)$ be the set of all functions from n to n bits. We then define the advantage of an iterated primitive distinguisher as follows (see also Fig. 1).

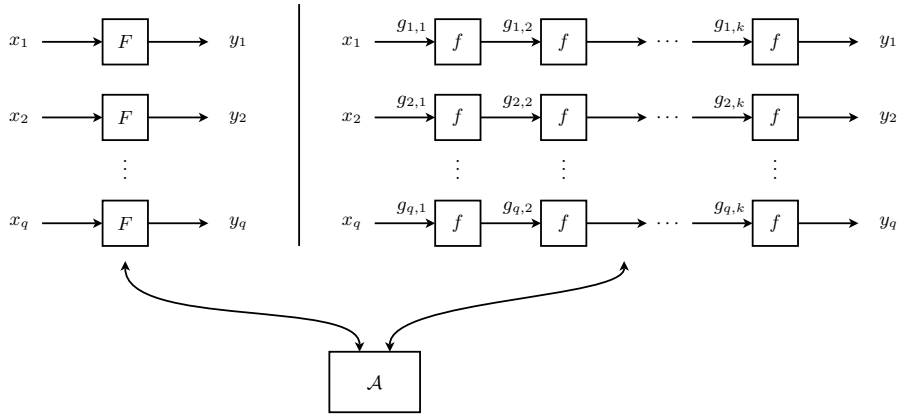

Fig. 1. Distinguishing between a random function (resp. permutation) F and a random function (resp. permutation) f that is iterated k times. The adversary $\mathcal{A}$ makes at most q queries. All queries are assumed to be distinct. In case F and f are permutations, the $\mathcal{A}$ can make both forward and inverse queries.

Definition 1 (Iterated Primitive Distinguisher). *Let F and f be a random variables over $\mathsf{perm}(n)$ (resp. over $\mathsf{func}(n)$). For an adversary $\mathcal{A}$, the advantage of distinguishing a random function (resp. random permutation) from the k-th*

iterate of a random function (resp. random permutation) using at most q queries is

$$\mathbf{Adv}_{F,f^k}(\mathcal{A}) = \left| \Pr\left[\mathcal{A}^F(q) \to 1\right] - \Pr\left[\mathcal{A}^{f^k}(q) \to 1\right] \right| , \tag{1}$$

where the adversary can make both forward and inverse queries in case F and f are permutations. Note that the adversary is only given access to F and f^k, and not to f directly.

For an iterated random function, the maximum adversarial advantage can be derived from the security bounds of CBC-MAC [3–5, 29], as was done by Wagner and Goldberg [32]. We will provide a simpler proof and a tighter bound for this construction. For the security bound for a iterated random permutation, we recall a recent result of Minaud and Seurin [27]. Theorem 1 considers the case where both F and f are random functions. In Theorem 2, both F and f are random permutations.

Theorem 1 (Iterated Random Function Security). *Let F be random function, and f^k be a random function that is iterated k times. Then for all $\mathcal{A}$ making at most q queries,*

$$\mathbf{Adv}_{F,f^k}(\mathcal{A}) \leq \frac{kq(kq-1)}{2^{n+1}} - \frac{q(q-1)}{2^{n+1}} \ . \tag{2}$$

Proof. *Let $\mathbf{E_1}$ be the event that there exist $g_{i,j} = g_{i',j'}$ such that $\neg(i = i' \wedge j = j')$ (see Fig. 1). Given that $\mathbf{E_1}$ does not happen, no input of f will be reused. Note that trivial reuse occurs when there exist $\ell, \ell' : \ell \neq \ell'$ such that $x_\ell = x_{\ell'}$, but we do not have to consider this as we assumed w.l.o.g. that the adversary makes distinct queries. Under the negation of $\mathbf{E_1}$, all queries are independent of each other, and the reply to every query follows a uniformly random distribution, so that F and f^k are indistinguishable. By the fundamental lemma of game playing, $\mathbf{Adv}_{F,f^k} \leq \Pr\left[\mathbf{E_1}\right]$. From the union bound, the probability that $g_{i,j} = g_{i',j'}$ where $\neg(i = i' \wedge j = j')$, is at most $0/2^n + 1/2^n + \cdots + (kq-1)/2^n = kq(kq-1)/2^{n+1}$. In fact, we can construct a tighter upper bound, by noting that since we assumed that all queries are distinct, we know that $g_{i,1} \neq g_{i',1}$ for $i \neq i'$. Let us therefore consider the event $\mathbf{E_2}$ that there exist $g_{i,j} = g_{i',j'}$ such that $\neg(i = i' \wedge j = j')$ and $\neg(j = j' = 1)$. Using a similar reasoning as before, F and f^k are indistinguishable under the negation of $\mathbf{E_2}$. Again by the union bound, the probability that $g_{i,j} = g_{i',j'}$ where $\neg(i = i' \wedge j = j')$ and $\neg(j = j' = 1)$, is at most $q/2^n + (q+1)/2^n + \cdots + (kq-1)/2^n$. This bound is similar to $\mathbf{E_1}$, but it does not contain the first q terms of $\mathbf{E_1}$ as they correspond to the cases where $\neg(i = i' \wedge j = j')$ but $j = j' = 1$. This sum is equal to $kq(kq-1)/2^{n+1}$ - $q(q-1)/2^{n+1}$, from which the theorem follows.* □

Theorem 2 (Iterated Random Permutation Security). *Let F be a random permutation, and f^k be a random permutation that is iterated k times. Then for all $\mathcal{A}$ making at most q queries,*

$$\mathbf{Adv}_{F,f^k}(\mathcal{A}) \leq \frac{(2k+1)q}{2^n} \ . \tag{3}$$

We refer to Minaud and Seurin [27] for a proof of this bound.

3.2 Matching Attacks for Iterated Primitives

Iterated Random Permutation. In [27], Minaud and Seurin proved that the bound of Theorem 1 is tight. When the number of iterations k is fixed, they provide an distinguisher that matches this bound up to a constant factor. We recall their distinguisher in Algorithm 1.

