Turning Online Ciphers Off

Elena Andreeva[1], Guy Barwell[2], Dan Page[2], and Martijn Stam[2]

[1] Department of Electrical Engineering, ESAT/COSIC, KU Leuven, Belgium.
elena.andreeva@esat.kuleuven.be ,
[2] Department of Computer Science, University of Bristol,
Merchant Venturers Building, Woodland Road,
Bristol, BS8 1UB, United Kingdom.
{guy.barwell, daniel.page, martijn.stam}@bris.ac.uk

Abstract. CAESAR has caused a heated discussion regarding the merits of one-pass encryption and online ciphers. The latter is a keyed, length preserving function which outputs ciphertext blocks as soon as the respective plaintext block is received. The immediacy of an online cipher gives a clear performance advantage, yet it comes at a price. Since ciphertext blocks cannot depend on later plaintext blocks, diffusion and hence security is limited. We show how one can attain the best of both worlds by providing provably secure constructions, achieving full cipher security, based on applying an online cipher and reordering blocks.
Explicitly, we show that with just two calls to the online cipher, security up to the birthday bound is both attainable and maximal. Moreover, we demonstrate that three calls to the online cipher suffice to obtain beyond birthday bound security, and (for suitably long messages) arbitrarily strong security. As part of our investigation, we extend an observation by Rogaway and Zhang, highlighting the close relationship between online ciphers and tweakable blockciphers with variable-length tweaks.

Keywords: beyond birthday bound, online ciphers, modes of operation, provable security, pseudorandom permutation, tweakable blockcipher

Table of Contents

1 Introduction

Modern understanding of symmetric cryptology has come a long way from a straightforward adaptation (cf. [24, Def. 3.30]) of the seminal definitions of probabilistic [public key] encryption [18]. Both authenticated encryption and variable input length ciphers have emerged as noteworthy primitives. From an efficiency perspective, a scheme is ideally one-pass and online, outputting ciphertexts as plaintext comes in. For nonce-based authenticated encryption, online schemes do not suffer in security, as long as nonces are indeed unique (and decryption of invalid ciphertexts only produces a single error message [2, 8, 21]). Once nonces do repeat, prefix patterns start leaking; the same is true for online ciphers. Two pass schemes become a necessity. This raises the question how easily one can boost the security of an online scheme. In this paper, we concentrate on turning online ciphers into fully fledged ciphers using only two or three passes (depending on the desired security level).

The original goal of cryptography was data confidentiality. From a modern perspective, this is addressed by authenticated encryption (AE), which provides both confidentiality and integrity (including of associated data [31]). Modern AE schemes are deterministic and rely on a nonce to ensure that encrypting the same message twice produces two unrelated ciphertexts: as long as nonces do not repeat, security is guaranteed. Once nonces do repeat, leaking plaintext equality patterns is inevitable, but for many schemes the damage is much worse [22, 25]. The security goal of *misuse resistant* AE [32] considers whether and how the security of an AE scheme degrades when a nonce is no longer used just once. There are many ways to construct authenticated encryption schemes [7, 29], but the number of options reduces drastically when misuse resistance is required. One approach is the encode-then-encipher (or pad-then-encipher) paradigm [6, 32, 36], where (public) redundancy is added to the message before it is being enciphered using a variable input length strong pseudorandom permutation ($\pm$PRP cipher).

Variable input length ciphers (either $\pm\mathsf{prp}$ or prp secure) are interesting in their own right, especially in scenarios where encryption has to occur *in situ* [5]. One example is adding confidentiality to an existing networking standard, where packet sizes are fixed and the expansion implicit when using authenticated encryption cannot be afforded; another application is disk encryption (possibly using tweaks so sectors can still be accessed independently).

A prp cipher will yield completely different ciphertexts if there is any difference between plaintexts. This forces at least two-passes, one to read the plaintext and one to write the ciphertext. Once the length of the input increases, a one-pass or online cipher might strike a better balance between the conflicting goals of efficiency and robust security. An *online cipher* [4] is a variable input length keyed permutation based on a blockcipher that outputs a ciphertext block as soon as it receives a plaintext block (but still based on all preceding plaintext blocks). In other words, it allows instant processing of plaintext and outputting ciphertext on the fly. Since online ciphers cannot be prp secure, relaxed security notions exist that capture "best possible" security. Online ciphers play a key role in achieving a similarly relaxed notion of online authenticated encryption with graceful security degradation against nonce reuse [15].

We believe there are many scenarios where an online cipher's security limitations are outweighed by their efficiency, but at the same time there will be situations where full cipher security is paramount. One could create tailor-made solutions for each of the primitives, but often it is more desirable to share components. This could be solved by using two modes of operation on say AES, but we imagine an environment where black-box use of an online cipher is already available (such as by API), and we are tasked to create a true cipher based on the access to the online cipher only.[3]

Our contribution. We consider schemes formed by composing calls to an online cipher around a simple (publicly known) mixing layer, and aim to minimize the number of calls made to the online cipher (Def. 4). We restrict the mixing layer to be blockwise-linear (defined in Sec. 2), with particular focus on linear layers that simply reorder the blocks, since these can be implemented most efficiently.

[3] Obviously if one would have direct access to whatever primitive underlies the online cipher, more efficient (and known) variable input length ciphers could be constructed. Nonetheless, minimizing the number of calls as imposed by an API is a metric that has previously shown its worth in the context of authenticated encryption [10].

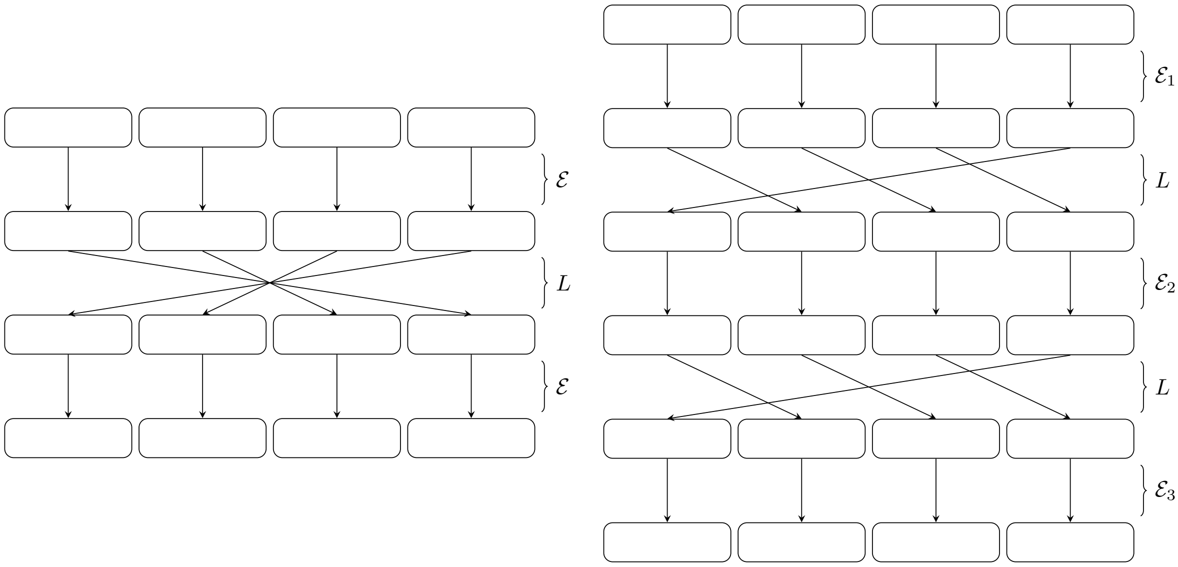


Fig. 1. Examples of the construction. On the left is the two layer reversing scheme, and on the right the three layer right shift instantiated with independent ciphers.

Fig. 1 highlights two typical constructions under consideration. Note that neither reversing the blocks nor cycling the final block to the front in itself is novel: both ideas have been suggested in one way or the other using more traditional IV-based encryption schemes [5] or in the context of key-wrap schemes [14]. Our novelty resides in using an online cipher as underlying primitive, and what we are able to prove as a result. Table 1 provides a summary of our results. The security bounds are simplifications of those in the paper, compromising tightness in favour of clarity (for stricter bounds please refer to the relevant theorems). As a boon, we describe an explicit correspondence between tweakable blockciphers and online ciphers (Thm. 7), extending an observation by Rogaway and Zhang [34].

We prove that only two calls to the online cipher are required to achieve security up to the birthday bound, in terms of indistinguishability from a random permutation. Even when the adversary is allowed to make inverse queries, this can be achieved by using a linear layer that reverses the message (Thm. 11). If one is not concerned about an adversary making queries of the construction's inverse, it suffices for the linear layer to move the final block to the start (Thm. 17), as long as the map remains invertible.

If one requires security beyond the birthday bound (something most symmetric schemes do not provide), one must make at least three queries to the online cipher (Thm. 10). Perhaps surprisingly, we find that three suffices: security is provided up until almost the blocksize by the construction making three calls to the online cipher around two calls to a linear layer that reverses the message (Thm. 13). We are not aware of any matching attacks against this construction, however the longer messages are, the higher the security guarantee the scheme provides (Thm. 14), defeating a number of common attack strategies. If one does not mind about inverse queries, simply moving the final block to the start suffices (Thm. 19).

Applications. We provide a concrete way for converting an online cipher into a true cipher. Our methods can trivially be extended to form tweakable ciphers from tweakable online ciphers with the tweaks and bounds of the non-tweak setting, or indeed from a non-tweakable online cipher to a tweakable cipher. There exist many ways to turn a true cipher into a secure AE scheme (e.g. Encode-then-Encipher [6,36]). Moreover, Hoang et al. demonstrate that with a tweakable cipher one may achieve the even stronger goal of Robust Authenticated Encryption [21, Thm. 5] (itself implying full misuse-resistant security [32]).

Construction			Input	Security			
Linear Layer	Cipher calls	Goal	Lengths	Advantage	Proof	Tight?	
1 block Right-shift	2	PRP	Any	$\frac{3}{2}q^2/2^n$	Thm. 17	Thm. 10	
1 block Right-shift	3	PRP	Any	$q^2/2^{2n}$	Thm. 19	Lem. 18	(†)
Blockwise Reverse	2	±PRP	Any	$4q^2/2^n$	Thm. 11	Thm. 10	
Blockwise Reverse	3	±PRP	Any	$nq/2^n$	Thm. 13		(†) (∗)
			$m \geq 2kn$	$4q^2/2^{kn}$	Thm. 14		

Table 1. Simplified upper bounds on adversarial advantage against our constructions, where a small advantage implies a secure scheme. Results are paramaterised by the maximum number of queries q, the blocksize n and an arbitrary integer $k > 0$. The input length column provides any limitations on the input length m (in bits). A bound is "tight" if there exists an attack that asymptotically (in q, n) matches the security bound. Notes: (†) Security proof requires independent ciphers (see note in section 1); (∗) The simplification requires $n \geq 4$ (full result does not, see theorem).

Incorporating our results plugs the gap to turn a secure online cipher into an Authenticated Encryption scheme meeting the strongest of security bounds.

For example, when instantiated with POE[AES4], the recommended online cipher of the POET CAESAR candidate [1], one can use the two layer reversal structure to build a RAE (or MRAE) scheme, against which the adversarial advantage is less than $l^2/2^{111}$, where the total number of blocks queried is l. This means the scheme is secure until at least 2^{50} blocks are queried, with the bound dominated by the difference between AES4 and an ideal AXU. Note that, although the RAE game allows decryption leakage from the final buffer, it would not be secure to leak between the online cipher calls: to do this would provision an indifferentiability attack on the construction, something that can be readily constructed (see App. A).

This further reinforces the assertion that online ciphers are an interesting object, meriting future study. As discussed by Hoang et al., there exist times when a user has to compromise security in return for other savings [21, Sec. 1: "Ciphertext Expansion"] such as reduced power consumption. Our construction provides a method by which real world devices may do this without requiring multiple primitives. This reduces the number of possible failure points and saves chip area. When optimal security is not required, the online cipher may be used directly. However, when security must be maximised, one may instead use our construction to provide Robust AE security.

Related work. The concept of an online cipher was first studied by Bellare et al. [4], providing the initial security definitions, against which they investigate some CBC variants. The security definitions and their relationships were developed through a number of papers [9, 16, 17, 23]. Later, Rogaway and Zhang exposed the close relationship between tweakable blockciphers and online ciphers [34], an observation that has since been exploited by others, yielding several explicit constructions (e.g. McOE [15]). There now exist a wide range of online cipher constructions, such as COPE [3], POE [1] and ELmE [12], the majority of which achieve birthday bound security. We are not aware of any online ciphers whose security might extend beyond the birthday bound.

One of our two-layer constructions is similar to CMC-core [20], but by building around an online cipher rather than CBC mode we achieve provable security without the need for a masking layer. The original AESKW algorithm [14] follows a similar design, since it can be decomposed into a series of calls to an online cipher and a linear layer, but is provided without proof; the KW1 algorithm [14] uses the cyclic shift instead. Our results are a next step towards proving the security of these standardized key wrap mechanisms.

As an alternative to our approach based on an online cipher, one can build a variable length cipher directly from a blockcipher (as TET [19] or AEZ [21] do), or extend the domain of a tweakable blockcipher (e.g. Minematsu's construction [28]). One could use an online cipher to emulate the blockcipher or tweakable blockcipher in these constructions but this would require excessively many calls to the online cipher, considerable less efficient than the three calls of our construction. Arguably a fairer comparison is possible in terms of blockcipher calls and overhead if the online cipher itself is bootstrapped from a blockcipher.

Context and caveats. We will model the online cipher used as having ideal properties, leading to an information-theoretic proof. Instantiating the scheme with any specific online cipher construction incurs an extra term (expressing the online-cipher security of the specific construction).

We express our results in terms of the blocks $\{0,1\}^n$ of a blockcipher, since most online ciphers are built around some internal block cipher, which is explicitly reflected in their syntax and security notions. For schemes built around AES, this means $n = 128$, which implies that our cipher only operates on input sizes a multiple of 128 bits. Essentially, we consider ciphers with domain $(\{0,1\}^n)^*$, as opposed to the preferable $\{0,1\}^*$. We ignore this subtle (but practically relevant) shortcoming, that has haunted other work on online ciphers as well [30, 34], and remark that existing domain completion techniques are not without issue.

Some of our results require independent online ciphers for every layer. These independent ciphers can be easily implemented with a single online cipher, courtesy of the close relationship we expose between tweakable blockciphers with arbitrary length tweaks and online ciphers (e.g. by prefixing each call with a marker corresponding to the appropriate cipher). Alternatively, keying or tweaking the ciphers independently suffices.

2 Preliminaries

Notation. Arrays and lists are indexed from 1. Within proofs and explanations, $X := Y$ means that X is defined to be Y. In the context of pseudocode, $T \leftarrow U$ means variable T takes value U, $X \leftarrow\!\!\$\, Y$ means that the variable X samples uniformly from the set Y, and $L \xleftarrow{\cup} x$ means the set $L \leftarrow L \cup \{x\}$.

A *world* is a collection of *oracles*, interfaces provided by a security game. For any set X of maps with the same interface, the world $\mathcal{W}[X]$ samples an element $\pi \leftarrow\!\!\$\, X$ uniformly, and provides access to π. The world $\pm\mathcal{W}[X]$ does the same, but also provides an interface to π^{-1}. Adversaries are information theoretic, being computationally unbounded. They make limited number of queries to a world $\mathcal{W}_*$ provided by the game, before outputting a value x, which we denote by $A^{\mathcal{W}_*} \to x$. Without loss of generality, we assume they are deterministic and minimal (so do not make queries equivalent to those already made, such as repeating queries). The *distinguishing advantage* between worlds $\mathcal{W}_0$ and $\mathcal{W}_1$ within q queries $\Delta^{\mathcal{W}_1}_{\mathcal{W}_0}(q)$ corresponds to the maximum *distinguisher*. Formally,

$$\Delta^{\mathcal{W}_1}_{\mathcal{W}_0}(q) := \max_{\substack{\mathcal{A}\in\text{Adversaries}\\ \mathcal{A}\text{ makes }q\text{ queries}}} \left|\mathbb{P}\left[\mathcal{A}^{\mathcal{W}_0} \to 1\right] - \mathbb{P}\left[\mathcal{A}^{\mathcal{W}_1} \to 1\right]\right|.$$

Blocks and strings. As discussed in Sec. 1, we constrain ourselves to working within $(\{0,1\}^n)^*$ rather than the more general $\{0,1\}^*$, and allow this to guide our definitions. The set of *blocks* is $\{0,1\}^n$, parametrised by n, the *blocksize* – usually $n = 128$ is inherited from an underlying blockcipher. A *string of blocks* (or simply *string*) is an element of $S \in (\{0,1\}^n)^*$– the length of a string $|S|$ is its length in blocks. We identify $(\{0,1\}^n)^l$ with $\{0,1\}^{ln}$ in the obvious way, allowing us to treat a string of blocks as a single bitstring, and vice versa. For a string (of blocks) X, denote by $X[i]$ the i^{th} block of X. Let $X[i..j] := X[i]||\dots||X[j]$, or the empty string ϵ if $j < i$, where $||$ denotes the concatenation of strings.

For any $x \in \{0,\dots,2^{mn}-1\}$, denote by $\langle x\rangle_m$ an m block string that unambiguously encodes x as an $(m\cdot n)$-bit number (the choice of encoding is not important, as long as it is injective). A function $f\colon (\{0,1\}^n)^* \to (\{0,1\}^n)^*$ is *length preserving* if $|f(X)| = |X|$ for any string X. It is *blockwise linear* if each output block is a linear combination of the input blocks.

2.1 Primitives

We use a number of standard primitives, in particular the notions of a cipher, tweakable blockcipher [26] and online cipher [4]. The keyspace (which we will assume to be the same for all our ciphers) is denoted by $\mathcal{K}$, and we assume all ciphers C to be *length preserving*.

Definition 1 (Cipher). *A* cipher E *is a family of permutations* $E.$ *on inputs* $X \in \mathcal{X} \subset \{0,1\}^*$ *indexed by a key* $k \in \mathcal{K}$*. If* $\mathcal{X} = \{0,1\}^n$*, we say it is a* block cipher*. If* $\mathcal{X} = (\{0,1\}^n)^+$ *and the construction is length preserving, it is a true cipher acting on blocks.*

So, a true cipher is a family of length-preserving permutations that contains an element for each length, also known as a VIL cipher (e.g. [5]).

Definition 2 (Tweakable blockcipher). *A* tweakable blockcipher *(a* TBC*)* $\tilde{E}$ *is a family of permutations of* $\{0,1\}^n$*, indexed by a key* $k \in \mathcal{K}$ *and a* tweak $T \in \mathcal{T}$*, where* $\mathcal{T}$ *is the tweak space. We denote applying this permutation to block* $M \in \{0,1\}^n$ *by* $M' \leftarrow \tilde{E}_k^T(M)$*, and its inverse by* $M \leftarrow \tilde{D}_k^T(M')$*.*

Thus a tweakable blockcipher can be thought of as a collection of blockciphers, the appropriate one of which is chosen by the tweak. Finally, we move on to the definition of an online cipher:

Definition 3 (Online cipher). *An* online cipher *is a cipher for which the* i^{th} *block of ciphertext depends only on the first* i *blocks of plaintext. Thus it is a family* $\mathcal{E}$ *of permutations on* $(\{0,1\}^n)^+$ *indexed by some* $k \in \mathcal{K}$*, where for any* $m > 0$ *and* $A \in \{0,1\}^{mn}$*,* $\mathcal{E}_k(A||B)[1..m] = \mathcal{E}_k(A)$ *for all* $B \in (\{0,1\}^n)^*$*.*

This formalisation of an online cipher (due to Bellare et al. [4]) describes a construction that can outputs ciphertext blocks as soon as the corresponding message blocks arrive. It does not necceserily define an online algorithm, since it imposes no limitations on the size of the internal state the construction may utilize, although in practice most schemes can be efficiently realised by an online algorithm.

Security notions. Intuitively, a cipher is secure if even given a large number of input–output pairs, virtually nothing is known about its behaviour on other values: every permutation that does not contradict already known information is equally likely. Motivated by this, let $\mathrm{Perm}(l)$ be the set of all permutations on l bits, meaning $\mathcal{W}[\mathrm{Perm}(l)]$ corresponds to the ideal cipher, since every possible permutation is equally likely. The actual security of a cipher E is measured by the likelihood an adversary can distinguish a randomly keyed instance of it from the ideal cipher (based on oracle access only).

Similarly, we define the ideal primitives for random function with inverse, TBC and online cipher by first defining the set of all such objects, following standard terminology neatly coalated by Halevi and Rogaway [20]. Define $\mathrm{Func}(l)$ to be the set of all functions from l bits to l-bit, $\mathrm{Perm}(\mathcal{T}, n)$ the set of functions $f\colon \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ where for any $T \in \mathcal{T}$ the map $M \to f(T, M)$ is a permutation, and $\mathrm{OPerm}(l)$ the set of all online permutations on l bit blocks. Thus $\mathcal{W}[\mathrm{Perm}(\mathcal{T}, n)]$ corresponds to the ideal TBC and $\mathcal{W}[\mathrm{OPerm}(l)]$ to the ideal online cipher.

Slightly more involved is the ideal random function with inverse, in which the encryption and decryption interfaces are instantiated with independently sampled random functions, subject to the condition that they never contradict oneanother. Commonly this is done by "lazy sampling", where values for the function (or its inverse) are selected as required: with each query, if the value is already defined it is returned, and if not a value is uniformly sampled and recorded (see Alg. C).

The security notions of PRF, PRP, TPRP and OPRP are defined by the adversarial advantage in distinguishing a primitive from the ideal random function, ideal cipher, ideal tweakable blockcipher and ideal online cipher respectively (when provided with oracle access to just the encryption interface). Analogously, we define $\pm$PRF, $\pm$PRP, $\pm$TPRP, $\pm$OPRP by providing oracle access to both the forward and inverse interfaces. The complete list is provided in Appendix B, but as an example,

$$\mathbf{Adv}^{\mathsf{oprp}}_{\mathcal{E}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow\!\!\$\ \mathcal{K}\ :\ \mathcal{A}^{\mathcal{E}_k} \to 1\right] - \mathbb{P}\left[\pi_* \leftarrow\!\!\$\ \mathrm{OPerm}(n)\ :\ \mathcal{A}^{\pi_*} \to 1\right].$$

The goals above are defined for primitives of just a single length, and we generalise to allow variable length constructions by providing an equivalent interface for every requested length. As previously, for some goal xxx, $\mathbf{Adv}^{\mathrm{xxx}}_{P(q)}$ is defined as the maximum across all adversaries making q queries. We say a scheme P is a *secure* xxx if $\mathbf{Adv}^{\mathrm{xxx}}_{P(q)}$ is sufficiently small. At times we use the term $\pm$PRP to refer to a secure $\pm$prp, and similarly for any other goals.

Use of ideal primitives. For conciseness, let $\mathcal{E}$ be the encryption routine of the ideal online cipher, with inverse $\mathcal{D}$. This means that $(\mathcal{E}, \mathcal{D}) = \pm\mathcal{W}[\mathrm{OPerm}]$, and our proofs will be constructed around this ideal primitive. Application of our results to real constructions requires swapping out the real online cipher for the ideal primitive. The cost of this switch depends on the overall objective, since the online cipher need only be secure under the equivalent notion to the overall construction. For example, for $\pm$prp security the online cipher must be a secure $\pm$oprp, but for prp security the online cipher need just be an oprp.

