The Iterated Random Permutation Problem with Applications to Cascade Encryption

Brice Minaud and Yannick Seurin

ANSSI, Paris, France
brice.minaud@gmail.com,yannick.seurin@m4x.org

Abstract. We introduce and study the *iterated random permutation problem*, which asks how hard it is to distinguish, in a black-box way, the r-th power of a random permutation from a uniformly random permutation of a set of size N. We show that this requires $\Omega(N)$ queries (even for a two-sided, adaptive adversary). As a direct application of this result, we show that cascading a block cipher with the same key cannot degrade its security (as a pseudorandom permutation) more than negligibly.

Keywords: iterated random permutation problem, block cipher, pseudorandom permutation, cascade encryption

1 Introduction

A Simple Question. Assume that, as a cautious and slightly paranoid cryptographer, you are not at ease with using AES (say, with 256-bit keys) as is. Instead, you define the block cipher myAES as

$$\mathsf{myAES}(k, x) \stackrel{\text{def}}{=} \mathsf{AES}(k, \mathsf{AES}(k, x)),$$

that is, you encipher the plaintext x twice with the same key k, hoping that this will increase security. After all, this seems like a cheap, "black-box" way of doubling the number of rounds of AES-256, and it is heuristically well established that increasing the number of rounds of a cipher improves its resistance to various attacks. Another motivation is some contexts could be to slow down brute force attacks.[1] How can you be sure that the security of your new custom block cipher does not suddenly collapse, becoming *much worse* than the security of AES-256? This seems quite implausible, but can we hope to *formally prove that this cannot happen*?

[1] For example, the traditional UNIX password protection mechanism `crypt` uses DES iterated 25 times. However this is in a hashing context and hence not directly relevant to our work.

Cascade Encryption. This question is obviously related to what is called *cascade encryption* (or *multiple encryption*), i.e., self-composition of a block cipher. Given a block cipher E, the cascade of length r associated with E encrypts a message x as

$$E'_{k_1,\ldots,k_r}(x) \stackrel{\text{def}}{=} (E_{k_r} \circ \cdots \circ E_{k_1})(x).$$

Cascade encryption has been extensively studied in the setting where the keys $(k_1, \ldots, k_r)$ for the r calls to the underlying block cipher E are independent: there are results in the computational setting [LR86, Mye99, MT09, Tes11], in the information-theoretic setting (where only computationally unbounded adversaries are considered) [Vau98, Vau99, Vau03, MP04, MPR07, CPS14], and in the ideal cipher model (in the context of key-length extension) [ABCV98, BR06, GM09, Lee13, DLMS14]. In particular, it is known that cascading a given block cipher with independent keys is *security-amplifying*: if E is a (q, t, ε)-pseudorandom permutation[2] (PRP), then the r-fold cascade with independent keys for the r calls to E is a $(q', t', r\varepsilon^r)$-PRP [Tes11], with $q' \simeq q$ and $t' \simeq t$. In the information-theoretic setting, the following slightly weaker result has been shown: if E is a (q, ε)-PRP, then the r-fold cascade of E with independent keys is a $(q, 2^{r-1}\varepsilon^r)$-PRP [Vau98, Vau99].

On the other hand, virtually nothing is known regarding the security of cascade encryption when the keys used for each call to the underlying block cipher are *not* independent.[3] Not only is it not known whether this might amplify security (and indeed, proving even a tiny security amplification result for cascade encryption without increasing the total key-length would be a major breakthrough), but there is absolutely no guarantee that this might not in some cases *dramatically deteriorate* security.

Our Result. In this short paper, we prove that *cascading a block cipher with the same key cannot degrade its security beyond negligible*. By security, we mean the standard notion of (strong) pseudorandomness, defined as follows.

Definition 1 (Strong Pseudorandom Permutation (SPRP)). *Let E be a block cipher with key space K and message space S, and $\mathsf{Perm}(S)$ be the set of all permutations of S. Let $\mathcal{D}$ be a distinguisher with oracle access to a permutation and its inverse, and returning a single bit. The SPRP-advantage of $\mathcal{D}$ against E*

[2] A block cipher $E = (E_k)_{k \in K}$ with key space K is a (q, t, ε)-PRP if any adversary making at most q oracle queries and running in time at most t can distinguish E_k (for a random key k) from a uniformly random permutation with advantage at most ε. See also Definition 1 below.

[3] This setting is sometimes called *product encryption* [Sha49, MM93], *cascade encryption* being reserved to the case where the keys are independent. Yet since the wording *product encryption* carries the idea of iterating a very weak round function rather than a entire block cipher, we will not use it here.

is defined as

$$\mathbf{Adv}^{\mathrm{sprp}}_{E}(\mathcal{D}) = \Big| \Pr\left[P \leftarrow_{\$} \mathsf{Perm}(S) : \mathcal{D}^{P,P^{-1}} = 1\right] - \Pr\left[k \leftarrow_{\$} K : \mathcal{D}^{E_k,(E_k)^{-1}} = 1\right] \Big|.$$

For integers q and t, the SPRP-advantage of E is defined as

$$\mathbf{Adv}^{\mathrm{sprp}}_{E}(q,t) = \max_{\mathcal{D}} \mathbf{Adv}^{\mathrm{sprp}}_{E}(\mathcal{D}),$$

where the maximum is taken over all distinguishers making at most q oracle queries and running in time at most t. E is a (q,t,ε)-SPRP if $\mathbf{Adv}^{\mathrm{sprp}}_{E}(q,t) \leq \varepsilon$.

A block cipher is deemed secure if its SPRP-advantage is "small" for all "reasonable" parameters q and t. We show the following.

Theorem 1. *Let E be a block cipher with message space of size N, and $r > 0$ be an integer. Let E^r be the block cipher obtained by r-fold self-composition of E with the same key. (Note that E and E^r have the same message and key spaces.) Then*

$$\mathbf{Adv}^{\mathrm{sprp}}_{E^r}(q,t) \leq \mathbf{Adv}^{\mathrm{sprp}}_{E}(rq,t') + \frac{(2r+1)q}{N},$$

with $t' = \mathcal{O}(t)$.

Hence, cascade encryption with the same key does not hurt security beyond negligible, or, to phrase it more positively, it can only improve the security of a given PRP. Theorem 1 follows straightforwardly from a purely information-theoretic result that we now expose in details.

The Iterated Random Permutation Problem. Let S be a set of size $N > 0$, and let $\mathsf{Perm}(S)$ be the group of all permutations of S. For a permutation $P \in \mathsf{Perm}(S)$ and an integer $r \geq 1$, we denote P^r the r-fold self-composition of P. Consider an adversary (later called distinguisher) $\mathcal{D}$ having two-sided oracle access to an element $P \in \mathsf{Perm}(S)$: it can either query $P(x)$ and receive the corresponding image y, or query $P^{-1}(y)$ and receive the corresponding antecedent x. We assume that $\mathcal{D}$ makes at most q (adaptive queries) before outputting a bit b. The *iterated random permutation problem* asks how many queries q needs $\mathcal{D}$ to distinguish with a noticeable probability the following two situations:

1. a permutation P is drawn at random from $\mathsf{Perm}(S)$, and $\mathcal{D}$ is given oracle access to P and P^{-1};
2. a permutation P is drawn at random from $\mathsf{Perm}(S)$, and $\mathcal{D}$ is given oracle access to P^r and $(P^r)^{-1}$.

