Turning Online Ciphers Off

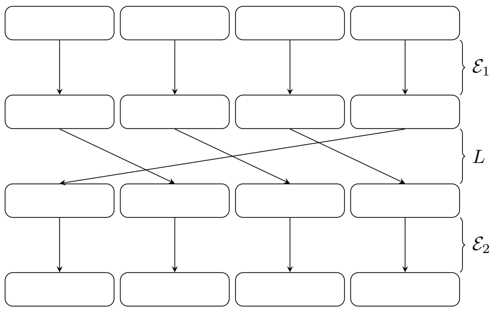
Elena Andreeva[1], Guy Barwell[2], Ritam Bhaumik[3], Mridul Nandi[3], Dan Page[2] and Martijn Stam[2]

[1] Department of Electrical Engineering, ESAT/COSIC, KU Leuven, Belgium.
elena.andreeva@esat.kuleuven.be ,
[2] Department of Computer Science, University of Brlistol,
Merchant Venturers Building, Woodland Road,
Brlistol, BS8 1UB, United Kingdom.
{guy.barwell,daniel.page,martijn.stam}@bris.ac.uk ,
[3] Indian Statistical Institute, Kolkata.
{bhaumik.ritam,mridul.nandi}@gmail.com

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 available as input. The immediacy of an online cipher affords a clear performance advantage, but it comes at a price: ciphertext blocks cannot depend on later plaintext blocks, limiting diffusion and hence security. We show how one can attain the best of both worlds by providing provably secure constructions, achieving full cipher security, based on applications of an online cipher around blockwise reordering layers.
Explicitly, we show that with just two calls to the online cipher, prp 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. We provide a full proof of this for a prp construction, and, in the $\pm$prp setting, security against adversaries who make queries of any single length. As part of our investigation, we extend an observation by Rogaway and Zhang by further 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

Contents



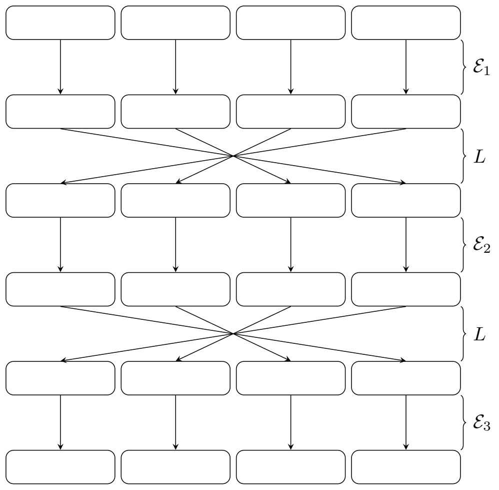


Figure 1: Examples of the construction. On the left is the two layer right shift scheme, and on the right the three layer reversing instantiated with independent ciphers.

1 Introduction

Modern understanding of symmetric cryptology has come a long way from a straightforward adaptation (cf. [?, Def. 3.30]) of the seminal definitions of probabilistic (public key) encryption [?]. In particular, both authenticated encryption (AE) 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 becomes available. 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).

Authenticated encryption provides both confidentiality and integrity (including of associated data [?]). 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 [?, ?]. The security goal of *misuse resistant* AE [?] 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 [?, ?], but the number of options reduces drastically when misuse resistance is required. One approach is the encode-then-encipher (or pad-then-encipher) paradigm [?, ?, ?], 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 (VIL) ciphers (either $\pm$prp or prp secure) are interesting in their own right, especially in scenarios where encryption has to occur *in situ* [?]. 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* [?] is a variable input length keyed permutation based on a blockcipher

Table 1: Simplified upper bounds on adversarial advantage against our constructions, where a small advantage implies a secure scheme. Results are parametrised by the maximum number of queries q of total length σ blocks, the blocksize N and $n = \log N$. Security bounds for adversaries making variable input length (VIL) or arbitrary input length (AIL) queries. A bound is "tight" if there exists an attack that asymptotically (in q, σ, N) matches the security bound.

Construction		Goal		Security		
Linear Layer	Ciphers	Access	Inputs	Advantage	Proof	Tight?
Right-shift	2	prp	VIL	$1.5\ q^2/N$	Theorem 2	Lemma 8
Right-shift	3	prp	VIL	$1.5(\sigma/N)^2$	Theorem 4	Lemma 11
Reversal	2	±prp	VIL	Insecure	—	Lemma 9
Reversal	2	±prp	AIL	q^2/N	Corollary 1	Lemma 8
Reversal	3	±prp	AIL	$n\,q/N$	Theorem 5	

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 [?].

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. In some circumstances, this might be solved by using two modes of operation on say AES, but this is not always possible. For example, HSMs do not generally allow direct access to their internal primitives, so direct access to the underlying blockcipher (or making single-block calls) may become prohibitively expensive (as each query typically carries a noticeable latency cost). So, when black-box use of an online cipher is already available (such as via an existing API), we are tasked to create a true cipher based on the access to the online cipher only.[1]

1.1 Our Contribution

We consider schemes formed by composing calls to an online cipher with a simple (publicly known) mixing layer, and aim to minimize the number of calls made to the online cipher (Definition 5). We restrict the mixing layer to be blockwise-linear (defined in Section 2.1), with particular focus on linear layers that simply reorder the blocks, since these can be implemented most efficiently. Figure 1 highlights two typical constructions under consideration. Note that neither reversing the blocks nor cycling the final block to the front is itself is novel: both ideas have been suggested in one way or the other, using more traditional IV-based encryption schemes [?] or in the context of key-wrap schemes [?]. The novelty of our contribution stems from 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

[1] Obviously if one had 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 [?].

describe an explicit correspondence between tweakable blockciphers and online ciphers (Theorem 1), extending an observation by Rogaway and Zhang [?].

If one is not concerned about an adversary making queries of the construction's inverse, then 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. Indeed, it suffices for the linear layer to move the final block to the start (Theorem 2), 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 (Lemma 8). Perhaps surprisingly, we find that three suffices: again using simply a right shift between layers leads to security until almost the blocksize (Theorem 4).

Security against adversaries making inverse queries (i.e. $\pm\mathsf{prp}$) is provided by using a linear layer that reverses the message. Unlike the prp case, using just two online cipher calls is not sufficient (Lemma 9), but security can be recovered by using a slightly modified construction (Theorem 3) or restricting the adversary to only making queries of a single length (Corollary 1). When three rounds are used, security is achieved beyond the birthday bound (Theorem 5), against adversaries who make queries of just a single length. We are not aware of any attacks matching this bound, and believe it to also hold in the VIL case (where adversaries may make queries of variable lengths), leaving these as open problems.

1.2 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 [?, ?]). Moreover, Hoang et al. demonstrate that with a tweakable cipher one may achieve the even stronger goal of Robust Authenticated Encryption [?, Theorem 5] (itself implying full misuse-resistant security [?]). Incorporating our results plugs the gap, allowing one to turn a secure online cipher into an Authenticated Encryption scheme meeting the strongest of security objectives.

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 [?, Section 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 may reduce hardware or code footprint, while decreasing the cost and complexity of certification or verification. 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.

1.3 Related Work

The concept of an online cipher was first studied by Bellare et al. [?], 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 [?, ?, ?, ?].

Later, Rogaway and Zhang exposed the close relationship between tweakable blockciphers and online ciphers [?], an observation that has since been exploited by others, yielding several explicit constructions (e.g. McOE [?]). There now exist a wide range of online cipher constructions, such as COPE [?], POE [?] and ELmE [?], the majority of which achieve birthday bound security. We are not aware of any online ciphers whose security might extend beyond the birthday bound; we consider this an interesting open problem in its own right, for which our research provides additional motivation.

As part of our study, we investigated some constructions similar to the CMC-core [?] (Section 4.2), finding that upgrading the CBC sections to secure online ciphers was not sufficient to allow removal of the mixing layer. One of the three round constructions is similar to the PIV construction of Shrimpton and Terashima [?], but as discussed in Section 5 their results cannot be applied in our setting due to the structure available to the adversary from the internal online cipher.

The original AESKW algorithm [?] 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 [?] uses the cyclic shift instead. Our results are a step towards proving the security of these standardized key wrap mechanisms, but are not directly applicable since their constructions also carry forward the state of the internal cipher.

As an alternative to our approach based on an online cipher, one can build a variable length cipher directly from a blockcipher (as TET [?] or AEZ [?] do), or extend the domain of a tweakable blockcipher (e.g. Minematsu's construction [?]). 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. If the online cipher itself is bootstrapped from a blockcipher to which a designer has direct access, arguably comparison in terms of blockcipher calls and overhead would be more relevant.

1.4 Context and Caveats

We will model the online cipher as an ideal online permutation, 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 chosen primitive). We will assume that the online ciphers are independent for every layer. This approximates the real constructions, and can itself be easily implemented with a single online cipher courtesy of the close relationship between tweakable blockciphers with arbitrary length tweaks and online ciphers (e.g. by prefixing each call with a marker corresponding to the appropriate cipher), or, alternatively keying or tweaking the ciphers independently suffices.

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

2 Preliminaries

2.1 Notation

Arrays and lists are indexed from 1, and initialised empty. 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. If L is a (finite) set, then $L \xleftarrow{\cup} x$ is shorthand for $L \leftarrow L \cup \{x\}$, whereas $X \xleftarrow{\$} L$ means that the variable X samples uniformly from L. $|L|$ denotes the number of elements of L. The symbol $\star$ denotes a wildcard that may take any value from the appropriate set.

We define bad as the union of bad events bad_i, so a game sets bad if bad_i is triggered for any i. At times bad^j (similarly bad_i^j) will be used to denote the event of triggering bad (resp. bad_i) while interacting with the world or oracle j as appropriate.

To aid verifiability and clarity, longer proofs will be broken up into a number of claims: each proof of a claim will end with ■, and □ will denote the end of the overall proof.

Finally, we will use the terms encryption and decryption (or variants thereof) to describe forward and backward queries to the construction. We will use the notation E^T to refer to a cipher with tweak T, and E^{-T} will be the inverse of the cipher under this same tweak (the negation should be viewed as acting on the E rather than on the tweak).

Blocks and strings The results in this paper are intrinsically block oriented, with an exclusive focus on Σ^* rather than the more general $\{0,1\}^*$, and we allow this to guide our definitions. The set of *blocks* is a finite set Σ, where the *blocksize* $N = |\Sigma|$ is often inherited from some underlying blockcipher—usually $N = 2^{128}$.

A *string of blocks* (or simply *string*) is an element $S \in \Sigma^*$, where $\Sigma^* := \bigcup_{n=0}^{\infty} \Sigma^n$. The length of a string $|S|$ is its length in blocks. For a string 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. Conversely, $\overline{X}$ is the blockwise reversal of X, so if $m = |X|$ then $\overline{X} := X[m]||X[m-1]||\dots||X[1]$. We will also let $\lceil X \rceil^k := X[1..k]$ be the first k blocks of X (its so-called *prefix*). The set of non-empty strings is $\Sigma^+ := \bigcup_{n=1}^{\infty} \Sigma^n$, and thus $\Sigma^+ = \{\Sigma||x : x \in \Sigma^*\}$.

For any $x \in \{0,\dots,N^m-1\}$, denote by $\langle x \rangle_m$ an m-block string that unambiguously encodes x (the choice of encoding is not important, as long as it is injective). A function $f\colon \Sigma^* \to \Sigma^*$ 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.

2.2 Primitives

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

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

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

Thus a tweakable blockcipher (TBC) can be thought of as a collection of blockciphers, an instance of which is chosen by the tweak.

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 $\mathring{\mathrm{E}}$ of permutations on Σ^+ indexed by some $k \in \mathcal{K}$, where for any $A, B \in \Sigma^*$, $\mathring{\mathrm{E}}_k(A||B)[1..|A|] = \mathring{\mathrm{E}}_k(A)$.

This formalisation of an online cipher (due to Bellare et al. [?]) describes a construction that can output ciphertext blocks as soon as the corresponding plaintext blocks are available. The final type of cipher we define is not widely studied in and of itself, but often occurs internally as part of the constructions of authenticated encryption modes based on online ciphers (e.g. [?]), since it is less susceptible to length extension attack.

Definition 4 (Online-but-last cipher)**.** An *online-but-last cipher* is a cipher that is online for all but the final block. Thus it is a family E of permutations on Σ^+ indexed by some

```
function $\$^T(M)$
    $C \leftarrow_\$ \Sigma^{|M|}$
    if $(T, M, \star) \in \mathcal{F}$ then
        $C \leftarrow \mathcal{F}^T(M)$
    $\mathcal{F} \leftarrow_\cup (T, M, C)$
    return $C$

function $\$^{-T}(C)$
    $M \leftarrow_\$ \Sigma^{|C|}$
    if $(T, \star, C) \in \mathcal{F}$ then
        $M \leftarrow \mathcal{F}^{-T}(M)$
    $\mathcal{F} \leftarrow_\cup (T, M, C)$
    return $M$
```

Figure 2: A lazily sampled tweakable random function with inverse, with internal table $\mathcal{F}$.

$k \in \mathcal{K}$, where for any $m > 0$ and $A \in \Sigma^m$, $E_k(A||B)[1..(m-1)] = E_k(A)[1..(m-1)]$ for all $B \in \Sigma^*$.

In a purely syntactical sense, an online cipher is also online-but-last. However, a randomly sampled online-but-last cipher E is unlikely to be online, since w.h.p. $E_k(A||B)[m] \neq E_k(A)[m]$ for any $B \in \Sigma^+$. To pre-empt ourselves somewhat, this means a secure online cipher is not a secure online-but-last cipher when queries can be made of multiple lengths (the VIL setting). However, when queries can only be made of a single length (AIL), the definitions coincide.

Ideal Objects An online permutation on blocks Σ is a length preserving permutation from $\mathring{\pi}\colon \Sigma^* \to \Sigma^*$ such that $\mathring{\pi}(A||B)[1..|A|] = \mathring{\pi}(A)$ for all $A, B \in \Sigma^*$. Define $\mathrm{OPerm}(\Sigma)$ as the set of all online permutations $\mathring{\pi}$ with blocks Σ. Then the ideal online permutation samples a permutation from $\mathrm{OPerm}(\Sigma)$ uniformly, making every possible online permutation equally likely.