2.2 Constructions

We seek a framework for efficiently converting an online cipher into a true cipher, ideally using only a small number of calls to the online cipher, sandwiched together by some highly efficient (invertible) mixing layer(s). Restricting to linear mixing layers leads us to the following definition.

Definition 4 (The Π_i^L construction). *Define Π_i^L to be the composition of i calls to an online cipher $\mathcal{E}$ around $(i-1)$ applications of a public family of blockwise linear layers L.*

So, for example, $\Pi_2^L(M) = \mathcal{E} \circ L \circ \mathcal{E}(M)$. We will consider various combinations of (L, i) and observe that some combinations lead to schemes with PRP or $\pm$PRP security. When clear, we omit the linear layer or number of rounds from the notation.

Blockwise linear layers. The first and most obvious candidates for linear layers are blockwise permutations: maps that simply reorder the blocks. In this paper we will focus on the blockwise reversal map, rev, and swap, the map the exchanges the first and last blocks of a message, and (pre-empting ourselves slightly) will show they suffice to obtain full $\pm$prp security. Later, inspired by the choices of AESKW [14], we will consider the right circular shift right, and by association its inverse, the left circular shift left.

Formally, for any $M \in (\{0,1\}^n)^m$, these maps are defined by:

$$\mathsf{right}(M) := M[m]||M_1||\ldots||M[m-1] \qquad \mathsf{rev}(M) := M[m]||M[m-1]||\ldots||M[2]||M[1]$$
$$\mathsf{left}(M) := M[2]||\ldots||M[m]||M[1] \qquad \mathsf{swap}(M) := M[m]||M[2]||M[3]||\ldots||M[m-1]||M[1]$$

2.3 Standard results

The $\pm$PRP–$\pm$PRF switching lemma, for which a proof is given by Halevi and Rogaway [20, App. C], will be used in several proofs to switch the ideal world from a random permutation to a random function with inverse. To bound final collision events, we use the well known birthday bounds.

Lemma 5 ($\pm$PRP–$\pm$PRF switch). *One cannot distinguish a random permutation from a random function with inverse any better than achieving collisions in the random function, even when given access to both interfaces. Therefore, if the shortest queries are m blocks long, $\Delta^{\mathcal{W}[\mathrm{Perm}]}_{\mathcal{W}[\mathrm{Func}]}(q) \leq q(q-1)/2^{mn+1}$.*

Lemma 6 (Birthday Bound). *The probability that a list of q independent random variables (each of t bits) contains a repeat is bounded. Explicitly,*

$$\frac{q(q-1)}{2^{t+2}} \leq \mathbb{P}\left[a_1, \ldots, a_q \leftarrow\!\!\$\ \{0,1\}^t \ :\ \exists i \neq j \text{ s.t. } a_i = a_j\right] \leq \frac{q(q-1)}{2^{t+1}}$$

where the lower bound requires $q \leq 2^{(t+1)/2}$, and the upper bound holds for all q.

3 Equating Online Ciphers and Tweakable Block Ciphers

Online ciphers can be formed from a chain of TBCs, an observation that allowed Rogaway and Zhang to simplify the analysis of online ciphers [34]. We observe that an even closer relationship exists: an online cipher *is* a TBC with variable length tweak.

The link between online ciphers and TBCs can be best understood by considering how online ciphers act on strings of blocks. Let $E(\cdot)$ be the encryption function of an online cipher and $A, B \in (\{0,1\}^n)^*$, then we define $E^A(B) := E(A||B)[x..m]$, where $x = |A| + 1$ and $m = |A| + |B|$. So, $E^A(B)$ returns the output blocks corresponding to B when processed with a prefix of A. Then, by the online property, $E(A||B) = E^{\epsilon}(A)||E^A(B)$. In the tweakable context, A is the tweak under which B is encrypted, and in the online context, we refer to A as the *prefix* under which B is encrypted. Thus the *prefix* is similar to the *state* in the incremental online cipher characterisation [33], except that prefixes may be arbitrarily large, whereas states have fixed length.

Similar to the encryption case, we define $\hat{\mathcal{D}}^A(B) := \mathcal{D}(A||B)[x..m]$. By setting $\mathcal{D}^A(B) := \hat{\mathcal{D}}^{E(A)}(B)$, we obtain the inverse of E^A, as $\mathcal{D}^A(E^A(B)) = B$. This is not a problem to compute, since to calculate $\mathcal{D}(A||B)$ one first calculates $M = \mathcal{D}(A)$, then $\mathcal{D}(A||B) = M||\mathcal{D}^M(B)$, something the online cipher does internally anyway. Our notation emphasises the correspondence with TBCs, since is A the tweak under which B is decrypted.

Theorem 7. *There is a security preserving one-to-one correspondence between online ciphers on blocks $\{0,1\}^n$ and tweakable blockciphers on $\{0,1\}^n$ with tweak space $(\{0,1\}^n)^*$.*

Proof. We begin by defining a map f from the set of online ciphers to the set of such TBCs. The tweakable blockcipher will call the online cipher on $T||M$, before throwing away all but the final block, effectively using the bulk of the cipher call preprocessing the tweak. So, if E is an online cipher then $f(E)$ is the TBC $f(E)^T_k(M) := E^T_k(M)$ for any $M \in \{0,1\}^n$ and $T \in (\{0,1\}^n)^*$.

Conversely, the map g from TBCs to online ciphers will call the TBC on each block, using previous blocks as the tweak. So, for TBC $\tilde{E}$, the online cipher $g(\tilde{E})$ is defined for all $M \in (\{0,1\}^n)^*$ with $m = |M|$ by

$$g(\tilde{E})_k(M) := \tilde{E}^{\epsilon}_k(M[1])||\tilde{E}^{M[1]}_k(M[2])||\dots||\tilde{E}^{M[1..(m-1)]}_k(M[m]).$$

We observe that for any TBC $\tilde{E}$ and online cipher E we have $f(g(\tilde{E}))_k = \tilde{E}_k$ and $g(f(E))_k = E_k$. Thus the maps are in fact inverses, defining a correspondence.

With the correspondence established, we move on to proving it preserves security. The key observation is that, because the map defines a correspondence between elements it must map the set of all TBCs onto the set of all online ciphers, and vice versa. That is, $f(\mathrm{OPerm}(n)) = \mathrm{Perm}((\{0,1\}^n)^*, n)$ and $g(\mathrm{Perm}((\{0,1\}^n)^*, n)) = \mathrm{OPerm}(n)$. So, if an online cipher is distinguishable from the ideal online cipher, by applying the f we see that the corresponding TBC is distinguishable from the ideal tweakable block cipher, and vice versa. Thus, security of one implies security of the other. □

Viewing an online cipher as a tweakable blockcipher, it is clear that after processing fresh prefixes, only uniform randomness will be output. In particular, we can ensure two calls to $\mathcal{E}^{\cdot}(\cdot)$ are independent by taking care with the length of tweaks: as long as $|t| \geq |u| + |y|$, $\mathcal{E}^t(x)$ is independent of $\mathcal{E}^u(y)$.

Corollary 8. *If no call to $\mathcal{E}$ has been made beginning or explicitly tweaked by A, then $\mathcal{E}^A(B)$ is uniformly sampled from all strings of length $|B|$.*

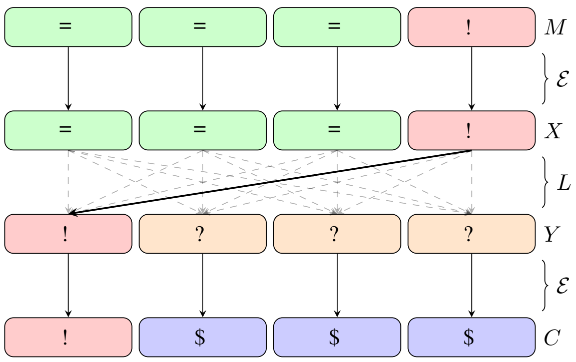
Finally, we provide a similar result about just the final block. Explicitly, when called with distinct inputs the final output blocks collide with probability at most that of colliding two blocks sampled uniformly at random.

Lemma 9. *Let $\mathcal{R} = (R_1, \dots, R_q)$ be a list of q blocks, where each $R_i = \mathcal{E}^{s_i}(t_i)$ is the output of the encryption of a unique input, meaning $s_i||t_i \neq s_j||t_j$ for any $i \neq j$. Then, the probability of a collision in the list (that $R_i = R_j$ for $i \neq j$) is bounded, with $\mathbb{P}[\mathsf{Collide}\ \mathcal{R}] \leq \frac{1}{2}q(q-1)2^{-n}$.*

Proof. Let $i \neq j$. Then, by construction, $R_i = R_j \iff \mathcal{E}^{t_i}(s_i) = \mathcal{E}^{t_j}(s_j)$. If $t_i = t_j$, then $R_i = R_j$ implies that $s_i = s_j$, which contradicts the assumption that all inputs were unique, and so cannot happen. If $t_i \neq t_j$, the tweakable cipher has different tweaks in instance, and so the two distributions are independent. Thus R_i and R_j are both sampled uniformly at random and independently, and so collide with probability 2^{-n}. So, taking the maximum of these probabilities, $\mathbb{P}[R_i = R_j \mid i \neq j] \leq 2^{-n}$. Applying the union bound, we get the required result. □

4 Reversing into a ±PRP

Any scheme with a single layer (and thus one call to the online cipher), can be trivially distinguished from a PRP with two queries. Explicitly, after querying $\mathcal{O}_{\text{Enc}}(\langle 0\rangle_1 || \langle i\rangle_1)$ for $i = 0, 1$, the two ciphertexts always agree on the first block of output for an online cipher, yet rarely for a true prp. In this section we investigate what can be done by using two or three layers of an online cipher when using blockwise reversal as the linear layer. In Thm. 10 we show that using two calls to the online cipher (and irrespective of the mixing layer), the best one can achieve is security up to the birthday bound. Effectively we reverse the logic of the attack, moving from guaranteed collisions in the first block of output for the construction to a scenario where the construction *never* collides on those blocks. We complement this result in Thm. 11, showing that two calls of an online cipher with reversal yields a ±prp up to the birthday bound, and extend further in Thm. 13, proving that an additional layer (and online cipher call) comes close to providing ±prp security up to the blocksize itself.



Key:

= Blocks that are the same across all queries.

! Blocks that are unique across all queries.

$ Blocks that are uniformly sampled.

? Blocks whose distribution is unknown/irrelevant.

Intuition: Since the first block output by L depends linearly on the final input block, varying only this block ensures that for the construction the first block of all ciphertexts will be distinct.

Fig. 2. An attack against PRP security of Π_2^L (Theorem 10)

Theorem 10. *The $\Pi = \Pi_2^L$ construction cannot achieve beyond birthday bound security for message lengths greater than 1, no matter what map is chosen for the blockwise linear layer L. In particular, $\mathbf{Adv}_{\Pi}^{\mathsf{prp}}(q) \geq \frac{q(q-1)}{8 \cdot 2^n}$ for all $q \leq 2 \cdot 2^{n/2}$.*

Proof. Any construction where $L(X)[1]$ is independent of $X[m]$ (where $|X| = m$), can trivially be distinguished, by two messages differing only in the final block (as they will share the same first ciphertext block). Henceforth, we assume that $L(X)[1]$ depends on $X[m]$.

Let $M^t := \langle 0\rangle_{m-1} || \langle t\rangle$, where $m \geq 2$ is chosen arbitrarily. The adversary $\mathcal{A}$ will vary t to make $q \leq 2^n$ queries of this form, and $\mathcal{A} \to 1$ if all q ciphertexts have distinct first blocks.

We begin by calculating $\mathbb{P}\left[\mathcal{A}^{\Pi} \to 1\right]$, following the logic shown in Figure 2, and label the internal variables as M, X, Y, C as per the diagram. By the online property, $X^t = \mathcal{E}(M^t)$ begins with a blocks that are the same across all queries. Since the final block is encrypted under the same prefix each time, the values of $X^t[m]$ are distinct between queries. By assumption on L, $Y^t[1]$ is linearly dependent on $X^t[m]$. Since the other blocks of X^t are constant through all queries, we must have that $Y^t[1] = A \oplus X^t[m]$ for some A that is independent of t. Since online cipher called on just one block is a permutation, equality in $C[1]$ blocks implies equality in $Y[1]$ variables. Overall then,

$$t = u \iff M^t = M^u \iff X^t[m] = X^u[m] \iff Y^t[1] = Y^u[1] \iff C^t[1] = C^u[1].$$

So, if $t \neq u$ the first blocks of the ciphertexts will differ, and thus $\mathbb{P}\left[\mathcal{A}^{\Pi} \to 1\right] = 1$.

On the other hand, one expects collisions on the first output block of an ideal cipher on $m > 1$ blocks after enough queries. In particular, the probability all q ciphertexts have distinct first blocks is simply the product of the probabilities that the first block of each ciphertext is distinct from those calculated before it. Thus

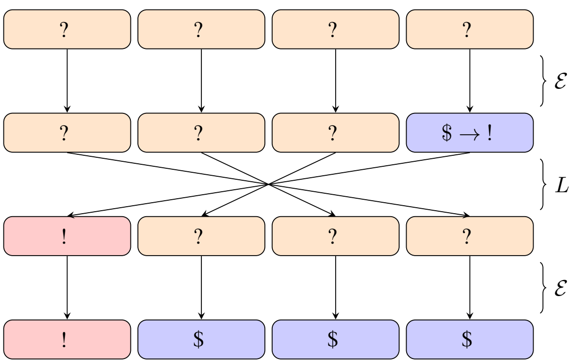
$$\mathbb{P}\left[\mathcal{A}^{\mathcal{W}[\text{Perm}]} \to 1\right] = \prod_{i=1}^{q} \left(1 - \frac{(i-1)(2^{(m-1)n} - 1)}{2^{mn} - (i-1)}\right) \leq \prod_{i=0}^{q-1} \left(1 - \frac{i}{2^{n+1}}\right) \leq 1 - \frac{q(q-1)}{8 \cdot 2^n}.$$

It is for the final inequality, that we require the bound on q. Combining, we have $\mathbf{Adv}_{\Pi}^{\mathsf{prp}}(q) \geq \frac{q(q-1)}{8 \cdot 2^n}$, which yields the stated bound. $\square$

4.1 Two layer ±PRP security to the birthday bound

We move on to considering a positive result: what are the minimum properties required of the linear layer to meet this bound? To prevent a similar attack to the one round construction, where there were elements of the message that could be changed without affecting large portions of the ciphertext, the linear layer must move blocks to and from each end of its input. This means that both $L(M)[1]$ and $L^{-1}(M)[1]$ must depend on $M[m]$. Though this condition appears to suffice, for clarity we will prove a slightly weaker result, instead assuming $L(M)[1] = L^{-1}(M)[1] = M[m]$. The intuition behind the proof is represented in Figure 3.

One possible instantiation of this form, the Π_2^{rev} construction, bears similarities to CMC mode [20], which combines two passes of CBC-mode requiring a masking layer—the “M” in the acronym—in between. Both CMC and our construction Π_2^{rev} provide security up to the birthday bound, which is asymptotically optimal due to the attack in Theorem 10. We reduce the amount of computation required outside the cipher call, plus we believe that when considering a single pass only, an online cipher provides a better security–efficiency tradeoff than CBC.



Key:

? Blocks whose distribution is unknown/irrelevant.

! Blocks that are unique across all queries.

$ Blocks that are uniformly sampled.

$ → ! Block takes different values with high probability.

Intuition: Inputs must be distinct, so with high probability the final block from the first layer is unique. If so, the first input block to the second layer is unique, and so the remaining output is sampled uniformly. By symmetry, a similar argument can be applied for inverse queries, checking for collisions in the same place.

Fig. 3. Intuition behind the PRP and ±PRP security of Π_2^L (Theorem 11)

Theorem 11. *Let L be a blockwise linear function that swaps the first and last blocks (i.e. $L(M)[1] = L^{-1}(M)[1] = M[m]$) such as* rev *or* swap*. Then $\Pi = \Pi_2^L$ is a secure ±PRP, with* $\mathbf{Adv}_{\Pi}^{\pm\mathsf{prp}}(q) \leq \frac{4q^2}{2^n}$.

Proof. We use domain separation to split the adversary's oracles into two pairs: the first two answering single block messages and the other two answering longer queries. The triangle inequality allows us to rewrite the advantage as

$$\mathbf{Adv}_{\Pi}^{\pm\mathsf{prp}}(q) = \Delta_{\$,\ \$\ ,\$,\ \$}^{\Pi,\Pi^{-1},\Pi,\Pi^{-1}}(q) \leq \Delta_{\Pi,\Pi^{-1},\$,\ \$}^{\Pi,\Pi^{-1},\Pi,\Pi^{-1}}(q) + \Delta_{\$,\$,\$,\$}^{\Pi,\Pi,\$,\$}(q)\ .$$

The second term relates just to messages of length 1, in which case the construction results in composing an $\pm$prp with itself. This composition does not reduce security, and so the two worlds are indistinguishable, meaning the second term is zero.

To bound the first term, we must be aware of the single block oracles, as they are present. However, we choose to focus on distributions (and events) that are necessarily independent of them, meaning they do not help the adversary distinguish between the worlds, and so for conciseness are omitted. The proof itself consists of a sequence of games, $\mathcal{W}_1$ to $\mathcal{W}_7$, code for which is provided in Appendix C.1.

We know that $m \geq 2$, allowing us to consider a "first" and a "final" block. Define the operator $\bar{\cdot}$ on strings such that for any $X, Y \in \{0,1\}^n$ and $A \in \{0,1\}^{n*}$, we have $L(X||A||Y) = Y||\overline{A}||X$. That is, $\overline{A}$ is the image of the central blocks under L. For completeness, our notation ought to expose the possible dependency of $\overline{A}$ on X and Y, yet this dependency does not affect our proofs and so is omitted for clarity. We let bad^j denote the event that game j sets flag bad.

The first world, $\mathcal{W}_0$, directly encodes the Π_2 construction, whilst $\mathcal{W}_1$ is adversarially indistinguishable from $\mathcal{W}_0$, since it just expands out some of the function calls for later use. Clearly, Worlds $\mathcal{W}_1$ and $\mathcal{W}_2$ are identical until $\mathcal{W}_2$ triggers bad^2, because the only difference between them is a resampling any prefixes that may repeat, which is in the same branch as (and so must set) the bad flag.

$\mathcal{W}_2$ and $\mathcal{W}_3$ are identical until bad^3 occurs. This is because their only differences are on Line 5.14 and Line 5.30, where an online cipher call is converted into a uniformly sampling. This is valid because, until bad^3, the prefix to this call is unique.

More obviously, $\mathcal{W}_3$ and $\mathcal{W}_4$ are identical until bad_4, since again the only change (the removal of a resampling) occurs within the same branch that sets bad.

Transitioning from $\mathcal{W}_4$ to $\mathcal{W}_5$ is merely notational. Combining these, we observe that each world $\mathcal{W}_0$ to $\mathcal{W}_5$ is indistinguishable from the next, until the latter sets bad. As such, $\mathcal{W}_0$ is indistinguishable from $\mathcal{W}_5$ until $\mathcal{W}_5$ sets bad.

Worlds $\mathcal{W}_5$ and $\mathcal{W}_6$ differ by a $\pm$PRP–$\pm$PRF switch to the first online cipher call. Any strategy that can trigger bad^5 may also be used to trigger bad^6, since the switch from an $\pm$PRP to an $\pm$PRF makes it easier for the adversary to generate internal collisions, the event required for bad^6.

Now, $\mathcal{W}_7$ is a $\pm$PRF, and is indistinguishable from $\mathcal{W}_6$ until bad^7. This is because $\mathcal{P}$ contains all values that the random function has been queried on, as well as all outputs from its inverse. Thus if a value $R_2 \notin \mathcal{P}$, we have not yet evaluated $F(R_2)$, and thus this value is uniformly sampled. Similarly for L_1 in the decryption case.

As before, strategies triggering bad^6 may be used to set bad^7, since until either game sets bad they are equivalent. Combining all such results, we see that $\mathbb{P}\left[\mathsf{bad}^5\right] \leq \mathbb{P}\left[\mathsf{bad}^7\right]$. Finally, we complete the series of games with a $\pm$PRP–$\pm$PRF switch on the overall construction. Thus,

$$\begin{aligned}\mathbf{Adv}_{\Pi_2}^{\pm\mathsf{prp}}(q) &\leq \mathbb{P}\left[\mathsf{bad}^5\right] + \Delta_{\pm\mathsf{prp}}^{\pm\mathsf{prf}}(q) + \mathbb{P}\left[\mathsf{bad}^7\right] + \Delta_{\pm\mathsf{prp}}^{\pm\mathsf{prf}}(q) \\ &\leq 2\frac{q(q-1)}{2^{n+1}} + 2\frac{q(3q+1)}{2^{n+1}} = \frac{4q^2}{2^n}\end{aligned}$$

Where we bound $\mathbb{P}\left[\mathsf{bad}^5\right] \leq \mathbb{P}\left[\mathsf{bad}^7\right]$ and $\mathbb{P}\left[\mathsf{bad}^7\right]$ using Lemma 12. □

Lemma 12. *Continuing the notation of Theorem 11,* $\mathbb{P}\left[\mathsf{bad}^7 \mid q\ \textit{queries}\right] \leq \frac{q(3q+1)}{2^{n+1}}$.

Proof. We now depart from the indistinguishability game, and switch to games in which the adversary interacts with a single oracle, trying to trigger the bad event, which are again presented in Appendix C.1.

Since the output from both oracles in $\mathcal{W}_7$ is uniformly sampled, it does not help the adversary in trying to set bad. Thus we may remove it, and any code that is then superfluous, to produce a new pair of oracles, $\mathcal{W}_8$, such that $\mathbb{P}\left[\mathsf{bad}^7\right] = \mathbb{P}\left[\mathsf{bad}^8\right]$.

Now consider the difference between the oracles in $\mathcal{W}_8$ and $\mathcal{W}_9$. The decryption oracle of $\mathcal{W}_8$ calculated L_3 as function of the input (input which is otherwise irrelevant), using the inverse cipher call $\mathcal{D}^\epsilon$. $\mathcal{W}_9$ replaces this by allowing the adversary to directly select L_3. To allow them to simulate $\mathcal{W}_8$, the adversary would require access to a $\mathcal{D}^\epsilon$ oracle. However, no other calls are made to either $\mathcal{D}^\epsilon$ or $\mathcal{E}^\epsilon$: all other block cipher calls are made to $\mathcal{E}^{L_1||A_1}$, which has a prefix length of at least one block. Thus access to $\mathcal{D}^\epsilon$ does not help the adversary in any way, and so can be omitted. The only other changes are the removal of new lines and splitting of $\mathcal{P}$ into 3 lists for notational reasons. Overall then, $\mathbb{P}\left[\mathsf{bad}^8\right] \leq \mathbb{P}\left[\mathsf{bad}^9\right]$, since any strategy triggering bad^8 also triggers bad^9.

Since the adversary receives no output from $\mathcal{W}_9$, adaptivity does not help forcing bad^9, thus we may restrict ourselves to non-adaptive adversaries. We extend the adversaries control by allowing them to directly submit $\mathcal{P}$ independent of their message requests (rather than half of it being blocks $M[0]$). Having done this, there is no input required for a decryption query, so we drop this. Thus the adversary submits the list $\mathcal{P}$, along with each of his encryption challenges. bad^9 is triggered when new elements (R_2 for encryption queries, L_1 for decryption queries) collide with previous elements, a check $\mathcal{W}_{10}$ makes at the end. Thus any strategy setting bad^9 may be used to set bad^{10}.