Algorithm 1: Cycle Distinguisher $\mathcal{D}^{\mathsf{Q}}_{\text{cycle}}(q)$

$s_0 \xleftarrow{\$} \{0,1\}^n$
for $i \leftarrow 0$ **to** $q-1$ **do**
  $s_{i+1} \leftarrow \mathsf{Q}(s_i)$
end
if *all* s_i *distinct* **then**
  return *0*
else
  return *1*
end

Iterated Random Function. When the number of iterations k is fixed, we now show that the bound of Theorem 1 is also tight. A distinguisher that matches the bound is provided in Algorithm 2. This distinguisher makes q queries to the interface Q, which can be either F or f^k, where F and f are random functions.

Algorithm 2: Collision Distinguisher $\mathcal{D}^{\mathsf{Q}}_{\text{coll}}(q)$

for $i \leftarrow 0$ **to** $q-1$ **do**
  $s_i \leftarrow \mathsf{Q}(i)$
end
if *all* s_i *distinct* **then**
  return *0*
else
  return *1*
end

For simplicity, let us assume that all f^i for $i \leq k$ are random functions. Without this assumption, it can be seen that the advantage of $\mathcal{D}^{\mathsf{Q}}_{\text{coll}}(q)$ will be even higher. This case will be dealt with in App. B, where use a more rigorous analysis to obtain a bound.

Theorem 3 (Collision Distinguisher for an Iterated Random Function). *Let F be a random function, and f^k be a random function that is iterated k times. Assume $q^2 \leq 2^n/k$. Then*

$$\mathbf{Adv}_{F,f^k}(\mathcal{D}^{\mathsf{Q}}_{\mathrm{coll}}(q)) \geq \frac{(k-1)q^2}{2^{n+3}} \ . \tag{4}$$

Proof. By Definition 1, the advantage of $\mathcal{D}^{\mathsf{Q}}_{\mathrm{coll}}(q)$ is given by

$$\mathbf{Adv}_{F,f^k}(\mathcal{D}^{\mathsf{Q}}_{\mathrm{coll}}(q)) = \left| \Pr\left[\mathcal{D}^{F}_{\mathrm{coll}}(q) \to 1\right] - \Pr\left[\mathcal{D}^{f^k}_{\mathrm{coll}}(q) \to 1\right] \right| \ . \tag{5}$$

Let p_c denote $\Pr\left[\mathcal{D}^{F}_{\mathrm{coll}}(q) \to 1\right]$, i.e. the probability that q queries to a random function f result in a collision. We have

$$\begin{aligned}
\Pr\left[\mathcal{D}^{f^k}_{\mathrm{coll}}(q) \to 1\right] &= 1 - \Pr\left[\forall i \leq k : \textit{all } q \textit{ outputs of } f^i \textit{ are distinct}\right] \ , && (6)\\
&= 1 - (1-p_c)^k \ , && (7)\\
&\geq 1 - (1 - kp_c + \binom{k}{2} p_c^2) \ , && (8)\\
&= kp_c - \binom{k}{2} p_c^2 \ . && (9)
\end{aligned}$$

As shown in App. A, $q^2/2^{n+1} \leq p_c \leq q^2/2^n$. Therefore,

$$\begin{aligned}
\mathbf{Adv}_{F,f^k}(\mathcal{D}^{\mathsf{Q}}_{\mathrm{coll}}(q)) &\geq (k-1)p_c - \binom{k}{2} p_c^2 \ , && (10)\\
&= (k-1)p_c \left(1 - \frac{kp_c}{2}\right) \ , && (11)\\
&\geq (k-1)\frac{q^2}{2^{n+1}} \left(1 - \frac{kq^2}{2^{n+1}}\right) \ , && (12)\\
&\geq \frac{(k-1)q^2}{2^{n+3}} \ . && (13)
\end{aligned}$$

□

4 Distinguishers on Non-Ideal Iterated Primitives

In Sect. 3, we stated security bounds for iterated permutations and iterated random functions. Both bounds were shown to be tight. For a fixed value of the iteration count k, respectively the cycle distinguisher (Algorithm 1) and the collision distinguisher (Algorithm 2) can be shown to match the bounds up to a constant factor.

Sect. 2 introduced two other distinguishers for iterated primitives, which we will refer to as the entropy-loss distinguisher for the case of random functions, and as the fixed-point distinguisher when random permutations are considered. As we explained, these distinguishers have already been applied to several iterated constructions, including the Unix password hashing algorithm and PBKDF1. For each of these constructions, the underlying building blocks were assumed to be ideal.

We now introduce a new result in this paper, by looking at a non-ideal building block that is iterated, such as a round of a block cipher. It is clear that one round of a block cipher cannot be modeled as a random permutation, as it is often easy to distinguish such a component from random. However, we claim that the aforementioned distinguishers also apply to the iteration of non-ideal building blocks, with an advantage that is at least as high as if they were random.

As all of the distinguishers presented in this paper have a very high complexity, it is not feasible to experimentally verify their complexity for real-world constructions. However, we can back up our claims by performing experiments on small-scale variants of ciphers. These distinguishers have a lower data complexity when the number of rounds is increased. They apply regardless of the underlying algorithm, and even when the algorithm is not known to the adversary. In the next section, we show an application of this idea to the Chaskey MAC function.

4.1 Application to Chaskey

Chaskey [28] is a Message Authentication Code (MAC) algorithm designed by Mouha et al. as a collaboration between KU Leuven-COSIC and Hitachi YRL. Currently, Chaskey is in a study period for standardization by ISO/IEC JTC 1/SC 27/WG 2. Standardization is expected to start in October 2015, when its maturity is confirmed. ITU-T SG17 has recently added new work items related to IoT and ITS security, for which Chaskey seems to be well-suited.