In other words, defining the advantage of $\mathcal{D}$ for the iterated random permutation problem as

$$\mathbf{Adv}_{P,P^r}(\mathcal{D}) = \Big| \Pr\left[P \leftarrow_{\$} \mathsf{Perm}(S) : \mathcal{D}^{P,P^{-1}} = 1\right] - \Pr\left[P \leftarrow_{\$} \mathsf{Perm}(S) : \mathcal{D}^{P^r,(P^r)^{-1}} = 1\right] \Big|,$$

and the best advantage at q queries as

$$\mathbf{Adv}_{P,P^r}(q) = \max_{\mathcal{D}} \mathbf{Adv}_{P,P^r}(\mathcal{D}),$$

where the maximum is taken over all distinguishers making at most q queries, we ask how q must grow with N for $\mathbf{Adv}_{P,P^r}(q)$ to be constant, say $\mathbf{Adv}_{P,P^r}(q) \geq 1/2$. We show that this requires $q = \Omega(N/r)$. More precisely, we have the following theorem.

Theorem 2. *For any integer q, one has*

$$\mathbf{Adv}_{P,P^r}(q) \leq \frac{(2r+1)q}{N}.$$

This theorem is proved in Section 2. Theorem 1 follows from Theorem 2 by a simple hybrid argument that we give for completeness.

Proof of Theorem 1. Let $\mathcal{D}$ be a distinguisher against the strong pseudorandomness of E^r making at most q oracle queries and running in time at most t. By definition,

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{sprp}}_{E^r}(\mathcal{D}) &= \Big| \Pr\left[P \leftarrow_{\$} \mathsf{Perm}(S) : \mathcal{D}^{P,P^{-1}} = 1\right] \\
&\qquad - \Pr\left[k \leftarrow_{\$} K : \mathcal{D}^{(E_k)^r,((E_k)^r)^{-1}} = 1\right] \Big| \\
&\leq \mathbf{Adv}_{P,P^r}(\mathcal{D}) + \mathbf{Adv}_{P^r,E^r}(\mathcal{D}) \\
&\leq \frac{(2r+1)q}{N} + \mathbf{Adv}_{P^r,E^r}(\mathcal{D}),
\end{aligned}$$

where the last inequality follows from Theorem 2 and where

$$\begin{aligned}
\mathbf{Adv}_{P^r,E^r}(\mathcal{D}) = \Big| &\Pr\left[P \leftarrow_{\$} \mathsf{Perm}(S) : \mathcal{D}^{P^r,(P^r)^{-1}} = 1\right] \\
&- \Pr\left[k \leftarrow_{\$} K : \mathcal{D}^{(E_k)^r,((E_k)^r)^{-1}} = 1\right] \Big|.
\end{aligned}$$

Consider the following distinguisher $\mathcal{D}'$ against the strong pseudorandomness of E. It has oracle access to some permutation oracle O (which is either a random permutation P or E_k for a random key k) and works as follows: it runs $\mathcal{D}$, answering each oracle query of $\mathcal{D}$ by querying its own oracle r times to return $O^r(x)$ for a direct query or $(O^r)^{-1}(y)$ for an inverse query, and outputting the same decision as $\mathcal{D}$. Clearly, the SPRP-advantage of $\mathcal{D}'$ against E is exactly $\mathbf{Adv}_{P^r,E^r}(\mathcal{D})$, and $\mathcal{D}'$ makes at most rq queries and runs in time $t' = \mathcal{O}(t)$. Hence

$$\mathbf{Adv}_{P^r,E^r}(\mathcal{D}) \leq \mathbf{Adv}^{\mathrm{sprp}}_{E}(rq, t'),$$

which concludes the proof. □

Remark 1. It can be noted in the proof above that even when $\mathcal{D}$ is non-adaptive (i.e., $\mathcal{D}$ chooses all its queries at the beginning of the security experiment and issues them all at once), $\mathcal{D}'$ seems to inherently have to query its oracle adaptively. And indeed, Theorem 1 does not extend to the non-adaptive variant of (strong) pseudorandomness. This can be seen from the following simple example: Consider a (single-key) Even-Mansour cipher [EM97], defined by $E_k(x) = k \oplus P(k \oplus x)$, where P is a public (efficiently computable and invertible) permutation. Assume that P is an involution (i.e., P^2 is the identity). Then, the block cipher E^2 obtained by composing E twice with the same key is highly insecure (even against non-adaptive adversaries making one single query) since it is equal to the identity for any key. On the other hand, modeling P as a public random involution oracle, it can be shown [DKS12] that E is secure against non-adaptive distinguishers making at most $q = 2^{n/2}$ encryption/decryption queries and evaluating P on at most $t = 2^{n/2}$ values.[4] This shows that, unlike what Theorem 1 ensures for adaptive security, cascading with the same key can completely ruin security against non-adaptive distinguishers.

We also exhibit a distinguisher whose advantage matches the upper bound of Theorem 2 (up to some constant term which depends on r), establishing the following lower bound.

Theorem 3. *For $q \leq N/r$, one has*

$$\mathbf{Adv}_{P,P^r}(q) \geq \frac{q}{2N} - \frac{r}{N}.$$

The adversary that we use to arrive at Theorem 3 simply picks a random message $x \in S$ and travels along the cycle on which this point lies, hoping to cycle back to x. Details of the analysis can be found in Section 3. A different attack, based on the search of a fixed point, has been analyzed by Courtois *et al.* [BAC12].

Perspectives. A natural question is whether it is possible to prove any kind of security amplification for cascade encryption with non-independent keys, which in its full generality would take the form

$$E'_k(x) = (E_{f_r(k)} \circ \cdots \circ E_{f_1(k)})(x),$$

where the f_i's are permutations of the key space of E.[5] However, in the particular scenario where the same key is reused (i.e., all f_i's are equal to the identity), this clearly requires additional assumptions on the underlying block cipher E, as indicated (again) by the simple example of a single-key Even-Mansour cipher

[4] But note that E can be distinguished from random by an *adaptive* adversary making two queries; namely, denoting O the adversary's oracle, it queries $y := O(x)$, $y' := O(y)$, and checks whether $y' = x$.

[5] Remark that, seeing E as a *round function* rather than a full-fledged block cipher and $(f_1, \ldots, f_r)$ as a key-schedule, this is exactly how most modern block ciphers are designed.