So, it remains just to bound this probability explicitly. Since $\mathcal{L}$ is uniformly sampled, the probability of a collision between its elements is 2^{-n} for each pair. Similarly, the probability of an element of $\mathcal{L}$ colliding with one in $\mathcal{R}$ is $|\mathcal{R}| \cdot 2^{-n}$. The event that two elements of $\mathcal{R}$ collide is precisely that discussed in Lemma 9, and thus is at most 2^{-n}. Let $\mathsf{Collide}\ X$ be the event that list X contains the same element twice. Then,

$$\begin{aligned}\mathbb{P}\left[\text{Repeats in } (\mathcal{L}||\mathcal{R})\right] &= \mathbb{P}\left[\mathsf{Collide}\ \mathcal{L}\right] + \mathbb{P}\left[\exists i, j \text{ s.t. } \mathcal{L}_i = \mathcal{R}_j\right] + \mathbb{P}\left[\mathsf{Collide}\ \mathcal{R}\right] \\ &\leq \binom{q_E}{2} 2^{-n} + q_D \cdot q_E \cdot 2^{-n} + \binom{q_D}{2} 2^{-n}.\end{aligned}$$

Let us now consider the probability that bad is set on Line 10.16 or Line 10.21. For each i such that $\mathrm{order}[i] = E$, we have the probability of this being set is simply that a single output from an encryption oracle collides with a list of length i. So, the probability of this occurring for the first time on the i^{th} query to the construction, itself the e^{th} encryption query, is $i/(2^n - e)$. Conversely, if $\mathrm{order}[i] = D$, this is the probability that a uniformly sampled element ($\mathcal{L}_d$) is in a list of length i, which again occurs with probability $i2^{-n}$. The sum of these two bounds is maximized by making all Dec-queries first, and so

$$\mathbb{P}\left[\mathsf{bad\ set\ here}\right] \leq \sum_{i=1}^{q_D} \frac{i}{2^n} + \sum_{i=1}^{q_E} \frac{q_D + i}{2^n - i} \leq \sum_{i=1}^{q_D} \frac{i}{2^n} + \sum_{i=q_D+1}^{q} \frac{i}{2^{n-1}} \leq \frac{q(q+1)}{2^n}.$$

Where we have assumed $q_E \leq 2^{n-1}$. Since these cover all possibilities,

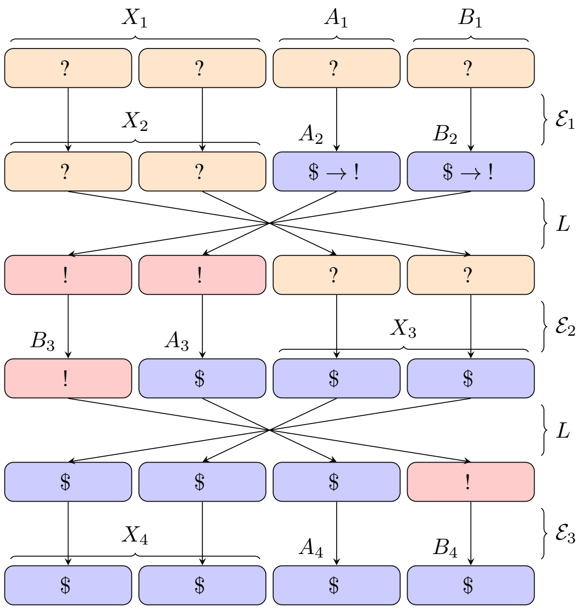
$$\mathbb{P}\left[\mathsf{bad}^7\right] \leq \mathbb{P}\left[\mathsf{bad}^{10}\right] \leq \frac{1}{2^n}\left[\binom{q_E}{2} + q_D \cdot q_E + \binom{q_D}{2} + q(q+1)\right] \leq \frac{q(3q+1)}{2^{n+1}}.$$

To generalize this result, we drop the requirement bounding q_E (and thus q) by observing that if $q_E > 2^{n-1}$, this bound is greater than 1 and so vacuously true. □

4.2 Three layer reverse: ±PRP beyond the birthday bound

We have shown that birthday bound security is both achievable and the best possible with just two layers. A natural question is whether security increases with more calls to the online cipher. We find in the affirmative: the Π_3^{rev} achieves security up until almost the blocksize, requiring just three calls to the online cipher.

The key observation behind the proof is that, until certain pairs of blocks repeat, the online ciphers act like tweakable random functions, which themselves act like independent uniform samplers. We provide a series of worlds that are perfectly indistinguishable until one of six bad events occurs. Each of these events is a collision, occurring across at least two blocks. The logic and variable naming scheme are represented in Figure 4.



Key:

? Strings of blocks whose distribution is unknown/irrelevant.

! Strings of k blocks that do not repeat too frequently.

\$ Strings of blocks that are uniformly sampled.

$\$ \to !$ Strings of blocks that are uniformly sampled and so don't repeat too frequently.

Intuition: The key observation behind the proof is that, until certain pairs of blocks repeat, the online ciphers act like tweakable random functions, which themselves act like independent uniform samplers. To formalise this, we use a series of worlds that are perfectly indistinguishable until one of six collision events occurs.

The diagram to the left provides a representation of this, and illustrates the naming system used in the proof. Each A_i, B_i is a single block, X_i a string of blocks of length $|X_i| = |M| - 2$. Explicitly, the collision events are on the prefix–value pairs (A_2, B_2), (A_2, B_3), $(X_1||A_1, B_1)$ and $(X_3||A_3, B_4)$, with some tested during both encryption and decryption. Finally, we prove it is hard for an adversary to trigger one of these events.

Fig. 4. Intuition behind the $\pm$PRP security of Π_3^{rev} (Theorem 13)

Theorem 13. *Let L be any blockwise linear function that when $|M| = m \geq 2$ satisfies $L(M)[1..2] = L^{-1}(M)[1..2] = M[m]||M[m-1]$, such as the* rev *map. Then, the adversarial advantage in distinguishing the Π_3^L construction from an $\pm$PRP within $q \leq 2^n$ queries is*

$$\mathbf{Adv}^{\pm\mathsf{prp}}_{\Pi_3}(q) \leq 1.5\frac{q(q-1)}{2^{2n}} + \left(\frac{q}{2^n}\right)^{\kappa} \frac{2^n}{(\kappa+1)!} + \frac{\kappa \cdot q}{2^n} \leq n\frac{q}{2^n}.$$

The first inequality holds for any $\kappa \in \mathbb{N}$, while the second assumes $n \geq 4$.

Proof. By the same argument as Thm. 11, the scheme is perfectly secure with regards to any adversary restricted to single block queries. What remains is bounding the probability that an adversary with access to two pairs of oracles (one pair answering single block queries and one answering any longer queries) can distinguish the scheme from random. For conciseness, we omit the single block oracles from our notation, because at no point will it assist the adversary in distinguishing between the worlds being compared or setting bad flags.

Throughout the remainder of this proof, A_i, B_i are blocks and $|X_i| = m - 2$ for all $i \in 1, \ldots, 4$, where $m = |M|$ is the length of M in blocks. The numbering scheme is represented in Figure 4. For any string of blocks $X \in (\{0,1\}^n)^*$, define $\overline{X}$ such that $B||A||\overline{X} = L(X||A||B)$. Again, $\overline{X}$ could conceivably depend on A or B (cf. similar notation used by Thm. 11), but this dependence does not invalidate the proofs (since the map is blockwise linear and publicly computable) and so is omitted for clarity.

We use a sequence of games, $\mathcal{W}_0, \ldots, \mathcal{W}_3$ as provided in Appendix C.3. Firstly, $\mathcal{W}_0$ directly encodes the Π_3^L construction. Then, after a series of identical until bad switches, we reach $\mathcal{W}_3$, which is perfectly indistinguishable from an $\pm$PRP until bad. Next, we proceed to explicitly bound the probability of the various bad flags, using auxiliary games $\mathcal{W}_4$ to $\mathcal{W}_6$. Let bad_i^j be the event that an adversary interacting with world $\mathcal{W}_j$ is able to set flag bad_i, and define $\mathsf{bad}^j := \bigvee_{i=1}^{6} \mathsf{bad}_i^j$.

Claim (1). $\mathbf{Adv}^{\pm\mathsf{prp}}_{\Pi_3^L} \leq \mathbb{P}\left[\mathsf{bad}^3\right]$.

Proof. $\mathcal{W}_0$ directly encodes the Π_3^L construction, while $\mathcal{W}_1$ simple expands this and adds code to support the bad flags.

Transitioning from $\mathcal{W}_1$ to $\mathcal{W}_2$ we perform a number of swaps, exchanging single block calls to the online cipher with tweakable random functions. An adversary cannot distinguish these switches until either the random function outputs the same value twice (which corresponds to repeating a tweak-output pair), or the inverse is called on a point that is already in the image of the random function. Explicitly: the first switch in the Enc oracle cannot be detected before bad_1, the second before bad_3 and the third before bad_5. Similarly, the decryption switches are undetectable: the first or second switches cannot be detected before bad_4 and the third before bad_6.

As long as the tweak–input pair does not repeat, the output from a tweakable random function is independently uniformly sampled. Until one of the bag flags is set, this holds for the six tweakable random function calls used in $\mathcal{W}_2$. Explicitly, ordering the six calls by their occurance in the code of $\mathcal{W}_2$, the input to the call is unique until (respectively) the event $\mathsf{bad}_3, \mathsf{bad}_1, \mathsf{bad}_3, \mathsf{bad}_2, \mathsf{bad}_4$ or bad_4 occurs. Thus until bad occurs, we may replace the tweakable random functions with independent samplings, which is precisely $\mathcal{W}_3$.

We observe that $\mathcal{W}_3$ is a random function with inverse. The Enc oracle samples $A_3||X_3||B_4$ uniformly at random, which (as $X_4||A_4$ is the image of $A_3||X_3$ under a permutation) is equivalent to sampling $X_4||A_4||B_4$ uniformly at random. Similarly, the Dec oracle samples $A_2||X_2||B_1$ which is equivalent to sampling $X_1||A_1||B_1$ uniformly at random. Now, a random function with inverse is perfectly indistinguishable from a random permutation until either one of the oracles repeats an output, or the output of one oracle corresponds an the input the adversary already made to the other. The output from the Enc oracle does not invalidate either of these requirements without first setting the flag bad_5, and Dec oracle does not without setting the bad_6 flag. Therefore, until bad^3, $\mathcal{W}_3$ is perfectly indistinguishable from a $\pm$PRP.

Combining these results, the worlds are all identical until the event bad occurs, and thus the advantage is at most $\mathbb{P}\left[\mathsf{bad}^3\right]$. ■

Claim (2). $\mathbb{P}\left[\mathsf{bad}_5^3 \vee \mathsf{bad}_6^3\right] \leq \frac{q(q-1)}{2 \cdot 2^{mn}}$

Proof. When the tests are made on the i^{th} query that might set the flags bad_5 or bad_6, the lists are of length $|\mathcal{L}_A| = |\mathcal{L}_D| = i - 1$. This query is either an encryption or a decryption query, but in either case the flag is set if a uniformly sampled element of $\{0,1\}^{mn}$ is present in the appropriate list, which occurs with probability $i - 1/2^{mn}$. Taking a union bound to sum across all queries gives the required result. ■

Claim (3). $\mathbb{P}\left[(q_E, q_D) \text{ queries } : \mathsf{bad}_1^3 \vee \mathsf{bad}_2^3\right] = \mathbb{P}\left[(q_D, q_E) \text{ queries } : \mathsf{bad}_3^3 \vee \mathsf{bad}_4^3\right]$

Proof. For clarity, consider $\mathcal{W}_4$ rather than $\mathcal{W}_3$, which differs only in that superfluous code has been removed. Then, by a simple symmetry argument we observe that bad_1 acts within the Enc oracle equivalently to how bad_4 does within the Dec oracle, and similarly bad_2 mirrors bad_3. Thus any strategy for triggering $\mathsf{bad}_1 \vee \mathsf{bad}_2$ that makes q_E encryption queries and q_D decryption queries can be used to trigger $\mathsf{bad}_3 \vee \mathsf{bad}_4$ with the same probability, after making q_D encryption queries and q_E decryption queries Since the relationship is symmetric, the opposite also holds and so the probabilities are equal. ■

Claim (4). $\mathbb{P}\left[\mathsf{bad}_3^3 \vee \mathsf{bad}_4^3\right] \leq \frac{q(q-1)}{2 \cdot 2^{2n}} + \mathbb{P}\left[\mathsf{bad}^6\right]$

Proof. Firstly, let us simplify the rather complex $\mathcal{W}_3$, by removing any code that cannot possible assist in setting $\mathsf{bad}_3 \vee \mathsf{bad}_4$. Since uniform sampling commutes with applying permutations, we can modify the decryption algorithm to sample X_1 directly, (rather than X_2). Making the simplifications and this change yields $\mathcal{W}_5$. Thus, $!\mathbb{P}\left[\mathsf{bad}_3^3 \vee \mathsf{bad}_4^3\right] = \mathbb{P}\left[\mathsf{bad}_3^5 \vee \mathsf{bad}_4^5\right]$.

Now, since the encryption oracle of $\mathcal{W}_5$ does not return anything, the adversary learns nothing from making Enc queries. Thus for any adversary there exists an equivalent one that makes his q_D decryption queries first (since these may affect his future inputs), and then makes $q_E = q - q_D$ encryption queries. It is this adversary we shall consider.

So, bad_4 can only be set as the result of two colliding Dec queries. bad_4 is set if two different calls to Dec repeat the pair (A_2, B_2). Two values of B_2 collide if and only if they were decrypted from colliding

B_3 values. Now, $B_3 = \mathcal{D}_3^{\mathcal{D}_3(X_4||A_4)}(B_4)$ is the output of an online cipher call for which the tweak–input pair must be unique (because the adversary never repeats $M = X_4||A_4||B_4$). So, by the same logic as Lemma 9, colliding values of B_3 (and thus B_2) is at least as hard as colliding independent uniformly random n-bit strings. Moreover, colliding values of A_2 is precisely that of colliding a one block uniform string, and is independent of the probability of colliding B_2.
So, $\mathbb{P}[\mathsf{bad}_4] = \mathbb{P}[\mathsf{Collide}\,(A_2, B_2)] \leq \frac{q_D(q_D-1)}{2 \cdot 2^{2n}}$.

Next, let us consider bad_3. This is set when either the variables colliding between two different encryption queries or during an encryption query they collide with those from a decryption query. The second of these is precisely the game described in $\mathcal{W}_8$, and as such is bounded by $\mathbb{P}\left[\mathsf{bad}^8\right]$. The pair (A_2, B_2) is set by $A_2||B_2 \leftarrow \mathcal{E}_1^{X_1}(A_1||B_1)$. Since the adversary never repeats inputs, we can again repeat the logic of Lemma 9, bounding this event by $\frac{q_E(q_E-1)}{2 \cdot 2^{2n}}$.

Summing up,

$$\mathbb{P}\left[\mathsf{bad}_3^3 \vee \mathsf{bad}_4^3\right] = \mathbb{P}\left[\mathsf{bad}_3^5 \vee \mathsf{bad}_4^5\right] \leq \frac{q_D(q_D-1)}{2 \cdot 2^{2n}} + \frac{q_E(q_E-1)}{2 \cdot 2^{2n}} + \mathbb{P}\left[\mathsf{bad}^6\right] \leq \frac{q(q-1)}{2 \cdot 2^{2n}} + \mathbb{P}\left[\mathsf{bad}^6\right]$$

This simplifies to the claimed bound since $q_D(q_D-1) + q_E(q_E-1) \leq q(q-1)$. ∎

Claim (5). For an adversary making q_D decryption queries followed by q_E encryption queries and any $\kappa \in \mathbb{N}$, $\mathbb{P}\left[\mathsf{bad}^6\right] \leq (\frac{q_D}{2^n})^\kappa \frac{2^n}{(\kappa+1)!} + \frac{q_E \cdot \kappa}{2^n}$

Proof. Let us consider the actions an adversary will take (given the information he knows from his decryption queries) to decide which encryption queries to make. The output he has (a series of $X_1||A_1$ strings) can be used to force a collision on the $A_2 = \mathcal{E}_1^{X_1}(A_1)$ value with one from a Dec query, but if he does this he does not know anything about the possible value of the corresponding B_2 value. Since he is never provided with the output of an $\mathcal{E}_1$ call, he cannot use this information to assist him in colliding on B_2 values. As such, there is no more effective strategy than ensuring he collides A_2 values and hopes to be lucky and collide the B_2 value.

So, assuming the adversary can always collide the A_2 component, what is the probability he succeeds in setting bad on any particular query? A B_2–collision between any single encryption $X_e||A_e||B_e$ and some decryption query $X_d||A_d||B_e$ occurs if $\mathcal{D}_2(\mathcal{D}_3^{X_d||A_d}(B_d)) = \mathcal{E}_1^{X_e||A_e}(B_e)$. Since $\mathcal{E}_1$ is a secure online cipher (and independent of $\mathcal{D}_2, \mathcal{D}_3$), this happens simply with the probability of colliding two independently uniformly sampled values: $1/2^n$. Thus for each encryption query, the probability that the adversary manages to set bad can be upper bounded (via union bound) by the number of B_2 values corresponding to the appropriate A_2 bound. Assuming $\#\mathcal{L}_C[A_2] \leq \kappa$, we can union bound once again (this time across all encryption queries) to deduce the probability an adversary triggers bad is at most $\mathbb{P}[\mathsf{bad} \mid \#\mathcal{L}_C[A_2] \leq \kappa] \leq \kappa \cdot q_E \cdot 2^{-n}$.

Consider then the probability that any of the $\mathcal{L}_C[x]$ contains more than κ elements. This corresponds to the number of times an A_2 value repeats during the adversary's Dec queries. Since A_2 is independently uniformly sampled by each Dec query, it follows from a standard balls and bins argument that for any $x \in \{0,1\}^n$, $\mathbb{P}[\#\mathcal{L}_C[x] > \kappa] \leq \left(\frac{q_D}{2^n}\right)^\kappa \frac{1}{(\kappa+1)!}$. Union bounding across all x, $\mathbb{P}[\exists x : \#\mathcal{L}_C[a] \geq \kappa] \leq (\frac{q_D}{2^n})^\kappa \frac{2^n}{(\kappa+1)!}$. Summing this with the bound from the previous paragraph proves the claim. ∎

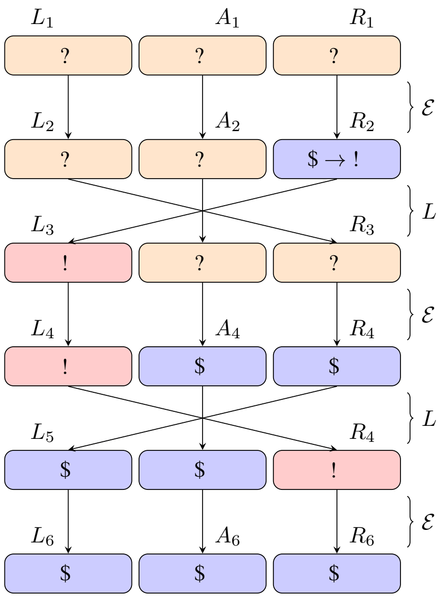
Claim (6). Combining these results, we have proven the stated result.

Proof. Firstly, combining the results from Claims (4) and (5),

$$\mathbb{P}\left[\mathsf{bad}_3^3 \vee \mathsf{bad}_4^3\right] \leq \frac{q(q-1)}{2 \cdot 2^{2n}} + \left(\frac{q_D}{2^n}\right)^\kappa \frac{2^n}{(\kappa+1)!} + \frac{\kappa \cdot q_E}{2^n}$$

So, applying the symmetry result of Claim (3),

$$\mathbb{P}\left[\bigvee_{i=1}^{4} \mathsf{bad}_i^3\right] \leq \frac{q(q-1)}{2^{2n}} + \frac{{q_D}^\kappa + {q_E}^\kappa}{2^{n\kappa}} \cdot \frac{2^n}{(\kappa+1)!} + \frac{\kappa \cdot q}{2^n}$$



Key:

? Strings of blocks whose distribution is unknown/irrelevant.

! Strings of k blocks that do not repeat too frequently.

\$ Strings of blocks that are uniformly sampled.

$\$ \to !$ Strings of blocks that are uniformly sampled and so don't repeat too frequently.

Intuition: In this diagram, the units are strings of blocks, rather than simply blocks as in others, with first and last boxes on each line k blocks long, the middle as appropriate.

Since inputs are unique, the final k blocks from the first call are almost certainly unique. Thus the second call has a unique prefix, so samples independently and uniformly. This means the majority of the output is sampled uniformly at random, and since random k block string is (with high probability) unique, the final output blocks have a unique prefix and so are sampled as required.

Fig. 5. Intuition behind the $\pm$PRP security of Π_3^{rev} (Theorem 14)

The second term in this can be upper bounded since ${q_D}^\kappa + {q_E}^\kappa \le q^\kappa$. Finally, we pull the whole bound together using Claims (1) and (2):

$$\begin{aligned}\mathbf{Adv}_{\Pi_3}^{\pm\mathsf{prp}}(q) \le \mathbb{P}\left[\mathsf{bad}^3\right] &\le \mathbb{P}\left[\mathsf{bad}_5^3 \vee \mathsf{bad}_6^3\right] + \mathbb{P}\left[\vee_{i=1}^{i=4}\mathsf{bad}_i^3\right]\\ &\le \frac{q(q-1)}{2\cdot 2^{mn}} + \frac{q(q-1)}{2^{2n}} + \left(\frac{q}{2^n}\right)^\kappa \cdot \frac{2^n}{(\kappa+1)!} + \frac{\kappa\cdot q}{2^n}\\ &\le 1.5\frac{q(q-1)}{2^{2n}} + \frac{\kappa\cdot q}{2^n} + \left(\frac{q}{2^n}\right)^\kappa \frac{2^n}{(\kappa+1)!}\end{aligned}$$

This completes the proof of the more specific result. ■

To write this more succinctly, suppose $q/2^n \le \frac{1}{\alpha}$ for some $\alpha > 0$. Then,

$$\mathbf{Adv}_{\Pi_3^L}^{\pm\mathsf{prp}} \le \frac{q}{2^n}\left[\frac{3}{2}\cdot\frac{1}{\alpha} + \kappa + \frac{1}{\alpha^{\kappa-1}}\cdot\frac{2^n}{(\kappa+1)!}\right] \le \frac{q}{2^n}\left[\frac{3}{2\alpha} + \kappa + \frac{e^\kappa\cdot 2^n}{\alpha^{\kappa-1}(\kappa+1)^{\kappa+1}}\right]. \tag{1}$$

Let us also assume $n \ge 4$, since blocksize security is meaningless if the blocks are this small anyway, and set $\kappa = n-1$ and $\alpha = 4$. This means $\alpha^{\kappa-1} > e^\kappa$, and $(\kappa+1)^{\kappa+1} \ge 4^n \ge 2\cdot 2^n$, and thus the final term is upper bounded by $\frac{1}{2}$. Thus the overall bound is less than $n\frac{q}{2^n}$. We observe that if $q/2^n > 1/\alpha$, this bound is vacuously true, completing the theorem. □

For any given n, if one wishes to find the maximal q such that $\mathbf{Adv}_{\Pi_3^L}^{\pm\mathsf{prp}}$ is still sufficiently small, this can be done by numerically selected (κ,α) to optimise Equation 1. In the common case of $n = 128$, putting $\kappa = 19$ and $\alpha = 16$ provides $\mathbf{Adv}_{\Pi_3^L}^{\pm\mathsf{prp}}(q) \le q2^{-123.7}$.