Although Chaskey uses a dedicated mode of operation to process variable-length messages, in this paper we will focus only on one-block messages. In that case, Chaskey uses a 128-bit key K to transform a 128-bit message M into a tag T of at most 128 bits. It does this using a variant of the Even-Mansour block cipher [16, 17]: $C = E_K(P) = K_1 \oplus \pi(P \oplus K \oplus K_1)$, where K_1 is related to K by a linear function.[5] Here, the plaintext P is equal to the one-block message, and the ciphertext C corresponds to the (untruncated) tag T. The Chaskey permutation π consists of eight identical rounds, one of which is shown in Fig. 2. A sixteen-round variant of Chaskey, named Chaskey-LTS, was proposed as well.

The derivation of the subkey K_1 from K goes as follows. Let us interchangeably consider an element a of $GF(2^{128})$ as the integer $2^{127}a[127] + 2^{126}a[126] + \ldots 2^0 a[0]$ in decimal notation, and as the polynomial $a(x) = a[n-1]x^{n-1} +$

[5] Chaskey also uses a subkey K_2 to process messages of an incomplete number of blocks, however this is not relevant to the analysis in this paper as we will only consider messages of one full block.

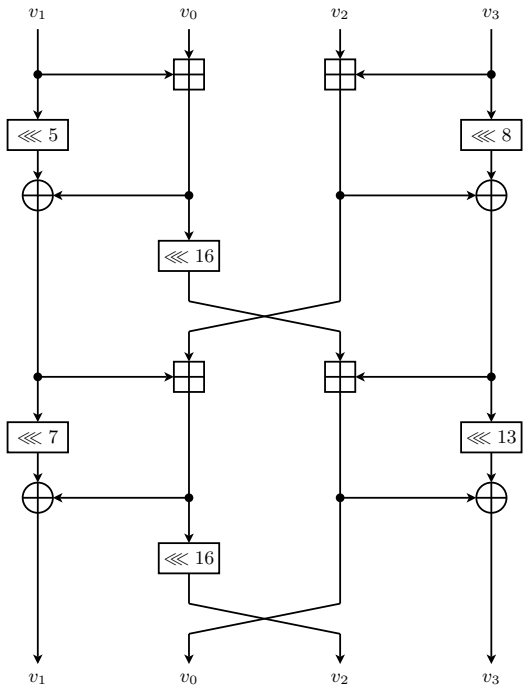

Fig. 2. One round of the Chaskey permutation π, defined as: $v_0\|v_1\|v_2\|v_3 \leftarrow \pi(v_0\|v_1\|v_2\|v_3)$. The values v_0 and v_1 are intentionally swapped, as this reduces the number of crossing lines in the figure.

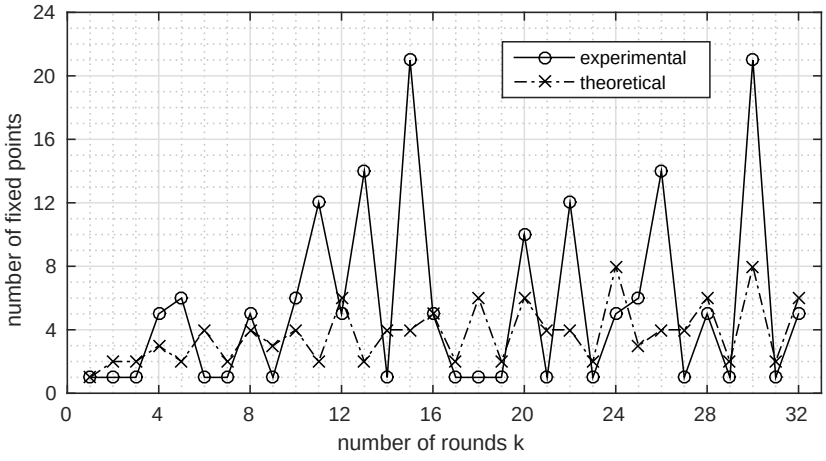

Fig. 3. Number of fixed points for $1 \leq k \leq 32$ rounds of SmallChaskey, compared to the expected number of fixed points if every round were a random permutation.

$a[n-2]x^{n-2} + \ldots + a[0]$ with binary coefficients. To multiply two elements a and b, we represent them as two polynomials $a(x)$ and $b(x)$, and calculate $a(x)b(x) \bmod f(x)$ where $f(x) = x^{128} + x^7 + x^2 + x + 1$. Using this notation, the subkey K_1 of Chaskey can be defined as $K_1 = 2K$.

The claimed security of the Chaskey block cipher is about $T = 2^{128}/D$ permutation evaluations when D plaintext-ciphertexts are available. We will refer to T and D as the time and the data complexity, respectively. The currently best known attack on Chaskey is by Leurent [24] on 7 rounds out of 8 rounds, and uses between 2^{45} and 2^{48} chosen plaintexts.

In the fixed-point distinguishers that we will now consider, we will have $T = 0$. This means that we will not perform a single permutation evaluation: the distinguisher does not depend on the permutation π, and even works if the algorithmic description of π is unknown to the adversary. Going back to the setting of Even-Mansour [16, 17], this means that the entire codebook (all 2^{128} plaintexts and ciphertexts) should appear as if it was generated by a random permutation.

Any number of rounds of the Chaskey permutation always has at least one fixed point: the all-zero input. The search for additional fixed points seems to be a computationally difficult problem, even for only one round of Chaskey. For this reason, we performed some experiments on SmallChaskey, a variant of Chaskey that we introduce in this paper. In SmallChaskey, all 32-bit words are replaced by 8-bit words, and the rotation constants 16, 8, 13, and 7 are replaced by 4, 2, 6, and 3, respectively.

The results are plotted in Fig. 3. As we can see from this figure, the number of fixed points is significantly higher than one, which is the number of fixed points expected for a random permutation. It also seems to be sometimes much higher than expected for an iterated random permutation. We conjecture that this behavior also holds for the full-size Chaskey and for other cryptographic round functions.

If this is the case, an adversary can distinguish Chaskey from a random permutation by collecting close to $D = 2^{128}$ plaintext-ciphertext pairs and counting the number of fixed points. This distinguisher has no impact on the Chaskey MAC function, which restricts the data complexity available to the adversary to at most 2^{64} plaintext-ciphertexts, because then a collision in the internal state of the mode of operation has a non-negligible probability.