$E_k(x) = k \oplus P(k \oplus x)$, where P is a public (efficiently computable and invertible) permutation. There is a generic[6] attack on any block cipher of this class requiring $q = 2^{n/2}$ queries to the encryption/decryption oracle and $t = 2^{n/2}$ evaluations of the inner permutation P [Dae91, DKS12]. Note that for any $r > 1$, the r-fold cascade with the same key E^r is again a one-round single-key Even-Mansour cipher, with inner permutation P^r, so that it can be generically attacked with $q = 2^{n/2}$ queries to the encryption/decryption oracle and $t = r2^{n/2}$ evaluations of P. Hence, under the assumption that P is such that the best attack against E is the generic one, composition with the same key does not amplify the security of such a block cipher. The same argument applies if the f_i's are of the form $f_i(k) = k \oplus c_i$ for public constants c_i. Indeed, this yields again a one-round single-key Even-Mansour cipher with inner permutation

$$P'(x) = c_r \oplus P(c_r \oplus c_{r-1} \oplus P(c_{r-1} \oplus \cdots \oplus c_1 \oplus P(c_1 \oplus x) \cdots)).$$

Besides, slide attacks [BW99] show that iterating a truly weak cipher cannot make it arbitrarily strong, independently of the number of iterations. For instance, in the information-theoretic setting, if E is so weak that it can be distinguished from random using a single plaintext/ciphertext pair with advantage $1 - 2^{n/2}$, then E^r can be distinguished from random using $2^{n/2}$ queries with constant probability of success, regardless of the value of r.[7]

We leave open the problem whether it is possible to find assumptions on the block cipher E (e.g. resistance to related-key attacks, resistance to key-dependent messages attacks, etc.) sufficient to prove that cascading with non-independent keys is security amplifying.

2 Proof of the Main Result

In this section, we prove Theorem 2. We rely on the game-playing framework, and we assume some familiarity of the reader with this technique (see [Sho04, BR06] for more details).

In all the following, given a non-empty set S, we denote $\mathrm{Card}(S)$ the number of elements in S. Let $\mathsf{Cycl}(S)$ denote the set of *cyclic permutations* of S, i.e., the subset of $\mathsf{Perm}(S)$ consisting of permutations with a single cycle. Overall, we will consider the following four games:

[6] In this context, an attack is said to be generic if it only uses the inner permutation P as a black-box.

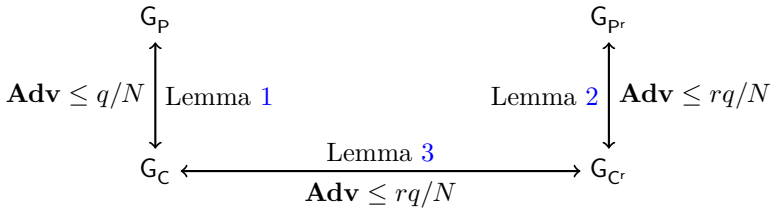
[7] Indeed, given $2^{n/2}$ plaintext/ciphertext pairs (p, c) for $(E_k)^r$, the distinguisher against E can be used to recognize so-called *slid pairs* $((p, c), (p', c'))$ satisfying $E_k(p) = p'$, and hence $E_k(c) = c'$. By the birthday paradox, such a slid pair is ensured to exist with constant probability when making $2^{n/2}$ random queries to $(E_k)^r$. Hence, the distinguisher between $(E_k)^r$ and a random permutation can count the number of plaintext/ciphertext pairs $((p, c), (p', c'))$, such that the distinguisher against E outputs 1 on both inputs (p, p') and (c, c'): the expected result is roughly 1 for a random permutation and 2 for $(E_k)^r$.

– G_{P}, which gives access to P and P^{-1} for $P \leftarrow_{\$} \mathsf{Perm}(S)$;
– $\mathsf{G}_{\mathsf{P}^r}$, which gives access to P^r and $(P^r)^{-1}$ for $P \leftarrow_{\$} \mathsf{Perm}(S)$;
– G_{C}, which gives access to C and C^{-1} for $C \leftarrow_{\$} \mathsf{Cycl}(S)$;
– $\mathsf{G}_{\mathsf{C}^r}$, which gives access to C^r and $(C^r)^{-1}$ for $C \leftarrow_{\$} \mathsf{Cycl}(S)$.

Each game provides two interfaces to the distinguisher, denoted Q and Q^{-1}, for querying the underlying permutation respectively in the direct and inverse direction. For example, the formal definition of G_{P} is:

Game G_{P}:
Initialization:
  $P \leftarrow_{\$} \mathsf{Perm}(S)$
procedure $\mathsf{Q}(x)$:
  return $P(x)$
procedure $\mathsf{Q}^{-1}(y)$:
  return $P^{-1}(y)$

For any games G, H, we write $\mathbf{Adv}_{\mathsf{G},\mathsf{H}}(q)$ to denote the maximal advantage attainable by distinguishers between G and H within q queries. We say that two games G and H are *equivalent* (within q queries) if $\mathbf{Adv}_{\mathsf{G},\mathsf{H}}(q) = 0$. Our goal is to bound $\mathbf{Adv}_{\mathsf{G}_{\mathsf{P}},\mathsf{G}_{\mathsf{P}^r}}(q)$. The layout of the proof is summarized by the following picture:



Lemma 1.

$$\mathbf{Adv}_{\mathsf{G}_{\mathsf{P}},\mathsf{G}_{\mathsf{C}}}(q) \le \frac{q}{N}.$$

Proof. We start with some useful definitions. A *partial permutation graph* (V, E) of size N is a directed graph (with loops allowed) with set of vertices V of size N and set of edges $E \subset V^2$, where each vertex has out- and in-degree 0 or 1. Given a partial permutation graph (V, E) containing no cycles and a vertex $z \in V$, the *source* of z, denoted $\mathsf{So}(z)$, is the unique $x \in V$ with in-degree 0 such that there is a path from x to z (with the convention that $\mathsf{So}(z) = z$ if z has in-degree 0), and the *sink* of z, denoted $\mathsf{Si}(z)$, is the unique $y \in V$ with out-degree 0 such that there is a path from z to y (with the convention that $\mathsf{Si}(z) = z$ if z has out-degree 0). The existence and uniqueness of $\mathsf{So}(z)$ and $\mathsf{Si}(z)$ when (V, E) is acyclic are straightforward to prove.