4.3 Three layer reverse: a $\pm$PRP beyond the blocksize

So, the Π_3^{rev} construction is a secure $\pm$PRP up until roughly $2^{n-\log(n)}$ queries, but if n is small this might not suffice. We address this shortcoming by proving that the scheme is in fact arbitrarily secure, as long as messages are sufficiently long. The intuition behind the proof is presented in Figure 5.

Theorem 14. *If all messages are at least $2k$ blocks long, the adversarial advantage of distinguishing $\Pi_3 = \Pi_3^{\mathsf{rev}}$ from an $\pm$PRP within q queries is* $\mathbf{Adv}_{\Pi_3}^{\pm\mathsf{prp}}(q) \le \frac{3q(q+1)}{2^{kn}}$.

Proof (Of Thm. 14). We will follow a sequence of worlds, depicted in Appendix C.4. From $\mathcal{W}_0$ to $\mathcal{W}_5$, we transition from the Π construction to a ±PRF. Then, Lemma 15 will use the further games to bound the probability of a bad event. As before, we let bad_i^j be the event that the adversary sets flag bad_i whilst interacting with world $\mathcal{W}_j$ and $\mathsf{bad}^j := \vee_i \mathsf{bad}_i^j$.

Let $\overline{M} := \mathsf{rev}(M)$ be the blockwise reversal of a string. In every oracle, setting $m = |M|$, we split M into strings $M = L_1||A_1||R_1$, notation we continue through each layer of the construction such that $|L_i| = |R_i| = kn$ and $|A_i| = (m-2k)n$ for all $i \in \{1, \ldots, 6\}$. The general concept behind the proof is represented by Figure 5, with strings have been labelled where practical.

World $\mathcal{W}_0$ precisely encodes the Π_3^{rev} construction. Moreover, $\mathcal{W}_0$ and $\mathcal{W}_1$ are perfectly indistinguishable, since the only difference is in the expansion of the encryption calls, the introduction of a list $\mathcal{P}$ and introducing bad flags.

Worlds $\mathcal{W}_1$ and $\mathcal{W}_2$ are indistinguishable until one of the bad flags is set. This is because, until one of the four flags is set, every query made by the second or third internal online cipher calls has a unique prefix. Thus until bad is set each online cipher call samples its output uniformly from all strings of the appropriate length.

The worlds $\mathcal{W}_2$ through to $\mathcal{W}_5$ all encode the same two oracles, albeit with slightly naming for the internal variables and modified code order. As such they behave identically, including the setting of the various bad flags.

Now, although it might not be immediately clear, $\mathcal{W}_5$ is a ±PRF. The return value of Dec queries is uniformly sampled on Line 33.21, and the Enc oracle uniformly samples the majority of its output on Line 33.17. The remaining output is simply the image of a uniformly sampled variable (L_5, Line 33.11) under a permutation, which is itself uniformly sampled. Thus both oracles output is uniformly and independently sampled: a ±PRF.

All together then,

$$\begin{aligned}\mathbf{Adv}_{\Pi_3}^{\pm\mathsf{prp}} &\leq \Delta_{\mathcal{W}_0}^{\Pi_3} + \Delta_{\mathcal{W}_1}^{\mathcal{W}_0} + \Delta_{\mathcal{W}_2}^{\mathcal{W}_1} + \Delta_{\mathcal{W}_5}^{\mathcal{W}_2} + \Delta_{\pm\mathsf{prf}}^{\mathcal{W}_5} + \Delta_{\pm\mathsf{prp}}^{\pm\mathsf{prf}} \\ &\leq 0 + 0 + \Pr[\mathsf{bad}^2] + 0 + 0 + \Delta_{\pm\mathsf{prp}}^{\pm\mathsf{prf}}\end{aligned}$$

Since $\mathcal{W}_2$ and $\mathcal{W}_5$ behave identically, $\mathbb{P}\left[\mathsf{bad}^2\right] = \mathbb{P}\left[\mathsf{bad}^5\right]$. The final term, $\Delta_{\pm\mathsf{prp}}^{\pm\mathsf{prf}}$, is simply the ±PRF–±PRP switch, as bounded in Lemma 5. Similarly, we can bound $\Pr[\mathsf{bad}^5]$ by Lemma 15. So, assuming $q \leq 2^{kn}$ (which we require for the final inequality),

$$\mathbf{Adv}_{\Pi_3}^{\pm\mathsf{prp}}(q) \leq \Pr[\mathsf{bad}^5] + \Delta_{\pm\mathsf{prp}}^{\pm\mathsf{prf}} \leq \frac{q}{2^{kn}}\left[(3q+1) + \frac{q-1}{2^{kn+1}}\right] \leq \frac{q(3q+2)}{2^{kn+1}}.$$

Noting this bound is vacuously true for $q \geq 2^{kn}$, we drop the limitation on q and simplify slightly to reach the general result. □

Lemma 15. *With high probability, the event* bad^5 *does not occur. Explicitly,* $\mathbb{P}\left[\mathsf{bad}^5\right] \leq q(3q+1)2^{-kn}$

Proof. We split the problem up using $\mathbb{P}\left[\mathsf{bad}^5\right] \leq \mathbb{P}\left[\mathsf{bad}_2^5 \vee \mathsf{bad}_4^5\right] + \mathbb{P}\left[\mathsf{bad}_1^5 \vee \mathsf{bad}_3^5\right]$.

Consider the first of these terms. The event bad_2^5 occurs if during an encryption query $L_5 \in \mathcal{P}$ on Line 33.13. Since L_5 is uniformly sampled, this occurs with probability $|\mathcal{P}| \cdot 2^{-kn}$, and similarly for decryption queries setting bad_4^5 on Line 33.32. Since every query increases $|\mathcal{P}|$ by 3, we observe that when the decision point is reached on query i, $|P| = 3i - 1$. As each query is either encryption or decryption, and the probability of each event is the same and independent of input, the overall probability can be easily bounded by applying a union bound. Thus $\mathbb{P}\left[\mathsf{bad}_2^5 \vee \mathsf{bad}_4^5\right] = \sum_{i=1}^{i=q}(3i-1)2^{-kn} = q(3q+1)2^{-(kn+1)}$.

The other term is rather more complicated to bound. Departing from the indistinguishability games, we provide a sequence of games where the adversary is given oracle access to a world $\mathcal{W}_j$ and seeks to trigger the event $\mathsf{bad}^j = \mathsf{bad}_1^j \vee \mathsf{bad}_3^j$. They are again provided in Appendix C.4, and continue the notation from Theorem 14.

Since we no longer care about indistinguishability, $\mathcal{W}_6$ dispenses with the uniformly sampled outputs of $\mathcal{W}_5$, restricting its output to variables used elsewhere. It also removes the code for setting the other now irrelevant flags. Moreover, we now give the adversary direct control over the randomly sampled elements that were added to the list $\mathcal{P}$. As such, by randomly choosing these values, the adversary may simulate $\mathcal{W}_5$ when given access to $\mathcal{W}_6$. Thus any strategy for setting bad^5 may be used to set bad^6, and so $\mathbb{P}\left[\mathsf{bad}_5\right] \leq \mathbb{P}\left[\mathsf{bad}^6\right]$.

Transitioning from $\mathcal{W}_6$ to $\mathcal{W}_7$ we abstract out the remaining output into separate oracles, to which we provide the adversary with arbitrary access. Given these $\mathcal{E}^\epsilon$ and $\mathcal{D}^\epsilon$ oracles, the adversary is able to simulate $\mathcal{W}_6$ with $\mathcal{W}_7$, and thus $\mathbb{P}\left[\mathsf{bad}^6\right] \leq \mathbb{P}\left[\mathsf{bad}^7\right]$.

In world $\mathcal{W}_8$, rather than keeping track of $\mathcal{P}$, we keep track of each element's blockwise reversal. This makes no difference for adversarially supplied elements (since he is aware of this), but simplifies the Enc oracle. Moreover, defining $x' := \overline{\mathcal{D}^\epsilon(\overline{x})}$ for any kn bit string x, we may simplify the Dec oracle also. Since the adversary has arbitrary oracle access to $\mathcal{D}^\epsilon$, $x \mapsto x'$ should be considered a public permutation, something we imply by denoting it through notation rather than a function call.

Ignoring calculation of x' then, we consider what value the extra oracles actually provide. They are limited to kn bit inputs, but all remaining online cipher calls used in the construction have a prefix of at least kn bits, and so their outputs are independent of those from the main $\mathcal{E}, \mathcal{D}$ oracles. So, the only place might may be of use is choosing which prefix to query under. Thus we give the adversary direct choice over which prefixes to query under, which can only assist the adversary. After these switches, the $\mathcal{E}^\epsilon$ and $\mathcal{D}^\epsilon$ oracles can no longer help the adversary, and so their removal will not inhibit the adversary. Overall then, $\mathbb{P}\left[\mathsf{bad}^7\right] = \mathbb{P}\left[\mathsf{bad}^8\right]$.

Since the adversary receives no output from their queries, adaptivity does not help in triggering bad^8. Thus we may assume he is non-adaptive, and this setting of the problem is presented by $\mathcal{W}_9$. The adversary submits a list $\mathcal{P}$ of elements he hopes to collide with (as per V_1, V_5 input variables in $\mathcal{W}_8$), along with lists of encryption and decryption queries in the form of $\mathcal{M}_E, \mathcal{M}_D$ and an order in which he wishes these queries be evaluated order. The non-adaptive game may then precompute all the values $\mathcal{R}_i$ from encryption queries and $\mathcal{L}_i$ from decryption queries. Having done so, we check for any collisions between these values, or between an encryption/decryption value and a value suitably early in the list $\mathcal{P}$.

Thus it remains to calculate $\mathbb{P}\left[\mathsf{bad}^9\right]$ explicitly, which we do through liberal use of the union bound.

Firstly, as per earlier proofs, we observe that no collisions may occur within either the list $\mathcal{R}$ or $\mathcal{L}$ from two elements encrypted (or decrypted) under the same prefix. If the prefix is different, the online ciphers are independent, and thus collisions occur simply with the probability of colliding two uniformly sampled elements: 2^{-kn}. Moreover, the probability of an element from $\mathcal{R}||\mathcal{L}$ being equal to any particular element of $\mathcal{P}$ is also 2^{-kn}.

Finally, we are left to bound the probability that an element occurs in both $\mathcal{L}$ and $\mathcal{R}$. Let $M_e = L_e||A_e||R_e$ and $M_d = L_d||A_d||R_e$ be a pair of encryption and decryption queries, and set $R = \mathcal{E}^{L_e||A_e}(R_e)$ and $L = \pi(\mathcal{D}_*^{L_d||A_d}(R_d))$. If the prefixes (that is, the values of $L_e||A_e$ and $L_d||A_d$) are different, then the encryption and decryption queries are independent, and so L independent of R. Thus the probability of $L = R$ is the probability of L being equal to some k block string: 2^{-kn}. If the prefixes are the same, both equal to some $T = L_e||A_e$ then

$$L = R \iff \mathcal{E}^T(R_e) = \pi(\mathcal{D}_*^T(R_d))) \iff R_d = \mathcal{E}^t(\pi^{-1}(\mathcal{E}^t(R_d)))$$

Since $\mathcal{E}$ is a secure online cipher and the adversary is non-adaptive, this also occurs with the probability of hitting an arbitrary element at random. Thus in fact, each of these individual events occurs with probability at most 2^{-kn}, so it remains just to count the total number of events listed and bound the overall value:

$$\mathbb{P}\left[\mathsf{bad}_1^5 \vee \mathsf{bad}_3^5\right] \leq \mathbb{P}\left[\mathsf{bad}^9\right] \leq \frac{1}{2^{kn}}\left[\binom{\#(\mathcal{L}||\mathcal{R})}{2} + \sum_{i=0}^{q} 2i\right] \leq \frac{1}{2^{kn}}\left[\binom{q}{2} + q(q+1)\right] = \frac{q(3q+1)}{2^{kn+1}}.$$

Combining this with the bound on the first term completes the proof. □

Extension through reduction: security with many layers. For completeness, we also consider what can be achieved with many layers. Since the Π_3^{rev} already provides beyond birthday bound security (and beyond blocksize security when messages are long enough), there is little utility in deriving ever higher security bounds. Instead, we provide an explicit reduction from many round cases to the smaller versions already studied, at the cost of requiring the ciphers be independent.

Lemma 16. *When the online ciphers are independent, the Π_i^L construction is no more distinguishable from a PRP or $\pm$PRP than Π_{i-1}^L.*

Proof (Of Lem. 16). Suppose there exists some adversary A who distinguishes Π_i^L from a PRP (or $\pm$PRP). Let us construct an adversary B who distinguishes Π_{i-1}^L from a PRP (or $\pm$PRP).

Let $\mathcal{O}_e$ be the encryption oracle B is provided with (which may be real or random). B chooses $\mathcal{E}_B$ uniformly from all online ciphers, and then simulates the Π_i^L encryption oracle with $\tilde{\Pi}_i^L := \mathcal{E}_B \circ L \circ \mathcal{O}_e$ (and similarly for the decryption oracle in the $\pm$PRP case).

The composition of a random permutation with an online cipher (itself a permutation) is again a random permutation. Thus $\tilde{\Pi}_i^L$ exactly simulates Π_i^L if $\mathcal{O}_e$ was the cipher or is a random permutation if $\mathcal{O}_e$ was. Therefore, B can run A against $\tilde{\Pi}_i^L$ and forward As result as his own, distinguishing with the same success probabilities A. □

5 Right Shifting towards a PRP

Two obvious candidates for the linear layer are the right and left rotations by one block, We denote them right and left respectively, and clearly they are each other's inverses. For messages of at least $i+1$ blocks, Π_i^{left} is not a PRP, since the first output block cannot possibly depend on the final input block. Moreover, this means Π_i^{right} cannot be an $\pm$PRP, since its inverse is the Π_i^{left} scheme instantiated around $\mathcal{D}$. Indeed, for any linear layer L, Π_2^L cannot be a secure PRP if the linear layer's first output block is independent of the final input block.[4]

Combining this limitation with Thm. 10 (two layer constructions cannot be indistinguishable from a PRP with beyond birthday bound security), at best Π_2^{right} is a PRP up to the birthday bound. We show this to be the case, and is formalised by Thm. 17. The proof is a direct simplification of that for Theorem 11 and so omitted.

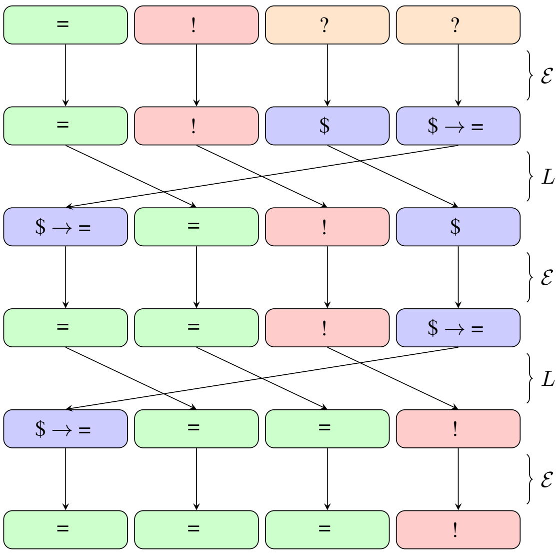
Theorem 17. *Let L be an invertible linear layer that satisfies $L(M[1]||\cdots||M[m])[1] = M[m]$, such as* right. *Then, the Π_2^L construction is indistinguishable from a PRP up to the birthday bound. Explicitly,* $\mathbf{Adv}^{\mathsf{prp}}_{\Pi_2^L}(q) \leq 3\frac{q(q-1)}{2^{n+1}}$.

Before considering what security is achieved with more layers, we observe that there exists an attack against the whole family of Π_i^{right} constructions. If an ideal online cipher is called with two messages that differ before the final block, the final ciphertext blocks are independently sampled. Now, if these independent random variables collide (which is likely to occur roughly every $2^{n/2}$ queries) the right layer will simply add a common prefix to both messages. From this observation, we build a distinguisher against Π_i^{right} (see proof of Lem. 18), following the logic shown in Figure 6.

Lemma 18. *With $\Pi = \Pi_i^{\mathsf{right}}$ for some $i \geq 2$, as long as messages contain at least $\lfloor \frac{3}{2} i \rfloor$ blocks and $n \geq 2$, there exists an attack demonstrating $\mathbf{Adv}^{\mathsf{prp}}_{\Pi}(q) \geq \frac{q(q-1)}{8 \cdot 2^{(i-1)n}}$ for any $q \leq 2^{\frac{i-1}{2}n}$.*

Proof (Of Lem. 18). Consider the adversary $\mathcal{A}$ that requests encryptions of malicious messages of the form $M_t = \langle 0 \rangle_1 || \langle t \rangle_a || \langle 0 \rangle_{i-1}$, with $a \geq \lceil (i-1)/2 \rceil = \lfloor i/2 \rfloor$ chosen to be minimal such that minimum message lengths are met. He will vary t, allowing him up to $2^a \geq 2^{\frac{i-1}{2}n}$ possible queries of this form (hence the bound on q). After making his queries, he returns 1 if there were two queries for which the ciphertexts began with the same i blocks. We claim $\mathcal{A}$ successfully distinguishes Π from an ideal PRP.

[4] At least, as long as messages are less than i blocks long



Key:

- = Blocks that are the same across all queries.
- ! Blocks that are unique across all queries.
- ? Blocks that may take any value.
- \$ Blocks that are uniformly sampled.
- \$ →= Blocks that are uniformly sampled, but that we expect to collide if we make enough queries.

Intuition: By making enough queries, we expect that the final blocks being uniformly sampled, will collide. If both layers collide at the same time, we end up building a surprisingly long prefix of colliding blocks, which is easily detectable from the output. Repeating the process, we can extend the attack to any number of rounds.

Fig. 6. An attack against Π_3^{right}

We first bound $\mathbb{P}\left[\mathcal{A}^{\Pi} \to 1\right]$, by considering the internal variables when encrypting M_t. During the first round, the first block is identical across all queries, and so encrypts to an identical value: $\mathcal{E}^{\epsilon}(0)$. Since encryption is an online permutation, the online encryption of the first $a+1$ blocks must be unique among all queries, since the counter was. As the first block is identical throughout, this in turn means the next a blocks must be unique amongst all queries. Given this unique prefix, the encryptions of the final $(i-1)$ blocks, $\mathcal{E}^{\langle 0\rangle_1 || \langle t\rangle_a}(\langle 0\rangle_{i-1})$ are independently uniformly sampled.

Notice that with precisely the probability of colliding two strings of n random bits, we have a collision on the final block. Thus, after the first linear layer, in which we shift the final block to the start, with this same probability there exist queries in which the first two blocks repeat. Since this output again consists of some repeated blocks, a unique section and then some arbitrary blocks, we may apply similar analysis. We do this for all but the final layer, albeit noting on round r there are now r repeated blocks rather than one, and $i-r$ arbitrary blocks after the unique section.

So, with the probability of colliding the independent and uniformly sampled final blocks on each of the first $(i-1)$, there are two queries for which the final block inputs collide on the first i blocks. As the cipher is online, this leads to an i block collision in the output, triggering $\mathcal{A} \to 1$. So, $\mathbb{P}\left[\mathcal{A}^{\Pi} \to 1\right]$ is at least this probability. Since the variables are independently sampled, this is equivalent to colliding a string of $(i-1)n$ independently sampled random bits, and so $\mathbb{P}\left[\mathcal{A}^{\Pi} \to 1\right] \geq \frac{q(q-1)}{2^{(i-1)n+2}}$.

Alternatively, consider $\mathbb{P}\left[\mathcal{A}^{\mathcal{W}[\mathrm{Perm}]} \to 1\right]$, the probability of getting a collision on the first i blocks of output from distinct calls to the ideal cipher. This is upper bounded by the probability of colliding outputs from the equivalent random function, which is simply the probability of colliding $i \cdot n$ random bits. Thus $\mathbb{P}\left[\mathcal{A}^{\mathcal{W}[\mathrm{Perm}]} \to 1\right] \leq \frac{q(q-1)}{2^{in+1}}$.

Combining these results,

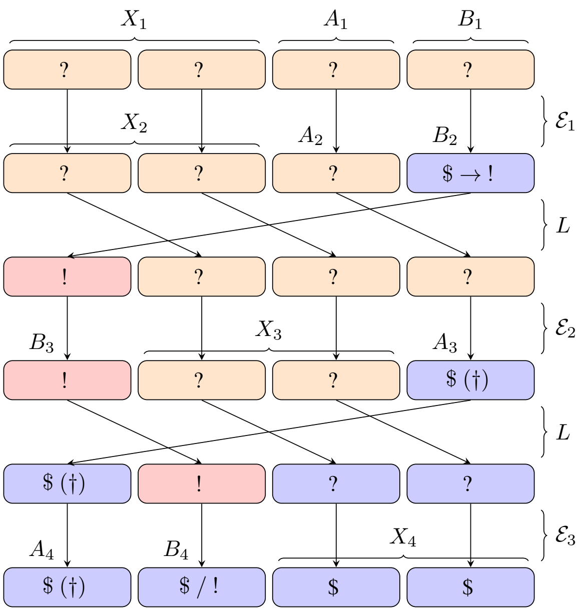
$$\mathbf{Adv}_{\Pi}^{\mathsf{prp}}(q) \geq \Delta_{\Pi}^{\mathcal{W}[\mathrm{Perm}]}(\mathcal{A}) \geq \frac{q(q-1)}{2^{(i-1)n+2}} - \frac{q(q-1)}{2^{in+1}} \geq \frac{q(q-1)}{2^{(i-1)n+2}}\left[1 - \frac{1}{2^{n-1}}\right]$$

Applying the hypothesis that $n \geq 2$, we bound $1 - 2^{-(n-1)} \geq 2^{-1}$ to yield the stated result. □

We note that, by increasing a (and thus requiring greater message lengths) we may increase the number of queries of this form, such that the attack succeeds with overwhelming probability.

5.1 Three layer shift: a PRP to almost blocksize

As with the two layer version, $\Pi = \Pi_3^{\mathsf{right}}$ cannot hope to achieve good $\pm$PRP security, since its inverse is trivially distinguishable. Applying Lem. 18, we see that for $n \geq 2$ and messages of length at least 4 blocks, then $\mathbf{Adv}_{\Pi}^{\mathsf{prp}}(q) \geq \frac{q(q-1)}{8 \cdot 2^{2n}} \approx (\frac{q}{2^n})^2$ for $q \leq 2^n$. Thus the best we can reasonably expect is PRP security up to the blocksize, something we achieve, asymptotically matching the attack. Again, we present the logic in a diagram (Fig. 7).



Key:

? Blocks whose distribution is unknown/irrelevant.

! Blocks that do not repeat too frequently.

\$ Blocks that are uniformly sampled.

\$ $\to$! Block that is uniformly sampled and so doesn't repeat too frequently.