However, we do not look into the mode of operation, but instead distinguish the Chaskey block cipher from an Even-Mansour cipher from a random permutation. This problem may be of independent academic interest. In Sect. 5.2, we will present results on Chaskey that also apply when a low number of plaintext-ciphertexts are available.

5 Speeding Up Exhaustive Search

The distinguishers that we presented in Sect. 3 and 4 have the property that the *data complexity* decreases when the number of rounds is increased. The time

complexity was not a parameter, in fact the distinguishers even work if the underlying algorithm is not known to the adversary.

We now look at cryptanalysis results where the *time complexity* decreases when the number of rounds is increased. These results hold the conventional model that is used in cryptanalysis: the time complexity is calculated in terms of the equivalent number of full-round computations.

However, we assume that the time complexity is only determined by the number of round evaluations. Any other computations that are performed by the adversary, are not taken into account. As a result of this, the speed-up over exhaustive search may be less than our simplified model shows. If the overhead of these other computations is significant, it may even be that there is only a speed-up over exhaustive search if the cipher uses a sufficiently large number of rounds.

Note that our analysis uses computational model that is common for meet-in-the-middle and biclique-style cryptanalysis. However, those types of cryptanalysis can be prevented by increasing the number of rounds. This is not the case for our observations. In fact, our observations have a lower complexity when the number of rounds is increased.

After recalling an observation by Dunkelman on the ESSENCE hash function, we will introduce new speed-ups over exhaustive search for the Chaskey MAC function, for the PRIDE block cipher, and for a variant of PBKDF1 without a salt.

5.1 ESSENCE

Let us recall the speed-up over exhaustive preimage search by Dunkelman [14] on the ESSENCE hash function. The ESSENCE compression function consists of 32 identical rounds, shown in Fig. 4. A preimage for the compression function (a "pseudo-preimage" for the hash function) can be found as follows. Let X be the target compression function value, so that we are looking for a message M and a chaining value CV for which $C(CV, M) = X$, where C is the ESSENCE compression function.

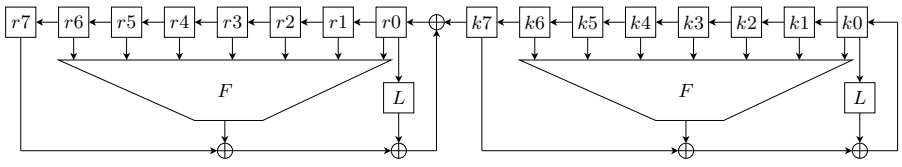

Fig. 4. One round of the ESSENCE hash function. The chaining value CV is loaded into the r_i registers, whereas the message M is loaded into the k_i registers. Every register is either 32 or 64 bits, resulting in a hash value of either 256 or 512 bits. Function F is a bitwise non-linear function, and L is a linear function operating on an entire word. ESSENCE consists of 32 rounds, after which a feed-forward is applied.

1. Initialize the message M and the chaining value CV with random values.
2. Iterate the ESSENCE round function 32 times, storing all intermediate values of M and CV.
3. Apply the feed-forward function, and check if the resulting compression output corresponds to the target value X.
4. If not, apply only one additional round of ESSENCE, and return to the previous step.

For the sake of completeness, we should note that a pseudo-preimage may not exist for the target value X, in which case this algorithm (and any other algorithm) will fail. The algorithm will also fail if it cycles back to the initial (M, CV) without encountering the target value X. This problem can be solved by selecting a new values of (M, CV) and restarting the algorithm.

If we assume that the round ESSENCE round function evaluations dominates any of the other calculations, then Dunkelman's observation effectively speeds up the search for a pseudo-preimage by the number of rounds. In the conventional security model where the attack complexity is determined by the equivalent number of compression function evaluations, this means that the time complexity goes down when the number of rounds is increased. This observation was independently rediscovered by Wang et al. [33] when they analyzed the 3D Cipher.

5.2 Chaskey

We now describe a speed-up of exhaustive search using known plaintexts on the Chaskey MAC algorithm. In the observation, all messages consist of one block, and the tags are not truncated. As explained in Sect. 4.1, the Chaskey MAC function is then equivalent to an Even-Mansour block cipher $C = E_K(P) = K_1 \oplus \pi(P \oplus K \oplus K_1)$, where $K_1 = 2K$. From this, it follows that $C = E_K(P) = 2K \oplus \pi(P \oplus 3K)$.

Unfortunately the chosen-plaintext attack by Daemen [12] and the known-plaintext attacks by Biryukov and Wagner [9] and by Dunkelman et al. [15] do not seem to apply here. In those attacks, the adversary must be able to evaluate the permutation π on inputs of its choosing. This is not possible when we apply the technique explained in Sect. 5.1 to speed up exhaustive search.

The observation goes as follows. First, D known plaintexts (P_i, C_i) are obtained, which are encrypted under the secret key K. Each known plaintext can be transformed into another known plaintext (P'_i, C'_i), encrypted under an unknown key K'_i, where $P'_i = 0$, $C'_i = C_i \oplus 2 \cdot 3^{-1} P_i$ and $K \oplus 3^{-1} P_i$. It is easy to check from the key schedule that both (P_i, C_i) and (P'_i, C'_i) are valid plaintext-ciphertext pairs (under keys K and K' respectively), as input and output to the underlying permutation π remain the same. Then, store (C'_i, P_i) into a hash table, indexed by the first coordinate.

We then obtain T input-output pairs (x_j, y_j) of the permutation π. This is done using the same trick as for ESSENCE (Sect. 5.1): once one input-output pair of the permutation is calculated, any additional input-output pairs can be

obtained from only one additional round evaluation. If an all-zero plaintext were encrypted, (x_j, y_j) would correspond to a plaintext (P_j, C_j) under a known key K_j, where $P_j = 0$, $C_j = y \oplus 2 \cdot 3^{-1}x$ and $K_j = 3^{-1}x$. We therefore check for each (x_j, y_j) if C_j appears as the first element in the hash table. If this is the case for element i, we can check the guess $K = 3^{-1}(P_i \oplus x_j)$.