We consider *lazily sampled* versions of G_{P} and G_{C}. To describe the lazy sampling procedure, we assume that G_{P} internally maintains a partial permutation graph over $V = S$ (with initially no edge). This graph represents the current state of the sampling process. We let $E \subset S^2$ denote the (time-dependent) set of edges of the graph. We also let X be the set of vertices with out-degree 1 and Y be

the set of vertices with in-degree 1, these two sets being time-dependent as well. Slightly abusing notation, for $x \in X$, we denote $E(x)$ the unique $y \in S$ such that $(x, y) \in E$, and for $y \in Y$, we denote $E^{-1}(y)$ the unique $x \in S$ such that $(x, y) \in E$. The lazy sampled version of G_{P} is as follows:

Game $\mathsf{G}_{\mathsf{P}}^{\mathsf{lazy}}$:
Variables:
  Set of edges E, initially empty
procedure $\mathsf{Q}(x)$:
  if $x \notin X$ **then**
    $y \leftarrow_{\$} S \setminus Y$
    $E := E \cup \{(x, y)\}$
  return $E(x)$
procedure $\mathsf{Q}^{-1}(y)$:
  if $y \notin Y$ **then**
    $x \leftarrow_{\$} S \setminus X$
    $E := E \cup \{(x, y)\}$
  return $E^{-1}(y)$

Claim. G_{P} and $\mathsf{G}_{\mathsf{P}}^{\mathsf{lazy}}$ are equivalent (for any number q of queries).

Proof. This is a folklore result (see e.g. [BR06, Section 7.4]). Proving it amounts to showing, with the previous notation, that if $P \leftarrow_{\$} \mathsf{Perm}(S)$ agrees with a partial permutation graph (S, E), for $x \in S \setminus X$, then $P(x)$ is uniformly distributed over $S \setminus Y$. Equivalently, for any $x, x_1, \ldots, x_n$ pairwise distinct in S, and $y_A, y_B, y_1, \ldots, y_n$ pairwise distinct in S, we have

$$\begin{aligned} &\mathrm{Card}\{P \in \mathsf{Perm}(S) : P(x) = y_A, P(x_1) = y_1, \ldots, P(x_n) = y_n\} \\ =&\,\mathrm{Card}\{P \in \mathsf{Perm}(S) : P(x) = y_B, P(x_1) = y_1, \ldots, P(x_n) = y_n\}. \end{aligned}$$

To see this, observe that left-hand side composition with transposition $(y_A\ y_B)$ is a bijection between the two sets. The reasoning for an inverse query is similar. ■

Similarly, the lazy version of G_{C} is:

Game $\mathsf{G}_{\mathsf{C}}^{\mathsf{lazy}}$:
Variables:
  Set of edges E, initially empty
procedure $\mathsf{Q}(x)$:
  if $x \notin X$ **then**
    $y \leftarrow_{\$} S \setminus (Y \cup \{\mathsf{So}(x)\})$
    $E := E \cup \{(x, y)\}$
  return $E(x)$
procedure $\mathsf{Q}^{-1}(y)$:
  if $y \notin Y$ **then**
    $x \leftarrow_{\$} S \setminus (X \cup \{\mathsf{Si}(y)\})$
    $E := E \cup \{(x, y)\}$
  return $E^{-1}(y)$

Claim. G_{C} and $\mathsf{G}_{\mathsf{C}}^{\mathsf{lazy}}$ are equivalent (for any number q of queries).

Proof. Here, we must show that for any partial permutation graph (S, E) containing no cycle, with the previous notation and letting $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_n\}$, $x \in S \setminus X$ and $y_A, y_B \in S \setminus (Y \cup \{\mathsf{So}(x)\})$, we have

$$\begin{aligned} &\mathrm{Card}\{C \in \mathsf{Cycl}(S) : C(x) = y_A, C(x_1) = y_1, \ldots, C(x_n) = y_n\} \\ =&\,\mathrm{Card}\{C \in \mathsf{Cycl}(S) : C(x) = y_B, C(x_1) = y_1, \ldots, C(x_n) = y_n\}. \end{aligned}$$

Once again, we prove this equality by building a bijection between the two sets. This bijection is: $C \mapsto (y_A\, y_B) \circ C \circ (\mathsf{Si}(y_A)\ \mathsf{Si}(y_B))$, where $(a\ b)$ denotes the transposition swapping a and b. If C is seen as a cyclic graph, this bijection swaps the position of the longest chain starting from y_A in E with the longest chain starting from y_B. Thus it preserves the cyclic structure and is an involutive bijection between the two sets considered. The reasoning for an inverse query is similar. ■

From the lazy sampling versions of the games, it becomes apparent that $\mathsf{G}_{\mathsf{C}}^{\mathsf{lazy}}$ and $\mathsf{G}_{\mathsf{P}}^{\mathsf{lazy}}$ are identical, unless the event $[\mathsf{Q}(x) = \mathsf{So}(x)$ or $\mathsf{Q}^{-1}(y) = \mathsf{Si}(y)]$ happens for some query in $\mathsf{G}_{\mathsf{P}}^{\mathsf{lazy}}$. More precisely, we can rewrite $\mathsf{G}_{\mathsf{C}}^{\mathsf{lazy}}$ using a flag bad as follows:

Game $\mathsf{G}_{\mathsf{C}}^{\mathsf{lazy2}}$:
Variables:
  Set of edges E, initially empty
  $\mathsf{bad} \leftarrow$ **false**
procedure $\mathsf{Q}(x)$:
  if $x \notin X$ **then**
    $y \leftarrow_{\$} S \setminus Y$
    if $y = \mathsf{So}(x)$ **then**
      $\mathsf{bad} \leftarrow$ **true**
      $y \leftarrow_{\$} S \setminus (Y \cup \{\mathsf{So}(x)\})$
    $E := E \cup \{(x, y)\}$
  return $E(x)$

procedure $\mathsf{Q}^{-1}(y)$:
  if $y \notin Y$ **then**
    $x \leftarrow_{\$} S \setminus X$
    if $x = \mathsf{Si}(y)$ **then**
      $\mathsf{bad} \leftarrow$ **true**
      $x \leftarrow_{\$} S \setminus (X \cup \{\mathsf{Si}(y)\})$
    $E := E \cup \{(x, y)\}$
  return $E^{-1}(y)$

Clearly, $\mathsf{G}_{\mathsf{C}}^{\mathsf{lazy}}$ and $\mathsf{G}_{\mathsf{C}}^{\mathsf{lazy2}}$ are equivalent (this technique is called *resampling*, see [BR06, Section 7.2]). Moreover, $\mathsf{G}_{\mathsf{P}}^{\mathsf{lazy}}$ and $\mathsf{G}_{\mathsf{C}}^{\mathsf{lazy2}}$ are syntactically identical unless bad is set to **true**. By the fundamental lemma of game-playing (see [BR06, Lemma 2]), one has

$$\mathbf{Adv}_{\mathsf{G}_{\mathsf{P}}^{\mathsf{lazy}},\mathsf{G}_{\mathsf{C}}^{\mathsf{lazy2}}}(q) \leq \max_{\mathcal{D}} \Pr\left[\mathcal{D} \text{ sets } \mathsf{bad} \text{ to } \mathbf{true} \text{ in } \mathsf{G}_{\mathsf{C}}^{\mathsf{lazy2}}\right],$$

where the maximum is taken over all distinguishers making at most q queries.