\$ (†) Blocks that are (perhaps surprisingly) independently uniformly sampled.

Intuition: To achieve beyond birthday bound security, we can no longer fail if single block collisions occur. Instead, we observe that, since the ! block does not repeat "too frequently", we can perform a PRP–PRF switch on the final block of the second call. Thus the block (†) is sampled uniformly, and crucially independently of !, leading to the result.

Then, since the first two blocks input to the final call almost never repeat (as a pair), the remaining blocks are uniform.

Since the linear layer is just a reordering, we use the same labels as on the previous line.

Fig. 7. Intuition behind the PRP security of Π_3^{right} (Theorem 19)

Theorem 19. *The $\Pi = \Pi_3^{\mathsf{right}}$ construction with independent online ciphers is a PRP, where $\mathbf{Adv}_{\Pi}^{\mathsf{prp}}(q) \leq \frac{q(q-1)}{2^{2n}}$.*

Proof. We begin by splitting the adversary's oracle in two: one that answers single block messages and one that answers longer messages. Clearly this has no effect on the advantage, being just a notational convenience. Since an online cipher on a single block is a random permutation, we have that (when restricted to single block queries) $\Delta_{\Pi}^{\mathrm{Perm}(n)}(q) = 0$.

So, if the first oracle is restricted to messages of one block and the second to messages of at least 2 blocks, with \$ denoting the appropriate random permutation,

$$\mathbf{Adv}_{\Pi}^{\mathsf{prp}}(q) = \Delta_{\$\,,\,\$}^{\Pi,\Pi}(q) \leq \Delta_{\Pi,\,\$}^{\Pi,\Pi}(q) + \Delta_{\$\,,\,\$}^{\Pi,\,\$}(q) = \Delta_{\Pi,\,\$}^{\Pi,\Pi}(q) + 0.$$

Thus it remains just to bound this final term. It roughly corresponds to $\mathbf{Adv}_{\Pi}^{\mathsf{prp}}(q)$ if all queries have length at least two blocks, except that the adversary is also given access to an oracle that simulates the a one-block OPRP. It happens that this oracle is of no use to an adversary, since it is independent from any of the important switches or calls made during the proof, so for conciseness we omit it from our notations and games.

To provide the result, we split the internal variables into three sections. In this case, we do so such that for an m block message M, we have $|A_i| = |B_i| = 1$ and $|X_i| = m - 2$, for all $i \in \{1, \ldots, 4\}$. We will start off with $M = X_1||A_1||B_1$, and will "track" the ordering of these sections through the linear layers. Each time the cipher is called, the appropriate blocks of its output will be labelled by the same letter (after incrementing i), meaning the ciphertext $C = A_4||B_4||X_4$. This labelling, and the logic described below, is represented in Figure 7, which shows how we convert from the Π_3^{right} scheme to a random function, a standard switch away from a random permutation.

To formalise this, we use a sequence of game–hops, comparing 10 oracles ($\mathcal{O}_0$ to $\mathcal{O}_9$), presented in Appendix C.2. While most of the transitions are simply bookkeeping, three will make significant changes.

The key observation behind the proof will be that the prefix $B_2||X_2$ for the second layers final block (encrypting $A_2 \to A_3$) does not repeat "too frequently". As such, we can perform a (tweakable) PRP–PRF switch on the final block of the second layer and simplify the construction (the transition $\mathcal{O}_1 \to \mathcal{O}_2$). After this, we observe a similar situation occurs when encrypting $B_3 \to B_4$, and so perform a PRP–PRF switch here also (the transition $\mathcal{O}_4 \to \mathcal{O}_5$). Having done so, we reach a scheme that is indistinguishable from a PRF, and so a final switch completes the sequence. To formalise this process, we will use a sequence of identical until bad transitions, keeping track of various bad flags, to bound the difference between games. As such, we define bad_i^j to be the event that flag bad_i is set by oracle j.

To begin, $\mathcal{O}_0$ directly encodes the Π construction, evaluating the linear layer implicitly by reordering the blocks. It is perfectly indistinguishable from $\mathcal{O}_1$, in which we partially expand the online cipher calls. We continue by comparing sequential pairs of oracles.

In $\mathcal{O}_2$, we replace a tweakable PRP of $\mathcal{O}_1$ with a tweakable PRF. A tweakable PRF is simply a family of PRFs indexed by the tweak, and so these two constructions are identical until there is an output collision on one of the PRFs. Thus, these function calls are equivalent until $\mathcal{O}_2$ repeats a tweak-output pair, which corresponds to the internal variables $(B_2||X_2, A_3)$. This is precisely the condition required to set bad_1, and thus the oracles are perfectly indistinguishable until the event bad_1^2 occurs[5].

There are three key differences between $\mathcal{O}_2$ and $\mathcal{O}_3$. Unlike $\mathcal{O}_2$, $\mathcal{O}_3$ samples A_3 and X_4 directly, and keeps track of a different set of strings. Firstly, sampling A_3 uniformly on Line 14.7 is perfectly indistinguishable, because the adversary always makes unique queries. As such, the string $B_2||X_2||A_2$ (which is the image of M under a permutation) must take a unique value on each query. Thus the random function $F^{B_2||X_2}$ is never called with the same parameter value A_2, which means all outputs are uniformly sampled. Secondly, until bad_1 is set the pair (B_2, A_3) have not repeated. Thus $(B_3||A_3)$ cannot have repeated (B_3 is the image of B_2 under a permutation), and so X_4 is set by an online cipher with a unique prefix, which Corollary 8 tells us is uniformly sampled. Finally, instead of setting bad_1 if there is a collision on the list of strings $B_2||X_2$, $\mathcal{O}_3$ keeps track of strings B_2. Since B_2 is clearly a substring of $B_2||X_2$, a collision on $B_2||X_2$ implies a collision on B_2. Thus any strategy that causes event bad_1^2 may be used to trigger bad_1^3. Thus $\mathbb{P}[\mathsf{bad}_2] \leq \mathbb{P}[\mathsf{bad}_3]$, and the oracles are adversarially indistinguishable until this event occurs.

Oracles $\mathcal{O}_3$ and $\mathcal{O}_4$ are perfectly indistinguishable, since the only difference is a removal of superfluous internal variables and reordering of code. Also, $\mathcal{O}_4$ and $\mathcal{O}_5$ are perfectly indistinguishable until bad_2 is set, for exactly the same logic as the transition from $\mathcal{O}_1$ to $\mathcal{O}_2$. Thus in $\mathcal{O}_5$, B_4 is set as the output of a tweakable random function, tweaked by A_3.

The difference from $\mathcal{O}_5$ to $\mathcal{O}_6$ is that we replace the random function with a uniform sampler. This would be distinguishable only if the adversary queries the random function with the same tweak and parameter, which means repeating the pair (A_3, B_3). However, since values of B_3 are in bijection with those of B_2, this is equivalent to repeating (A_3, B_2), which is the requirement to set bad_1. Thus until an input sequence that would trigger bad_1^6 occurs, the oracles $\mathcal{O}_5$ and $\mathcal{O}_6$ are perfectly indistinguishable. Moreover, any sequence of queries triggering bad_1^5 will trigger bad_1^6. Thus until the event bad_1^6 occurs, the oracles are equivalent.

Since A_4 is uniquely determined by A_3, keeping track of A_3 values is equivalent to keeping track of A_4. Moreover, rather than sampling A_3 and defining $A_4 \leftarrow \mathcal{E}_3^\epsilon(A_3)$, we may sample A_4 and set $A_3 \leftarrow \mathcal{D}_3^\epsilon(A_4)$. Performing these changes to $\mathcal{O}_6$ form $\mathcal{O}_7$, and thus have that $\mathcal{O}_6$ and $\mathcal{O}_7$ are perfectly indistinguishable.

[5] Formally, they are indistinguishable until a sequence of queries is made to the oracle that would trigger bad_1^2 if the oracle was $\mathcal{O}_2$

Set $\mathcal{O}_8$ to be simply a reordering and simplification of $\mathcal{O}_7$, which is done in such a way that $\mathcal{O}_8$ is clearly a random function. Thus $\mathcal{O}_7$ and $\mathcal{O}_8$ compute exactly the same function, and so are perfectly indistinguishable.

Oracle $\mathcal{O}_9$ is perfectly indistinguishable from an ideal cipher until bad_2^9 occurs (when two outputs collide). Clearly the outputs of $\mathcal{O}_8$ and $\mathcal{O}_9$ are indistinguishable, and any strategy that sets bad_2^9 may be used to set bad_2^8. Thus $\mathcal{O}_8$ is perfectly indistinguishable from an ideal cipher until bad_2^8 occurs.

Since the probability of bad occurring does not decrease in any of the transitions up to $\mathcal{O}_8$, wherever it makes sense $\mathbb{P}\left[\mathsf{bad}_i^j\right] \leq \mathbb{P}\left[\mathsf{bad}_i^8\right]$. So, putting these together and writing Collide X for the event that an adversary interacting with $\mathcal{O}_8$ can find two distinct inputs such that the variable X takes the same value during each calculation,

$$\begin{aligned}\mathbf{Adv}_{\Pi}^{\mathsf{prp}} \leq \Delta_{\mathcal{O}_8}^{\Pi} + \Delta_{\mathsf{prp}}^{\mathcal{O}_8} &\leq \mathbb{P}\left[\mathsf{bad}_1^8\right] + \mathbb{P}\left[\mathsf{bad}_2^8\right] \\ &= \mathbb{P}\left[\mathsf{Collide}\ (A_4||B_2)\right] + \mathbb{P}\left[\mathsf{Collide}\ (A_4||B_4)\right].\end{aligned}$$

Since both A_4 and B_4 are independently uniformly sampled, a the collision events on $A_4||B_4$ is simply birthday collision over a 2^{2n} block. Similarly, as discussed in Lemma 9, collision of independent random samples upper bounds the probability of colliding strings B_2. So, as A_4 is independent of B_2, the probability of Collide $A_4||B_2$ is also bounded by a birthday collision on a 2^{2n} block. Thus both values are bounded by the probability of this collision, namely $\frac{q(q-1)}{2^{2n+1}}$, and substituting this into the formula above yields the stated result. □

6 Conclusion

We have shown how one can efficiently turn an online cipher in a fully fledged cipher, using two types of mixing layer: reversing which leads to ±prp security, and a right cyclic shift, providing prp security. For birthday bound security using two calls to the online cipher suffices, whereas for close to blocksize security three calls are both necessary and sufficient. As far as we are aware, the construction of online ciphers with beyond birthday bound security itself is still an open problem. We hope our work will spur on the study of these versatile primitives.

Extensions and reformulations. Our results extend to tweakable online ciphers, forming tweakable ciphers with the tweaks and bounds of the non-tweak setting (this is mainly an exercise in notation). Similarly, our proofs can easily be adapted to cover a large set of mixing layers: in particular bit-,byte- or word-wise reversal maps can be used in place of blockwise reversal (for any word size dividing the block size). Generalising the results to cover incomplete final blocks should not be too arduous, although the notation becomes rather cumbersome.

Our characterisation of an online cipher (due to Bellare et al. [4]) is at its most general. The more specific definition of Rogaway et al. [33] additionally imposes a finite amount of state that the online cipher may use. Our results may be recast into this context by considering the state as a hash of the prefix, for the penalty of an adversary colliding two states.

There are several schemes for converting a true cipher into an authenticated encryption scheme (e.g. [6]), and even to achieve the recent, stronger goal of robust authenticated encryption [21]. By instantiating these modes with our construction, one can build a very secure scheme from an online cipher.

Further research. All our results are stated relative to an indistinguishability notion. A stronger notion is the indifferentiability framework [27], where an adversary would also have access to the online cipher itself (in addition to the cipher one attempts to construct). Indifferentiability is a much more challenging goal, and existing impossibility results relating to the self-composition of hash functions [13] extend to the PRP case of online ciphers (curiously, the ±PRP situation seems less straightforward). We provide a more detailed discussion in Appendix A.

From the CMC and EME constructions, it is clear that more involved mixing layers may reduce the security required of the cryptographic primitive. An interesting question is whether our work can be extended to show beyond birthday security of a 'CMCMC' or 'EMEME' like construction. Relatedly, how much can we relax the security notion of the underlying primitive and still retain good security (this question is relevant for practical key wrap schemes). Another question is whether changing the mixing layer will boost security when using three calls to an online cipher. We conjecture that among blockwise linear schemes, the scheme Π_3^{rev} is essentially optimal. The level of security achieved by a shift-based scheme with more layers than blocks remains a tantalizing open problem: conceivably they may achieve $\pm$PRP security.

7 Acknowledgments

The authors would like to thank Daniel Martin for thoughtful comments on early drafts of this document.

Elena Andreeva is supported by a Postdoctoral Fellowship of the Research Foundation Flanders (FWO).

This work was conducted whilst Guy Barwell was a PhD student at the University of Bristol, supported by an EPSRC grant.

References

[1] Abed, F., Fluhrer, S.R., Forler, C., List, E., Lucks, S., McGrew, D.A., Wenzel, J.: Pipelineable on-line encryption. In: FSE 2014. pp. 205–223. LNCS, Springer, Berlin, Germany (2015) Cited on page 5.

[2] Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Mouha, N., Yasuda, K.: How to securely release unverified plaintext in authenticated encryption. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 105–125. Springer, Berlin, Germany, Kaoshiung, Taiwan, R.O.C. (Dec 7–11, 2014) Cited on page 3.

[3] Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: Parallelizable and authenticated online ciphers. In: Sako and Sarkar [35], pp. 424–443 Cited on page 5.

[4] Bellare, M., Boldyreva, A., Knudsen, L.R., Namprempre, C.: Online ciphers and the hash-CBC construction. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 292–309. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 19–23, 2001) Cited on pages 3, 5, 7, and 24.

[5] Bellare, M., Rogaway, P.: On the construction of variable-input-length ciphers. In: Knudsen, L.R. (ed.) FSE'99. LNCS, vol. 1636, pp. 231–244. Springer, Berlin, Germany, Rome, Italy (Mar 24–26, 1999) Cited on pages 3, 4, and 7.

[6] Bellare, M., Rogaway, P.: Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 317–330. Springer, Berlin, Germany, Kyoto, Japan (Dec 3–7, 2000) Cited on pages 3, 4, and 24.

[7] Bernstein, D.J.: CAESAR competition call (2013), `http://competitions.cr.yp.to/caesar-call-3.html` Cited on page 3.

[8] Boldyreva, A., Degabriele, J.P., Paterson, K.G., Stam, M.: On symmetric encryption with distinguishable decryption failures. In: Moriai, S. (ed.) FSE 2013. LNCS, vol. 8424, pp. 367–390. Springer, Berlin, Germany, Singapore (Mar 11–13, 2014) Cited on page 3.

[9] Boldyreva, A., Taesombut, N.: Online encryption schemes: New security notions and constructions. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 1–14. Springer, Berlin, Germany, San Francisco, CA, USA (Feb 23–27, 2004) Cited on page 5.

[10] Bond, M., French, G., Smart, N.P., Watson, G.J.: The low-call diet: Authenticated encryption for call counting HSM users. In: Dawson, E. (ed.) CT-RSA 2013. LNCS, vol. 7779, pp. 359–374. Springer, Berlin, Germany, San Francisco, CA, USA (Feb 25 – Mar 1, 2013) Cited on page 3.

[11] Canteaut, A. (ed.): FSE 2012, LNCS, vol. 7549. Springer, Berlin, Germany, Washington, DC, USA (Mar 19–21, 2012) Cited on pages 25 and 26.

[12] Datta, N., Nandi, M.: ELmE: A misuse resistant parallel authenticated encryption. In: Susilo, W., Mu, Y. (eds.) ACISP 14. LNCS, vol. 8544, pp. 306–321. Springer, Berlin, Germany, Wollongong, NSW, Australia (Jul 7–9, 2014) Cited on page 5.

[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.) CRYPTO 2012. LNCS, vol. 7417, pp. 348–366. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 19–23, 2012) Cited on pages 24 and 27.

[14] Dworkin, M.: Request for review of key wrap algorithms. Cryptology ePrint Archive, Report 2004/340 (2004), `http://eprint.iacr.org/2004/340` Cited on pages 4, 5, and 8.

[15] Fleischmann, E., Forler, C., Lucks, S.: McOE: A family of almost foolproof on-line authenticated encryption schemes. In: Canteaut [11], pp. 196–215 Cited on pages 3 and 5.

[16] Fouque, P.A., Joux, A., Martinet, G., Valette, F.: Authenticated on-line encryption. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 145–159. Springer, Berlin, Germany, Ottawa, Ontario, Canada (Aug 14–15, 2004) Cited on page 5.

[17] Fouque, P.A., Martinet, G., Poupard, G.: Practical symmetric on-line encryption. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 362–375. Springer, Berlin, Germany, Lund, Sweden (Feb 24–26, 2003) Cited on page 5.

[18] Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984) Cited on page 3.

[19] Halevi, S.: Invertible universal hashing and the TET encryption mode. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 412–429. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 19–23, 2007) Cited on page 5.

[20] Halevi, S., Rogaway, P.: A tweakable enciphering mode. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 482–499. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 17–21, 2003) Cited on pages 5, 7, 8, and 11.

[21] Hoang, V.T., Krovetz, T., Rogaway, P.: Robust authenticated-encryption AEZ and the problem that it solves. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 15–44. Springer, Berlin, Germany, Sofia, Bulgaria (Apr 26–30, 2015) Cited on pages 3, 4, 5, and 24.

[22] Joux, A.: Authentication failures in nist version of gcm. NIST Comment (2006) Cited on page 3.

[23] Joux, A., Martinet, G., Valette, F.: Blockwise-adaptive attackers: Revisiting the (in)security of some provably secure encryption models: CBC, GEM, IACBC. In: Yung [37], pp. 17–30 Cited on page 5.

[24] Katz, J., Lindell, Y.: Introduction to Modern Cryptography. Chapman & Hall/CRC (2008) Cited on page 3.

[25] Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 306–327. Springer, Berlin, Germany, Lyngby, Denmark (Feb 13–16, 2011) Cited on page 3.

[26] Liskov, M., Rivest, R.L., Wagner, D.: Tweakable block ciphers. In: Yung [37], pp. 31–46 Cited on page 7.

[27] Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Berlin, Germany, Cambridge, MA, USA (Feb 19–21, 2004) Cited on pages 24 and 27.

[28] Minematsu, K.: Beyond-birthday-bound security based on tweakable block cipher. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 308–326. Springer, Berlin, Germany, Leuven, Belgium (Feb 22–25, 2009) Cited on page 5.

[29] Namprempre, C., Rogaway, P., Shrimpton, T.: Reconsidering generic composition. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 257–274. Springer, Berlin, Germany, Copenhagen, Denmark (May 11–15, 2014) Cited on page 3.

[30] Nandi, M.: Two new efficient CCA-secure online ciphers: MHCBC and MCBC. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 350–362. Springer, Berlin, Germany, Kharagpur, India (Dec 14–17, 2008) Cited on page 6.

[31] Rogaway, P.: Authenticated-encryption with associated-data. In: Atluri, V. (ed.) ACM CCS 02. pp. 98–107. ACM Press, Washington D.C., USA (Nov 18–22, 2002) Cited on page 3.

[32] Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Berlin, Germany, St. Petersburg, Russia (May 28 – Jun 1, 2006) Cited on pages 3 and 4.

[33] Rogaway, P., Wooding, M., Zhang, H.: The security of ciphertext stealing. In: Canteaut [11], pp. 180–195 Cited on pages 9 and 24.

[34] Rogaway, P., Zhang, H.: Online ciphers from tweakable blockciphers. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 237–249. Springer, Berlin, Germany, San Francisco, CA, USA (Feb 14–18, 2011) Cited on pages 4, 5, 6, and 9.

[35] Sako, K., Sarkar, P. (eds.): ASIACRYPT 2013, Part I, LNCS, vol. 8269. Springer, Berlin, Germany, Bengalore, India (Dec 1–5, 2013) Cited on pages 25 and 26.

[36] Shrimpton, T., Terashima, R.S.: A modular framework for building variable-input-length tweakable ciphers. In: Sako and Sarkar [35], pp. 405–423 Cited on pages 3 and 4.

[37] Yung, M. (ed.): CRYPTO 2002, LNCS, vol. 2442. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 18–22, 2002) Cited on page 26.

A Impracticality of Indifferentiability

The security definitions given in Section 2.1 are the standard indistinguishability notions for symmetric primitives, but are less strong than the *indifferentiability* notions of Maurer et al. [27]. In the indistinguishability game, the adversary is provided with oracle access to the overall construction (and possibly its inverse) or the ideal construction. In contrast, the indifferentiability game provides the adversary with access to the overall construction and also the internal primitive, or to the ideal construction and a simulator of the internal primitive. Thus the indifferentiability setting of the prp security game for the $\Pi = \Pi_i^L$ construction instantiated around the online cipher $\mathcal{E}$ is $\Delta_{E[\mathcal{E}],\mathcal{E}}^{\pi,S[\pi]}$, where $S[\cdot]$ is a simulator that provided with access to the permutation π simulates an online cipher $S[\pi]$.

Allowing leakage on the intermediate layers of the construction would allow the adversary to query the online cipher and overall construction in a somewhat independent manner, effectively allowing them to play the indifferentiability game.

Recent work by Dodis et al. [13] showed that the composition of two calls to a hash function is not indifferentiable from the original hash unless the simulator makes an unreasonably large number of queries. Broadly speaking, their attack depends on calling the random oracle to derive a chain of secret values. Then, with using two calls to the primitive, this is used to generate a second,non-overlapping chain. In the real world, to ensure this relationship holds, any simulator must make a large number of queries, effectively by calculating such chains themselves.

A similar result can be found when we consider whether the Π_i^L construction is indifferentiable from a in ideal cipher (with respect to the online cipher). We assume $L(M)[1]$ is linearly dependant on $M[m]$ for all $M \in \{0,1\}^{mn}$, since otherwise the scheme is trivially distinguishable. Then, the distinguisher can simply consider $H(M) := L \circ \mathcal{E}(M)$, from which (with high probability) the first output block is uniformly sampled. Using this, he can conduct an equivalent experiment, efficiently building two long chains and forcing the simulator to link them. Since the simulator is not provided with access to the inverse permutation, they are unable to invert the chains, leading to a similar analysis.

Let us denote the simulator of $\mathcal{E}$ by $S[\cdot]$, with inverse $T[\cdot]$, taking as parameters the oracles to which it is provided access, the Π_i^L scheme instantiated with online cipher $\mathcal{E}$ by $E[\mathcal{E}]$ with inverse $D[\mathcal{D}]$, and an ideal cipher by π with inverse π^{-1}. Then, by the above attack, $\Delta_{E[\mathcal{E}],\mathcal{E}}^{\pi,S[\pi]}$, which corresponds to indifferentiability from an ideal cipher, is large (in terms of simulator queries).

However, a simulator can defend against this attack with only a small number of queries if provided with the inverse of the permutation, since he may "unwind" any chains the adversary created. Thus, $\Delta_{E[\mathcal{E}],D[\mathcal{D}],\mathcal{E},\mathcal{D}}^{\pi,\pi^{-1},S[\pi,\pi^{-1}],T[\pi,\pi^{-1}]}$ (corresponding to indifferentiability from an ideal cipher under the $\pm$PRP game) cannot be bounded below by this attack. This leaves the rather counter-intuitive situation that a scheme might be indifferentiable from an ideal cipher with inverse, yet not from an ideal cipher when not provided with inverse acess. Other situations exist, such as bounding $\Delta_{E[\mathcal{E}],\mathcal{E},\mathcal{D}}^{\pi,S[\pi,\pi^{-1}],T[\pi,\pi^{-1}]}$, which reflects the scenario of a system providing interfaces for both directions of the online cipher, but only provides an interface for encryption queries of the true cipher. Whilst we suspect it unlikely, these constructions may yet be proven indifferentiable, but such results are beyond the scope of this paper.