Similarly, we define the ideal primitives for random functions, tweakable block ciphers and ciphers by first defining the set of all such objects, following standard terminology neatly collated by Halevi and Rogaway [?]. Define $\mathrm{Perm}(\Sigma^*)$ as the set of all length-preserving permutations $\pi : \Sigma^* \to \Sigma^*$; $\mathrm{Func}(\Sigma^*)$ to be the set of all length-preserving functions $\$\colon \Sigma^* \to \Sigma^*$; $\mathrm{Func}(\mathcal{T}, \Sigma)$ the set of all functions $\$\colon \mathcal{T} \times \Sigma \to \Sigma$ where for any $T \in \mathcal{T}$ the map $M \to \tilde{\pi}(T, M)$ is length-preserving; and $\mathrm{Perm}(\mathcal{T}, \Sigma)$ the set of functions $\tilde{\pi}\colon \mathcal{T} \times \Sigma \to \Sigma$, where for any $T \in \mathcal{T}$ the map $M \to \tilde{\pi}(T, M)$ is a length-preserving permutation. In each case the ideal construction samples an element from the set uniformly at random.

Slightly more involved is the ideal (tweakable) random function with inverse, in which the encryption and decryption interfaces are instantiated with independently sampled (tweakable) random functions, subject to the condition that they never contradict oneanother. This is commonly 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 Figure 2 for a code-based definition).

2.2.1 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. We measure this in terms of the probability an adversary can distinguish between the real scheme and an idealised version.

Adversarial advantages. To capture indistinguishability, we will provide an adversary access to one of two worlds; the adversary's task is to determine with which he is communicating. Each *world* is a collection of *oracles*—in this paper interfaces to a function (such as a permutation) and possibly its "inverse" . Eventually an adversary $\mathbb{A}$ interacting with $\mathcal{W}$ will terminate with output x, which we denote by $\mathbb{A}^{\mathcal{W}} \to x$.

Adversaries are computationally unbounded, but only allowed a limited number of queries to the available oracles. Without loss of generality, we assume these information theoretic adversaries are deterministic and minimal (so do not make queries equivalent to those already made, such as repeating queries).

The *distinguishing advantage* for adversary $\mathbb{A}$ between worlds $\mathcal{W}_0$ and $\mathcal{W}_1$ is

$$\underset{\mathbb{A}}{\Delta}\begin{pmatrix}\mathcal{W}_0\\ \mathcal{W}_1\end{pmatrix} := \left|\mathbb{P}\left[\mathbb{A}^{\mathcal{W}_0} \to 1\right] - \mathbb{P}\left[\mathbb{A}^{\mathcal{W}_1} \to 1\right]\right|.$$

We generalise this to classes of adversaries by taking the maximum advantage of any adversary in the class. In particular, the maximum advantage for all adversaries making at most q queries is

$$\underset{q}{\Delta}\begin{pmatrix}\mathcal{W}_0\\ \mathcal{W}_1\end{pmatrix} := \max_{\substack{\mathbb{A}\in\text{Adversaries}\\ \mathbb{A}\text{ makes }\leq q\text{ queries}}} \underset{\mathbb{A}}{\Delta}\begin{pmatrix}\mathcal{W}_0\\ \mathcal{W}_1\end{pmatrix}.$$

Standard security notions The security notions of prf, prp, tprf, tprp and oprp are defined by the adversarial advantage in distinguishing a primitive from the ideal random function, ideal permutation, ideal tweakable permutation and ideal online permutation respectively (when provided with oracle access to just the encryption interface). Analogously, we define $\pm\mathsf{prf}$, $\pm\mathsf{prp}$, $\pm\mathsf{tprf}$, $\pm\mathsf{tprp}$, $\pm\mathsf{oprp}$ by providing oracle access to both the forward and inverse interfaces. We delay the security definitions for an online-but-last cipher to Section 3.2. The complete list is provided in Appendix B, but as an example,

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

For some goal xxx, $\mathbf{Adv}^{\mathrm{xxx}}_{P}(q)$ is defined as the maximum across all adversaries making q queries. Informally, we say a scheme P is a secure xxx if $\mathbf{Adv}^{\mathrm{xxx}}_{P}(q)$ is sufficiently small.

AIL and VIL For constructions allowing variable input lengths (such as online ciphers), the default security notions allow an adversary to make queries of any (adaptively chosen) lengths. This security notion is referred to as the Variable Input Length (VIL) setting (e.g. [?]). The Arbitrary Input Length (AIL) setting is a restriction of this, allowing the adversary to make queries of any *single* length. Effectively, the first query an adversary makes, fixes this length.

The AIL flavour can be thought of as a point-wise security definition, in that it holds for any (single) length, but not necessarily multiple lengths at once. The stronger VIL notion corresponds to a uniform security definition, holding across all lengths simultaneously. We differentiate between the two settings by denoting AIL instances by (for example) $\pm\mathsf{prp}\bullet$, letting $\pm\mathsf{prp}$ denote the VIL case.

2.3 Composition Constructions

We seek a framework for efficiently converting a secure online cipher into a fully secure cipher, ideally using only a small number of calls to the online cipher, sandwiched together around some highly efficient invertible mixing layer(s). We ignore pre- or post-whitening layers, because an adversary can trivially remove them.

Our proofs will be in the standard model, and will all begin by switching out the online cipher for an ideal online permutation. This step is essentially unavoidable, and requires the online cipher be secure against adversaries with the access corresponding to that which they will have to the overall construction. To construct a $\pm$prp, we require the online cipher be an $\pm$oprp; for a prp it suffices for the online cipher to be an oprp. Henceforth, we will take this step as read, and instead directly bound the remaining term: the advantage of distinguishing a construction built around ideal online permutations from a true cipher.

Together, these observations motivate the following definition:

Definition 5 (The Π_i^L construction)**.** Define Π_i^L to be the composition of i calls to ideal online permutations $\mathcal{E}_1, \ldots, \mathcal{E}_i$ around $(i-1)$ applications of a public family of blockwise linear layers L.

So, for example, $\Pi_2^L(M) = \mathcal{E}_2 \circ L \circ \mathcal{E}_1(M)$. By considering various combinations of (L, i), we will observe that some combinations lead to schemes with prp or $\pm$prp• security. When clear, we may omit the linear layer from the notation.

Just using one call to the online permutation layer implies there is no linear layer, thus Π_1 equals $\mathcal{E}_1$, and so is online. Thus it can be trivially distinguished from a prp• with two queries. Explicitly, the ciphertexts of $\langle 0\rangle_1||\langle 0\rangle_1$ and $\langle 0\rangle_1||\langle 1\rangle_1$ will always agree on the first block of output for an online cipher, yet rarely for a true prp. Similarly, when L is the identity function, for any i the composition Π_i^L is simply that of multiple online permutation calls, and thus also an online permutation, meaning the same attack applies.

Candidate linear layers The 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 as the simplest map that may reasonably lead to a $\pm$prp. Inspired by the choices of AESKW [?], we will also consider the right circular shift right and by association its inverse, the left circular shift left. Formally, for any $M \in \Sigma^m$, these maps are defined by:

$$\begin{aligned}
\mathsf{right}(M) &:= M[m] \;||\; M[1] \;||\; \ldots \;||\; M[m-2] \;||\; M[m-1] \\
\mathsf{left}(M) &:= M[2] \;||\; M[3] \;||\; \ldots \;||\; M[m] \;||\; M[1] \\
\mathsf{rev}(M) &:= M[m] \;||\; M[m-1] \;||\; \ldots \;||\; M[2] \;||\; M[1]\ .
\end{aligned}$$

3 Initial Observations and Standard Results

Before our main investigation, we first cover a number of auxiliary results that later proofs will build on. After recalling a number of well-known results, we move on to explore the close relationship between online ciphers and tweakable blockciphers.

3.1 Standard Proof Techniques

3.1.1 Classical Bounds

To bound various collision events, we'll use the well-known birthday bound (Lemma 1). Lemma 2 reproduces the $\pm$prp–$\pm$prf switching lemma by Halevi and Rogaway [?, Appendix C], which we use in several proofs to hop from a random permutation to a random function with inverse.

Lemma 1 (Birthday bound)**.** *The probability that a list of q independent random variables sampled uniformly from Σ^m contains a repeat is bounded. Explicitly,*

$$\frac{q(q-1)}{4 \cdot N^m} \le \mathbb{P}\left[a_1, \ldots, a_q \leftarrow\!\!\$\ \Sigma^m \;:\; \exists i \neq j \text{ s.t. } a_i = a_j\right] \le \frac{q(q-1)}{2 \cdot N^m}$$

where the lower bound requires $q \le \sqrt{2N^m}$, and the upper bound holds for all q.

Lemma 2 (±prp–±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,* $\underset{q}{\Delta}\left(\begin{smallmatrix}\pi\ ,\ \pi^{-1}\\ \$\ ,\ \$^{-1}\end{smallmatrix}\right) \leq \frac{1}{2}q(q-1)/N^m$.

3.1.2 Game-Based Proof Techniques and Lazy Sampling

In this section we discuss a number of game-based proof techinques, based on work by Bellare and Rogaway [?]. We will model standard primitives through "Lazy Sampling", where random elements are only sampled when required. Internal variables of a construction are shared between the corresponding forward and backward interfaces. As an example, Figure 3 defines an oracle that implements an ideal tweakable permutation through lazy sampling, with both $\tilde{\mathrm{E}}$ and $\tilde{\mathrm{D}}$ having access to the same internal table π that records previously determined values. Similarly, we will later encounter Figure 5, defining a lazily sampled ideal online permutation.

Triples as functions To concisely describe the partial state of a lazily sampled function, we will allow sets of triples to represent partially defined tweakable functions, slightly overloading the notation. Specifically, the current state of the lazily sampled, tweakable function π is identified with the set π representing its internal table, so $\pi^T(M) = C \iff (T, M, C) \in \pi$ and similarly $\pi^{-T}(C) = M \iff (T, M, C) \in \pi$. Note that neither π as a set nor as a function incorporate any lazy sampling on its own: any changes to π will have to be explicit (and we'll use separate notation to denote the function which additionally takes care of the sampling). In this work, all functions represented in this way will be of the form $\pi\colon \Sigma^* \times \Sigma \to \Sigma$ (the syntax of a tweakable blockcipher introduced in Definition 2), so we will introduce the following shorthand, allowing ourselves to add a string of blocks in one go. For any k and $M, C \in \Sigma^k$, and any $T \in \Sigma^*$, we let $\pi \xleftarrow{\cup} (T, M, C)$ concisely represent "For every $i \in \{1, \ldots, k\}$, $\pi \xleftarrow{\cup} (T||\lceil M\rceil^{i-1}, M[i], C[i])$".

Identical-until-bad Several of our proofs will demonstrate two worlds are indistinguishable until the adversary triggers some event bad, known as identical-until-bad. This allows us to bound the (in)security of the construction by bounding the probability that an adversary communicating with either of the two (until this point identical) games can cause a bad event. We generalise the traditional terminology somewhat, defining what it means for two worlds to be identical-until-bad, and being less restrictive on our definition of "identical".

Bellare and Rogaway define identical-until-bad as two code-based games that only differ in branches that necessarily set bad. We soften this demand by instead requiring that the games act identically until a query that sets bad. This means that, until bad is set, all output variables are identically distributed, as are any variables made available to other oracles or interfaces, such as bad flags or internal tables π. This subtle difference allows us to merge the identical-until-bad and code-reordering steps of the seminal work, and allows us to consider a whole sequence of games to be identical-until-bad, as long as every one of them acts identically until bad is set.

Two worlds are identical-until-bad if for any adversary the games induced by the adversary interacting one of the worlds are identical-until-bad. We might also refer to two oracles being identical-until-bad, in which case the corresponding worlds that are identical-until-bad should be clear from the context.

Implications A direct result of using lazy sampling is that it becomes apparent how similar several of the ideal primitives are to one-another. Our next lemma states this, providing an alternative description of the afore-mentioned ±prp–±prf switch in terms of actual code-based games, as well as incorporating tweaks: it is this version we will

```
function $\tilde{\mathrm{E}}^T(M)$
    if $(T, M, \star) \notin \pi$ then
        $C \leftarrow_{\$} \Sigma \setminus \mathsf{image}(\pi^T)$
        $\pi \leftarrow_{\cup} (T, M, C)$
    return $\pi^T(M)$

function $\tilde{\mathrm{D}}^T(C)$
    if $(T, \star, C) \notin \pi$ then
        $M \leftarrow_{\$} \Sigma \setminus \mathsf{domain}(\pi^T)$
        $\pi \leftarrow_{\cup} (T, M, C)$
    return $\pi^{-T}(C)$
```

Figure 3: Modelling the tweakable blockcipher $\tilde{\mathrm{E}}$ with inverse $\tilde{\mathrm{D}}$ via lazy sampling.

use for the actual switching within later proofs. The associated code, given in Figure 4, defines a tweakable random permutation, a random function, or a pair of random samplers, depending on which sections of optional code are included. Moreover, since the oracles given are all identical-until-bad, any combination of the four bad events can be used for switching. Thus one may switch a random permutation for something that is a random permutation in the forward direction, and a random sampler in the inverse direction.

If the bad events are bounded directly this technique often leads to excessively large adversarial advantages (discussed further in Section 3.3). However, when carefully combined such that bad events align between switches it can be an effective and efficient tool.