For any distinguisher $\mathcal{D}$, the probability that bad is set to **true** at the i-th query of $\mathcal{D}$ in $\mathsf{G}_{\mathsf{C}}^{\mathsf{lazy2}}$ is exactly $1/(N-i)$. Hence, we finally obtain

$$\mathbf{Adv}_{\mathsf{G}_{\mathsf{P}},\mathsf{G}_{\mathsf{C}}}(q) = \mathbf{Adv}_{\mathsf{G}_{\mathsf{P}}^{\mathsf{lazy}},\mathsf{G}_{\mathsf{C}}^{\mathsf{lazy2}}}(q) \leq 1 - \prod_{i=0}^{q-1}\left(1 - \frac{1}{N-i}\right) = \frac{q}{N}. \qquad \square$$

Lemma 2.

$$\mathbf{Adv}_{\mathsf{G}_{\mathsf{P}^r},\mathsf{G}_{\mathsf{C}^r}}(q) \leq \frac{rq}{N}.$$

Proof. Any distinguisher between P^r and C^r can be used to distinguish between P and C at the cost of multiplying the number of queries by r. More formally, given a distinguisher $\mathcal{D}$ between P^r and C^r making at most q queries, consider

the distinguisher $\mathcal{D}'$ with oracle access to some permutation oracle O (which is either P or C) working as follows: it runs $\mathcal{D}$, answering each oracle query of $\mathcal{D}$ by querying its own oracle r times to return $O^r(x)$ for a direct query or $(O^r)^{-1}(y)$ for an inverse query, and outputting the same decision as $\mathcal{D}$. Clearly, the advantage of $\mathcal{D}'$ in distinguishing G_{P} and G_{C} is equal to the advantage of $\mathcal{D}$ in distinguishing $\mathsf{G}_{\mathsf{P}^r}$ and $\mathsf{G}_{\mathsf{C}^r}$, and $\mathcal{D}'$ makes at most rq queries if $\mathcal{D}$ makes at most q queries. Hence, by Lemma 1,

$$\mathbf{Adv}_{\mathsf{G}_{\mathsf{P}^r},\mathsf{G}_{\mathsf{C}^r}}(q) \leq \mathbf{Adv}_{\mathsf{G}_{\mathsf{P}},\mathsf{G}_{\mathsf{C}}}(rq) \leq \frac{rq}{N}. \qquad \square$$

Lemma 3.

$$\mathbf{Adv}_{\mathsf{G}_{\mathsf{C}},\mathsf{G}_{\mathsf{C}^r}}(q) \leq \frac{rq}{N}.$$

Proof. Let $d = \gcd(N, r)$. The key observation is that $\mathsf{G}_{\mathsf{C}^r}$ is equivalent to querying a random permutation with d cycles of equal length.[8] This follows from the fact that the mapping $C \mapsto C^r$ sends $\mathsf{Cycl}(S)$ onto the set of permutations with exactly d cycles of the same length, and that each such permutation has the same number of preimages in $\mathsf{Cycl}(S)$ under this mapping (the interested reader can refer to Appendix A where we prove this claim). In particular, if $d = 1$ (when N and r are coprime), the games G_{C} and $\mathsf{G}_{\mathsf{C}^r}$ are identical and we are done. If $d > 1$, we need to upper bound the advantage of an adversary distinguishing between a random permutation with a single cycle, and a random permutation with d cycles of equal length.

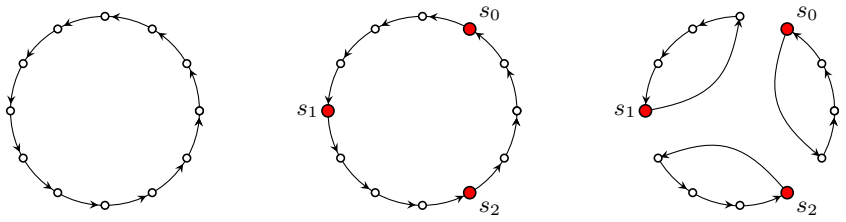
We now describe a new game $\mathsf{G}^*_{\mathsf{C}^r}$, which we claim is an equivalent description of $\mathsf{G}_{\mathsf{C}^r}$.

Game $\mathsf{G}^*_{\mathsf{C}^r}$:
Initialization:
  $C \leftarrow_{\$} \mathsf{Cycl}(S)$
  $s_0 \leftarrow_{\$} S$
  for $i < d$, $s_i = C^{N/d}(s_{i-1})$
procedure $\mathsf{Q}(x)$:
  if $x = s_i$ **for** some i **then**
    return $C(s_{(i-1) \bmod d})$
  else
    return $C(x)$
procedure $\mathsf{Q}^{-1}(y)$:
  if $y = C(s_i)$ **for** some i **then**
    return $s_{(i+1) \bmod d}$
  else
    return $C^{-1}(y)$

Intuitively, $\mathsf{G}^*_{\mathsf{C}^r}$ may be pictured as shown on Fig. 1.

We show in Appendix A that the sampling process underlying game $\mathsf{G}^*_{\mathsf{C}^r}$ is also equivalent to sampling a random permutation with d cycles of equal length. Meanwhile, we define the game $\mathsf{G}^*_{\mathsf{C}}$ as being identical to $\mathsf{G}^*_{\mathsf{C}^r}$, except queries $\mathsf{Q}(x)$ (resp. $\mathsf{Q}^{-1}(y)$) simply return $C(x)$ (resp. $C^{-1}(y)$): the s_i's play no special role. This corresponds to step 2 in the picture above. The point is that $\mathsf{G}^*_{\mathsf{C}}$ is clearly an equivalent description of G_{C} (since procedures Q and Q^{-1} are syntactically

[8] When we say "a random permutation with some property", more formally we mean "a uniformly random element among permutations with this property".



1. Pick a random cycle. 2. Pick a random point s_0. Split into d equal chains. 3. Redirect the s_i's to form d cycles.

Fig. 1. Representation of the game $\mathsf{G}^*_{\mathsf{C}^r}$.

the same in both games), while $\mathsf{G}^*_{\mathsf{C}^r}$ is an equivalent description of $\mathsf{G}_{\mathsf{C}^r}$ (indeed, by the two claims proved in Appendix A, they are both equivalent to querying a random permutation with d cycles of length N/d).

Thus $\mathbf{Adv}_{\mathsf{G}_{\mathsf{C}},\mathsf{G}_{\mathsf{C}^r}}(q) = \mathbf{Adv}_{\mathsf{G}^*_{\mathsf{C}},\mathsf{G}^*_{\mathsf{C}^r}}(q)$. The following claim completes the proof.

Claim.

$$\mathbf{Adv}_{\mathsf{G}^*_{\mathsf{C}},\mathsf{G}^*_{\mathsf{C}^r}}(q) \leq \frac{dq}{N}.$$

Proof. The only difference between $\mathsf{G}^*_{\mathsf{C}}$ and $\mathsf{G}^*_{\mathsf{C}^r}$ occurs when $\mathsf{Q}(s_i)$ is queried for some i (or $\mathsf{Q}^{-1}(C(s_i))$ for backward queries). So $\mathbf{Adv}_{\mathsf{G}_{\mathsf{C}},\mathsf{G}_{\mathsf{C}^r}}(q)$ is upper bounded by the advantage of an adversary playing the following game: she queries $\mathsf{G}^*_{\mathsf{C}}$, and wins iff one of the queries is an s_i (or $C(s_i)$ for a backward query). We now prove that the advantage of such an adversary is at most dq/N.