Overall then, there are impossibility results limiting the scope for security under the indifferentiability game in this area. As such, there are clear limitations for when access can be provided to the online cipher under the same keying scheme as to the overall construction. Thus for viable security results, we are limited to the indistinguishability setting, meaning any instantiations of the Π_i^L construction should be keyed (or tweaked) independently from interfaces provided to the online cipher.

B Security Definitions

We provide here the formal security notions described in Section 2.1. Let E be a cipher on n bits, $\tilde{E}$ a tweakable block cipher with n-bit blocks and tweakspace $\mathcal{T}$ and $\mathcal{E}$ an online cipher, all with keyspace $\mathcal{K}$.

Then, advantage of some adversary $\mathcal{A}$ against the security goals of the various schemes are as follows:

$$\mathbf{Adv}_{E}^{\mathsf{prp}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow_{\$} \mathcal{K} \,:\, \mathcal{A}^{E_k} \to 1\right] - \mathbb{P}\left[\pi \leftarrow_{\$} \mathrm{Perm}(n) \,:\, \mathcal{A}^{\pi} \to 1\right]$$
$$\mathbf{Adv}_{E}^{\pm\mathsf{prp}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow_{\$} \mathcal{K} \,:\, \mathcal{A}^{E_k, E_k^{-1}} \to 1\right] - \mathbb{P}\left[\pi \leftarrow_{\$} \mathrm{Perm}(n) \,:\, \mathcal{A}^{\pi,\pi^{-1}} \to 1\right]$$
$$\mathbf{Adv}_{\tilde{E}}^{\mathsf{tprp}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow_{\$} \mathcal{K} \,:\, \mathcal{A}^{\tilde{E}_k} \to 1\right] - \mathbb{P}\left[\tilde{\pi} \leftarrow_{\$} \mathrm{Perm}(\mathcal{T}, n) \,:\, \mathcal{A}^{\tilde{\pi}} \to 1\right]$$
$$\mathbf{Adv}_{\tilde{E}}^{\pm\mathsf{tprp}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow_{\$} \mathcal{K} \,:\, \mathcal{A}^{\tilde{E}_k, \tilde{E}_k^{-1}} \to 1\right] - \mathbb{P}\left[\tilde{\pi} \leftarrow_{\$} \mathrm{Perm}(\mathcal{T}, n) \,:\, \mathcal{A}^{\tilde{\pi},\tilde{\pi}^{-1}} \to 1\right]$$
$$\mathbf{Adv}_{\mathcal{E}}^{\mathsf{oprp}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow_{\$} \mathcal{K} \,:\, \mathcal{A}^{\mathcal{E}_k} \to 1\right] - \mathbb{P}\left[\pi_* \leftarrow_{\$} \mathrm{OPerm}(n) \,:\, \mathcal{A}^{\pi_*} \to 1\right]$$
$$\mathbf{Adv}_{\mathcal{E}}^{\pm\mathsf{oprp}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow_{\$} \mathcal{K} \,:\, \mathcal{A}^{\mathcal{E}_k, \mathcal{E}_k^{-1}} \to 1\right] - \mathbb{P}\left[\pi_* \leftarrow_{\$} \mathrm{OPerm}(n) \,:\, \mathcal{A}^{\pi_*,\pi_*^{-1}} \to 1\right]$$
$$\mathbf{Adv}_{F}^{\mathsf{prf}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow_{\$} \mathcal{K} \,:\, \mathcal{A}^{F_k} \to 1\right] - \mathbb{P}\left[\phi \leftarrow_{\$} \mathrm{Func}(n) \,:\, \mathcal{A}^{\phi} \to 1\right]$$
$$\mathbf{Adv}_{F}^{\pm\mathsf{prf}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow_{\$} \mathcal{K} \,:\, \mathcal{A}^{F_k, F_k^{-1}} \to 1\right] - \mathbb{P}\left[\phi, \phi' \leftarrow_{\$} \mathrm{Func}(n) \,:\, \mathcal{A}^{\phi,\phi'} \to 1\right]$$

Note that since the adversary is prevented from making queries to which he already knows the answer, this definition of an $\pm$PRF is equivalent to that presented in the main body of the paper and much simpler to work with. These are generalised to functions of the number of queries by defining

$$\mathbf{Adv}_{\mathcal{W}_1}^{\mathrm{xxx}} := \max_{\substack{\text{Adversaries } \mathcal{A} \\ \mathcal{A} \text{ makes } q \text{ queries}}} \left|\mathbf{Adv}_{\mathcal{W}_1}^{\mathrm{xxx}}(\mathcal{A})\right|$$

A primitive P is a secure xxx if $\mathbf{Adv}_{P}^{\mathrm{xxx}}(q)$ is sufficiently small.

C Games & Oracles

In this appendix we provide the code for the oracles and games used in the proofs. Wherever possible, line numbers have been preserved between pairs of games in a sequence. In these cases, lines which change in the transition from one world to the next are indicated on the first of these by "▷ ▷ ▷" at the end of the line. If no lines are marked as such, the code has been rearranged or simplified, and making such a comparison is meaningless.

This paper makes use of the indistinguishability game, where an adversary is provided access to an unknown world and must distinguish which world he is communicating with. The other game used is the SetBad game, in which the adversary communicates with a single world and aims to set the flag bad.

Internal Oracles. At times, the oracles may themselves call out to internal oracles, modelling standard primitives, access to which is not given to the adversary. To minimise repetition, such oracles provided here, and are available to all oracles and games in the paper. Internal variables are shared between sets of private oracles given in the same algorithm box, but not between different instantiations or primitives. Following on from our decision to omit the key from our notation, we use the lower index to specify an independent instantiation of a primitive.

Alg. 0: (Tweakable) Random Function with inverse

for $T \in (\{0,1\}^n)^*$ **do**
  $\mathcal{F}_T \leftarrow \emptyset$
end for
function $F^T(M)$
  $C \leftarrow_{\$} \{0,1\}^n$
  if $\exists x$ s.t. $(M, x) \in \mathcal{F}_T$ **then**
    $C \leftarrow x$
  end if
  $\mathcal{F}_T \leftarrow_{\cup} (M, C)$
  return C
end function
function $F^{-1}(C)$
  $M \leftarrow_{\$} \{0,1\}^n$
  if $\exists x$ s.t. $(x, C) \in \mathcal{F}_T$ **then**
    $M \leftarrow x$
  end if
  $\mathcal{F}_T \leftarrow_{\cup} (M, C)$
  return M
end function

Alg. 1: Tweakable Block Cipher $(\tilde{\mathrm{E}}, \tilde{\mathrm{D}})$

for $T \in (\{0,1\}^n)^*$ **do**
  $\pi_T \leftarrow \emptyset \subset \{0,1\}^n \times \{0,1\}^n$
end for
function $\tilde{\mathrm{E}}^T(M)$
  if $M \notin \mathrm{Dom}(\pi_T)$ **then**
    $C \leftarrow_{\$} \{0,1\}^n \setminus \mathrm{Im}(\pi_T)$
    $\pi_T \leftarrow_{\cup} (M, C)$
  end if
  return $\pi_T(C)$
end function
function $\tilde{\mathrm{D}}^T(C)$
  if $C \notin \mathrm{Im}(\pi_T)$ **then**
    $M \leftarrow_{\$} \{0,1\}^n \setminus \mathrm{Dom}(\pi_T)$
    $\pi_T \leftarrow_{\cup} (M, C)$
  end if
  return $\pi_T^{-1}(C)$
end function

Alg. 2: Online en/de-cryption oracles $\mathcal{E}^T$,$\mathcal{D}^T$

function $\mathcal{E}^T(M)$
  $m \leftarrow |M|/n$
  for $i \leftarrow 1 \ldots m$ **do**
    $T_i \leftarrow T || M[1..(i-1)]$
    $C_i \leftarrow \tilde{\mathrm{E}}^{T_i}(M[i])$
  end for
  return $C_1 || \ldots || C_m$
end function
function $\mathcal{D}^T(C)$
  $m \leftarrow |C|/n$
  $M_0 \leftarrow \epsilon$
  for $i \leftarrow 1 \ldots m$ **do**
    $T_i \leftarrow T || M_{i-1}$
    $M_i \leftarrow M_{i-1} || \tilde{\mathrm{D}}^{T_i}(C[i])$
  end for
  return M_m
end function

C.1 $\Pi_2^{\mathbf{rev}}$ is a $\pm$PRP

This appendix provides a thorough list of worlds used in the proof of Theorem 11. Throughout, $|L_i| = |R_i| = 1$ for all i, with $|A_i| = (m-2)$, where m is the length of $|M|$ in blocks.

Alg. 3: $\mathcal{W}_0$ is the Π_2 construction

function $\text{Enc}(M)$
$L_1||A_1||R_1 \leftarrow M$
$L_2||A_2||R_2 \leftarrow \mathcal{E}^{\epsilon}(L_1||A_1||R_1)$
$L_3||A_3||R_3 \leftarrow R_2||\overline{A_2}||L_2$
$L_4||A_4||R_4 \leftarrow \mathcal{E}^{\epsilon}(L_3||A_3||R_3)$
return $L_4||A_4||R_4$
end function
function $\text{Dec}(M)$
$L_4||A_4||R_4 \leftarrow M$
$L_3||A_3||R_3 \leftarrow \mathcal{D}^{\epsilon}(L_4||A_4||R_4)$
$L_2||A_2||R_2 \leftarrow R_3||\overline{A_3}||L_3$
$L_1||A_1||R_1 \leftarrow \mathcal{D}^{\epsilon}(L_2||A_2||R_2)$
return $L_1||A_1||R_1$
end function

Alg. 4: $\mathcal{W}_1$ and $\boxed{\mathcal{W}_2}$

$\mathcal{P} \leftarrow \emptyset$
function $\text{Enc}(M)$
$L_1||A_1||R_1 \leftarrow M$
$L_2||A_2 \leftarrow \mathcal{E}^{\epsilon}(L_1||A_1)$ $\triangleright\triangleright\triangleright$
$R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
$\mathcal{P} \xleftarrow{\cup} L_1$
$L_3 \leftarrow R_2$
$A_3||R_3 \leftarrow \overline{A_2}||L_2$ $\triangleright\triangleright\triangleright$
$L_4 \leftarrow \mathcal{E}^{\epsilon}(L_3)$
if $L_3 \in \mathcal{P}$ **then**
$\mathtt{bad} \leftarrow \mathtt{true}$
$\boxed{L_3 \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{P}}$
end if
$A_4||R_4 \leftarrow \mathcal{E}^{L_3}(A_3||R_3)$ $\triangleright\triangleright\triangleright$
$\mathcal{P} \xleftarrow{\cup} L_3$
return $L_4||A_4||R_4$
end function
function $\text{Dec}(M)$
$L_4||A_4||R_4 \leftarrow M$
$L_3||A_3 \leftarrow \mathcal{D}^{\epsilon}(L_4||A_4)$
$\mathcal{P} \xleftarrow{\cup} L_3$
$R_3 \leftarrow \mathcal{D}^{L_3||A_3}(R_4)$
$L_2 \leftarrow R_3$
$A_2||R_2 \leftarrow \overline{A_3}||L_3$ $\triangleright\triangleright\triangleright$
$L_1 \leftarrow \mathcal{D}^{\epsilon}(L_2)$
if $L_1 \in \mathcal{P}$ **then**
$\mathtt{bad} \leftarrow \mathtt{true}$
$\boxed{L_1 \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{P}}$
end if
$A_1||R_1 \leftarrow \mathcal{D}^{L_1}(A_2||R_2)$ $\triangleright\triangleright\triangleright$
$\mathcal{P} \xleftarrow{\cup} L_1$
return $L_1||A_1||R_1$
end function

Alg. 5: $\boxed{\mathcal{W}_3}$ and $\mathcal{W}_4$

$\mathcal{P} \leftarrow \emptyset$
function $\text{Enc}(M)$
$L_1||A_1||R_1 \leftarrow M$

$R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
$\mathcal{P} \xleftarrow{\cup} L_1$
$L_3 \leftarrow R_2$

$L_4 \leftarrow \mathcal{E}^{\epsilon}(L_3)$
if $L_3 \in \mathcal{P}$ **then**
$\mathtt{bad} \leftarrow \mathtt{true}$
$\boxed{L_3 \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{P}}$
end if
$A_4||R_4 \xleftarrow{\$} \{0,1\}^{(m-1)n}$
$\mathcal{P} \xleftarrow{\cup} L_3$
return $L_4||A_4||R_4$
end function
function $\text{Dec}(M)$
$L_4||A_4||R_4 \leftarrow M$
$L_3||A_3 \leftarrow \mathcal{D}^{\epsilon}(L_4||A_4)$
$\mathcal{P} \xleftarrow{\cup} L_3$
$R_3 \leftarrow \mathcal{D}^{L_3||A_3}(R_4)$
$L_2 \leftarrow R_3$

$L_1 \leftarrow \mathcal{D}^{\epsilon}(L_2)$
if $L_1 \in \mathcal{P}$ **then**
$\mathtt{bad} \leftarrow \mathtt{true}$
$\boxed{L_1 \xleftarrow{\$} \{0,1\}^n \setminus \mathcal{P}}$
end if
$A_1||R_1 \xleftarrow{\$} \{0,1\}^{(m-1)n}$
$\mathcal{P} \xleftarrow{\cup} L_1$
return $L_1||A_1||R_1$
end function

Alg. 6: $\boxed{\mathcal{W}_5}$ and $\boxed{\mathcal{W}_6}$

$\mathcal{P} \leftarrow \emptyset$
function ENC(M)
 $L_1||A_1||R_1 \leftarrow M$
 $R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
 $\mathcal{P} \xleftarrow{\cup} L_1$
 $L_4 \leftarrow \boxed{\mathcal{E}^{\epsilon}(R_2)}\ \boxed{F(R_2)}$ ▷ ▷ ▷
 if $R_2 \in \mathcal{P}$ **then**
  bad $\leftarrow$ `true`
 end if
 $A_4||R_4 \xleftarrow{\$} \{0,1\}^{(m-1)n}$
 $\mathcal{P} \xleftarrow{\cup} R_2$
 return $L_4||A_4||R_4$
end function
function DEC(M)
 $L_4||A_4||R_4 \leftarrow M$
 $L_3||A_3 \leftarrow \mathcal{D}^{\epsilon}(L_4||A_4)$
 $\mathcal{P} \xleftarrow{\cup} L_3$
 $R_3 \leftarrow \mathcal{D}^{L_3||A_3}(R_4)$
 $L_1 \leftarrow \boxed{\mathcal{D}^{\epsilon}(R_3)}\ \boxed{F^{-1}(R_3)}$ ▷ ▷ ▷
 if $L_1 \in \mathcal{P}$ **then**
  bad $\leftarrow$ `true`
 end if
 $A_1||R_1 \xleftarrow{\$} \{0,1\}^{(m-1)n}$
 $\mathcal{P} \xleftarrow{\cup} L_1$
 return $L_1||A_1||R_1$
end function

Alg. 7: $\mathcal{W}_7$: A PRF with inverse

$\mathcal{P} \leftarrow \emptyset$
function ENC(M)
 $L_1||A_1||R_1 \leftarrow M$
 $R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
 $\mathcal{P} \xleftarrow{\cup} L_1$
 $L_4 \xleftarrow{\$} \{0,1\}^n$ ▷ ▷ ▷
 if $R_2 \in \mathcal{P}$ **then**
  bad $\leftarrow$ `true`
 end if
 $A_4||R_4 \xleftarrow{\$} \{0,1\}^{(m-1)n}$ ▷ ▷ ▷
 $\mathcal{P} \xleftarrow{\cup} R_2$
 return $L_4||A_4||R_4$ ▷ ▷ ▷
end function
function DEC(M)
 $L_4||A_4||R_4 \leftarrow M$
 $L_3||A_3 \leftarrow \mathcal{D}^{\epsilon}(L_4||A_4)$ ▷ ▷ ▷
 $\mathcal{P} \xleftarrow{\cup} L_3$
 $R_3 \leftarrow \mathcal{D}^{L_3||A_3}(R_4)$ ▷ ▷ ▷
 $L_1 \xleftarrow{\$} \{0,1\}^n$
 if $L_1 \in \mathcal{P}$ **then**
  bad $\leftarrow$ `true`
 end if
 $A_1||R_1 \xleftarrow{\$} \{0,1\}^{(m-1)n}$ ▷ ▷ ▷
 $\mathcal{P} \xleftarrow{\cup} L_1$
 return $L_1||A_1||R_1$ ▷ ▷ ▷
end function

Alg. 8: $\mathcal{W}_8$: $\mathbb{P}\left[\mathsf{bad}^7\right] = \mathbb{P}\left[\mathsf{bad}^8\right]$

$\mathcal{P} \leftarrow \emptyset$
function ENC(M)
 $L_1||A_1||R_1 \leftarrow M$
 $R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
 $\mathcal{P} \xleftarrow{\cup} L_1$
 if $R_2 \in \mathcal{P}$ **then**
  bad $\leftarrow$ `true`
 end if
 $\mathcal{P} \xleftarrow{\cup} R_2$
end function
function DEC(M)
 $L_4||A_4||R_4 \leftarrow M$
 $L_3 \leftarrow \mathcal{D}^{\epsilon}(L_4)$
 $\mathcal{P} \xleftarrow{\cup} L_3$
 $L_1 \xleftarrow{\$} \{0,1\}^n$
 if $L_1 \in \mathcal{P}$ **then**
  bad $\leftarrow$ `true`
 end if
 $\mathcal{P} \xleftarrow{\cup} L_1$
end function

Alg. 9: $\mathcal{W}_9$: $\mathbb{P}\left[\mathsf{bad}^8\right] \leq \mathbb{P}\left[\mathsf{bad}^9\right]$

$\mathcal{P} \leftarrow \emptyset, \mathcal{L} \leftarrow \emptyset, \mathcal{R} \leftarrow \emptyset$
function ENC(M)
  $L_1 || A_1 || R_1 \leftarrow M$
  $R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
  $\mathcal{P} \leftarrow_\cup L_1$
  if $R_2 \in \mathcal{P} \cup \mathcal{L} \cup \mathcal{R}$ **then**
    $\mathsf{bad} \leftarrow \mathtt{true}$
  end if
  $\mathcal{R} \leftarrow_\cup R_2$
end function
function DEC(L_3)
  $\mathcal{P} \leftarrow_\cup L_3$
  $L_1 \leftarrow_\$ \{0,1\}^n$
  if $L_1 \in \mathcal{P} \cup \mathcal{L} \cup \mathcal{R}$ **then**
    $\mathsf{bad} \leftarrow \mathtt{true}$
  end if
  $\mathcal{L} \leftarrow_\cup L_1$
end function

Alg. 10: $\mathcal{W}_{10}$: Adversary non-adaptive

Require: $|\mathcal{P}| = q, q_E + q_D = q, |\mathcal{M}| = q_E$.
Require: order $\in \{E, D\}^q$
$\mathcal{L}, \mathcal{R}$ empty lists
function CHALLENGE($\mathcal{P}, \mathcal{M}$, order)
  for $i \in [1, \ldots, q_E]$ **do**
    $L_1 || A_1 || R_1 \leftarrow \mathcal{M}[i]$
    $\mathcal{R}_i \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
  end for
  for $i \in [1, \ldots, q_D]$ **do**
    $\mathcal{L}_i \leftarrow_\$ \{0,1\}^n$
  end for
  if List $\mathcal{L}||\mathcal{R}$ contains repeats **then**
    $\mathsf{bad} \leftarrow \mathtt{true}$
  end if
  for $i \in [1, \ldots, q]$ **do**
    if order$[i] = E$ **then**
      if $\exists j \leq i$ s.t. $\mathcal{P}_j = \mathcal{R}_e$ **then**
        $\mathsf{bad} \leftarrow \mathtt{true}$
      end if
      $e \leftarrow e + 1$
    else
      if $\exists j \leq i$ s.t. $\mathcal{P}_j = \mathcal{L}_d$ **then**
        $\mathsf{bad} \leftarrow \mathtt{true}$
      end if
      $d \leftarrow d + 1$
    end if
  end for
end function

C.2 Π_3^{right} is a PRP

This appendix provides the oracles used during the proof of Theorem 19. A_i, B_i are blocks, with X_i the appropriate length to ensure $|M| = |(X_i||A_i||B_i)|$. We evaluate the shift implicitly, by reordering blocks in the following query.