Lemma 3 (Code-based $\pm\mathsf{prp}$ switches)**.** *Until either tweak–input or tweak–output pairs repeat, an ideal tweakable permutation is indistinguishable from a pair of tweakable random samplers. Moreover, a partial switch can also be performed to exchange the two oracles of an ideal tweakable permutation for a tweakable random permutation in one direction and a random sampler in the other. In particular, let $\tilde{\pi}$ be an ideal tweakable permutation, $\$$ an ideal tweakable function, and $\tilde{r}$ a tweakable random sampler. Then, for any adversary $\mathbb{A}$,*

$$\underset{\mathbb{A}}{\Delta}\begin{pmatrix}\tilde{\pi}, \tilde{\pi}^{-1} \\ \$, \$^{-1}\end{pmatrix} \leq \mathbb{P}\left[\mathsf{bad}_2 \vee \mathsf{bad}_4\right],$$

$$\underset{\mathbb{A}}{\Delta}\begin{pmatrix}\tilde{\pi}, \tilde{\pi}^{-1} \\ \tilde{r}, \tilde{r}^{-1}\end{pmatrix} \leq \mathbb{P}\left[\mathsf{bad}_1 \vee \mathsf{bad}_2 \vee \mathsf{bad}_3 \vee \mathsf{bad}_4\right],$$

$$\underset{\mathbb{A}}{\Delta}\begin{pmatrix}\tilde{\pi}, \tilde{\pi}^{-1} \\ \tilde{\pi}, \tilde{r}^{-1}\end{pmatrix} \leq \mathbb{P}\left[\mathsf{bad}_3 \vee \mathsf{bad}_4\right],$$

and similarly for other combinations.

Proof. This result follows by inspection of Figure 4. □

3.2 Equating Online Ciphers and Tweakable Block Ciphers

Online ciphers can be formed from a chain of tweakable blockciphers, an observation that allowed Rogaway and Zhang to simplify the analysis of online ciphers [**?**]. We observe that an even closer relationship exists: an online cipher *is* a tweakable blockcipher with variable length tweak. That an online cipher induces a tweakable blockcipher is simply setting a result of Bellare et al. into modern terminology [**?**, Proposition 1], while the converse is an extension of the result of Rogaway and Zhang. So, the result is not particularly surprising, but is nonetheless worth highlighting, since it yields neater terminology to study online ciphers.

We introduce some additional notation to better expose the relationship. Let $\mathring{\mathrm{E}}(\cdot)$ be an online cipher and $A, B \in \Sigma^*$ with $|A| = a$ and $|B| = b$. Then, define $\mathring{\mathrm{E}}^A(B) := \mathring{\mathrm{E}}(A||B)[(a+1)..(a+b)]$. So, $\mathring{\mathrm{E}}^A(B)$ returns the output blocks corresponding to B when processed with a *prefix* of A. By the online property, $\mathring{\mathrm{E}}(A||B) = \mathring{\mathrm{E}}^\epsilon(A)||\mathring{\mathrm{E}}^A(B)$. If we

```
function $\tilde{\mathrm{E}}^T(M)$
    $C \leftarrow_\$ \Sigma$
    if $(T, M, \star) \in \pi$ then
        $\mathsf{bad}_1 \leftarrow \mathtt{true}$
        $\boxed{C \leftarrow \pi^T(M)}$
    else if $(T, \star, C) \in \pi$ then
        $\mathsf{bad}_2 \leftarrow \mathtt{true}$
        $\boxed{C \leftarrow_\$ \Sigma \setminus \mathsf{image}(\pi^T)}$
    $\pi \leftarrow_\cup (T, M, C)$
    return $C$

function $\tilde{\mathrm{D}}^T(C)$
    $M \leftarrow_\$ \Sigma$
    if $(T, \star, C) \in \pi$ then
        $\mathsf{bad}_3 \leftarrow \mathtt{true}$
        $\boxed{M \leftarrow \pi^{-T}(C)}$
    else if $(T, M, \star) \in \pi$ then
        $\mathsf{bad}_4 \leftarrow \mathtt{true}$
        $\boxed{M \leftarrow_\$ \Sigma \setminus \mathsf{domain}(\pi^T)}$
    $\pi \leftarrow_\cup (T, M, C)$
    return $M$
```

Figure 4: Comparing tweakable random functions. The list π is initialised empty. To be a uniform sampler, we include none of the boxed code. A tweakable random function includes the unboxed code and also the code in dashed boxes. Finally, a tweakable random permutation uses all of the code. Thus we see the three constructions (and the various hybrids of them) are all identical-until-bad.

think of $\mathring{\mathrm{E}}^{\cdot}(\cdot)$ as a tweakable cipher, 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 [**?**], except that prefixes may be arbitrarily large, whereas states have fixed length.

Similarly, we define $\mathring{\mathrm{D}}^A(B) := \mathring{\mathrm{D}}(\mathring{\mathrm{E}}(A)||B)[(a+1)..(a+b)]$, which is the inverse of $\mathring{\mathrm{E}}^A(B)$, as $\mathring{\mathrm{D}}^A(\mathring{\mathrm{E}}^A(B)) = B$. Computing $\mathring{\mathrm{D}}^A(\cdot)$ is not unreasonably onerous, since to calculate $\mathring{\mathrm{D}}(A||B)$ one first calculates $M = \mathring{\mathrm{D}}(A)$, then $\mathring{\mathrm{D}}(A||B) = M||\mathring{\mathrm{D}}^M(B)$, something an online cipher tends to do internally anyway. Our notation emphasises the correspondence with tweakable blockciphers, since is A the tweak under which B is decrypted.

Theorem 1. *There is a security-preserving, one-to-one correspondence between online ciphers on blocks Σ and tweakable blockciphers on Σ with tweak space Σ^*.*

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

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

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

We observe that for any tweakable blockcipher $\tilde{\mathrm{E}}$ and online cipher $\mathring{\mathrm{E}}$ we have $f(g(\tilde{\mathrm{E}}))_k = \tilde{\mathrm{E}}_k$ and $g(f(\mathring{\mathrm{E}}))_k = \mathring{\mathrm{E}}_k$. 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 tweakable blockciphers onto the set of all online ciphers, and vice versa. That is, $f(\mathrm{OPerm}(\Sigma)) = \mathrm{Perm}(\Sigma^*, \Sigma)$ and $g(\mathrm{Perm}(\Sigma^*, \Sigma)) = \mathrm{OPerm}(\Sigma)$. So, if an online cipher is distinguishable from the ideal online cipher, by applying f we see that the corresponding tweakable blockcipher is distinguishable from the ideal tweakable blockcipher, and vice versa. Thus, security of one implies security of the other. $\square$

```
function $\mathcal{E}^P(M)$
    $C \leftarrow_\$ \Sigma$
    if $(P, M[1], \star) \in \mathring{\pi}$ then
        $C \leftarrow \mathring{\pi}^P(M[1])$
    else if $(P, \star, C) \in \mathring{\pi}$ then
        $C \leftarrow_\$ \Sigma \setminus \mathsf{image}(\mathring{\pi}^P)$
    $\mathring{\pi} \leftarrow_\cup (P, M[1], C)$
    $C' \leftarrow \mathcal{E}^{P||M[1]}(M[2..|M|])$
    return $C||C'$
```

```
function $\mathcal{D}^P(C)$
    $M \leftarrow_\$ \Sigma$
    if $(P, \star, C[1]) \in \mathring{\pi}$ then
        $M \leftarrow \mathring{\pi}^{-P}(C[1])$
    else if $(P, M, \star) \in \mathring{\pi}$ then
        $M \leftarrow_\$ \Sigma \setminus \mathsf{image}(\mathring{\pi}^P)$
    $\mathring{\pi} \leftarrow_\cup (P, M, C[1])$
    $C' \leftarrow \mathcal{D}^{P||M}(C[2..|C|])$
    return $M||M'$
```

Figure 5: A mechanism for lazily sampling an ideal online permutation $\mathcal{E}$ with inverse $\mathcal{D}$ and internal table $\mathring{\pi}$ after processing prefix P. The table $\mathring{\pi}$ is initialised empty. We observe that if $(P, M, \star) \notin \mathring{\pi}$ then $(P||M, \star, \star) \notin \mathring{\pi}$ (because of how $\mathring{\pi}$ is filled in). The routines above could be simplified by directly sampling from outside the image of $\mathring{\pi}$ rather than first sampling uniformly; this mechanism was chosen because to highlight the relationship with tweakable permutations and facilitate later switches.

Online-but-last ciphers An online-but-last cipher is a cipher that is online for all but the final block. We can expose this feature by extending the tweakable blockcipher correspondence by adding a second tweakable blockcipher for the final block. So, an online-but-last cipher $\mathcal{O}_{obl}$ is the concatenation of an online cipher $\mathring{\mathrm{E}}$, and a separate tweakable block cipher $\widetilde{\mathcal{E}}$. That is, for a query $M \in \Sigma^m$,

$$\mathcal{O}_{obl}(M) = \mathring{\mathrm{E}}(M[1..(m-1)])||\widetilde{\mathcal{E}}^{M[1..(m-1)]}(M[m]).$$

3.3 Ideal Online Permutations

For the remainder of the paper, let $\mathcal{E}$ be the encryption routine of a lazily sampled ideal online permutation, with inverse $\mathcal{D}$, and internal (partial) table $\mathring{\pi}$. Figure 5 presents a code-based definition for how $\mathcal{E}$ and $\mathcal{D}$ operate and use $\mathring{\pi}$ internally. When multiple ciphers $\mathcal{E}_i$ are used, the corresponding internal tables will be denoted $\mathring{\pi}_i$.

Immediate properties of ideal online permutations Applying the correspondence between tweakable blockciphers and online ciphers, we can view an ideal online permutation as an ideal tweakable single-block permutation. Consequently, after processing a *fresh* prefix, the (remaining) output will appear uniformly random: if no call to $\mathcal{E}$ has been made explicitly or implicitly tweaked by A, then $\mathcal{E}^A(B)$ is uniformly sampled from all strings of length $|B|$. That is, if the adversary has not yet queried $\mathcal{E}^A(\cdot)$, either directly or indirectly as part of a longer query, the output is uniformly random. Moreover, we can ensure independence between two calls to an ideal online permutation $\mathcal{E}$ by taking care with the length of tweaks: as long as $|t| \geq |u| + |y|$, $\mathcal{E}^t(x)$ is independent of $\mathcal{E}^u(y)$.

Even if tweaks repeat, the final blocks behave sufficiently random to derive meaningful results. 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 4. *Let $\mathcal{R} = (R_1, \ldots, R_q)$ be a list of q blocks, where each $R_i = \mathcal{E}^{t_i}(x_i)$ is the output of the encryption of a unique input by a random online permutation $\mathcal{E}$, meaning $t_i||x_i \neq t_j||x_j$ 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 $\mathbb{P}[\exists i \neq j \text{ s.t. } R_i, R_j \in \mathcal{R}] \leq \frac{1}{2}q(q-1)/N$, where the probability is taken over the choice of online permutation $\mathcal{E}$.*

Proof. Let $i \neq j$. Then, by construction, $R_i = R_j \iff \mathcal{E}^{t_i}(x_i) = \mathcal{E}^{x_j}(s_j)$. If $t_i = t_j$, then $R_i = R_j$ implies that $x_i = x_j$, 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 of oneanother. Thus R_i and R_j are both sampled uniformly at random and independently, and so collide with probability N^{-1}. So, in either case, $\mathbb{P}[R_i = R_j \mid i \neq j] \leq N^{-1}$. Applying the union bound, we get the required result. $\square$

3.3.1 Identical-until-bad switching lemmas

Compared to the previous results, the next two lemmas are slightly more cumbersome to define, but are essentially the extension of Lemma 3 (the code-based ±tprp-±tprf switches) into the online setting. They will allow us to replace an ideal online permutation with an alternative routine that is easier to reason about such that the replacement is undetectable until a bad event occurs. Lemma 5 demonstrates that we may switch between the oracles of Figure 6, and use $\mathcal{O}_2$ instead of $\mathcal{O}_1$, while Lemma 6 shows that the oracles of Figure 7 may be exchanged, in particular allowing $\mathcal{O}_4$ to replace $\mathcal{O}_3$, as long as bad does not occur.

We will require that all queries have a prefix of length p, and total length at least $p+1$, and will simplify the code by assuming that the prefix–message pair does not repeat, capturing this unwanted behaviour with a bad event. In each case, we begin by parsing the input into a triple (P, Q, R) such that $|P| = p$ and $|Q| = 1$. The choice of variable names (P, Q, R) is inspired partially by using P for "prefix", but also to reduce the number of variables using the same names in later results, making it easier to state which variable in the lemmas corresponds to which variable in the later theorems.

Direct and indirect adversaries An adversary with direct unrestricted access to any of the oracles $\mathcal{O}_1$, $\mathcal{O}_2$, $\mathcal{O}_3$ or $\mathcal{O}_4$ can easily trigger bad. Taking $\mathcal{O}_1$ as an example, querying (P, Q, R) then (P, Q, R') will trigger bad_A. So, when we apply these lemmas, we cannot simply bound the probability of bad in an abstract setting, and sum it with other partial bounds. Rather, we must more clearly describe the access the relevant adversary has, before bounding their ability of triggering bad.

Ultimately, we will be interested in the indistinguishability of constructions built around calls to ideal online permutations. An adversary will then only have access to the overall construction which itself triggers internal calls to the ideal online permutation, the inputs to which are not wholly controlled by the adversary. Lemma 5 and 6 tell us that it is acceptable to replace these ideal online permutation calls with slightly different objects, so long as the bad events do not occur. In this significantly more restricted setting, we will find that the events are sufficiently hard to trigger that they lead to meaningful security results.

Unique prefix queries Lemma 5 compares $\mathcal{O}_1$, which essentially describes access to $\mathcal{E}$, with $\mathcal{O}_2$, which replaces the second and third sections of the online permutation call with simpler routines. So, the lemma essentially allows replacement the later parts of an online permutation call with random sampling, as long as the prefix and first non-prefix block do not repeat.