To show this, we give extra information to the adversary: we grant her full knowledge of the cycle C before queries begin. Clearly this can only increase her advantage. The point is that queries no longer provide any new information. Thus the game becomes equivalent to the adversary simply trying to guess one of the s_i's within q tries.

Notice that the position of the s_i's in the cycle C is essentially defined modulo $a = N/d$. Guessing the position of one of the s_i's in the cycle amounts to guessing a value modulo a. Thus the game is equivalent to guessing a value among a possibilities, within q tries. The advantage of an adversary in this game is:

$$1 - \prod_{i=0}^{q-1} \left(1 - \frac{1}{a-i}\right) = 1 - \frac{a-q}{a} = 1 - \frac{N-dq}{N} = \frac{dq}{N}.$$

By the previous reasoning, this is an upper bound for $\mathbf{Adv}_{\mathsf{G}^*_{\mathsf{C}},\mathsf{G}^*_{\mathsf{C}^r}}(q)$. ■

Thus, we have

$$\mathbf{Adv}_{\mathsf{G}_{\mathsf{C}},\mathsf{G}_{\mathsf{C}^r}}(q) = \mathbf{Adv}_{\mathsf{G}^*_{\mathsf{C}},\mathsf{G}^*_{\mathsf{C}^r}}(q) \leq \frac{dq}{N} \leq \frac{rq}{N}.$$ □

The proof of Theorem 2 is now complete. Combining Lemmas 1, 2, and 3, we obtain

$$\mathbf{Adv}_{\mathsf{G_P},\mathsf{G_{Pr}}}(q) \leq \mathbf{Adv}_{\mathsf{G_P},\mathsf{G_C}}(q) + \mathbf{Adv}_{\mathsf{G_C},\mathsf{G_{Cr}}}(q) + \mathbf{Adv}_{\mathsf{G_{Cr}},\mathsf{G_{Pr}}}(q) \leq \frac{(2r+1)q}{N}.$$

3 A Matching Attack

In this section, we describe a simple attack matching the bound in Theorem 2 within a constant factor, when the number of iterations r is constant. Our attack uses the following distinguisher $\mathcal{D}_{\text{cycle}}$ between $\mathsf{G_P}$ and $\mathsf{G_{Pr}}$. It makes q queries to the interface Q (corresponding to P in $\mathsf{G_P}$ and P^r in $\mathsf{G_{Pr}}$), ignoring Q^{-1}.

Distinguisher $\mathcal{D}^{\mathsf{Q}}_{\text{cycle}}(q)$
$s_0 \leftarrow_{\$} S$
for i in $\{0, \ldots, q-1\}$:
  $s_{i+1} \leftarrow \mathsf{Q}(s_i)$
end for
if all s_i's are distinct
  return 0
else
  return 1

Thus, $\mathcal{D}^{\mathsf{Q}}_{\text{cycle}}$ returns 1 iff the point $s_0 \leftarrow_{\$} S$ belongs to a cycle of length at most q. We have the following result (from which Theorem 3 is a direct application).

Lemma 4. *Assume $q \leq N/r$. Then*

$$C(r)\frac{q}{N} - \frac{r}{N} \leq \mathbf{Adv}_{\mathsf{G_P},\mathsf{G_{Pr}}}(\mathcal{D}_{\text{cycle}}) \leq C(r)\frac{q}{N} + \frac{r}{N} \quad \text{with } C(r) = \sum_{d|r} \frac{\phi(d)}{d} - 1$$

where $d|r$ denotes "d divides r", and ϕ is Euler's totient function. Moreover $C(r) \geq 1/2$ for $r \geq 2$.

Proof. By definition:

$$\mathbf{Adv}_{\mathsf{G_P},\mathsf{G_{Pr}}}(\mathcal{D}_{\text{cycle}}) = \Big| \Pr\left[P \leftarrow_{\$} \mathsf{Perm}(S) : \mathcal{D}^{P^r}_{\text{cycle}} = 1\right] - \Pr\left[P \leftarrow_{\$} \mathsf{Perm}(S) : \mathcal{D}^{P}_{\text{cycle}} = 1\right] \Big|.$$

We now set out to compute these two probabilities.

If we pick a random point in a random permutation on N points, and look at the length of the cycle it belongs to, all lengths $1 \leq k \leq N$ are equally probable. This is a standard result. It can be shown, for instance, using $\mathsf{G}^{\mathsf{lazy}}_{\mathsf{P}}$: if we choose $s_0 \leftarrow_{\$} S$ and query q times along a chain, and assume the first i queries do not

create a cycle, then the probability that the next query does is exactly $1/(N-i)$. Thus, the probability that s_0 belongs to a cycle of length k is precisely

$$\prod_{i=0}^{k-1}\left(1-\frac{1}{N-i}\right)\cdot\frac{1}{N-k}=\frac{1}{N}.$$

As a consequence, one has

$$\Pr\left[P\leftarrow_{\$}\mathsf{Perm}(S):\mathcal{D}^{P}_{\mathrm{cycle}}(q)=1\right]=\frac{q}{N}.$$

We now turn to the case where $\mathcal{D}_{\mathrm{cycle}}$ interacts with P^r instead of P. We let

$$p\stackrel{\mathrm{def}}{=}\Pr[P\leftarrow_{\$}\mathsf{Perm}(S):\mathcal{D}^{P^r}_{\mathrm{cycle}}(q)=1].$$

First, we recall two classic equalities regarding the totient function:

$$\sum_{d|n}\phi(d)=n\quad(1)\qquad\qquad\phi(n)=n\prod_{p|n,p\in\mathbb{P}}\left(1-\frac{1}{p}\right)\quad(2)$$

where $\mathbb{P}$ is the set of prime numbers. Now let k be the length of the cycle containing s_0. In P^r this cycle is broken up into $d=\gcd(k,r)$ cycles of length k/d. Hence $\mathcal{D}_{\mathrm{cycle}}(q)$ detects a cycle iff $q\geq k/d$. Since all lengths k are equally probable, we have

$$\begin{aligned}
p&=\frac{1}{N}\mathrm{Card}\left\{k\leq N:q\geq\frac{k}{\gcd(k,r)}\right\}\\
&=\frac{1}{N}\mathrm{Card}\{k\leq N:\exists d|r,\gcd(k,r)=d\text{ and }k\leq dq\}\\
&=\frac{1}{N}\mathrm{Card}\{k:\exists d|r,\gcd(k,r/d)=1\text{ and }k\leq\min(q,N/d)\}\qquad\text{with }k\leftarrow k/d\\
&=\frac{1}{N}\mathrm{Card}\{k:\exists d|r,\gcd(k,r/d)=1\text{ and }k\leq q\}\qquad\text{using }q\leq N/r\\
&=\frac{1}{N}\sum_{d|r}\mathrm{Card}\{k:\gcd(k,d)=1\text{ and }k\leq q\}\text{ since }d\mapsto r/d\text{ is 1-to-1 over }d|r\\
&\geq\frac{1}{N}\sum_{d|r}\mathrm{Card}\left\{k:\gcd(k,d)=1\text{ and }k\leq d\left\lfloor\frac{q}{d}\right\rfloor\right\}\\
&=\frac{1}{N}\sum_{d|r}\phi(d)\left\lfloor\frac{q}{d}\right\rfloor\\
&\geq\frac{1}{N}\sum_{d|r}\phi(d)\left(\frac{q}{d}-1\right)\\
&=\frac{q}{N}\sum_{d|r}\frac{\phi(d)}{d}-\frac{r}{N}\qquad\text{by (1).}
\end{aligned}$$