Alg. 11: $\mathcal{O}_0$ is the $\Pi = \Pi_3^{\mathsf{right}}$ construction

function ENC(M)
  $X_1 || A_1 || B_1 \leftarrow M$
  $X_2 || A_2 || B_2 \leftarrow \mathcal{E}_1^{\epsilon}(X_1 || A_1 || B_1)$
  $B_3 || X_3 || A_3 \leftarrow \mathcal{E}_2^{\epsilon}(B_2 || X_2 || A_2)$
  $A_4 || B_4 || X_4 \leftarrow \mathcal{E}_3^{\epsilon}(A_3 || B_3 || X_3)$
  return $A_4 || B_4 || X_4$
end function

Alg. 12: $\mathcal{O}_1$ expands out $\mathcal{O}_0$

function $\textsc{Enc}(M)$
    $X_1||A_1||B_1 \leftarrow M$
    $X_2||A_2 \leftarrow \mathcal{E}_1^{\epsilon}(X_1||A_1)$
    $B_2 \leftarrow \mathcal{E}_1^{X_1||A_1}(B_1)$
    $B_3||X_3 \leftarrow \mathcal{E}_2^{\epsilon}(B_2||X_2)$
    $A_3 \leftarrow \mathcal{E}_2^{B_2||X_2}(A_2)$ ▷ ▷ ▷
▷ ▷ ▷
▷ ▷ ▷
▷ ▷ ▷
▷ ▷ ▷
    $A_4||B_4 \leftarrow \mathcal{E}_3^{\epsilon}(A_3||B_3)$
    $X_4 \leftarrow \mathcal{E}_3^{A_3||B_3}(X_3)$
    return $A_4||B_4||X_4$
end function

Alg. 13: $\mathcal{O}_2$: TPRF switch on $\mathcal{E}_2$ final block

$\mathcal{C} \leftarrow \emptyset$
function $\textsc{Enc}(M)$
    $X_1||A_1||B_1 \leftarrow M$
    $X_2||A_2 \leftarrow \mathcal{E}_1^{\epsilon}(X_1||A_1)$
    $B_2 \leftarrow \mathcal{E}_1^{X_1||A_1}(B_1)$
    $B_3||X_3 \leftarrow \mathcal{E}_2^{\epsilon}(B_2||X_2)$
    $A_3 \leftarrow F_2^{B_2||X_2}(A_2)$ ▷ ▷ ▷
    if $(B_2||X_2, A_3) \in \mathcal{C}$ **then** ▷ ▷ ▷
        $\mathtt{bad}_1 \leftarrow \mathtt{true}$
    end if
    $\mathcal{C} \xleftarrow{\cup} (B_2||X_2, A_3)$ ▷ ▷ ▷
    $A_4||B_4 \leftarrow \mathcal{E}_3^{\epsilon}(A_3||B_3)$
    $X_4 \leftarrow \mathcal{E}_3^{A_3||B_3}(X_3)$ ▷ ▷ ▷
    return $A_4||B_4||X_4$
end function

Alg. 14: $\mathcal{O}_3$: Identical to $\mathcal{O}_2$ until $\mathtt{bad}_1$

$\mathcal{C} \leftarrow \emptyset$
function $\textsc{Enc}(M)$
    $X_1||A_1||B_1 \leftarrow M$
    $X_2||A_2 \leftarrow \mathcal{E}_1^{\epsilon}(X_1||A_1)$
    $B_2 \leftarrow \mathcal{E}_1^{X_1||A_1}(B_1)$
    $B_3||X_3 \leftarrow \mathcal{E}_2^{\epsilon}(B_2||X_2)$
    $A_3 \xleftarrow{\$} \{0,1\}^n$
    if $(B_2, A_3) \in \mathcal{C}$ **then**
        $\mathtt{bad}_1 \leftarrow \mathtt{true}$
    end if
    $\mathcal{C} \xleftarrow{\cup} (B_2, A_3)$
    $A_4||B_4 \leftarrow \mathcal{E}_3^{\epsilon}(A_3||B_3)$
    $X_4 \xleftarrow{\$} \{0,1\}^{(m-2)n}$
    return $A_4||B_4||X_4$
end function

Alg. 15: $\mathcal{O}_4$: Simplify & reorder

$\mathcal{C} \leftarrow \emptyset$
function $\textsc{Enc}(M)$
  $X_1||A_1||B_1 \leftarrow M$
  $A_3 \leftarrow\!\!\$\ \{0,1\}^n$
  $X_4 \leftarrow\!\!\$\ \{0,1\}^{(m-2)n}$
  $B_2 \leftarrow \mathcal{E}_1^{X_1||A_1}(B_1)$
  $B_3 \leftarrow \mathcal{E}_2^{\epsilon}(B_2)$
  if $(B_2, A_3) \in \mathcal{C}$ **then**
    $\mathtt{bad}_1 \leftarrow \mathtt{true}$
  end if
  $\mathcal{C} \leftarrow\!\!\cup\ (B_2, A_3)$
  $A_4 \leftarrow \mathcal{E}_3^{\epsilon}(A_3)$
  $B_4 \leftarrow \mathcal{E}_3^{A_3}(B_3)$ ▷ ▷ ▷
  ▷ ▷ ▷
  ▷ ▷ ▷
  ▷ ▷ ▷
  ▷ ▷ ▷
  return $A_4||B_4||X_4$
end function

Alg. 16: $\mathcal{O}_5$: TPRP–TPRF switch on $\mathcal{E}_3$

$\mathcal{C}, \mathcal{C}' \leftarrow \emptyset$
function $\textsc{Enc}(M)$
  $X_1||A_1||B_1 \leftarrow M$
  $A_3 \leftarrow\!\!\$\ \{0,1\}^n$
  $X_4 \leftarrow\!\!\$\ \{0,1\}^{(m-2)n}$
  $B_2 \leftarrow \mathcal{E}_1^{X_1||A_1}(B_1)$
  $B_3 \leftarrow \mathcal{E}_2^{\epsilon}(B_2)$
  if $(B_2, A_3) \in \mathcal{C}$ **then**
    $\mathtt{bad}_1 \leftarrow \mathtt{true}$
  end if
  $\mathcal{C} \leftarrow\!\!\cup\ (B_2, A_3)$
  $A_4 \leftarrow \mathcal{E}_3^{\epsilon}(A_3)$
  $B_4 \leftarrow F_3^{A_3}(B_3)$ ▷ ▷ ▷
  if $(A_3, B_4) \in \mathcal{C}'$ **then**
    $\mathtt{bad}_2 \leftarrow \mathtt{true}$
  end if
  $\mathcal{C}' \leftarrow\!\!\cup\ (A_3, B_4)$
  return $A_4||B_4||X_4$
end function

Alg. 17: $\mathcal{O}_6$: Sample B_4 uniformly

$\mathcal{C}, \mathcal{C}' \leftarrow \emptyset$
function $\textsc{Enc}(M)$
  $X_1||A_1||B_1 \leftarrow M$
  $A_3 \leftarrow\!\!\$\ \{0,1\}^n$ ▷ ▷ ▷
  $X_4 \leftarrow\!\!\$\ \{0,1\}^{(m-2)n}$
  $B_2 \leftarrow \mathcal{E}_1^{X_1||A_1}(B_1)$
  $B_3 \leftarrow \mathcal{E}_2^{\epsilon}(B_2)$
  if $(B_2, A_3) \in \mathcal{C}$ **then** ▷ ▷ ▷
    $\mathtt{bad}_1 \leftarrow \mathtt{true}$
  end if
  $\mathcal{C} \leftarrow\!\!\cup\ (B_2, A_3)$ ▷ ▷ ▷
  $A_4 \leftarrow \mathcal{E}_3^{\epsilon}(A_3)$ ▷ ▷ ▷
  $B_4 \leftarrow\!\!\$\ \{0,1\}^n$
  if $(A_3, B_4) \in \mathcal{C}'$ **then** ▷ ▷ ▷
    $\mathtt{bad}_2 \leftarrow \mathtt{true}$
  end if
  $\mathcal{C}' \leftarrow\!\!\cup\ (A_3, B_4)$ ▷ ▷ ▷
  return $A_4||B_4||X_4$
end function

Alg. 18: $\mathcal{O}_7$: Sample A_4 not A_3

$\mathcal{C}, \mathcal{C}' \leftarrow \emptyset$
function $\textsc{Enc}(M)$
  $X_1||A_1||B_1 \leftarrow M$
  $A_4 \leftarrow_{\$} \{0,1\}^n$
  $X_4 \leftarrow_{\$} \{0,1\}^{(m-2)n}$
  $B_2 \leftarrow \mathcal{E}_1^{X_1||A_1}(B_1)$
  $B_3 \leftarrow \mathcal{E}_2^{\epsilon}(B_2)$
  if $(B_2, A_4) \in \mathcal{C}$ **then**
    $\mathsf{bad}_1 \leftarrow \mathtt{true}$
  end if
  $\mathcal{C} \leftarrow_{\cup} (B_2, A_4)$
  $A_3 \leftarrow \mathcal{D}_3^{\epsilon}(A_4)$
  $B_4 \leftarrow_{\$} \{0,1\}^n$
  if $(A_4, B_4) \in \mathcal{C}'$ **then**
    $\mathsf{bad}_2 \leftarrow \mathtt{true}$
  end if
  $\mathcal{C}' \leftarrow_{\cup} (A_4, B_4)$
  return $A_4||B_4||X_4$
end function

Alg. 19: $\mathcal{O}_8$: Simplify & reorder

$\mathcal{C}, \mathcal{C}' \leftarrow \emptyset$
function $\textsc{Enc}(M)$
  $X_1||A_1||B_1 \leftarrow M$
  $A_4 \leftarrow_{\$} \{0,1\}^n$
  $B_4 \leftarrow_{\$} \{0,1\}^n$
  $X_4 \leftarrow_{\$} \{0,1\}^{(m-2)n}$
  $B_2 \leftarrow \mathcal{E}_1^{X_1||A_1}(B_1)$ $\triangleright\triangleright\triangleright$
  if $(B_2, A_4) \in \mathcal{C}$ **then** $\triangleright\triangleright\triangleright$
    $\mathsf{bad}_1 \leftarrow \mathtt{true}$ $\triangleright\triangleright\triangleright$
  end if $\triangleright\triangleright\triangleright$
  $\mathcal{C} \leftarrow_{\cup} (B_2, A_4)$ $\triangleright\triangleright\triangleright$
  if $(A_4, B_4) \in \mathcal{C}'$ **then** $\triangleright\triangleright\triangleright$
    $\mathsf{bad}_2 \leftarrow \mathtt{true}$
  end if
  $\mathcal{C}' \leftarrow_{\cup} (A_4, B_4)$ $\triangleright\triangleright\triangleright$
  return $A_4||B_4||X_4$
end function

Alg. 20: $\mathcal{O}_9$: A PRP until bad_2

$\mathcal{C}' \leftarrow \emptyset$
function $\textsc{Enc}(M)$
  $X_1||A_1||B_1 \leftarrow M$
  $A_4 \leftarrow_{\$} \{0,1\}^n$
  $B_4 \leftarrow_{\$} \{0,1\}^n$
  $X_4 \leftarrow_{\$} \{0,1\}^{(m-2)n}$
  if $(A_4||B_4||X_4) \in \mathcal{C}'$ **then**
    $\mathsf{bad}_2 \leftarrow \mathtt{true}$
  end if
  $\mathcal{C}' \leftarrow_{\cup} (A_4||B_4||X_4)$
  return $A_4||B_4||X_4$
end function

C.3 Π_3^{rev} is a $\pm$PRP

These are the worlds used in the proof of Theorem 13. Throughout this section, A_i, B_i are blocks and $|X_i| = m - 2$ for all $i \in 1, \dots, 6$, where m is the length of M in blocks. For any string of blocks $X \in (\{0,1\}^n)^*$, $\overline{X}$ is the blockwise reversal of X.

Alg. 21: $\mathcal{W}_0$ is the Π_3^{rev} construction

function $\textsc{Enc}(M)$
 $X_1||A_1||B_1 \leftarrow M$
 $X_2||A_2||B_2 \leftarrow \mathcal{E}^\epsilon(X_1||A_1||B_1)$
 $B_3||A_3||X_2 \leftarrow \mathcal{E}^\epsilon(B_2||A_2||\overline{X}_2)$
 $X_4||A_4||B_4 \leftarrow \mathcal{E}^\epsilon(\overline{X}_3||A_3||B_3)$
 return $X_4||A_4||B_4$
end function
function $\textsc{Dec}(M)$
 $X_4||A_4||B_4 \leftarrow M$
 $X_3||A_3||B_3 \leftarrow \mathcal{D}^\epsilon(X_4||A_4||B_4)$
 $B_2||A_2||X_2 \leftarrow \mathcal{D}^\epsilon(B_3||A_3||\overline{X}_3)$
 $X_1||A_1||B_1 \leftarrow \mathcal{D}^\epsilon(\overline{X}_1||A_2||B_2)$
 return $X_1||A_1||B_1$
end function

Alg. 22: $\mathcal{W}_1$ expands out $\mathcal{W}_0$

$\mathcal{L}_A, \mathcal{L}_B, \mathcal{L}_C, \mathcal{L}_D \leftarrow \emptyset$
function $\textsc{Enc}(M)$
 $X_1||A_1||B_1 \leftarrow M$
 $X_2||A_2||B_2 \leftarrow \mathcal{E}_1^\epsilon(X_1||A_1||B_1)$
 $B_3 \leftarrow \mathcal{E}_2^\epsilon(B_2)$
 $A_3 \leftarrow \mathcal{E}_2^{B_2}(A_2)$ ▷▷▷
 $X_3 \leftarrow \mathcal{E}_2^{B_2||A_2}(\overline{X}_2)$ ▷▷▷
 $X_4||A_4 \leftarrow \mathcal{E}_3^\epsilon(\overline{X}_3||A_3)$
 $B_4 \leftarrow \mathcal{E}_3^{\overline{X}_3||A_3}(B_3)$ ▷▷▷
 if $(B_2, A_3) \in \mathcal{L}_B$ **then**
  $\mathsf{bad}_1 \leftarrow \mathtt{true}$
 end if
 if $(B_2, A_2) \in \mathcal{L}_C$ **then**
  $\mathsf{bad}_3 \leftarrow \mathtt{true}$
 end if
 if $(X_3||A_3, B_4) \in \mathcal{L}_D$ **then**
  $\mathsf{bad}_5 \leftarrow \mathtt{true}$
 end if
 $\mathcal{L}_A \xleftarrow{\cup} (X_1||A_1, B_1)$
 $\mathcal{L}_B \xleftarrow{\cup} (B_2, A_3)$
 $\mathcal{L}_C \xleftarrow{\cup} (B_2, A_2)$
 $\mathcal{L}_D \xleftarrow{\cup} (X_3||A_3, B_4)$
 return $X_4||A_4||B_4$
end function
function $\textsc{Dec}(M)$
 $X_4||A_4||B_4 \leftarrow M$
 $X_3||A_3||B_3 \leftarrow \mathcal{D}_3^\epsilon(X_4||A_4||B_4)$
 $B_2 \leftarrow \mathcal{D}_2^\epsilon(B_3)$
 $A_2 \leftarrow \mathcal{D}_2^{B_2}(A_3)$ ▷▷▷
 $X_2 \leftarrow \mathcal{D}_2^{B_2||A_2}(\overline{X}_3)$ ▷▷▷
 $X_1||A_1 \leftarrow \mathcal{D}_1^\epsilon(\overline{X}_2||A_2)$
 $B_1 \leftarrow \mathcal{D}_1^{X_1||A_1}(B_2)$ ▷▷▷
 if $(B_2, A_3) \in \mathcal{L}_B$ **then**
  $\mathsf{bad}_2 \leftarrow \mathtt{true}$
 end if
 if $(B_2, A_2) \in \mathcal{L}_C$ **then**
  $\mathsf{bad}_4 \leftarrow \mathtt{true}$
 end if
 if $(X_1||A_1, B_1) \in \mathcal{L}_A$ **then**
  $\mathsf{bad}_6 \leftarrow \mathtt{true}$
 end if
 $\mathcal{L}_A \xleftarrow{\cup} (X_1||A_1, B_1)$
 $\mathcal{L}_B \xleftarrow{\cup} (B_2, A_3)$
 $\mathcal{L}_C \xleftarrow{\cup} (B_2, A_2)$
 $\mathcal{L}_D \xleftarrow{\cup} (X_3||A_3, B_4)$
 return $X_1||A_1||B_1$
end function

Alg. 23: $\mathcal{W}_2$: ±PRP–±PRF switch on $\mathcal{E}_2$

$\mathcal{L}_A, \mathcal{L}_B, \mathcal{L}_C, \mathcal{L}_D \leftarrow \emptyset$
function $\textsc{Enc}(M)$
  $X_1||A_1||B_1 \leftarrow M$
  $X_2||A_2||B_2 \leftarrow \mathcal{E}_1^{\epsilon}(X_1||A_1||B_1)$
  $B_3 \leftarrow \mathcal{E}_2^{\epsilon}(B_2)$
  $A_3 \leftarrow \mathsf{F}_2^{B_2}(A_2)$ $\triangleright\triangleright\triangleright$
  $X_3 \leftarrow \mathsf{F}_2^{B_2||A_2}(\overline{X}_2)$ $\triangleright\triangleright\triangleright$
  $X_4||A_4 \leftarrow \mathcal{E}_3^{\epsilon}(\overline{X}_3||A_3)$
  $B_4 \leftarrow \mathsf{F}_3^{\overline{X}_3||A_3}(B_3)$ $\triangleright\triangleright\triangleright$
  if $(B_2, A_3) \in \mathcal{L}_B$ **then**
    $\mathsf{bad}_1 \leftarrow \mathtt{true}$
  end if
  if $(B_2, A_2) \in \mathcal{L}_C$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
  end if
  if $(X_3||A_3, B_4) \in \mathcal{L}_D$ **then**
    $\mathsf{bad}_5 \leftarrow \mathtt{true}$
  end if
  $\mathcal{L}_A \xleftarrow{\cup} (X_1||A_1, B_1)$
  $\mathcal{L}_B \xleftarrow{\cup} (B_2, A_3)$
  $\mathcal{L}_C \xleftarrow{\cup} (B_2, A_2)$
  $\mathcal{L}_D \xleftarrow{\cup} (X_3||A_3, B_4)$
  return $X_4||A_4||B_4$
end function
function $\textsc{Dec}(M)$
  $X_4||A_4||B_4 \leftarrow M$
  $X_3||A_3||B_3 \leftarrow \mathcal{D}_3^{\epsilon}(X_4||A_4||B_4)$
  $B_2 \leftarrow \mathcal{D}_2^{\epsilon}(B_3)$
  $A_2 \leftarrow \mathsf{G}_2^{B_2}(A_3)$ $\triangleright\triangleright\triangleright$
  $X_2 \leftarrow \mathsf{G}_2^{B_2||A_2}(X_2)$ $\triangleright\triangleright\triangleright$
  $X_1||A_1 \leftarrow \mathcal{D}_1^{\epsilon}(\overline{X}_2||A_2)$
  $B_1 \leftarrow \mathsf{G}_1^{X_1||A_1}(B_2)$ $\triangleright\triangleright\triangleright$
  if $(B_2, A_3) \in \mathcal{L}_B$ **then**
    $\mathsf{bad}_2 \leftarrow \mathtt{true}$
  end if
  if $(B_2, A_2) \in \mathcal{L}_C$ **then**
    $\mathsf{bad}_4 \leftarrow \mathtt{true}$
  end if
  if $(X_1||A_1, B_1) \in \mathcal{L}_A$ **then**
    $\mathsf{bad}_6 \leftarrow \mathtt{true}$
  end if
  $\mathcal{L}_A \xleftarrow{\cup} (X_1||A_1, B_1)$
  $\mathcal{L}_B \xleftarrow{\cup} (B_2, A_3)$
  $\mathcal{L}_C \xleftarrow{\cup} (B_2, A_2)$
  $\mathcal{L}_D \xleftarrow{\cup} (X_3||A_3, B_4)$
  return $X_1||A_1||B_1$
end function

$\mathsf{F}_1, \mathsf{F}_2, \mathsf{G}_2, \mathsf{G}_3$ are tweakable random functions.

Alg. 24: $\mathcal{W}_3$: They're actually samplings

$\mathcal{L}_A, \mathcal{L}_B, \mathcal{L}_C, \mathcal{L}_D \leftarrow \emptyset$ $\triangleright\triangleright\triangleright$
function $\textsc{Enc}(M)$
  $X_1||A_1||B_1 \leftarrow M$
  $X_2||A_2||B_2 \leftarrow \mathcal{E}_1^{\epsilon}(X_1||A_1||B_1)$ $\triangleright\triangleright\triangleright$
  $B_3 \leftarrow \mathcal{E}_2^{\epsilon}(B_2)$ $\triangleright\triangleright\triangleright$
  $A_3 \leftarrow_{\$} \{0,1\}^n$
  $X_3 \leftarrow_{\$} \{0,1\}^{(m-2)n}$
  $X_4||A_4 \leftarrow \mathcal{E}_3^{\epsilon}(\overline{X}_3||A_3)$
  $B_4 \leftarrow_{\$} \{0,1\}^n$ $\triangleright\triangleright\triangleright$
  if $(B_2, A_3) \in \mathcal{L}_B$ **then**
    $\mathsf{bad}_1 \leftarrow \mathtt{true}$
  end if
  if $(B_2, A_2) \in \mathcal{L}_C$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
  end if
  if $(X_3||A_3, B_4) \in \mathcal{L}_D$ **then** $\triangleright\triangleright\triangleright$
    $\mathsf{bad}_5 \leftarrow \mathtt{true}$ $\triangleright\triangleright\triangleright$
  end if $\triangleright\triangleright\triangleright$
  $\mathcal{L}_A \xleftarrow{\cup} (X_1||A_1, B_1)$ $\triangleright\triangleright\triangleright$
  $\mathcal{L}_B \xleftarrow{\cup} (B_2, A_3)$
  $\mathcal{L}_C \xleftarrow{\cup} (B_2, A_2)$
  $\mathcal{L}_D \xleftarrow{\cup} (X_3||A_3, B_4)$ $\triangleright\triangleright\triangleright$
  return $X_4||A_4||B_4$ $\triangleright\triangleright\triangleright$
end function
function $\textsc{Dec}(M)$
  $X_4||A_4||B_4 \leftarrow M$
  $X_3||A_3||B_3 \leftarrow \mathcal{D}_3^{\epsilon}(X_4||A_4||B_4)$ $\triangleright\triangleright\triangleright$
  $B_2 \leftarrow \mathcal{D}_2^{\epsilon}(B_3)$
  $A_2 \leftarrow_{\$} \{0,1\}^n$
  $X_2 \leftarrow_{\$} \{0,1\}^{(m-2)n}$
  $X_1||A_1 \leftarrow \mathcal{D}_1^{\epsilon}(\overline{X}_2||A_2)$
  $B_1 \leftarrow_{\$} \{0,1\}^n$ $\triangleright\triangleright\triangleright$
  if $(B_2, A_3) \in \mathcal{L}_B$ **then**
    $\mathsf{bad}_2 \leftarrow \mathtt{true}$
  end if
  if $(B_2, A_2) \in \mathcal{L}_C$ **then**
    $\mathsf{bad}_4 \leftarrow \mathtt{true}$
  end if
  if $(X_1||A_1, B_1) \in \mathcal{L}_A$ **then** $\triangleright\triangleright\triangleright$
    $\mathsf{bad}_6 \leftarrow \mathtt{true}$ $\triangleright\triangleright\triangleright$
  end if $\triangleright\triangleright\triangleright$
  $\mathcal{L}_A \xleftarrow{\cup} (X_1||A_1, B_1)$ $\triangleright\triangleright\triangleright$
  $\mathcal{L}_B \xleftarrow{\cup} (B_2, A_3)$
  $\mathcal{L}_C \xleftarrow{\cup} (B_2, A_2)$
  $\mathcal{L}_D \xleftarrow{\cup} (X_3||A_3, B_4)$ $\triangleright\triangleright\triangleright$
  return $X_1||A_1||B_1$ $\triangleright\triangleright\triangleright$
end function

Alg. 25: $\mathcal{W}_4$: Remove Superfluous Code

$\mathcal{L}_B, \mathcal{L}_C \leftarrow \emptyset$
function $\text{Enc}(M)$
  $X_1||A_1||B_1 \leftarrow M$
  $A_2||B_2 \leftarrow \mathcal{E}_1^{X_1}(A_1||B_1)$
  $A_3 \leftarrow_{\$} \{0,1\}^n$ ▷ ▷ ▷
  $X_3 \leftarrow_{\$} \{0,1\}^{(m-2)n}$ ▷ ▷ ▷
  $X_4||A_4 \leftarrow \mathcal{E}_3^{\epsilon}(\overline{X}_3||A_3)$ ▷ ▷ ▷
  if $(B_2, A_3) \in \mathcal{L}_B$ **then** ▷ ▷ ▷
    $\mathsf{bad}_1 \leftarrow \mathtt{true}$ ▷ ▷ ▷
  end if ▷ ▷ ▷
  if $(B_2, A_2) \in \mathcal{L}_C$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
  end if
  $\mathcal{L}_B \leftarrow_{\cup} (B_2, A_3)$ ▷ ▷ ▷
  $\mathcal{L}_C \leftarrow_{\cup} (B_2, A_2)$
  return $X_4||A_4$
end function
function $\text{Dec}(M)$
  $X_4||A_4||B_4 \leftarrow M$
  $A_3||B_3 \leftarrow \mathcal{D}_3^{\mathcal{D}_3(X_4)}(A_4||B_4)$ ▷ ▷ ▷
  $B_2 \leftarrow \mathcal{D}_2^{\epsilon}(B_3)$
  $A_2 \leftarrow_{\$} \{0,1\}^n$
  $X_2 \leftarrow_{\$} \{0,1\}^{(m-2)n}$
  $X_1||A_1 \leftarrow \mathcal{D}_1^{\epsilon}(\overline{X}_2||A_2)$
  if $(B_2, A_3) \in \mathcal{L}_B$ **then** ▷ ▷ ▷
    $\mathsf{bad}_2 \leftarrow \mathtt{true}$ ▷ ▷ ▷
  end if ▷ ▷ ▷
  if $(B_2, A_2) \in \mathcal{L}_C$ **then**
    $\mathsf{bad}_4 \leftarrow \mathtt{true}$
  end if
  $\mathcal{L}_B \leftarrow_{\cup} (B_2, A_3)$ ▷ ▷ ▷
  $\mathcal{L}_C \leftarrow_{\cup} (B_2, A_2)$
  return $X_1||A_1$
end function

Alg. 26: $\mathcal{W}_5$: Focus on $\mathsf{bad}_3, \mathsf{bad}_4$; sample X_1

$\mathcal{L}_C \leftarrow \emptyset$
function $\text{Enc}(M)$
  $X_1||A_1||B_1 \leftarrow M$
  $A_2||B_2 \leftarrow \mathcal{E}_1^{X_1}(A_1||B_1)$
  if $(A_2, B_2) \in \mathcal{L}_C$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
  end if
  $\mathcal{L}_C \leftarrow_{\cup} (A_2, B_2)$
end function
function $\text{Dec}(M)$
  $X_4||A_4||B_4 \leftarrow M$
  $B_3 \leftarrow \mathcal{D}_3^{\mathcal{D}_3(X_4||A_4)}(B_4)$
  $B_2 \leftarrow \mathcal{D}_2^{\epsilon}(B_3)$
  $A_2 \leftarrow_{\$} \{0,1\}^n$
  $X_1 \leftarrow_{\$} \{0,1\}^{(m-2)n}$
  $A_1 \leftarrow \mathcal{D}_1^{X_1}(A_2)$
  if $(A_2, B_2) \in \mathcal{L}_C$ **then**
    $\mathsf{bad}_4 \leftarrow \mathtt{true}$
  end if
  $\mathcal{L}_C \leftarrow_{\cup} (A_2, B_2)$
  return $X_1||A_1$
end function

In the final game, the adversary makes q_d Dec queries, followed by q_e Enc queries.