The probability that $C_i = C_j$ for any i, j is 2^{-128}. As our data contains $TD = 2^{128}$ pairs (C_i, C_j), we will find a match with a non-negligible probability of success. As each of the T permutation evaluations can be generated with an equivalent time complexity of T/k, where k is the number of rounds of the cipher, exhaustive search is effectively sped up with a factor of k. Of course, this analysis assumes that any other calculations besides the Chaskey round function evaluations are negligible. If this is not the case, the speed-up over exhaustive search will be smaller than k. Regardless of this assumption, the time complexity our observation will decrease when the number of rounds is increased.

All results in this section were experimentally verified on a small-scale variant of Chaskey.

5.3 PRIDE

PRIDE is a lightweight block cipher proposed by Albrecht et al. [1]. It processes a 64-bit plaintext P using a 128-bit key K, which is split into two 64-bit keys K_0 and K_1. Similar to PRINCE [11], PRIDE is also based on the FX-construction [22, 23] and also claims a security of $TD = 2^{128}$, where T and D are again, respectively, the time and data complexity of any cryptanalysis result. The PRIDE construction is illustrated in Fig. 5. Our description of PRIDE omits a bit permutation that is applied to the plaintext and ciphertext, and adds a diffusion layer to the last round, so that every round becomes identical. As such, our description of PRIDE is equivalent up to linear functions that are applied to the plaintext and ciphertext. An adversary can easily apply these functions to any plaintext-ciphertext, as they do not require knowledge of the key.

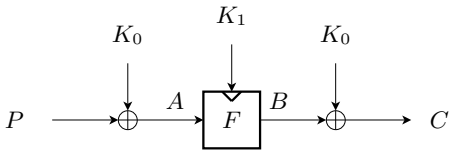

Fig. 5. The PRIDE block cipher, which processes 64-bit plaintext P using a 128-bit key K, which is split into two 64-bit keys K_0 and K_1. Key K_0 is used for prewhitening and postwhitening. The block cipher F is a 20-round Substitution-Permutation Network (SPN) that uses K_1.

PRIDE has a very simple key schedule. The designers claim no resistance against related-key attacks, and in fact note that PRIDE can be distinguished

trivially in this setting. For this reason, we will tackle the security bound $TD = 2^{128}$, by showing a speed-up over exhaustive search to go below this bound in the cryptanalysis model used in this paper. But first, we describe the round function and key schedule of PRIDE.

The block cipher F used inside PRIDE is a 20-round Substitution-Permutation Network (SPN). It consists of a subkey addition (the XOR of $f_i(K_1)$ for round i), followed by a substitution layer S and a permutation layer P, as shown in Fig. 6. The subkeys $f_i(K_1)$ are derived from K_1 as follows. Let $\|$ denote the concatenation of binary strings. First, we split K_1 into eight bytes: $K_1 = u_1\| \ldots \|u_8$. Then,

$$f_i(K_1) = u_1\|u_2 + 193i\|u_3\|u_4 + 165i\|u_5\|u_6 + 81i\|u_7\|u_8 + 197i \ , \qquad (14)$$

where all operations are performed modulo 2^8.

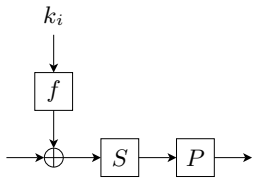

Fig. 6. One round of the PRIDE Substitution-Permutation Network (SPN), where S denotes the substitution layer and P refers to the permutation layer.

This results in the following pair of slid keys K_1, K_1':

$$K_1 = a_1\|a_2 \qquad \|a_3\|a_4 \qquad \|a_5\|a_6 \qquad \|a_7\|a_8 \qquad , \qquad (15)$$
$$K_1' = a_1\|a_2 + 193\|a_3\|a_4 + 165\|a_5\|a_6 + 81\|a_7\|a_8 + 197 \ , \qquad (16)$$

where again all calculations are performed modulo 2^8.

This observation could be used to construct a slide attack under related keys K_1, K_1'. However, related-key attacks do not violate the PRIDE design criteria. We can, however, use this pair of slid keys to speed up exhaustive search. This can be done as follows.

At the core of the attack, we will speed up the generation of (A, B)-values for the block cipher F, shown in Fig. 5. A full-round encryption is required to obtain one (A, B)-value under an adversary-chosen key K_1, after which each additional round will result in another (A', B'), where the relation between K_1 and K_1' is given by (15)-(16).

For the FX-construction used in PRIDE, the existing attacks proceed as follows: iterate over all keys K_1, and then apply an attack of the Even-Mansour block cipher. This strategy is used in the chosen-plaintext attack by Kilian and Rogaway [22, 23], and in the known plaintext attack for $D = 2^{n/2}$ by Biryukov and Wagner [9], where n denotes the block size. In a straightforward way, the

attack by Dunkelman et al. [15] on Even-Mansour also can be turned into a known plaintext attack on the FX-construction for any value of D.

However, none of these attacks seem to be applicable to speed up exhaustive search for PRIDE. This is because all of them require a loop over K_1, whereas our speed-up will generate (A, B)-values under random keys K_1. For this reason, we now introduce a new cryptanalysis result on the FX-construction. First, D known plaintexts (P_i, C_i) are collected, each encrypted under the secret key K. The values $(P_i \oplus C_i, P_i)$ are stored in a hash table, using the first coordinate as the index. Then for each of the T values $(A_j, B_j, K_{1,j})$, look up $A_j \oplus B_j$ in the hash table. If a match is found, we can check the key guess $K_0 = P_i \oplus A_j$, $K_1 = K_{1,j}$. To reach a non-negligible probability of success, this key must be checked with at least one other plaintext-ciphertext pair.

The probability that $P_i \oplus C_i = A_j \oplus B_j$ is 2^{-64}. As the data consists of $TD = 2^{128}$ pairs, we expect to find 2^{64} matches. When checking each of these against another plaintext-ciphertext, we expect that one candidate key will survive. Note that each of the T block cipher evaluations were generated with an equivalent time complexity of T/k. Therefore, this effectively speeds up exhaustive search by a factor of k.