Lemma 5. *Fix $p \geq 0$, let $\mathcal{E}$ be an ideal online permutation with internal table π, and let $\mathcal{O}_1$ and $\mathcal{O}_2$ be the routines given in Figure 6. Consider an adversary $\mathbb{A}$ with access to either $\mathcal{O}_1$ or $\mathcal{O}_2$. Assume all queries are at least $p+1$ blocks in length, parsed such that $|P_1| = p$ and $|Q_1| = 1$. Then, the oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ are identical-until-bad.*

Proof. $\mathcal{O}_1$ is simply the result of separating out a call to the random permutation via the online property. $\mathcal{O}_2$ replaces the second and third of these permutation calls with random sampling, neglecting to carry over the consistency checks.

```
function $\mathcal{O}_1(P_1||Q_1||R_1)$
    $\mathring{\pi}' \leftarrow \mathring{\pi}$
    $P_2 \leftarrow \mathcal{E}^{\epsilon}(P_1)$
    $Q_2 \leftarrow \mathcal{E}^{P_1}(Q_1)$
    $R_2 \leftarrow \mathcal{E}^{P_1||Q_1}(R_1)$
    if $(P_1, Q_1, \star) \in \mathring{\pi}'$ then
        $\mathsf{bad}_A \leftarrow$ true
    if $(P_1, \star, Q_2) \in \mathring{\pi}'$ then
        $\mathsf{bad}_{A'} \leftarrow$ true

    return $P_2||Q_2||R_2$
```

```
function $\mathcal{O}_2(P_1||Q_1||R_1)$

    $P_2 \leftarrow \mathcal{E}^{\epsilon}(P_1)$
    $Q_2 \leftarrow_\$ \Sigma$
    $R_2 \leftarrow_\$ \Sigma^{|R_1|}$
    if $(P_1, Q_1, \star) \in \mathring{\pi}$ then
        $\mathsf{bad}_A \leftarrow$ true
    if $(P_1, \star, Q_2) \in \mathring{\pi}$ then
        $\mathsf{bad}_{A'} \leftarrow$ true
    $\mathring{\pi} \leftarrow_\cup (P_1, Q_1||R_1, Q_2||R_2)$
    return $P_2||Q_2||R_2$
```

Figure 6: The oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ are identical until bad. In each case they have access to the ideal online permutation $\mathcal{E}$ and its internal table $\mathring{\pi}$ (See Figure 5). The bad events are equivalent, with the duplicate table $\mathring{\pi}'$ only used because the $\mathcal{E}$ calls in $\mathcal{O}_1$ update $\mathring{\pi}$, rather than it being updated after the bad tests. $\mathcal{O}_1$ is simply the result of expanding out the call to an online cipher via the online property, while $\mathcal{O}_2$ expands out some of the calls before simplifying to something that is identical-until-bad. We use shorthand from Section 3.1.2 to describe adding each input-output pair to $\mathring{\pi}$ under the appropriate tweak, and use a blank line in $\mathcal{O}_1$ to keep corresponding code aligned.

The first of these switches (that Q_2 may be uniformly sampled) is precisely the case $\mathsf{bad}_1 \vee \mathsf{bad}_2$ of Lemma 3. Since (P_1, Q_1) was fresh (else bad_A was set), R_1 is encrypted under a new prefix and thus sampled uniformly at random. Finally, these values are added to the table $\mathring{\pi}$ to ensure any future calls to the inverse function are consistent. Together, until bad occurs, $\mathcal{O}_2$ is identical to $\mathcal{O}_1$, as required. □

Random prefix queries Lemma 6 extends lemma 5 showing that if the prefix is randomly chosen outside the adversary's control then the output is essentially uniformly sampled from all strings of the appropriate rate.

The most useful application of it is that if the first p blocks of a message are randomly selected outside of the adversary's control, the output is indistinguishable from that of a random permutation until a collision occurs in $\mathcal{L}_A$ or $\mathcal{L}_{A'}$. A collision in $\mathcal{L}_{A'}$ is simply a random event, occurring with probability roughly $q^2/|\Sigma|^{p+1}$. Our constructions will be such that in general it is hard for an adversary to force a collision on Q_1 without ensuring uniqueness of P_1, meaning that the best strategy for triggering bad_A will also be a random collision, occurring with roughly the same probability. Thus, by ensuring the conditions for Lemma 6 are met, it will lead to an overall security bound of around $q^2/|\Sigma|^{p+1}$ for some appropriate value of p.

Lemma 6. *Fix $p \geq 0$, let $\mathcal{E}$ be an ideal online permutation with internal table π and inverse $\mathcal{D}$, and let $\mathcal{O}_3$ and $\mathcal{O}_4$ be the routines described in Figure 7. Consider an adversary $\mathbb{A}$ who may repeat queries, given access to either $\mathcal{O}_3$ or $\mathcal{O}_4$, making queries of at least 1 block in length, parsed such that $|Q_1| = 1$. Then, the oracles $\mathcal{O}_3$ and $\mathcal{O}_4$ are identical-until-bad. Moreover, until* bad*, the outputs of $\mathcal{O}_3$ or $\mathcal{O}_4$ are randomly sampled and non-repeating.*

Proof. Oracle $\mathcal{O}_3$ is the same as $\mathcal{O}_1$ except for an additional line that randomly samples P_1. So, applying Lemma 5, we can swap the corresponding code for the contents of $\mathcal{O}_2$. Now, since P_1 is sampled uniformly and $\mathcal{E}$ is a permutation with inverse $\mathcal{D}$, we can equivalently sample P_2 and calculate P_1. This yields $\mathcal{O}_4$, completing the first part of the claim.

For the second, we observe that $\mathcal{O}_4$ is a random function, since the output is independently uniformly sampled in three sections. Moreover, any output collisions imply

```
function $\mathcal{O}_3(Q_1||R_1)$
    $\mathring{\pi}' \leftarrow \mathring{\pi}$
    $P_1 \leftarrow_\$ \Sigma^p$
    $P_2 \leftarrow \mathcal{E}^\epsilon(P_1)$
    $Q_2 \leftarrow \mathcal{E}^{P_1}(Q_1)$
    $R_2 \leftarrow \mathcal{E}^{P_1||Q_1}(R_1)$
    if $(P_1, Q_1, \star) \in \mathring{\pi}'$ then
        $\mathsf{bad}_A \leftarrow$ true
    if $(P_1, \star, Q_2) \in \mathring{\pi}'$ then
        $\mathsf{bad}_{A'} \leftarrow$ true

    return $P_2||Q_2||R_2$
```

```
function $\mathcal{O}_4(Q_1||R_1)$

    $P_2 \leftarrow_\$ \Sigma^p$
    $P_1 \leftarrow \mathcal{D}^\epsilon(P_2)$
    $Q_2 \leftarrow_\$ \Sigma$
    $R_2 \leftarrow_\$ \Sigma^{|R_1|}$
    if $(P_1, Q_1, \star) \in \mathring{\pi}$ then
        $\mathsf{bad}_A \leftarrow$ true
    if $(P_1, \star, Q_2) \in \mathring{\pi}$ then
        $\mathsf{bad}_{A'} \leftarrow$ true
    $\mathring{\pi} \leftarrow_\cup (P_1, Q_1||R_1, Q_2||R_2)$
    return $P_2||Q_2||R_2$
```

Figure 7: Oracles $\mathcal{O}_3$ and $\mathcal{O}_4$ are identical-until-bad. $\mathcal{E}$ is an ideal online permutation, with inverse $\mathcal{D}$ and internal table $\mathring{\pi}$. As with figure 6, $\mathring{\pi}'$ duplicates $\mathring{\pi}$ to ensure tests are run against the input list, not an updated one. Inputs are at least 1 block long, parsed such that $|Q_1| = 1$, and $p \geq 0$ is a parameter. Oracle $\mathcal{O}_3$ corresponds to calling an ideal online permutation with a randomised prefix. The output of Oracle $\mathcal{O}_4$ is randomly sampled from $\Sigma^{p+1+|R_1|}$, the set of strings p blocks longer than its input. It does not repeat an output until bad, and until bad the output of $\mathcal{O}_4$ is indistinguishable from the output of a random permutation on $p + 1 + |R_1|$ blocks.

repeated values of (P_2, Q_2), which implies that (P_1, Q_2) must have repeated because P_1 is the image of P_2 under a permutation. This would have set $\mathsf{bad}_{A'}$, meaning outputs do not collide until $\mathsf{bad}_{A'}$ occurs. Thus $\mathcal{O}_4$, and so also $\mathcal{O}_3$, is identical-until-bad to a random permutation. □

3.4 An Identical Until Bad Funnel

Suppose we wish to bound the advantage $\mathbb{A}$ has at distinguishing two worlds $\mathcal{W}_0$ and $\mathcal{W}_2$. Assume we already have that $\mathcal{W}_0$ and $\mathcal{W}_1$ are identical-until-bad with bad event bad_A, and $\mathcal{W}_1$ and $\mathcal{W}_2$ are identical-until-bad with bad event bad_B. Applying the traditional game-hopping mechanism, one would then apply the triangle inequality to bound

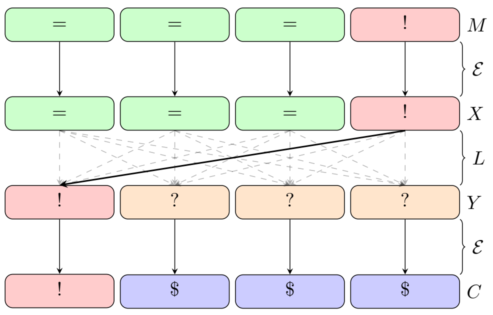
$$\underset{\mathbb{A}}{\Delta}\begin{pmatrix}\mathcal{W}_0\\ \mathcal{W}_2\end{pmatrix} \leq \underset{\mathbb{A}}{\Delta}\begin{pmatrix}\mathcal{W}_0\\ \mathcal{W}_1\end{pmatrix} + \underset{\mathbb{A}}{\Delta}\begin{pmatrix}\mathcal{W}_1\\ \mathcal{W}_2\end{pmatrix} \leq \mathbb{P}\left[\mathsf{bad}_A^1\right] + \mathbb{P}\left[\mathsf{bad}_B^1\right].$$

The identical-until-bad funnel mechanism is based on the observation that until $\mathsf{bad}_A \vee \mathsf{bad}_B$ occurs, $\mathcal{W}_1$ is identical to *both* $\mathcal{W}_0$ *and* $\mathcal{W}_2$. Thus, we can more directly bound their distinguishing distance, with

$$\underset{\mathbb{A}}{\Delta}\begin{pmatrix}\mathcal{W}_0\\ \mathcal{W}_2\end{pmatrix} \leq \mathbb{P}\left[\mathsf{bad}_A^1 \vee \mathsf{bad}_B^1\right] = \mathbb{P}\left[\mathsf{bad}_A^2 \vee \mathsf{bad}_B^2\right] \leq \mathbb{P}\left[\mathsf{bad}^2\right].$$

In cases where bad_A and bad_B are independent this might not gain us anything, but if bad_A and bad_B are statistically dependent this immediately leads to an improved security bound. Applying induction to this result leads to the following wholly unsurprising (but certainly useful) result.

Lemma 7 (Identical-until-bad funnels). *Let $\mathcal{W}_0, \ldots, \mathcal{W}_n$ be oracles or worlds. Suppose for all $i \in \{1, \ldots, n\}$ and any adversary $\mathbb{A}$ the worlds $\mathcal{W}_{i-1}$ and $\mathcal{W}_i$ are identical-until-*bad



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.

Figure 8: An attack against the $\mathsf{prp}\bullet$ security of Π_2^L (Lemma 8)

for some bad event $\mathsf{bad}_1 \vee \cdots \vee \mathsf{bad}_i$. *Then,*

$$\underset{\mathbb{A}}{\Delta}\begin{pmatrix}\mathcal{W}_0\\ \mathcal{W}_n\end{pmatrix} \leq \mathbb{P}\left[\mathsf{bad}_1 \vee \cdots \vee \mathsf{bad}_n\right] \leq \mathbb{P}\left[\mathsf{bad}^n\right].$$

We term the mechanism a *funnel* because each step further restricts the region over which the equivalences all hold, "funneling" us into a more restrictive region bounded by all the bad events. Observe that a requirement for this routine is that each step beyond the first preserves every bad events used previously, so that all required events continue to make sense. That said, maintaining the previous requirements is not entirely unhelpful.

Firstly, the funnel allows us to delay bounding the probability of the bad event until the final oracle, which tends to be simpler to reason about.

It also allows the designer to "reuse" previous bad events "for free", without collecting additional terms into our final security bound. To illustrate this, suppose $\mathcal{W}_0$ and $\mathcal{W}_1$ are identical until bad_A; $\mathcal{W}_1$ and $\mathcal{W}_2$ are identical until $\mathsf{bad}_A \vee \mathsf{bad}_B$; $\mathcal{W}_2$ and $\mathcal{W}_3$ are also identical until $\mathsf{bad}_A \vee \mathsf{bad}_B$; and $\mathcal{W}_3$ and $\mathcal{W}_\epsilon$ are identical until $\mathsf{bad}_A \vee \mathsf{bad}_B \vee \mathsf{bad}_C$. Then,

$$\underset{\mathbb{A}}{\Delta}\begin{pmatrix}\mathcal{W}_0\\ \mathcal{W}_\epsilon\end{pmatrix} \leq \mathbb{P}\left[\mathsf{bad}_A^3 \vee \mathsf{bad}_B^3 \vee \mathsf{bad}_C^3\right] \leq \mathbb{P}\left[\mathsf{bad}^3\right],$$

which may be substantially smaller than the four terms present in the traditional bound.

4 Two Layer Constructions

In this section we ask the natural question: what can be achieved using two layers? The most intuitive candidate for a $\pm\mathsf{prp}$ is Π_2^{rev}, but in Lemma 9 we show that this construction is not secure against adversaries who can vary input lengths. By instead using an online-but-last permutation as our primitive, we are able to achieve full $\pm\mathsf{prp}$ security (Theorem 3), even against variable length queries. This implies the $\mathsf{prp}\bullet$ security of the original Π_2^{rev} construction (Corollary 1).

First, in Lemma 8 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 previous attack, moving from guaranteed collisions in the first block of output to a scenario where the construction *never* collides on those blocks. When instantiating the linear layer with a simple one-block right shift, we get prp security up to this bound (Theorem 2).

Lemma 8. *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,* $\mathbf{Adv}_{\Pi}^{\mathsf{prp}\bullet}(q) \geq \frac{q(q-1)}{8\cdot N}$ *for all* $q \leq \sqrt{N}$.

Proof. Suppose $L(X)[1]$ is independent of $X[m]$ for messages of length $m = |X|$. Then the first input block to the second round is independent of the final input block. Thus the system can trivially be distinguished by querying two messages differing only in the final block and checking if they share the same first ciphertext block. Henceforth, we assume that $L(X)[1]$ depends on $X[m]$.