Alg. 27: $\mathcal{W}_8$: Final game $E \hookrightarrow D$

$\forall a \in \{0,1\}^n : \mathcal{L}_C[a] \leftarrow \emptyset$
function $\text{Dec}(M)$
  $X_4||A_4||B_4 \leftarrow M$
  $B_3 \leftarrow \mathcal{D}_3^{\mathcal{D}_3(X_4||A_4)}(B_4)$
  $B_2 \leftarrow \mathcal{D}_2^{\epsilon}(B_3)$
  $A_2 \leftarrow_{\$} \{0,1\}^n$
  $X_1 \leftarrow_{\$} \{0,1\}^{(m-2)n}$
  $A_1 \leftarrow \mathcal{D}_1^{X_1}(A_2)$
  $\mathcal{L}_C[A_2] \leftarrow_{\cup} B_2$
  return $X_1||A_1$
end function
Then...
function $\text{Enc}(M)$
  $X_1||A_1||B_1 \leftarrow M$
  $A_2 \leftarrow \mathcal{E}_1^{X_1}(A_1)$
  $B_2 \leftarrow \mathcal{E}_1^{X_1||A_1}(B_1)$
  if $B_2 \in \mathcal{L}_C[A_2]$ **then**
    $\mathsf{bad} \leftarrow \mathtt{true}$
  end if
end function

C.4 Π_3^{rev} with long messages

These are the worlds used in the proof of Theorem 14. Throughout, $|L_i| = |R_i| = k$ for all $i \in 1, \ldots, 6$, with $|A_i| = (m - 2k)$, where $m = |M|$ is the length of M in blocks.

To keep both worlds each of the main transitions on the same page as each other, this column is intentionally blank.

Alg. 28: $\mathcal{W}_0$ is the Π_3^{rev} construction

function ENC(M)
  $L_1||A_1||R_1 \leftarrow M$
  $L_2||A_2||R_2 \leftarrow \mathcal{E}^\epsilon(L_1||A_1||R_1)$
  $L_3||A_3||R_3 \leftarrow \overline{R_2}||\overline{A_2}||\overline{L_2}$
  $L_4||A_4||R_4 \leftarrow \mathcal{E}^\epsilon(L_3||A_3||R_3)$
  $L_5||A_5||R_5 \leftarrow \overline{R_4}||\overline{A_4}||\overline{L_4}$
  $L_6||A_6||R_6 \leftarrow \mathcal{E}^\epsilon(L_5||A_5||R_5)$
  return $L_6||A_6||R_6$
end function
function DEC(M)
  $L_6||A_6||R_6 \leftarrow M$
  $L_5||A_5||R_5 \leftarrow \mathcal{D}^\epsilon(L_6||A_6||R_6)$
  $L_4||A_4||R_4 \leftarrow \overline{R_5}||\overline{A_5}||\overline{L_5}$
  $L_3||A_3||R_3 \leftarrow \mathcal{D}^\epsilon(L_4||A_4||R_4)$
  $L_2||A_2||R_2 \leftarrow \overline{R_3}||\overline{A_3}||\overline{L_3}$
  $L_1||A_1||R_1 \leftarrow \mathcal{D}^\epsilon(L_2||A_2||R_2)$
  return $L_1||A_1||R_1$
end function

Alg. 29: $\mathcal{W}_1$ expands out $\mathcal{W}_0$ and is equivalent

$\mathcal{P} \leftarrow \emptyset$
function $\textsc{Enc}(M)$
  $L_1||A_1||R_1 \leftarrow M$
  $L_2||A_2 \leftarrow \mathcal{E}^{\epsilon}(L_1||A_1)$
  $\mathcal{P} \xleftarrow{\cup} L_1$
  $R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
  $L_3 \leftarrow \overline{R_2}$
  $A_3||R_3 \leftarrow \overline{A_2}||\overline{L_2}$
  $L_4 \leftarrow \mathcal{E}^{\epsilon}(L_3)$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_1 \leftarrow$ True
  end if
  $\mathcal{P} \xleftarrow{\cup} L_3$
  $A_4||R_4 \leftarrow \mathcal{E}^{L_3}(A_3||R_3)$ $\triangleright\triangleright\triangleright$
  $L_5 \leftarrow \overline{R_4}$
  $A_5||R_5 \leftarrow \overline{A_4}||\overline{L_4}$
  $L_6 \leftarrow \mathcal{E}^{\epsilon}(L_5)$
  if $L_5 \in \mathcal{P}$ **then**
    $\mathsf{bad}_2 \leftarrow$ True
  end if
  $\mathcal{P} \xleftarrow{\cup} L_5$
  $A_6||R_6 \leftarrow \mathcal{E}^{L_5}(A_5||R_5)$ $\triangleright\triangleright\triangleright$
  return $L_6||A_6||R_6$
end function
function $\textsc{Dec}(M)$
  $L_6||A_6||R_6 \leftarrow M$
  $L_5||A_5 \leftarrow \mathcal{D}^{\epsilon}(L_6||A_6)$
  $\mathcal{P} \xleftarrow{\cup} L_5$
  $R_5 \leftarrow \mathcal{D}^{L_5||A_5}(R_6)$
  $L_4 \leftarrow \overline{R_5}$
  $A_4||R_4 \leftarrow \overline{A_5}||\overline{L_5}$
  $L_3 \leftarrow \mathcal{D}^{\epsilon}(L_4)$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \xleftarrow{\cup} L_3$
  $A_3||R_3 \leftarrow \mathcal{D}^{L_3}(A_4||R_4)$ $\triangleright\triangleright\triangleright$
  $L_2 \leftarrow \overline{R_3}$
  $A_2||R_2 \leftarrow \overline{A_3}||\overline{L_3}$
  $L_1 \leftarrow \mathcal{D}^{\epsilon}(L_2)$
  if $L_1 \in \mathcal{P}$ **then**
    $\mathsf{bad}_4 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \xleftarrow{\cup} L_1$
  $A_1||R_1 \leftarrow \mathcal{D}^{L_1}(A_2||R_2)$ $\triangleright\triangleright\triangleright$
  return $L_1||A_1||R_1$
end function

Alg. 30: $\mathcal{W}_2$ is identical to $\mathcal{W}_1$ until bad

$\mathcal{P} \leftarrow \emptyset$
function $\textsc{Enc}(M)$
  $L_1||A_1||R_1 \leftarrow M$
  $L_2||A_2 \leftarrow \mathcal{E}^{\epsilon}(L_1||A_1)$ $\triangleright\triangleright\triangleright$
  $\mathcal{P} \xleftarrow{\cup} L_1$
  $R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
  $L_3 \leftarrow \overline{R_2}$
  $A_3||R_3 \leftarrow \overline{A_2}||\overline{L_2}$ $\triangleright\triangleright\triangleright$
  $L_4 \leftarrow \mathcal{E}^{\epsilon}(L_3)$ $\triangleright\triangleright\triangleright$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_1 \leftarrow$ True
  end if
  $\mathcal{P} \xleftarrow{\cup} L_3$
  $A_4||R_4 \xleftarrow{\$} \{0,1\}^{(m-k)n}$ $\triangleright\triangleright\triangleright$
  $L_5 \leftarrow \overline{R_4}$
  $A_5||R_5 \leftarrow \overline{A_4}||\overline{L_4}$ $\triangleright\triangleright\triangleright$
  $L_6 \leftarrow \mathcal{E}^{\epsilon}(L_5)$
  if $L_5 \in \mathcal{P}$ **then**
    $\mathsf{bad}_2 \leftarrow$ True
  end if
  $\mathcal{P} \xleftarrow{\cup} L_5$
  $A_6||R_6 \xleftarrow{\$} \{0,1\}^{(m-k)n}$
  return $L_6||A_6||R_6$
end function
function $\textsc{Dec}(M)$
  $L_6||A_6||R_6 \leftarrow M$
  $L_5||A_5 \leftarrow \mathcal{D}^{\epsilon}(L_6||A_6)$
  $\mathcal{P} \xleftarrow{\cup} L_5$
  $R_5 \leftarrow \mathcal{D}^{L_5||A_6}(R_6)$
  $L_4 \leftarrow \overline{R_5}$
  $A_4||R_4 \leftarrow \overline{A_5}||\overline{L_5}$ $\triangleright\triangleright\triangleright$
  $L_3 \leftarrow \mathcal{D}^{\epsilon}(L_4)$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \xleftarrow{\cup} L_3$
  $A_3||R_3 \xleftarrow{\$} \{0,1\}^{(m-k)n}$ $\triangleright\triangleright\triangleright$
  $L_2 \leftarrow \overline{R_3}$
  $A_2||R_2 \leftarrow \overline{A_3}||\overline{L_3}$ $\triangleright\triangleright\triangleright$
  $L_1 \leftarrow \mathcal{D}^{\epsilon}(L_2)$
  if $L_1 \in \mathcal{P}$ **then**
    $\mathsf{bad}_4 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \xleftarrow{\cup} L_1$
  $A_1||R_1 \xleftarrow{\$} \{0,1\}^{(m-k)n}$
  return $L_1||A_1||R_1$
end function

Alg. 31: $\mathcal{W}_3$: Remove superfluous code

$\mathcal{P} \leftarrow \emptyset$
function ENC(M)
  $L_1||A_1||R_1 \leftarrow M$
  $\mathcal{P} \leftarrow_{\cup} L_1$
  $R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
  $L_3 \leftarrow \overline{R_2}$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_1 \leftarrow$ True
  end if
  $\mathcal{P} \leftarrow_{\cup} L_3$
  $R_4 \leftarrow_{\$} \{0,1\}^{kn}$ ▷ ▷ ▷
  $L_5 \leftarrow \overline{R_4}$ ▷ ▷ ▷
  $L_6 \leftarrow \mathcal{E}^{\epsilon}(L_5)$
  if $L_5 \in \mathcal{P}$ **then**
    $\mathsf{bad}_2 \leftarrow$ True
  end if
  $\mathcal{P} \leftarrow_{\cup} L_5$
  $A_6||R_6 \leftarrow_{\$} \{0,1\}^{(m-k)n}$
  return $L_6||A_6||R_6$
end function
function DEC(M)
  $L_6||A_6||R_6 \leftarrow M$
  $L_5||A_5 \leftarrow \mathcal{D}^{\epsilon}(L_6||A_6)$
  $\mathcal{P} \leftarrow_{\cup} L_5$
  $R_5 \leftarrow \mathcal{D}^{L_5||A_5}(R_6)$
  $L_4 \leftarrow \overline{R_5}$
  $L_3 \leftarrow \mathcal{D}^{\epsilon}(L_4)$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \leftarrow_{\cup} L_3$
  $R_3 \leftarrow_{\$} \{0,1\}^{kn}$ ▷ ▷ ▷
  $L_2 \leftarrow \overline{R_3}$ ▷ ▷ ▷
  $L_1 \leftarrow \mathcal{D}^{\epsilon}(L_2)$ ▷ ▷ ▷
  if $L_1 \in \mathcal{P}$ **then**
    $\mathsf{bad}_4 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \leftarrow_{\cup} L_1$
  $A_1||R_1 \leftarrow_{\$} \{0,1\}^{(m-k)n}$
  return $L_1||A_1||R_1$
end function

Alg. 32: $\mathcal{W}_4$: Sampling commutes with perm.

$\mathcal{P} \leftarrow \emptyset$
function ENC(M)
  $L_1||A_1||R_1 \leftarrow M$
  $\mathcal{P} \leftarrow_{\cup} L_1$
  $R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
  $L_3 \leftarrow \overline{R_2}$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_1 \leftarrow$ True
  end if
  $\mathcal{P} \leftarrow_{\cup} L_3$
  $L_5 \leftarrow_{\$} \{0,1\}^{kn}$
  $L_6 \leftarrow \mathcal{E}^{\epsilon}(L_5)$
  if $L_5 \in \mathcal{P}$ **then**
    $\mathsf{bad}_2 \leftarrow$ True
  end if
  $\mathcal{P} \leftarrow_{\cup} L_5$
  $A_6||R_6 \leftarrow_{\$} \{0,1\}^{(m-k)n}$
  return $L_6||A_6||R_6$
end function
function DEC(M)
  $L_6||A_6||R_6 \leftarrow M$
  $L_5||A_5 \leftarrow \mathcal{D}^{\epsilon}(L_6||A_6)$
  $\mathcal{P} \leftarrow_{\cup} L_5$
  $R_5 \leftarrow \mathcal{D}^{L_5||A_5}(R_6)$
  $L_4 \leftarrow \overline{R_5}$
  $L_3 \leftarrow \mathcal{D}^{\epsilon}(L_4)$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \leftarrow_{\cup} L_3$
  $L_1 \leftarrow_{\$} \{0,1\}^{kn}$
  if $L_1 \in \mathcal{P}$ **then**
    $\mathsf{bad}_4 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \leftarrow_{\cup} L_1$
  $A_1||R_1 \leftarrow_{\$} \{0,1\}^{(m-k)n}$
  return $L_1||A_1||R_1$
end function

Alg. 33: $\mathcal{W}_5$: Simplify & reorder: A $\pm$PRF

$\mathcal{P} \leftarrow \emptyset$
function $\textsc{Enc}(M)$ $\triangleright\triangleright\triangleright$
  $L_1||A_1||R_1 \leftarrow M$
  $\mathcal{P} \leftarrow_{\cup} L_1$
  $R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
  $L_3 \leftarrow \overline{R_2}$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_1 \leftarrow$ True
  end if
  $\mathcal{P} \leftarrow_{\cup} L_3$
  $L_5 \leftarrow_{\$} \{0,1\}^{kn}$ $\triangleright\triangleright\triangleright$
  $L_6 \leftarrow \mathcal{E}^{\epsilon}(L_5)$
  if $L_5 \in \mathcal{P}$ **then** $\triangleright\triangleright\triangleright$
    $\mathsf{bad}_2 \leftarrow$ True $\triangleright\triangleright\triangleright$
  end if $\triangleright\triangleright\triangleright$
  $\mathcal{P} \leftarrow_{\cup} L_5$
  $A_6||R_6 \leftarrow_{\$} \{0,1\}^{(m-k)n}$ $\triangleright\triangleright\triangleright$
  return $L_6||A_6||R_6$ $\triangleright\triangleright\triangleright$
end function
function $\textsc{Dec}(M)$ $\triangleright\triangleright\triangleright$
  $L_6||A_6||R_6 \leftarrow M$
  $L_1||A_1||R_1 \leftarrow_{\$} \{0,1\}^{mn}$ $\triangleright\triangleright\triangleright$
  $L_5||A_5 \leftarrow \mathcal{D}^{\epsilon}(L_6||A_5)$
  $\mathcal{P} \leftarrow_{\cup} L_5$
  $R_5 \leftarrow \mathcal{D}^{L_5||A_5}(R_6)$
  $L_4 \leftarrow \overline{R_5}$
  $L_3 \leftarrow \mathcal{D}^{\epsilon}(L_4)$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \leftarrow_{\cup} L_3$
  if $L_1 \in \mathcal{P}$ **then** $\triangleright\triangleright\triangleright$
    $\mathsf{bad}_4 \leftarrow \mathtt{true}$ $\triangleright\triangleright\triangleright$
  end if $\triangleright\triangleright\triangleright$
  $\mathcal{P} \leftarrow_{\cup} L_1$
  return $L_1||A_1||R_1$ $\triangleright\triangleright\triangleright$
end function

Alg. 34: $\mathcal{W}_6$: Remove useless output

$\mathcal{P} \leftarrow \emptyset$
function $\textsc{Enc}(M, L_5)$
  $L_1||A_1||R_1 \leftarrow M$
  $\mathcal{P} \leftarrow_{\cup} L_1$
  $R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
  $L_3 \leftarrow \overline{R_2}$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_1 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \leftarrow_{\cup} L_3$
  $L_6 \leftarrow \mathcal{E}^{\epsilon}(L_5)$
  $\mathcal{P} \leftarrow_{\cup} L_5$
  return L_6
end function
function $\textsc{Dec}(M, L_1)$
  $L_6||A_6||R_6 \leftarrow M$
  $L_5||A_5 \leftarrow \mathcal{D}^{\epsilon}(L_6||A_6)$
  $\mathcal{P} \leftarrow_{\cup} L_5$
  $R_5 \leftarrow \mathcal{D}^{L_5||A_5}(R_6)$
  $L_4 \leftarrow \overline{R_5}$
  $L_3 \leftarrow \mathcal{D}^{\epsilon}(L_4)$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \leftarrow_{\cup} L_3$
  $\mathcal{P} \leftarrow_{\cup} L_1$
end function

Alg. 35: $\mathcal{W}_7$: Additional Oracles & simplify

$\mathcal{P} \leftarrow \emptyset$
function ENC(M, V_1, V_5)
  $\mathcal{P} \leftarrow_\cup V_1$
  $\mathcal{P} \leftarrow_\cup V_5$
  $L_1||A_1||R_1 \leftarrow M$
  $R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
  $L_3 \leftarrow \overline{R_2}$ ▷▷▷
  if $L_3 \in \mathcal{P}$ **then** ▷▷▷
    $\mathsf{bad}_1 \leftarrow$ True
  end if
  $\mathcal{P} \leftarrow_\cup L_3$
end function
function DEC(M, V_1, V_5)
  $\mathcal{P} \leftarrow_\cup V_1$
  $\mathcal{P} \leftarrow_\cup V_5$
  $L_6||A_6||R_6 \leftarrow M$
  $L_5||A_5 \leftarrow \mathcal{D}^\epsilon(L_6||A_6)$ ▷▷▷
  $R_5 \leftarrow \mathcal{D}^{L_5||A_5}(R_6)$
  $L_4 \leftarrow \overline{R_5}$ ▷▷▷
  $L_3 \leftarrow \mathcal{D}^\epsilon(L_4)$ ▷▷▷
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \leftarrow_\cup L_3$
end function

function $\mathcal{E}$-ACCESS(S)
  ENSURE$(|S| = kn)$
  return $\mathcal{E}^\epsilon(S)$
end function
function $\mathcal{D}$-ACCESS(S)
  ENSURE$(|S| = kn)$
  return $\mathcal{D}^\epsilon(S)$
end function

Alg. 36: $\mathcal{W}_8$: Additional Oracles do not help

$\mathcal{P} \leftarrow \emptyset$
function ENC(M, V_1, V_5)
  $\mathcal{P} \leftarrow_\cup V_1$
  $\mathcal{P} \leftarrow_\cup V_5$
  $L_1||A_1||R_1 \leftarrow M$
  $R_2 \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$

  if $R_2 \in \mathcal{P}$ **then**
    $\mathsf{bad}_1 \leftarrow$ True
  end if
  $\mathcal{P} \leftarrow_\cup L_3$
end function
function DEC(M, V_1, V_5)
  $\mathcal{P} \leftarrow_\cup V_1$
  $\mathcal{P} \leftarrow_\cup V_5$
  $L_6||A_6||R_6 \leftarrow M$
  $L_5||A_5 \leftarrow L_6||A_6$
  $R_5 \leftarrow \mathcal{D}^{L_5||A_5}(R_6)$

  $L_3 \leftarrow (R_5)'$ ▷ $x' := \overline{\mathcal{D}^\epsilon(\overline{x})}$
  if $L_3 \in \mathcal{P}$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
  end if
  $\mathcal{P} \leftarrow_\cup L_3$
end function

function $\mathcal{E}$-ACCESS(S)
  ENSURE$(|S| = kn)$
  return $\mathcal{E}^\epsilon(S)$
end function
function $\mathcal{D}$-ACCESS(S)
  ENSURE$(|S| = kn)$
  return $\mathcal{D}^\epsilon(S)$
end function

Alg. 37: $\mathcal{W}_9$: Adversary non-adaptive

Require: $|\mathcal{P}| = 2q, |\mathcal{M}_E| = q_E, |\mathcal{M}_D| = q_D$
Require: order $\in \{E, D\}^q$
$\mathcal{R}, \mathcal{L}$ empty lists
function CHALLENGE($\mathcal{P}, \mathcal{M}_E, \mathcal{M}_D$)
    for $i \in [1, \ldots, q_E]$ **do**
        $L_1||A_1||R_1 \leftarrow \mathcal{M}_E[i]$
        $\mathcal{R}_i \leftarrow \mathcal{E}^{L_1||A_1}(R_1)$
    end for
    for $i \in [1, \ldots, q_D]$ **do**
        $L_1||A_1||R_1 \leftarrow \mathcal{M}_D[i]$
        $L_2 \leftarrow \mathcal{D}_*^{L_1||A_1}(R_1)$
        $\mathcal{L}_i \leftarrow \pi(L_2)$
    end for
    if List $\mathcal{L}||\mathcal{R}$ contains repeats **then**
        bad $\leftarrow$ `true`
    end if
    for $i \in [1, \ldots, q]$ **do**
        if order$[i] = E$ **then**
            if $\mathcal{R}_e \in \mathcal{P}[1..2i]$ **then**
                bad $\leftarrow$ `true`
            end if
            $e \leftarrow e + 1$
        else
            if $\mathcal{L}_d \in \mathcal{P}[1..2i]$ **then**
                bad $\leftarrow$ `true`
            end if
            $d \leftarrow d + 1$
        end if
    end for
end function