We have experimentally verified all parts of this observation on a small-scale variant of PRIDE.

5.4 Password Hashing: PKCS #5's PBKDF1 Without a Salt

We now show how exhaustive search for password hashing algorithms can be sped up, in particular for a variant of PKCS #5's PBKDF1. Recall the PBKDF1 construction from Sect. 2: the password and salt are concatenated, after which a hash function is iteratively applied k times to obtain the password hash. The PKCS #5 standard recommends to use at least $k = 1000$, however in recent real-world applications, k is often ten or a hundred times higher. More specifically, in this section we will consider a variant of PBKDF1 that does not use a salt value. Our findings therefore do not apply to PBKDF1 directly, but may nevertheless lead to theoretical insights into its construction.

Let n be the output size of the hash function. Given one password hash, a classical preimage search requires about $T = 2^n$ evaluations of PBKDF1 to obtain a non-negligible success probability. If D password hashes are given, a straightforward calculation shows that recovering any of these requires about $T = 2^n/D$ PBKDF1 evaluations.

Now observe that the time complexity can be reduced by a factor of k. This is because evaluating one password guess requires k evaluations of the hash function used inside PBKDF1, but every additional guess has an additional cost of only one hash function evaluation. This effectively speeds up exhaustive search by a factor of k: given D password hashes, recovering any of them has a time complexity of $2^n/(D \cdot k)$. Yet again, this cryptanalysis result has a time complexity that decreases when the number of rounds is increased.

We constructed a small-scale variant of this password hashing function to verify our results experimentally.

6 Conclusion and Future Work

This paper focused on cryptanalysis results on symmetric-key primitives for which the complexity goes down when the number of rounds is increased. To the best of our knowledge, this paper provided the first comprehensive study of these types of observations. We investigated two classes of observations for iterated symmetric-key constructions: distinguishers and speed-ups of exhaustive search.

The distinguishers exploited the fact that iterating identical round functions always leads to security erosion. The expected number of fixed points increases when a random permutation is iterated, whereas iterating a random function reduces the expected number of image points. We explained how these effects also appear for non-ideal primitives, and used this to construct a new observation for the Chaskey MAC function.

When the underlying primitives are ideal, we recalled the security bounds of iterated primitives. When the number of iterations k is fixed, there exist attacks that match the bounds up to a constant factor. The bound for an iterated permutation is linear in k, whereas the bound for an iterated random function is quadratic in k. An interesting open problem to prove a linear bound in k for this case. This which would mean that the distinguishers in this paper provide no "real" speed-up when k increases, as the number of round evaluations would remain constant.

For ciphers that consist of identical rounds, we recalled an unpublished observation by Dunkelman, which was later rediscovered by Wang et al. They noted that one output has been evaluated, any subsequent input-output pairs can be obtained at the cost of only one additional round evaluation. We showed how to use this observation to speed up exhaustive search for several algorithms, including Chaskey, PRIDE, and an unsalted variant of the PKCS #5's PBKDF1 password hashing algorithm.

As we explained, the actual speed-up of exhaustive search is difficult to evaluate, but nevertheless decreases when the number of rounds is increased. As a prerequisite for our speed-ups over exhaustive search, developed two new variants of existing attacks: one on the Even-Mansour construction, and another on the FX-construction.

As our analysis of the PRIDE block cipher showed, our results also apply to a block cipher with non-identical round functions. An interesting direction for future work is to see if our speed-ups of exhaustive search can be applied to other ciphers with non-identical rounds. We hope that this will eventually lead to a better understanding of the security of various symmetric-key primitives, when small speed-ups over exhaustive search are taken into account.

Acknowledgments. Thanks to the anonymous reviewers for their useful comments and suggestions, as well as to Mridul Nandi and Somitra Kumar Sanadhya. We also thank Atul Luykx, Brice Minaud, Kazuhiko Minematsu, Yu Sasaki and Yannick Seurin for bringing related work to our attention. This work was supported in part by the Research Council KU Leuven: GOA TENSE (GOA/11/007),

by Research Fund KU Leuven, OT/13/071, and by the French Agence Nationale de la Recherche through the BLOC project under Contract ANR-11-INS-011. Nicky Mouha is supported by a Postdoctoral Fellowship from the Flemish Research Foundation (FWO-Vlaanderen), and by a JuMo grant from KU Leuven (JuMo/14/48CF). Jérémy Jean and Ivica Nikolić are supported by the Singapore National Research Foundation Fellowship 2012 (NRF-NRFF2012-06).

References

1. Albrecht, M.R., Driessen, B., Kavun, E.B., Leander, G., Paar, C., Yalçin, T.: Block Ciphers - Focus on the Linear Layer (feat. PRIDE). In Garay, J.A., Gennaro, R., eds.: Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I. Volume 8616 of LNCS., Springer (2014) 57–76
2. Bard, G.V., Ault, S.V., Courtois, N.T.: Statistics of Random Permutations and the Cryptanalysis of Periodic Block Ciphers. Cryptologia **36**(3) (2012) 240–262
3. 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. Volume 839 of LNCS., Springer (1994) 341–358
4. Bellare, M., Kilian, J., Rogaway, P.: The Security of the Cipher Block Chaining Message Authentication Code. J. Comput. Syst. Sci. **61**(3) (2000) 362–399
5. Bernstein, D.J.: A short proof of the unpredictability of cipher block chaining (January 2005) `http://cr.yp.to/antiforgery/easycbc-20050109.pdf`.
6. Biham, E., Chen, R.: Near-Collisions of SHA-0. In Franklin, M.K., ed.: Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings. Volume 3152 of LNCS., Springer (2004) 290–305
7. Biryukov, A.: Slide Attack. In van Tilborg, H.C.A., Jajodia, S., eds.: Encyclopedia of Cryptography and Security, 2nd Ed. Springer (2011) 1221–1222
8. Biryukov, A., Wagner, D.: Slide Attacks. In Knudsen, L.R., ed.: Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings. Volume 1636 of LNCS., Springer (1999) 245–259
9. Biryukov, A., Wagner, D.: Advanced Slide Attacks. In Preneel, B., ed.: Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. Volume 1807 of LNCS., Springer (2000) 589–606
10. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique Cryptanalysis of the Full AES. In Lee, D.H., Wang, X., eds.: ASIACRYPT. Volume 7073 of LNCS., Springer (2011) 344–371
11. 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., Yalçin, T.: PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract. In Wang, X., Sako, K., eds.: Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings. Volume 7658 of LNCS., Springer (2012) 208–225
12. Daemen, J.: Limitations of the Even-Mansour Construction. [21] 495–498