Let $M^t := \langle 0\rangle_{m-1}||\langle t\rangle_1$, where $m \geq 2$ is chosen arbitrarily to meet length requirements. The adversary $\mathbb{A}$ will vary t to make $q \leq N$ queries of this form, and $\mathbb{A} \to 1$ iff all q ciphertexts have distinct first blocks. Since $\mathbb{A}$ makes q queries, we have that $\mathbf{Adv}_{\Pi}^{\mathsf{prp}\bullet}(q) \geq \mathbb{P}\left[\mathbb{A}^{\Pi} \to 1\right] - \mathbb{P}\left[\mathbb{A}^{\pi} \to 1\right]$, terms we now bound directly.

We begin by calculating $\mathbb{P}\left[\mathbb{A}^{\Pi} \to 1\right]$, following the logic shown in Figure 8, 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 $m-1$ 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, $Y^t[1] = L(X^t)[1]$ depends 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. An online cipher called on just one block is a permutation, so equality of $C[1]$ blocks occurs if and only if there is 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 for all pairs $t \neq u$, the first blocks of the ciphertexts will differ and thus $\mathbb{P}\left[\mathbb{A}^{\Pi} \to 1\right] = 1$.

On the other hand, for a random permutation on $m > 1$ blocks, one expects collisions on the first output block after enough queries. In particular, the probability all q ciphertexts have distinct first blocks is simply the product of the probabilities that the first collision does not occur on the i^{th} query for all $i \leq q$. Thus, in the ideal case,

$$\mathbb{P}\left[\mathbb{A}^{\pi} \to 1\right] = \prod_{i=1}^{q}\left(1 - \frac{(i-1)(N^{m-1}-1)}{N^m-(i-1)}\right) \leq \prod_{i=0}^{q-1}\left(1 - \frac{i}{2N}\right) \leq 1 - \frac{q(q-1)}{8\cdot N}.$$

It is for the final inequality that we require the bound on q.

Taking the difference between these terms, we have $\mathbf{Adv}_{\Pi}^{\mathsf{prp}\bullet}(q) \geq \frac{q(q-1)}{8\cdot N}$, as claimed. □

4.1 Right Shifting Towards a prp

Two obvious candidates for the linear layer are the right and left rotations by one block. For messages of at least $i+1$ blocks, Π_i^{left} is not a $\mathsf{prp}\bullet$, since the first output block cannot possibly depend on the final input block. Moreover, this means Π_i^{right} cannot be an $\pm\mathsf{prp}\bullet$, since its inverse is the Π_i^{left} scheme instantiated around $\mathcal{D}$ unless the message length is smaller than the number of rounds.

Combining this limitation with Lemma 8 (two layer constructions cannot be indistinguishable from a $\mathsf{prp}\bullet$ with beyond birthday bound security), Π_2^{right} is at best a prp up to the birthday bound. This is in fact the case, a statement formalised as follows:

Theorem 2. *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 random permutation up to the birthday bound. Explicitly, $\mathbf{Adv}_{\Pi_2^L}^{\mathsf{prp}}(q) \leq \frac{q(q-1)}{N}$.*

Proof. We will use a very simple identical-until-bad style argument, moving from the real world to the ideal world. Assume that the final block of output from the first online permutation call is always unique (at the cost of a birthday bound collision term, by

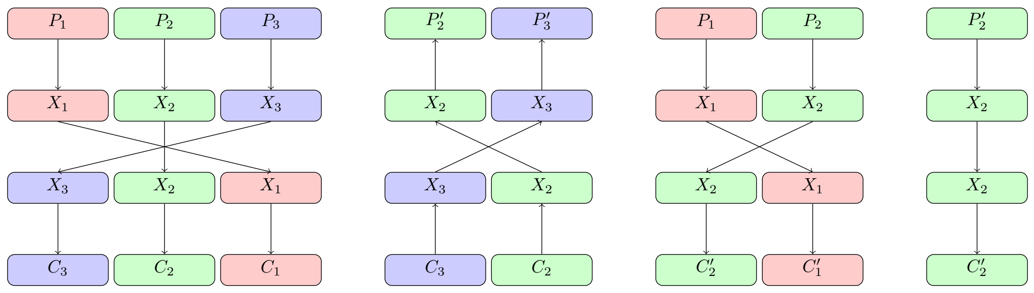


Figure 9: Attack against Π_2^{rev}, as described in Lemma 9. The adversary $\mathbb{A}$ makes the four queries shown in this diagram, and returns 1 if the two values labelled C_2' agree. Shown in the diagram are the internal variables corresponding to this in the real case, demonstrating that the collision will always occur, an event that almost never occurs in the ideal case.

Lemma 4). Then, by assumption on the linear layer, the first input block to the second online permutation call is unique. Thus we can apply Lemma 6 in the case $p = 0$ to switch out the second online permutation call for routine whose output is uniformly sampled and non-repeating (moving from the $\mathcal{O}_3$ case to the $\mathcal{O}_4$ case). After this switch, the construction is identical-until-bad to an ideal permutation.

Therefore, it remains to measure the two bad events from the application of Lemma 6. The first bad event from the lemma is that the first block of input repeats, which by assumption never occurs. The second bad event is that the first block of output repeats, which is again a birthday bound collision event.

Thus the construction is identical-until-bad to a random permutation until either of these two birthday bound collision events occur, and their sum gives the claimed result. □

4.2 Two Layers Versus ±prp Security

We move on to consider $\pm\mathsf{prp}$ security. Such a scheme will still be susceptible to the birthday attack given in Lemma 8, but what are the minimum properties required of the linear layer to meet this bound? To prevent a similar attack to the single layer construction, where certain message blocks 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 for any strings $X, Y \in \Sigma^m$, $L(X)[1]$ must depend on $X[m]$, and $L^{-1}(Y)[1]$ must depend on $Y[m]$.

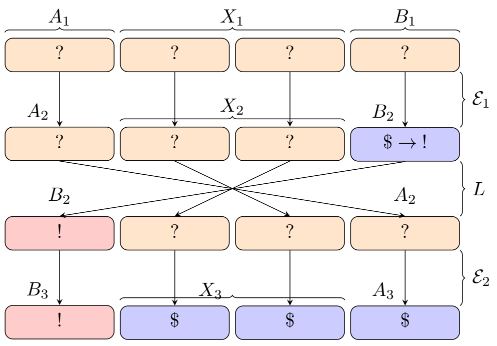
So, the most intuitive candidate is Π_2^{rev}: the two layer construction instantiated around a linear layer that reverses the order of the blocks. However this turns out to be insecure in the VIL setting, as shown by the following attack.

Lemma 9. *The* $\pm\mathsf{prp}$ *advantage against* Π_2^{rev} *is* $\mathbf{Adv}^{\pm\mathsf{prp}}_{\Pi_2^{\mathsf{rev}}}(q) \geq 1 - N^{-1}$ *for all* $q \geq 4$.

Proof. We will provide an explicit adversary $\mathbb{A}$, making 4 queries, each of at most 3 blocks. Note that the attack can be easily generalised to comply with most reasonable message-length requirements by replacing the blocks P_1, P_2 and P_3 with strings of blocks of an appropriate length.

Let Π and Π^{-1} be oracles corresponding to the construction, and let $\mathcal{E}$ be the internal online permutation. We will use "Enc" to refer to the adversaries forward oracle, and "Dec" for their inverse oracle.

Adversary $\mathbb{A}$ makes four queries of varying lengths to the construction, as visualised in Figure 9. First, $\mathbb{A}$ picks an arbitrary three-block string $P_1||P_2||P_3$, and queries $C_1||C_2||C_3 \leftarrow$



Key:

? Blocks whose distribution is unknown/irrelevant.

! Blocks that are unique across all queries.

\$ Blocks that are uniformly sampled.

$\$ \to !$ 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 equivalent place.

Figure 10: Intuition behind the AIL-$\pm\mathsf{prp}$ security of Π_2^{rev}, as shown by the VIL security of an Online-but-last scheme (Theorem 3).

$\mathrm{Enc}(P_1||P_2||P_3)$. Next, $\mathbb{A}$ queries $P_2'||P_3' \leftarrow \mathrm{Dec}(C_3||C_2)$ and then $C_2'||C_1' \leftarrow \mathrm{Enc}(P_1||P_2)$. Finally, $\mathbb{A}$ queries $\alpha \leftarrow \mathrm{Enc}(P_2')$ and returns 1 if and only if $\alpha = C_2'$.

As shown in the figure, we will always have $\mathbb{A}^{\Pi_2^{\mathsf{rev}}} \to 1$. Let $X_1||X_2||X_3 := \mathcal{E}(P_1||P_2||P_3)$. Then, by the online property, $C_3||C_2 = \mathcal{E}(X_3||X_2)$. So, $P_2'||P_3' = \mathcal{E}^{-1}(\mathsf{rev}(\mathcal{E}^{-1}(C_3||C_2)) = \mathcal{E}^{-1}(X_2||X_3)$, and thus $P_2' = \mathcal{E}^{-1}(X_2)$. Similarly, $C_2'||C_1' = \mathcal{E}(\mathsf{rev}(\mathcal{E}(P_1||P_2))) = \mathcal{E}(X_2||X_1)$, and so $C_2' = \mathcal{E}(X_2)$. Therefore $\Pi(P_2') = \mathcal{E}(\mathsf{rev}(\mathcal{E}(P_2'))) = \mathcal{E}(X_2) = C_2'$.

In the ideal case, when interacting with an ideal permutation π rather than the construction Π, the coincidence will almost certainly not occur, with $\mathbb{P}\left[\mathbb{A}^{\pi,\pi^{-1}} \to 1\right] = N^{-1}$. Taking the difference between these two terms, we see that $\mathbb{A}$ almost always distinguishes the scheme from an ideal permutation. □

A natural follow-up question is what minor modifications are required to make the scheme secure. To answer this, we show that the scheme is secure if built around an "online-but-last" permutation. This is a permutation acting on blocks where the final block acts like an independent tweakable permutation for each query length, but is online for all previous blocks.

Since an online permutation is identical to an online-but-last permutation when only queried with messages a fixed length, this proves the AIL security of Π_2^{rev}.

Theorem 3. *The two round construction* Π*, built from an online-but-last permutation around the linear layer that reverses the order of the blocks, is a secure VIL–*$\pm\mathsf{prp}$ *up to the birthday bound. Explicitly,* $\mathbf{Adv}_{\Pi}^{\pm\mathsf{prp}}(q) \leq q(q-1)/N$.

Corollary 1. Π_2^{rev} *is a secure* $\pm\mathsf{prp}\bullet$ *until the birthday bound,* $\mathbf{Adv}_{\Pi_2^{\mathsf{rev}}}^{\pm\mathsf{prp}\bullet}(q) \leq q(q-1)/N$.

Proof of Theorem 3. We will use an identical-until-bad proof mechanism, following the intuition shown in Figure 10 and using the variable naming given there. So, upon query M, we set $B||X_1||A \leftarrow M$, where $|X_1| = |M| - 2$ and $|A| = |B| = 1$. Note that this excludes single-block queries, but an adversary can learn nothing from these since single-block queries to an online-but-last cipher are independent from queries of any other length.

Consider first the encryption routine. As discussed in Section 3.2, an online-but-last cipher is an online cipher whose final block call has been replaced with an independent TBC, and we swap out this independent TBC for a TPRF. Since the adversary never repeats a query, the inputs to the TBC never repeat, and so its output (B_2) is uniformly sampled, independent of the input. This means that the second call to the online-but-last cipher always begins with a uniformly sampled block, and so (by the online-but-last version of Lemma 6 with $x = 0$), its output ($B_3||X_3$) is indistinguishable from the output of a random permutation until there is a collision on B_2.

The equivalent switches in decryption are undetectable until the adversary either can distinguish the tprp–tprf switch or collide on the randomly sampled A_2. Making both sets of switches at the same time we reach a scheme that is indistinguishable from a $\pm\mathsf{prp}$, so it remains to bound the probability of detecting these switch events. The tprp-tprf switches are (collectively) bounded by Lemma 2, since the bound is maximised by an adversary who makes only queries of a single length. The probability of colliding the random variables with those from any previous query (encryption or decryption) is bounded by Lemma 1. Summing these two terms, the overall advantage is bounded by $q(q-1)/N$. □

5 Three Round Constructions: Moving Beyond the Birthday Bound

We have shown that birthday bound security is 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: there exist schemes making three calls to the online cipher that achieve markedly better security. In particular, Π_3^{right} achieves prp security beyond the birthday bound, and Π_3^{rev} achieves $\pm\mathsf{prp}\bullet$ security up until almost the blocksize.

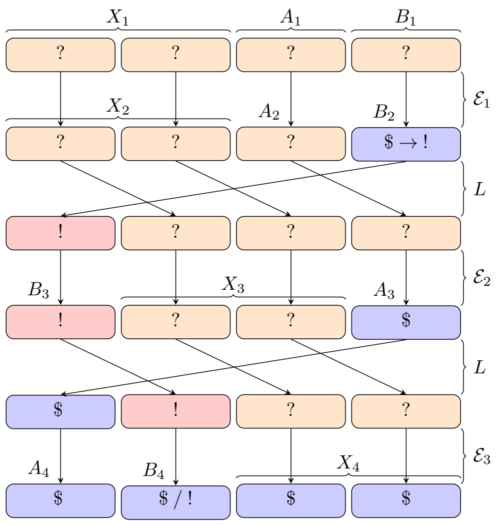
Similarity to PIV The Π_3^{rev} construction can be rephrased in a way that is similar to the PIV wide-block tweakable blockcipher design of Shrimpton and Terashima [?], and the proofs follow a similar overall design. In each case, the first round encrypts a few blocks of input tweaked by the whole message: either through an explicit tweakable blockcipher (as in PIV) or implicitly via the final blocks of the online cipher (as with Π_3^{rev}). These encrypted blocks are then used as a unique tweak to encrypt the remaining message blocks, before the encrypted blocks are then re-encrypted to ensure the system cannot be broken with inverse queries. Their result is not directly applicable because the final blocks of an online cipher are *not* a secure $\pm\mathsf{prp}$: they may leak plaintext repetition patterns. As such, if messages are too short, a PIV-style construction would also leak these patterns.