One can upper bound p in a very similar manner, and we obtain the main inequality.

Finally, we show that $C(r) \geq 1/2$ for $r \geq 2$. In fact it holds that for all r, $C(r) \geq 1 - 1/r$. To see this, observe that if $r > 2$ is not prime, we have:

$$\begin{aligned} C(r) &= \sum_{d|r} \frac{\phi(d)}{d} - 1 \\ &= \frac{\phi(1)}{1} + \sum_{d|r, 1<d<r} \frac{\phi(d)}{d} + \prod_{p|r, p\in\mathbb{P}} \left(1 - \frac{1}{p}\right) - 1 \quad \text{by (2)} \\ &\geq \sum_{d|r, 1<d<r} \frac{\phi(d)}{d} + \left(1 - \sum_{p|r, p\in\mathbb{P}} \frac{1}{p}\right) \quad \text{by the union bound} \\ &\geq 1 \quad \text{since } \{p \in \mathbb{P} : p|r\} \subseteq \{1 < d < r : d|r\}. \end{aligned}$$

On the other hand, if r is prime then $C(r) = 1 - 1/r$, hence this is the lower bound. □

Corollary 1. *For constant r, the best distinguisher between G_{P} and $\mathsf{G}_{\mathsf{P}^r}$ has advantage $\Theta(q/N)$ as $N \to \infty$ and $q \leq N$ is any function of N.*

Proof. Theorem 2 shows that the advantage is $\mathcal{O}(q/N)$. Theorem 3 shows that it is $\Omega(q/N)$ if $q \leq N/r$. On the other hand if $q > N/r$, as the advantage can only increase with q, it is at least $C(r)\frac{N/r}{N} + o(1) \geq \frac{1}{2r} + o(1) = \Omega(1) = \Omega(q/N)$. Hence overall the advantage is $\Theta(q/N)$. □

For concreteness, if $r = 2$, Theorem 3 exhibits a distinguisher with advantage $0.5q/N$ (under the assumption $q < N/2$), while the main theorem upper bounds the advantage of any such distinguisher by $5q/N$. Note that if r is not constant, the behavior is more complex; informally, only cycles whose length is not coprime with r are affected by the transformation $P \mapsto P^r$. In particular, if r is prime and $r > N$, $P \mapsto P^r$ is a permutation of $\mathsf{Perm}(S)$ and G_{P} is indistinguishable from $\mathsf{G}_{\mathsf{P}^r}$.

The problem of finding a tight bound for variable r is interesting from a purely theoretical standpoint, although we do not know of a situation where such a result would be applicable.

References

[ABCV98] William Aiello, Mihir Bellare, Giovanni Di Crescenzo, and Ramarathnam Venkatesan. Security Amplification by Composition: The Case of Doubly-Iterated, Ideal Ciphers. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO '98*, volume 1462 of *LNCS*, pages 390–407. Springer, 1998.

[BAC12] Gregory V. Bard, Shaun Van Ault, and Nicolas T. Courtois. Statistics of Random Permutations and the Cryptanalysis of Periodic Block Ciphers. *Cryptologia*, 36(3):240–262, 2012.

[BR06] Mihir Bellare and Phillip Rogaway. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, 2006. Full version available at http://eprint.iacr.org/2004/331.

[BW99] Alex Biryukov and David Wagner. Slide Attacks. In Lars R. Knudsen, editor, *Fast Software Encryption - FSE '99*, volume 1636 of *LNCS*, pages 245–259. Springer, 1999.

[CPS14] Benoit Cogliati, Jacques Patarin, and Yannick Seurin. Security Amplification for the Composition of Block Ciphers: Simpler Proofs and New Results. In Antoine Joux and Amr M. Youssef, editors, *Selected Areas in Cryptography - SAC 2014*, volume 8781 of *LNCS*, pages 129–146. Springer, 2014.

[Dae91] Joan Daemen. Limitations of the Even-Mansour Construction. In Hideki Imai, Ronald L. Rivest, and Tsutomu Matsumoto, editors, *Advances in Cryptology - ASIACRYPT '91*, volume 739 of *LNCS*, pages 495–498. Springer, 1991.

[DKS12] Orr Dunkelman, Nathan Keller, and Adi Shamir. Minimalism in Cryptography: The Even-Mansour Scheme Revisited. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 336–354. Springer, 2012.

[DLMS14] Yuanxi Dai, Jooyoung Lee, Bart Mennink, and John P. Steinberger. The Security of Multiple Encryption in the Ideal Cipher Model. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 (Proceedings, Part I)*, volume 8616 of *LNCS*, pages 20–38. Springer, 2014.

[EM97] Shimon Even and Yishay Mansour. A Construction of a Cipher from a Single Pseudorandom Permutation. *Journal of Cryptology*, 10(3):151–162, 1997.

[GM09] Peter Gazi and Ueli M. Maurer. Cascade Encryption Revisited. In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 37–51. Springer, 2009.

[Lee13] Jooyoung Lee. Towards Key-Length Extension with Optimal Security: Cascade Encryption and Xor-cascade Encryption. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 405–425. Springer, 2013.

[LR86] Michael Luby and Charles Rackoff. Pseudo-random Permutation Generators and Cryptographic Composition. In *Symposium on Theory of Computing - STOC '86*, pages 356–363. ACM, 1986.

[MM93] Ueli M. Maurer and James L. Massey. Cascade Ciphers: The Importance of Being First. 6(1):55–61, 1993.

[MP04] Ueli M. Maurer and Krzysztof Pietrzak. Composition of Random Systems: When Two Weak Make One Strong. In Moni Naor, editor, *Theory of Cryptography Conference - TCC 2004*, volume 2951 of *LNCS*, pages 410–427. Springer, 2004.

[MPR07] Ueli M. Maurer, Krzysztof Pietrzak, and Renato Renner. Indistinguishability Amplification. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, volume 4622 of *LNCS*, pages 130–149. Springer, 2007. Full version available at http://eprint.iacr.org/2006/456.