13. Dodis, Y., Ristenpart, T., Steinberger, J.P., Tessaro, S.: To Hash or Not to Hash Again? (In)Differentiability Results for H^2 and HMAC. In Safavi-Naini, R., Canetti, R., eds.: Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings. Volume 7417 of LNCS., Springer (2012) 348–366
14. Dunkelman, O.: Preimages for the ESSENCE Compression Function. Presented at the Rump Session of the ECRYPT II Hash Function Retreat (2009)
15. Dunkelman, O., Keller, N., Shamir, A.: Minimalism in Cryptography: The Even-Mansour Scheme Revisited. In Pointcheval, D., Johansson, T., eds.: EUROCRYPT. Volume 7237 of LNCS., Springer (2012) 336–354
16. Even, S., Mansour, Y.: A Construction of a Cipher From a Single Pseudorandom Permutation. [21] 210–224
17. Even, S., Mansour, Y.: A Construction of a Cipher from a Single Pseudorandom Permutation. J. Cryptology **10**(3) (1997) 151–162
18. Flajolet, P., Odlyzko, A.M.: Random Mapping Statistics. In Quisquater, J., Vandewalle, J., eds.: Advances in Cryptology - EUROCRYPT '89, Workshop on the Theory and Application of of Cryptographic Techniques, Houthalen, Belgium, April 10-13, 1989, Proceedings. Volume 434 of LNCS., Springer (1989) 329–354
19. Gligoroski, D., Klima, V.: Practical Consequences of the Aberration of Narrow-Pipe Hash Designs from Ideal Random Functions. In Gusev, M., Mitrevski, P., eds.: ICT Innovations 2010. Volume 83 of Communications in Computer and Information Science., Springer (2011) 81–93
20. Hong, J., Kim, W.: TMD-Tradeoff and State Entropy Loss Considerations of Streamcipher MICKEY. In Maitra, S., Madhavan, C.E.V., Venkatesan, R., eds.: Progress in Cryptology - INDOCRYPT 2005, 6th International Conference on Cryptology in India, Bangalore, India, December 10-12, 2005, Proceedings. Volume 3797 of LNCS., Springer (2005) 169–182
21. Imai, H., Rivest, R.L., Matsumoto, T., eds.: Advances in Cryptology - ASIACRYPT '91, International Conference on the Theory and Applications of Cryptology, Fujiyoshida, Japan, November 11-14, 1991, Proceedings. In Imai, H., Rivest, R.L., Matsumoto, T., eds.: ASIACRYPT. Volume 739 of LNCS., Springer (1993)
22. Kilian, J., Rogaway, P.: How to Protect DES Against Exhaustive Key Search. In Koblitz, N., ed.: Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings. Volume 1109 of LNCS., Springer (1996) 252–267
23. Kilian, J., Rogaway, P.: How to Protect DES Against Exhaustive Key Search (an Analysis of DESX). J. Cryptology **14**(1) (2001) 17–35
24. Leurent, G.: On Chaskey. Presented at the Early Symmetric Crypto - ESC 2015 (2015)
25. Ma, B., Li, B., Hao, R., Li, X.: Improved (Pseudo) Preimage Attacks on Reduced-Round GOST and Grøstl-256 and Studies on Several Truncation Patterns for AES-like Compression Functions. In Tanaka, K., Suga, Y., eds.: Advances in Information and Computer Security - 10th International Workshop on Security, IWSEC 2015, Nara, Japan, August 26-28, 2015, Proceedings. Volume 9241 of LNCS., Springer (2015) 79–96
26. Martin, J.W.: ESSENCE: A Family of Cryptographic Hashing Algorithms. Submission to the NIST SHA-3 Competition (Round 1) (2008) `http://csrc.nist.gov/groups/ST/hash/sha-3/Round1/submissions_rnd1.html`.
27. Minaud, B., Seurin, Y.: The Iterated Random Permutation Problem with Applications to Cascade Encryption. In Gennaro, R., Robshaw, M., eds.: Advances in

Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I. Volume 9215 of LNCS., Springer (2015) 351–367
28. Mouha, N., Mennink, B., Herrewege, A.V., Watanabe, D., Preneel, B., Verbauwhede, I.: Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers. In Joux, A., Youssef, A.M., eds.: Selected Areas in Cryptography - SAC 2014 - 21st International Conference, Montreal, QC, Canada, August 14-15, 2014, Revised Selected Papers. Volume 8781 of LNCS., Springer (2014) 306–323
29. Nandi, M.: A Simple and Unified Method of Proving Indistinguishability. In Barua, R., Lange, T., eds.: Progress in Cryptology - INDOCRYPT 2006, 7th International Conference on Cryptology in India, Kolkata, India, December 11-13, 2006, Proceedings. Volume 4329 of LNCS., Springer (2006) 317–334
30. Nandi, M.: A Simple Proof of a Distinguishing Bound of Iterated Uniform Random Permutation. Cryptology ePrint Archive, Report 2015/579 (2015) `http://eprint.iacr.org/`.
31. Röck, A.: Stream Ciphers Using a Random Update Function: Study of the Entropy of the Inner State. In Vaudenay, S., ed.: Progress in Cryptology - AFRICACRYPT 2008, First International Conference on Cryptology in Africa, Casablanca, Morocco, June 11-14, 2008. Proceedings. Volume 5023 of LNCS., Springer (2008) 258–275
32. Wagner, D., Goldberg, I.: Proofs of Security for the Unix Password Hashing Algorithm. In Okamoto, T., ed.: Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings. Volume 1976 of LNCS., Springer (2000) 560–572
33. Wang, L., Sasaki, Y., Sakiyama, K., Ohta, K.: Polynomial-Advantage Cryptanalysis of 3D Cipher and 3D-Based Hash Function. In Hanaoka, G., Yamauchi, T., eds.: Advances in Information and Computer Security - 7th International Workshop on Security, IWSEC 2012, Fukuoka, Japan, November 7-9, 2012. Proceedings. Volume 7631 of LNCS., Springer (2012) 170–181
34. Yao, F.F., Yin, Y.L.: Design and Analysis of Password-Based Key Derivation Functions. In Menezes, A., ed.: Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings. Volume 3376 of LNCS., Springer (2005) 245–261
35. Yao, F.F., Yin, Y.L.: Design and Analysis of Password-Based Key Derivation Functions. IEEE Transactions on Information Theory **51**(9) (2005) 3292–3297