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 $\pm\mathsf{prp}$ security because its inverse is trivially distinguishable. Moreover, there exists an attack against the $\mathsf{prp}\bullet$ security of all the Π_i^{right} schemes (described later as Lemma 11), and substituting in the appropriate parameters shows that if $N \geq 4$ and messages are at least 4 blocks long, $\mathbf{Adv}_{\Pi}^{\mathsf{prp}\bullet}(q) \geq \frac{q(q-1)}{8N^2} \approx (\frac{q}{N})^2$ for $q \leq N$. Thus the best we can reasonably expect is prp security up to the blocksize, something we now demonstrate is attainable. This constitutes a significant improvement over Theorem 2's birthday bound. Again, we present the logic behind our proof in a diagram (Figure 11).

The key observation behind the proof is that B_2 does not repeat "too frequently", allowing us to perform a (tweakable) random permutation–random function switch on the final block of the second layer and simplify the construction. Then we apply Lemma 6 to demonstrate the scheme similar to a random permutation, and measure the appropriate events.

Theorem 4. *The $\Pi = \Pi_3^{\mathsf{right}}$ construction is a variable input length prp, where for all adversaries making queries totalling at most σ blocks, $\mathbf{Adv}_{\Pi}^{\mathsf{prp}}(\sigma) \leq 1.5\frac{\sigma(\sigma-1)}{N^2}$.*



Key:

?	Blocks whose distribution is unknown/irrelevant.
!	Blocks that do not repeat too frequently.
\$	Blocks that are uniformly sampled.
$\$ \to !$	Block that does not repeat frequently.

Intuition: To achieve beyond birthday bound security, we can no longer fail if single block collisions occur. Instead, we observe that, since B_2 does not repeat "too frequently", we can perform a PRP–PRF switch on the call defining A_3. Thus the block A_3 is sampled uniformly, and independently of B_2. Then, since the first two blocks input to the final call almost never repeat (as a pair), the remaining blocks are uniform.

Figure 11: Intuition behind the prp security of Π_3^{right} (Theorem 4)

Proof. We split the internal variables into three sections (A_i, X_i and B_i) such that for an m block message M, $|A_i| = |B_i| = 1$ and $|X_i| = m-2$, for all $i \in \{1,\ldots,4\}$. We set $X_1||A_1||B_1 \leftarrow M$, and "track" the ordering of these sections through the linear layers. Each time the permutation is called, the appropriate blocks of its output will be labelled by the same letter (and incremented index), meaning the ciphertext $C = A_4||B_4||X_4$. We label the three ideal online permutations $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$, and they will be lazily sampled, with internal tables $\mathring{\pi}_1, \mathring{\pi}_2, \mathring{\pi}_3$. This labelling, and the logic described below, are represented in Figure 11, which sketches how we convert from the Π_3^{right} scheme to a random function, a standard switch away from a random permutation.

Formally, we consider a series of claims and use an identical-until-bad style argument, stepping between collections of oracles, before applying the funnelling techinque to bound the adversarial advantage. First we justify that single block queries do not assist the adversary (Claim 1). Next, we define a routine (Figure 12) and demonstrate it identical-until-bad to one of the random permutations in the construction (Claim 2). Then, we show that substituting this into Π_3^{right} and applying Lemma 6 leads to a construction perfectly secure until bad is set (Claim 3). Finally, we measure the probability of bad (Claim 4) to complete the proof.

We will assume before querying any long messages, the adversary first queries on all prefix strings. This assumption allows us to make statements about the freshness of inputs/outputs of the final block of a query, but it leads to a loss of tightness when considering adversaries who do not make queries of many (if any) different lengths. We assumed the adversary first queried all prefixes of their chosen query, so to switch back our overall bound must be given in term of the total number of blocks σ, rather than the number of queries q.

Claim 1. *Single block queries do not assist the adversary. For any adversary $\mathbb{A}$ there exists an adversary $\mathbb{B}$ making no queries of length 1 such that* $\mathbf{Adv}_{\Pi}^{\mathsf{prp}}(\mathbb{A}) = \mathbf{Adv}_{\Pi}^{\mathsf{prp}}(\mathbb{B})$.

Proof. First, $\mathbb{B}$ samples a random single-block permutation P. Then, $\mathbb{B}$ runs $\mathbb{A}$, answering single-block queries with P and forwarding all of $\mathbb{A}$s longer queries to his own oracle. When $\mathbb{A}$ terminates, $\mathbb{B}$ forwards the response as his own.

In the ideal case, $\mathbb{B}$ perfectly simulates the ideal world for $\mathbb{A}$, and so $\mathbb{P}[\mathbb{A}^{\pi} \to 1] = \mathbb{P}[\mathbb{B}^{\pi} \to 1]$. In the real case, we observe that the restriction of an ideal online permutation

```
**function** $\widetilde{\mathcal{E}}_2(B_2||X_2||A_2)$
    $B_3 \leftarrow \mathcal{E}_2(B_2)$
    $X_3 \leftarrow \mathcal{E}_2^{B_2}(X_2)$
    $A_3 \leftarrow\!\!\$\ \Sigma$
    **if** $(B_2, A_2) \in \mathcal{L}_A$ **then**
        $\mathsf{bad}_A \leftarrow \mathtt{true}$
    **if** $(B_2, A_3) \in \mathcal{L}_B$ **then**
        $\mathsf{bad}_B \leftarrow \mathtt{true}$
    $\mathcal{L}_A \leftarrow\!\!\cup\ (B_2, A_2)$
    $\mathcal{L}_B \leftarrow\!\!\cup\ (B_2, A_3)$
    $\mathring{\pi}_2 \leftarrow\!\!\cup\ (B_2||X_2, A_2, A_3)$
    **return** $B_3||X_3||A_3$
```

Figure 12: The function $\widetilde{\mathcal{E}}_2$ is identical-until-bad to $\mathcal{E}_2$. As well as updating the lists $\mathcal{L}_A, \mathcal{L}_B$ to track bad events, $\widetilde{\mathcal{E}}_2$ must record the sampled output A_3 in $\mathring{\pi}_2$ in case a later (longer) query is made for which this is a prefix.

to a single block is an ideal permutation. Since all the later claims are independent of the distribution of the first block of $\mathcal{E}_1$, we may switch this for a random permutation without loss. Thus $\mathbb{B}$ also perfectly simulates the real world to $\mathbb{A}$, and $\mathbb{P}\left[\mathbb{A}^{\Pi} \to 1\right] = \mathbb{P}\left[\mathbb{B}^{\Pi} \to 1\right]$. Taking the difference completes the claim. ∎

Henceforth, we assume that the adversary never makes single block queries, and thus label the internal variables as in Figure 11, with the possibility that $|X_i| = 0$.

Claim 2. *$\widetilde{\mathcal{E}}_2$ (Figure 12) is identical to $\mathcal{E}_2$ until* bad.

Proof. $\widetilde{\mathcal{E}}_2$ expands the online permutation call slightly, separating the final block and then applying a random permutation to random function switch. Thus it replaces the line $A_3 \leftarrow \mathcal{E}_2^{B_2||X_2}(A_2)$ with $A_3 \leftarrow\!\!\$\ \Sigma$ and updates the table $\mathring{\pi}_2$ accordingly. Until (B_2, A_2) repeats (setting bad_A), either the input or tweak is fresh, and thus the output is not pre-determined.[2] The check for bad_B confirms that the output has not been chosen to contradict the permutation property. Thus until bad is set, this is identical to the random online permutation $\mathcal{E}_2$. ∎

Claim 3. *Consider the oracles given in Figure 13. Until* bad *is set,* Π *behaves identically to both the random function defined by* Π'' *and a random permutation.*

Proof. To move from Π to Π' one substitutes $\widetilde{\mathcal{E}}_2$ in for $\mathcal{E}_2$, since all the bad events are already provided by the CheckForBad procedure. Thus by Claim 2, until bad is set these two routines are identical. Now, consider the later part of Π': the lines $A_3 \leftarrow\!\!\$\ \Sigma$ and its successor, which applies $\mathcal{E}_3$. These are the form required to apply Lemma 6 with $p = 1$, where the bad events of that lemma are equivalent to bad_C and bad_{out} of the CheckForBad procedure. So, Π' is identical-until-bad to Π'', and thus Π is identical-until-bad to Π''.

Similar to the second part of Lemma 6, Π'' is a random function and thus a random permutation until the output repeats, which would be captured by the bad_{out} event, since a repeated output of $A_4||B_4||X_4$ necessarily implies repetition of (A_4, B_4). ∎

[2] Indeed, this is true until $(B_2||X_2, A_2)$ repeats, but we chose to record the smaller (more probable) event as it lines up more neatly with other bad events we will consider.

```
function $\Pi(M)$
    $X_1||A_1||B_1 \leftarrow M$
    $X_2||A_2||B_2 \leftarrow \mathcal{E}_1(X_1||A_1||B_1)$
    $B_3||X_3||A_3 \leftarrow \mathcal{E}_2(B_2||X_2||A_2)$
    $A_4||B_4||X_4 \leftarrow \mathcal{E}_3(A_3||B_3||X_3)$
    CheckForBad(*)
    return $A_4||B_2||X_4$

procedure CheckForBad(*)
    if $(B_2, A_2) \in \mathcal{L}_A$ then
        $\mathsf{bad}_A \leftarrow$ true
    if $(B_2, A_3) \in \mathcal{L}_B$ then
        $\mathsf{bad}_B \leftarrow$ true
    if $(A_3, B_3) \in \mathcal{L}_C$ then
        $\mathsf{bad}_C \leftarrow$ true
    if $(A_4, B_4) \in \mathcal{L}_{out}$ then
        $\mathsf{bad}_{out} \leftarrow$ true
    $\mathcal{L}_A \overset{\cup}{\leftarrow} (B_2, A_2)$
    $\mathcal{L}_B \overset{\cup}{\leftarrow} (B_2, A_3)$
    $\mathcal{L}_C \overset{\cup}{\leftarrow} (A_3, B_3)$
    $\mathcal{L}_{out} \overset{\cup}{\leftarrow} (A_4, B_4)$
```

```
function $\Pi'(M)$
    $X_1||A_1||B_1 \leftarrow M$
    $X_2||A_2||B_2 \leftarrow \mathcal{E}_1(X_1||A_1||B_1)$
    $B_3 \leftarrow \mathcal{E}_2(B_2)$
    $X_3 \leftarrow \mathcal{E}_2^{B_2}(X_2)$
    $A_3 \overset{\$}{\leftarrow} \Sigma$
    $A_4||B_4||X_4 \leftarrow \mathcal{E}_3(A_3||B_3||X_3)$
    $\mathring{\pi}_2 \overset{\cup}{\leftarrow} (B_2||X_2, A_2, A_3)$
    CheckForBad(*)
    return $A_4||B_2||X_4$

function $\Pi''(M)$
    $X_1||A_1||B_1 \leftarrow M$
    $X_2||A_2||B_2 \leftarrow \mathcal{E}_1(X_1||A_1||B_1)$
    $B_3 \leftarrow \mathcal{E}_2(B_2)$
    $X_3 \leftarrow \mathcal{E}_2^{B_2}(X_2)$
    $A_4 \overset{\$}{\leftarrow} \Sigma$
    $A_3 \leftarrow \mathcal{D}_3(A_4)$
    $B_4 \overset{\$}{\leftarrow} \Sigma$
    $X_4 \overset{\$}{\leftarrow} \Sigma^{m-2}$
    $\mathring{\pi}_2 \overset{\cup}{\leftarrow} (B_2||X_2, A_2, A_3)$
    CheckForBad(*)
    return $A_4||B_4||X_4$
```

Figure 13: Π is the code for the original scheme, with the online permutation calls separated and the blocks reordered according to right. Moving from Π to Π' corresponds to expanding out the second permutation call and substituting $\widetilde{\mathcal{E}}_2$ in for $\mathcal{E}_2$. The switch from Π' to Π'' is application of Lemma 6 to the line $A_3 \overset{\$}{\leftarrow} \Sigma$ and the call to $\mathcal{E}_3$. Overall, Π is identical to both Π' and Π'' until procedure CheckForBad triggers bad.

Claim 4. *The probability an adversary interacting with* Π'' *sets* bad *is* $\mathbb{P}[\mathsf{bad}] \leq 1.5q(q-1)/N^2$.

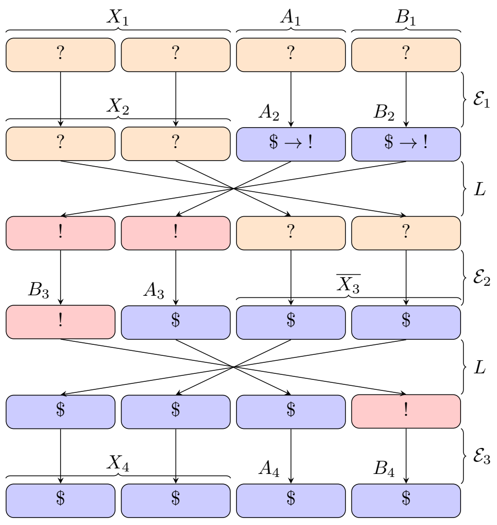
Proof. The event bad_A is that of colliding on the last two blocks output by an online permutation. By the same argument as Lemma 4, the probability of this event is bounded by $\frac{1}{2}q(q-1)/N^2$. The probability of colliding on B_2 on any particular query is at most N^{-1}, and also for A_3 since they are independent. Thus the probability of bad_B is at most $\frac{1}{2}q(q-1)/N^2$. Since B_3 is the image of B_2 under a permutation, collisions on B_3 imply collisions of B_2. So, bad_C cannot occur without bad_B first occurring. Finally, bad_{out} occurs if the pair (A_4, B_4) repeats, which is simply a collision of independently sampled values, again at the cost of $\frac{1}{2}q(q-1)/N^2$. Collecting these values,

$$\mathbb{P}[\mathsf{bad}] \leq \mathbb{P}[\mathsf{bad}_A] + \mathbb{P}[\mathsf{bad}_B] + \mathbb{P}[\mathsf{bad}_{out}] \leq 3 \cdot \frac{q(q-1)}{2N^2}. \qquad \blacksquare$$

All together then, the difference between Π and a random permutation is at most that of setting bad, which is bounded as stated. □

5.2 Three Layer Reverse: ±prp Beyond the Birthday Bound

So, by Theorem 4, there exist three layer constructions with security beyond the birthday bound, and in this section we investigate whether the Π_3^{rev} construction is a ±prp beyond



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.
$\$ \rightarrow !$	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, perform a sequence of switches on the scheme, demonstrating it is an ±PRP until certain bad events occur.