[MT09] Ueli M. Maurer and Stefano Tessaro. Computational Indistinguishability Amplification: Tight Product Theorems for System Composition. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *LNCS*, pages 355–373. Springer, 2009.

[Mye99] Steven Myers. *On the Development of Block-Ciphers and Pseudo-Random Function Generators Using the Composition and XOR Operators*. PhD thesis, University of Toronto, 1999.

[Sha49] Claude Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[Sho04] Victor Shoup. Sequences of Games: A Tool for Taming Complexity in Security Proofs. IACR ePrint Archive, Report 2004/332, 2004. Available at http://eprint.iacr.org/2004/332.pdf.

[Tes11] Stefano Tessaro. Security Amplification for the Cascade of Arbitrarily Weak PRPs: Tight Bounds via the Interactive Hardcore Lemma. In Yuval Ishai, editor, *Theory of Cryptography - TCC 2011*, volume 6597 of *LNCS*, pages 37–54. Springer, 2011.

[Vau98] Serge Vaudenay. Provable Security for Block Ciphers by Decorrelation. In Michel Morvan, Christoph Meinel, and Daniel Krob, editors, *Symposium on Theoretical Aspects of Computer Science, STACS 98*, volume 1373 of *LNCS*, pages 249–275. Springer, 1998.

[Vau99] Serge Vaudenay. Adaptive-Attack Norm for Decorrelation and Super-Pseudorandomness. In Howard M. Heys and Carlisle M. Adams, editors, *Selected Areas in Cryptography - SAC '99*, volume 1758 of *LNCS*, pages 49–61. Springer, 1999.

[Vau03] Serge Vaudenay. Decorrelation: A Theory for Block Cipher Security. *Journal of Cryptology*, 16(4):249–286, 2003.

A Omitted Proofs

We prove here two claims that we used in the proof of Lemma 3. We denote $\mathsf{Cycl}_d(S)$ the set of permutations of S with exactly d cycles of length N/d (note that $\mathsf{Cycl}(S) = \mathsf{Cycl}_1(S)$).

Claim. Let S be a set of size N, $r \geq 1$ be an integer, and $d = \gcd(N, r)$. Let ϕ be the mapping

$$\begin{aligned} \phi : \mathsf{Cycl}(S) &\to \mathsf{Perm}(S) \\ P &\mapsto P^r. \end{aligned}$$

Then $\phi(\mathsf{Cycl}(S)) = \mathsf{Cycl}_d(S)$ and all permutations in $\mathsf{Cycl}_d(S)$ have exactly the same number of preimages by ϕ.

Proof. First, we show that for any $C \in \mathsf{Cycl}(S)$, $\phi(C) \in \mathsf{Cycl}_d(S)$. Let $a = N/d$. Denote

$$C = (x_1\ x_2\ \cdots\ x_N).$$

Then it is easy to see that C^r is the product of d disjoint cycles C_i, $1 \leq i \leq d$, with

$$C_i = (x_i\ x_{(i+r) \bmod N}\ x_{(i+2r) \bmod N} \cdots\ x_{(i+(a-1)r) \bmod N}).$$

For $A \in \mathsf{Cycl}_d(S)$, we denote $\phi^{-1}(A)$ the set of preimages of A by ϕ. We now show that for any $A, B \in \mathsf{Cycl}_d(S)$, $|\phi^{-1}(A)| = |\phi^{-1}(B)|$. For $P \in \mathsf{Perm}(S)$, we denote f_P the conjugation by P, namely $f_P(Q) = P \circ Q \circ P^{-1}$. Since A and B have the same cycle structure, they belong to the same conjugacy class, i.e., there exists a permutation P such that $f_P(A) = B$. Hence, for any $C \in \phi^{-1}(A)$, $f_P(C) \in \mathsf{Cycl}(S)$ since conjugation preserves the cycle structure, and one has

$$f_P(C)^r = (P \circ C \circ P^{-1})^r = P \circ C^r \circ P^{-1} = P \circ A \circ P^{-1} = B.$$

This implies that $f_P(\phi^{-1}(A)) \subseteq \phi^{-1}(B)$, and hence $|\phi^{-1}(A)| \leq |\phi^{-1}(B)|$ since f_P is one-to-one. By symmetry, $|\phi^{-1}(A)| = |\phi^{-1}(B)|$. ■

Claim. Let ψ denote the mapping which sends a pair $(C, s_0) \in \mathsf{Cycl}(S) \times S$ to the permutation defined by game $\mathsf{G}^*_{\mathsf{C}^r}$. Then $\psi(\mathsf{Cycl}(S) \times S) = \mathsf{Cycl}_d(S)$ and all permutations in $\mathsf{Cycl}_d(S)$ have exactly the same number of preimages by ψ.

Proof. The fact that $\psi(C, s_0) \in \mathsf{Cycl}_d(S)$ for any $(C, s_0) \in \mathsf{Cycl}(S) \times S$ is clear. We now show that for any $A, B \in \mathsf{Cycl}_d(S)$, $|\psi^{-1}(A)| = |\psi^{-1}(B)|$. As in the previous claim, there exists a permutation P such that $f_P(A) = B$. We show that for any $(C, s_0) \in \psi^{-1}(A)$, $(f_P(C), P(s_0)) \in \psi^{-1}(B)$. First, $f_P(C) \in \mathsf{Cycl}(S)$ since conjugation preserves the cycle structure. For $i < d$, let $s_i = C^{iN/d}(s_0)$. By definition of ψ, $A(s_i) = C(s_{(i-1) \bmod d})$ and $A(x) = C(x)$ for $x \notin \{s_0, \ldots, s_{d-1}\}$. Let $s'_0 = P(s_0)$ and for $i < d$, $s'_i = f_P(C)^{iN/d}(s'_0) = P(s_i)$. Then

$$\begin{aligned}\psi(f_P(C), P(s_0))(s'_i) &= f_P(C)(s'_{(i-1) \bmod d}) \\ &= P \circ C(s_{(i-1) \bmod d}) = P \circ A(s_i) = f_P(A)(s'_i) = B(s'_i),\end{aligned}$$

and for $x \notin \{s'_0, \ldots, s'_{d-1}\}$, since $P^{-1}(x) \notin \{s_0, \ldots, s_{d-1}\}$, one has

$$\psi(f_P(C), P(s_0))(x) = f_P(C)(x) = P \circ C \circ P^{-1}(x) = P \circ A \circ P^{-1}(x) = B(x),$$

which shows that $\psi(f_P(C), P(s_0)) = B$. Hence, the image of $\psi^{-1}(A)$ by the one-to-one mapping $(C, s_0) \mapsto (f_P(C), P(s_0))$ is a subset of $\psi^{-1}(B)$, thus $|\psi^{-1}(A)| \leq |\psi^{-1}(B)|$. By symmetry, $|\psi^{-1}(A)| = |\psi^{-1}(B)|$. ■