A Collision Probability of a Random Function

In this section, we bound the probability p_c that at least one collision is found after q queries to a random function.

Theorem 4 (Random Function Collision Probability). *Let f be a random n-bit function. Assume $q^2 \leq 2^n/k$. The probability p_c that a collision is found after q queries is bounded by*

$$\frac{q^2}{2^{n+1}} \leq p_c \leq \frac{q^2}{2^n} \tag{17}$$

Proof. Let $N = 2^n$, and let P_j^i denote $i(i-1)...(i-j+1)$, which is the number of j-permutations of i distinct objects.

An upper bound on p_c is given by

$$p_c = 1 - \frac{P_q^N}{N^q} \ , \tag{18}$$

$$\leq 1 - \frac{(1 - \frac{q^2}{N})N^q}{N^q} \ , \tag{19}$$

$$\leq \frac{q^2}{N} \ . \tag{20}$$

A lower bound on p_c can be calculated as follows.

$$p_c = 1 - \frac{P_q^N}{N^q} \ , \tag{21}$$

$$\geq 1 - (1 - \frac{q}{N})^q \ , \tag{22}$$

$$\geq 1 - (1 - q\frac{q}{N} + \binom{q}{2}\frac{q^2}{N^2}) \ , \tag{23}$$

$$= q\frac{q}{N} - \binom{q}{2}\frac{q^2}{N^2} \ , \tag{24}$$

$$\geq \frac{q^2}{N} - \frac{q^4}{2N^2} \ , \tag{25}$$

$$\geq \frac{q^2}{N} - \frac{q^2}{2N} \ , \tag{26}$$

$$\geq \frac{q^2}{2N} \ . \tag{27}$$

□

B Iterated Random Function: a More Rigorous Bound

We now derive a more rigorous bound for Theorem 1 through a sequence of generous relaxations. First, observe that one particular iteration of f will generate uniform independent random outputs if the inputs are all distinct and different from all the other inputs in previous iterations, including the first iteration (which has inputs that are chosen by the adversary).

Let $N = 2^n$. The probability of this being satisfied at the i-th iteration when there has been no collision so far, is $P^{N-q}_{(i-1)q}/N^{(i-1)q}$. Thus,

$$\Pr\left[\mathcal{D}^{f^k}_{\text{coll}}(q) \to 1\right] \geq p_c + \frac{P^{N-q}_q}{N^q}p_c + \frac{P^{N-q}_{2q}}{N^{2q}}p_c + \cdots + \frac{P^{N-q}_{(k-1)q}}{N^{(k-1)q}}p_c \ , \tag{28}$$

$$\begin{aligned} &\geq p_c\left(1 + \left(1 - \frac{q^2}{N-q}\right)\left(1 - \frac{q}{N}\right)^q\right) \\ &\quad + \left(1 - \frac{4q^2}{N-q}\right)\left(1 - \frac{q}{N}\right)^{2q} + \ldots \\ &\quad + \left(1 - \frac{(k-1)^2q^2}{N-q}\right)\left(1 - \frac{q}{N}\right)^{(k-1)q} \ . \end{aligned} \tag{29}$$

Therefore,

$$\begin{aligned} \mathbf{Adv}_{F,f^k}(\mathcal{D}^{\mathsf{Q}}_{\text{coll}}(q)) \geq\ & p_c\left(1 - \frac{q^2}{N-q}\right)\left(1 - \frac{q^2}{N}\right) \\ &+ p_c\left(1 - \frac{4q^2}{N-q}\right)\left(1 - \frac{2q^2}{N}\right) + \ldots \\ &+ p_c\left(1 - \frac{(k-1)^2q^2}{N-q}\right)\left(1 - \frac{(k-1)q^2}{N}\right) \ . \end{aligned} \tag{30}$$

Now, $p_c \geq q^2/2N$ (see App. A), and for $q < N/2$, we have

$$\left(1 - \frac{i^2q^2}{N-q}\right)\left(1 - \frac{iq^2}{N}\right) \geq \left(1 - \frac{2i^2q^2}{N}\right)\left(1 - \frac{iq^2}{N}\right) \ , \tag{31}$$

$$\geq 1 - \frac{(2i^2+i)}{N}q^2 \ , \tag{32}$$

$$\geq 1 - \frac{3i^2}{N}q^2 \ . \tag{33}$$

Plugging these two inequalities in gives us

$$\mathbf{Adv}_{F,f^k}(\mathcal{D}^{\mathsf{Q}}_{\text{coll}}(q)) \geq \frac{q^2}{2N}\left(k - 1 - \frac{3q^2}{N}\left(1 + 4 + ... + (k-1)^2\right)\right) \ , \tag{34}$$

$$= \frac{q^2}{2N}k(k-1)(2k-1) \ , \tag{35}$$

$$\geq \frac{q^2}{2N}\left(k - \frac{q^2}{N}k^3\right) \ . \tag{36}$$

For $q^2 \leq N/2k^2$, we have

$$\mathbf{Adv}_{F,f^k}(\mathcal{D}^{\mathsf{Q}}_{\text{coll}}(q)) \geq \frac{kq^2}{4N} \ . \tag{37}$$