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 $|X_i| = |M| - 2$. The most important collision event is on (A_2, B_2). Finally, we prove it is hard for an adversary to trigger one of these events.

Figure 14: Intuition behind the $\pm\mathsf{prp}$ security of Π_3^{rev} (Theorem 5)

the birthday bound. Similar to the previous result, we will prove the construction secure until a birthday-style collision on pairs of blocks, and so achieve security up to almost the blocksize.

We will provide a proof for the AIL case, because, perhaps surprisingly, the VIL case appears significantly more nuanced. Roughly speaking, this is because an adaptive adversary can perform an attack similar to that against the two-round construction (Lemma 9), which force any modelling or simulation attempts within our proofs to define internal variables early, then make later queries in which the proof depends on freshness of these values. Unlike the two-round case however, we have not been able to extend this into an attack against the actual scheme, leaving the question of VIL security open at present.

As such, we will provide an AIL proof that we hope can eventually be extended to the VIL case: the majority of the internal claims are true in both settings, hopefully leaving a smaller problem for future work. Given this limitation, we favour clarity over tightness, meaning our bound can trivially be improved by a small factor.

Theorem 5. *Set* $\Pi_3 = \Pi_3^{\mathsf{rev}}$ *to be the construction built from three independent online permutations, around two calls of* rev*. Then the adversarial advantage in distinguishing* Π_3 *from a AIL random permutation is bounded. For any adversary making* $q \leq N/8$ *queries,*

$$\mathbf{Adv}_{\Pi_3}^{\pm\mathsf{prp}\bullet}(q) \leq \frac{q}{N} \log_2 N.$$

Proof. Applying Lemma 5 to Π_3, we will derive a construction (Figure 15) that is identical-until-bad to both Π_3^{rev} and a random permutation (Claim 1). Then, we bound the probability of bad, (Claims 2 to 5) to complete the result.

The variable naming scheme and parsing will follow those given in Figure 14. $\mathcal{E}_1$ is an ideal random permutations with inverse $\mathcal{D}_1$ and will be lazily sampled with internal table $\mathring{\pi}_1$, and similarly for $\mathcal{E}_2$ and $\mathcal{E}_3$. The case where $|M| = 1$ is trivially secure, since the restriction of an ideal online permutation to a single block *is* a random permutation, so assume all queries are of at least two blocks. Then, parsing of M is done such that for a message of length $m \geq 2$, $|X| = m - 2$ and $|A| = |B| = 1$.

Claim 1. *The pair* (Enc, Dec) *given in Figure 15 are identical-until-bad to the appropriate routines of* Π_3^{rev}*. Moreover, they are identical-until-bad to a random permutation.*

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

```
function $\Pi^{-1}(M)$
    $X_4||A_4||B_4 \leftarrow C$
    $X_3||A_3||B_3 \leftarrow \mathcal{D}_3^\epsilon(X_4||A_4||B_4)$
    $B_2||A_2||\overline{X}_2 \leftarrow \mathcal{D}_2^\epsilon(B_3||A_3||\overline{X}_3)$
    $X_1||A_1||B_1 \leftarrow \mathcal{D}_1^\epsilon(X_2||A_2||B_2)$
    return $X_1||A_1||B_1$
```

```
function $\textsc{Enc}(M)$
    $X_1||A_1||B_1 \leftarrow M$
    - - - - - - - - - - - - - - -
    $X_2||A_2 \leftarrow \mathcal{E}_1^\epsilon(X_1||A_1)$
    $B_2 \leftarrow \mathcal{E}_1^{X_1||A_1}(B_1)$
    - - - - - - - - - - - - - - -
    $B_3 \leftarrow \mathcal{E}_2^\epsilon(B_2)$
    $A_3 \leftarrow_\$ \Sigma$
    $\overline{X}_3 \leftarrow_\$ \Sigma^{|M|-2}$
    if $(B_2, A_2, \star) \in \mathring{\pi}_2$ then
        $\mathsf{bad}_B \leftarrow \mathtt{true}$
    if $(B_2, \star, A_3) \in \mathring{\pi}_2$ then
        $\mathsf{bad}_{B'} \leftarrow \mathtt{true}$
    $\mathring{\pi}_2 \leftarrow_\cup (B_2, A_2||\overline{X}_2, A_3||\overline{X}_3)$
    - - - - - - - - - - - - - - -
    $X_4||A_4 \leftarrow \mathcal{E}_3^\epsilon(X_3||A_3)$
    $B_4 \leftarrow_\$ \Sigma$
    if $(X_3||A_3, B_3, \star) \in \mathring{\pi}_3$ then
        $\mathsf{bad}_C \leftarrow \mathtt{true}$
    if $(X_3||A_3, \star, B_4) \in \mathring{\pi}_3$ then
        $\mathsf{bad}_{C'} \leftarrow \mathtt{true}$
    $\mathring{\pi}_3 \leftarrow_\cup (X_3||A_3, B_3, B_4)$
    - - - - - - - - - - - - - - -
    return $X_4||A_4||B_4$
```

```
function $\textsc{Dec}(M)$
    $X_4||A_4||B_4 \leftarrow M$
    - - - - - - - - - - - - - - -
    $X_3||A_3 \leftarrow \mathcal{D}_3^\epsilon(X_4||A_4)$
    $B_3 \leftarrow \mathcal{D}_3^{X_3||A_3}(B_4)$
    - - - - - - - - - - - - - - -
    $B_2 \leftarrow \mathcal{D}_2^\epsilon(B_3)$
    $A_2 \leftarrow_\$ \Sigma$
    $\overline{X}_2 \leftarrow_\$ \Sigma^{|M|-2}$
    if $(B_2, A_2, \star) \in \mathring{\pi}_2$ then
        $\mathsf{bad}_B \leftarrow \mathtt{true}$
    if $(B_2, \star, A_3) \in \mathring{\pi}_2$ then
        $\mathsf{bad}_{B'} \leftarrow \mathtt{true}$
    $\mathring{\pi}_2 \leftarrow_\cup (B_2, A_2||\overline{X}_2, A_3||\overline{X}_3)$
    - - - - - - - - - - - - - - -
    $X_1||A_1 \leftarrow \mathcal{D}_1^\epsilon(X_2||A_2)$
    $B_1 \leftarrow_\$ \Sigma$
    if $(X_1||A_1, \star, B_2) \in \mathring{\pi}_1$ then
        $\mathsf{bad}_A \leftarrow \mathtt{true}$
    if $(X_1||A_1, B_1, \star) \in \mathring{\pi}_1$ then
        $\mathsf{bad}_{A'} \leftarrow \mathtt{true}$
    $\mathring{\pi}_1 \leftarrow_\cup (X_1||A_1, B_1, B_2)$
    - - - - - - - - - - - - - - -
    return $X_1||A_1||B_1$
```

Figure 15: The scheme Π_3^{rev} as oracles (E,D), followed by a construction (Enc, Dec) that is identical-until-bad. To generate (Enc, Dec) out of (E,D) one applies Lemma 5 to swap out $\mathcal{E}_2$, $\mathcal{D}_2$, $\mathcal{E}_3$ and $\mathcal{D}_1$. The dashed lines are for guidance only, and separate the elements of the routine coming from each of the three ideal online permutations. The parsing of M is done such that for a message of length m, $|X| = m - 2 \geq 0$ and $|A| = |B| = 1$. Since (Enc, Dec) are identical to random permutation until bad, bounding bad also bounds the distinguishing advantage against Π_3^{rev}.

Proof. We will prove that the pair (Enc, Dec) in Figure 15 is identical to Π_3^{rev}, shown as (Π, Π^{-1}) in the same figure, until an adversary can trigger one of the bad events given in (Enc, Dec). Converting from (E,D) to the (Enc, Dec) is achieved through repeated application of Lemma 5. To swap out $\mathcal{E}_2$ we use the case $p = 1$, and thus $(P_1, Q_1, R_1) = (B_3, A_3, \overline{X}_3)$, We also switch out $\mathcal{E}_3$, using the case $p = |M| - 1$, which corresponds to setting $(P_1, Q_1, R_1) = (X_3||A_3, B_3, \epsilon)$, then removing the superfluous code. We also apply the corresponding switches to $\mathcal{D}_2$ and $\mathcal{D}_1$. Since each of these (four) switches was identical-until-bad, applying all of them is as well.

Following the same logic used in Lemma 6, (Enc, Dec) are a random function with inverse, since their outputs are sampled uniformly and independent of input. In the Enc case, B_4 is sampled directly, while $(X_4||A_4)$ is the image of $(X_3||A_3)$, which is itself uniformly sampled. Similarly, in Dec B_1 is sampled directly, with $X_1||A_1$ the the image of a uniformly sampled variable under permutation. Moreover, output collisions do not occur without triggering bad, specifically $\mathsf{bad}_{C'}$ for Enc and $\mathsf{bad}_{A'}$ for Dec. Thus it is also a random permutation until bad. ■

Claim 2. *Assuming* $\neg\mathsf{bad}_B \wedge \neg\mathsf{bad}_{B'}$*, the probability of setting* $\mathsf{bad}_A \vee \mathsf{bad}_{A'} \vee \mathsf{bad}_C \vee \mathsf{bad}_{C'}$ *is small, with* $\mathbb{P}\left[\mathsf{bad}_A \vee \mathsf{bad}_{A'} \vee \mathsf{bad}_C \vee \mathsf{bad}_{C'} | \neg\mathsf{bad}_B \wedge \neg\mathsf{bad}_{B'}\right] \leq \frac{1}{2}q(q-1)/N^2$.

Proof. We observe a symmetry between the two pairs of bad events: only encryption can trigger bad_C or $\mathsf{bad}_{C'}$, and only decryption can trigger bad_A or $\mathsf{bad}_{A'}$. Let us assume the i^{th} query is made to Enc, and that no bad events have yet triggered, to bound the probability of setting $\mathsf{bad}_C \vee \mathsf{bad}_{C'}$. We consider two cases, depending on $|M|$.

Firstly, consider the case $|M| > 2$, and thus $|X| \geq 1$. Then, the probability of setting $\mathsf{bad}_C \vee \mathsf{bad}_{C'}$ is upper bounded by that of colliding on the prefix $(X_3||A_3)$, since without a prefix collision neither event can occur. Thus $\mathsf{bad}_C \vee \mathsf{bad}_{C'}$ requires that $X_3||A_3$ takes a value that it has taken previously, the probability of which is upper bounded by $(i-1)/N^2$ since X_3 and A_3 are uniformly sampled.

Alternatively, suppose $|M| = 2$ and thus $|X| = 0$. Then, the probability of setting $\mathsf{bad}_{C'}$ is $(i-1)/N^2$, since it requires the pair of uniformly sampled blocks (A_3, B_4) to take a value already present in a list of length $i-1$. The event bad_C can only occur if on a previous query $A_3||B_3$ occurred as internal variables. Since B_3 is the image of B_2 under a permutation, this implies that the pair (B_2, A_3) also repeated, meaning $\mathsf{bad}_{B'}$ had already been set. Given the assumption on $\mathsf{bad}_{B'}$, this cannot occur, and thus such queries cannot trigger bad_C.

So, in either case, the probability of an encryption query setting $\mathsf{bad}_C \vee \mathsf{bad}_{C'}$ is at most $(i-1)/N^2$.

Applying the same logic to decryption queries, we also bound the probability the i^{th} query is a decryption query and triggers $\mathsf{bad}_A \vee \mathsf{bad}_{A'}$ by $(i-1)/N^2$. Since every query is either encryption or decryption, but not both, the probability of any single query triggering any one of the four events must also be this value, and applying the union bound completes the claim. ■

Claim 3. *For any* $\alpha \in \mathbb{N}$*, the probability of setting* $\mathsf{bad}_B \vee \mathsf{bad}_{B'}$ *on the* i^{th} *query if it has not previously been set is bounded, with*

$$\mathbb{P}\left[\mathsf{bad}_B \vee \mathsf{bad}_{B'} \textit{ first set on query } i\right] \leq \frac{\alpha}{N} + \frac{\alpha}{N-i+1} + \frac{1}{(\alpha+1)!}\left(\frac{i-1}{N}\right)^{\alpha+1}.$$

Proof. For all j, let A_1^j be the value A_1 took on the j^{th} query, and similarly for all other variables. Let $\mathcal{Q}_i := \left\{(A_2^j, A_3^j) \mid j < i \wedge B_2^j = B_2^i\right\}$. Finally, let $\alpha \in \mathbb{N}$ be a parameter, which will bound how large we expect the largest of the $\mathcal{Q}_i$s to be.

With this notation in hand, we can rewrite the event probability that an adversary interacting with (Enc, Dec) can trigger bad_B or $\mathsf{bad}_{B'}$ on their i^{th} query, and upper bound it by

$$\begin{aligned}
&\mathbb{P}\left[\mathsf{bad}_B \vee \mathsf{bad}_{B'} \text{ first set on query } i\right] \\
&\quad = \mathbb{P}\left[(A_2^i, \star) \in \mathcal{Q}_i \vee (\star, A_3^i) \in \mathcal{Q}_i\right] \\
&\quad \leq \mathbb{P}\left[(A_2^i, \star) \in \mathcal{Q}_i\right] + \mathbb{P}\left[(\star, A_3^i) \in \mathcal{Q}_i\right] \\
&\quad \leq \mathbb{P}\left[(A_2^i, \star) \in \mathcal{Q}_i | \#\mathcal{Q}_i \leq \alpha\right] + \mathbb{P}\left[(\star, A_3^i) \in \mathcal{Q}_i | \#\mathcal{Q}_i \leq \alpha\right] + \mathbb{P}\left[\#\mathcal{Q}_i > \alpha\right].
\end{aligned}$$

We now bound these terms for an encryption query, as decryption is equivalent.

Immediately, as A_3^i is sampled uniformly at random on every query we can bound the first term, with $\mathbb{P}\left[(\star, A_3^i) \in \mathcal{Q}_i | \#\mathcal{Q}_i \leq \alpha\right] \leq \alpha/N$.

Given the adversary does not repeat queries, for any j such that $(A_2^j, \star) \in \mathcal{Q}_i$ then $X_1^i || A_1^i \neq X_1^j || A_1^j$, because otherwise the online property would ensure $B_2 \neq B_2^j$ (and thus $A_2^j \neq \mathcal{Q}_i$). Now, if $X_1 = X_1^j$ this means that $A_1 \neq A_1^j$, and so by the online property $A_2 \neq A_2^j$. So, the only way $(A_2^i, \star) \in \mathcal{Q}_i$ is if there exists a previous query j with $B_2^i = B_2^j$ such that $X_1^i \neq X_1^j$ and $\mathcal{E}^{X_1^i}(A_1^i) = \mathcal{E}^{X_1^j}(A_1^j)$.

The adversary cannot detect a collision $A_2^i = A_2^j$ if there was not also a collision on either X_2 or B_2. By assumption this is the first time bad_B has occurred, and we know that $X_1^i \neq X_1^j$, so the adversary has no information that can further assist them in triggering this collision.[3] So, this is simply the probability that an element of a partial random permutation with at most $i-1$ terms defined is already specified in a separate list of length at most α, which is upper bounded by $\alpha/(N-i)$.

Finally, we bound $\mathbb{P}[\#\mathcal{Q}_i > \alpha]$. It is maximised if each query is made with a different prefix $X_1^i || A_1^i$, since for any two queries sharing a prefix the online property forces the final output blocks to differ. So, $\mathbb{P}[\#\mathcal{Q}_i > \alpha] \leq \left(\frac{i-1}{N}\right)^{\alpha+1}/(\alpha+1)!$ by a standard collision-counting argument.

Collecting these three terms together gives the claimed bound. ■

Claim 4. *Assume $q \leq N/8$ and let $n = \log_2 N$. Then the probability of setting $\mathsf{bad}_B \vee \mathsf{bad}_{B'}$ within q queries is bounded, with $\mathbb{P}[\mathsf{bad}_B \vee \mathsf{bad}_{B'}] \leq \frac{q}{N}\left(\frac{5n}{7} + \frac{1}{2}\right)$.*

Proof. By assumption, $q/N \leq 1/8$. So, by Claim 3, we can upper bound the probability of setting $\mathsf{bad}_B \vee \mathsf{bad}_{B'}$ by

$$\begin{aligned}\mathbb{P}[\mathsf{bad}_B \vee \mathsf{bad}_{B'} \text{ first set on query } i] &\leq \frac{\alpha}{N} + \frac{\alpha}{N-q} + \frac{1}{(\alpha+1)!}\left(\frac{q}{N}\right)^{\alpha+1} \\ &\leq \frac{\alpha}{N} + \frac{\alpha}{N}\cdot\frac{8}{7} + \frac{1}{(\alpha+1)!\cdot 8^{\alpha+1}}.\end{aligned}$$

So, applying the union bound and collecting like terms,

$$\mathbb{P}[\mathsf{bad}_B \vee \mathsf{bad}_{B'}] \leq \frac{15\alpha}{7}\cdot\frac{q}{N} + \frac{1}{8^{\alpha+1}(\alpha+1)!}q.$$

We may assume $n = \log_2 N \geq 3$, since otherwise $q \leq N/8$ would imply $q = 0$ and $\mathbb{P}[\mathsf{bad}] = 0$. So, fixing $\alpha = \lfloor n/3 \rfloor$, we have that $8^{\alpha+1} \geq 8^{n/3} \geq N$ and $(\alpha+1)! \geq \alpha+1 \geq 2$. Substituting these in yields $\mathbb{P}[\mathsf{bad}_B \vee \mathsf{bad}_{B'}] \leq \frac{q}{N}\left(\frac{5n}{7} + \frac{1}{2}\right)$, which is the bound claimed. ■

Claim 5. *Let $n = \log_2 N$. Then the probability of setting bad within $q < N/8$ queries is bounded, with $\mathbb{P}[\mathsf{bad}] \leq n\frac{q}{N}$.*

Proof. We will bound bad by

$$\mathbb{P}[\mathsf{bad}] \leq \mathbb{P}[\mathsf{bad}_B \vee \mathsf{bad}_{B'}] + \mathbb{P}[\mathsf{bad}_A \vee \mathsf{bad}_{A'} \vee \mathsf{bad}_C \vee \mathsf{bad}_{C'} | \neg\mathsf{bad}_B \wedge \neg\mathsf{bad}_B]$$

Substituting in the bounds from Claim 2 and Claim 4,

$$\mathbb{P}[\mathsf{bad}] \leq \frac{q}{N}\left(\frac{5n}{7} + \frac{1}{2}\right) + \frac{q(q-1)}{2N^2} \leq \frac{q}{N}\left(\frac{5n}{7} + \frac{1}{2} + \frac{1}{16}\right) \leq n\frac{q}{N},$$

where we have used that $(q-1)/N \leq 1/8$ for the second inequality, and for the third we have used that (since $q \leq N/8$) either $n \geq 3$ or $q = 0$. ■

Combining Claim 1 (Π^3_{rev} is identical-until-bad to an $\pm\mathsf{prp}$) with Claim 5 (bounding the probability of bad) completes the theorem. □

[3] This step that does not hold in the VIL case. If the adversary may make variable length queries, it is conceivable that they may make a length-extension style attack: first finding a pair $X_1 || A_1$ and $X_1' || A_1'$ that lead to a collision $\mathcal{E}^{X_1}(A_1) = \mathcal{E}^{X_1'}(A_1')$. With this in hand, varying values B_1, B_1' until the collision extends to $\mathcal{E}^{X_1}(A_1 || B_1) = \mathcal{E}^{X_1'}(A_1' || B_1')$ despite only ever having collisions between two elements.

6 Towards Security with Many Layers

For completeness, let us consider what can be achieved by the Π^{rev} scheme by using many layers. Since the Π_3^{rev} already provides beyond birthday bound security, 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, demonstrating that the larger schemes inherit the security of their smaller counterparts.

Lemma 10. *The Π_i^L construction is no more distinguishable from a random permutation than Π_{i-1}^L. For any adversary $\mathbb{A}$, there exists adversary $\mathbb{B}$ running within similar resources (up to a constant factor) such that*

$$\mathbf{Adv}_{\Pi_i^L}^{\pm\mathsf{prp}}(\mathbb{A}) = \mathbf{Adv}_{\Pi_{i-1}^L}^{\pm\mathsf{prp}}(\mathbb{B})$$

and similarly for prp, $\pm\mathsf{prp}\bullet$ *and* $\mathsf{prp}\bullet$.

Proof. We will proof just the prp case: the other three cases are similar.

Let $\mathbb{A}$ be a prp adversary. Let us construct an adversary $\mathbb{B}$ who uses $\mathbb{A}$ to distinguish Π_{i-1}^L if $\mathbb{A}$ can distinguish Π_i^L.

Let Enc be the encryption oracle $\mathbb{B}$ is provided with (which may be real or random). $\mathbb{B}$ constructs an ideal online permutation $\mathcal{E}_B$ via lazy sampling, and then simulates a Π_i^L encryption oracle with $\tilde{\Pi}_i^L := \mathcal{E}_B \circ L \circ \mathrm{Enc}$ (and similarly for the decryption oracle in the $\pm\mathsf{prp}$ case). To distinguish the scheme, $\mathbb{B}$ runs $\mathbb{A}$, answering all of $\mathbb{A}$s oracle queries with $\tilde{\Pi}_i^L$, then forwards $\mathbb{A}$s result as his own.

The composition of a random permutation with an online cipher (itself a permutation) is again a random permutation. So, $\mathbb{P}\left[\mathbb{B}^{\pi} \to 1\right] = \mathbb{P}\left[\mathbb{A}^{\pi'} \to 1\right]$. Conversely, if $\mathrm{Enc} = \Pi_{i-1}^L$, then $\tilde{\Pi}_i^L = \Pi_i^L$, and so $\mathbb{P}\left[\mathbb{B}^{\Pi_{i-1}^L} \to 1\right] = \mathbb{P}\left[\mathbb{A}^{\Pi_i^L} \to 1\right]$. Taking the difference between the these terms completes the bound. □

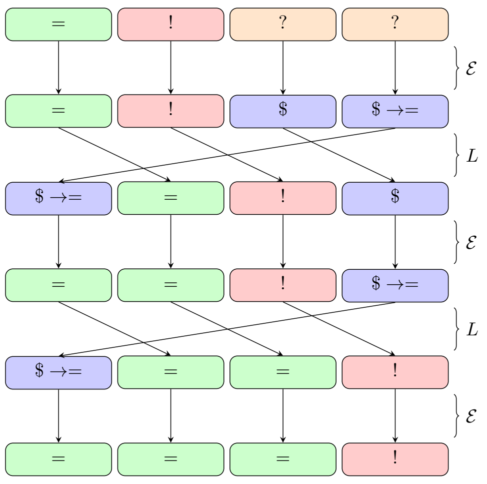
We observe that there exists an attack against the whole family of Π_i^{right} constructions. If an ideal online permutation is called with two messages that differ before the final block, the final ciphertext blocks are independently sampled. Now, if these independent random variables collide (which is likely to occur roughly every $\sqrt{N}$ queries) the right layer will simply add a common prefix to both messages. From this observation, we build a distinguisher against Π_i^{right}, following the logic shown in Figure 16.

Lemma 11. *Consider Π_i^{right} for some i, and assume we may make queries of at least $a+c+i$ blocks, with c such that $N^c \geq 4$. Then for any $q \leq \min(N^a, N^{(i-1)/2})$, $\mathbf{Adv}_{\Pi}^{\mathsf{prp}\bullet}(q) \geq \frac{q(q-1)}{8N^{i-1}}$.*

Proof. Consider the adversary $\mathbb{A}$ that requests encryptions of malicious messages of the form $M_t = \langle 0\rangle_{c+1}||\langle t\rangle_a||\langle 0\rangle_{i-1}$. He will vary t, allowing him up to N^a possible queries of this form (hence the first bound on q). After making his queries, he returns 1 if there were two queries for which the ciphertexts began with the same $i+c$ blocks. We claim $\mathbb{A}$ successfully distinguishes Π from an ideal prp.

We first bound $\mathbb{P}\left[\mathbb{A}^{\Pi} \to 1\right]$, by considering the internal variables when encrypting M_t. During the first round, the first c blocks are identical across all queries, and so encrypts to an identical value: $\mathcal{E}^{\epsilon}(\langle 0\rangle_{c+1})$. Since $\mathcal{E}$ is an online permutation, the online encryption of the first $a+c+1$ blocks must be unique among all queries, since the counter t was. Then, as the first $c+1$ blocks are 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, $\mathcal{E}^{\langle 0\rangle_{c+1}||\langle t\rangle_a}(\langle 0\rangle_{i-1})$ 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



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.

Figure 16: An attack against Π_3^{right}.

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 that at the r^{th} layer there are now $c + r$ repeated blocks rather than $c + 1$, 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 $(i - 1)$ internal rounds, there are two queries for which the final block inputs collide on the first $i + c$ blocks. As the cipher is online, this leads to an $i + c$ block collision in the output, triggering $\mathbb{A} \to 1$. So, $\mathbb{P}\left[\mathbb{A}^{\Pi} \to 1\right]$ is at least this probability. Since the variables are independently sampled, this is equivalent to colliding a string of $(i - 1)$ independently sampled blocks, $\mathbb{P}\left[\mathbb{A}^{\Pi} \to 1\right] \geq \frac{q(q-1)}{4N^{i-1}}$ as long as $q \leq \sqrt{2N^{i-1}}$ (Lemma 1).

Alternatively, consider $\mathbb{P}\left[\mathbb{A}^{\pi} \to 1\right]$. the probability of getting a collision on the first $i + c$ 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 $i + c$ random blocks. Thus $\mathbb{P}\left[\mathbb{A}^{\pi} \to 1\right] \leq \frac{q(q-1)}{2N^{i+c}}$.

Combining these results,

$$\mathbf{Adv}_{\Pi}^{\mathsf{prp}}(q) \geq \mathbb{P}\left[\mathbb{A}^{\Pi} \to 1\right] - \mathbb{P}\left[\mathbb{A}^{\pi} \to 1\right] \geq \frac{q(q-1)}{4N^{i-1}} - \frac{q(q-1)}{2N^{i+c}} = (1-\epsilon)\frac{q(q-1)}{4N^{i-1}},$$

where $\epsilon = 2/N^{c+1}$ is small for reasonable parameter sizes. Applying the hypothesis that $N^{c+1} \geq 4$, we have that $\epsilon \leq 1/2$ yielding the stated result. In the common $N = 2^{128}$ case, $\epsilon \leq 2^{-127}$. $\square$

7 Conclusion

We have shown how one can (relatively) efficiently turn an online cipher in a fully fledged cipher, using two types of mixing layer. To achieve birthday bound security, only two calls to the online cipher are required, with a linear layer that shifts blocks one step right achieving prp security, and blockwise reversal leading to a $\pm\mathsf{prp}\bullet$. For close to blocksize security three calls to the online cipher are both necessary and sufficient; a right-shift mixing layer yields a prp, while reversing achieves $\pm\mathsf{prp}\bullet$ security. 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.

7.1 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).

Our characterisation of an online cipher (due to Bellare et al. [?]) is at its most general. The more specific definition of Rogaway et al. [?] 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. [?]), and even to achieve the recent, stronger goal of robust authenticated encryption [?]. By instantiating these modes with our construction, one can build a very secure scheme from an online cipher.

7.2 Further Research

The most important open problem left by this work is the security of the three round reversal scheme under variable input length attacks, without any additional restrictions on the lengths of these queries. Resolving this is an important step towards truly understanding the utility of an online cipher as a base primitive.

All our results are stated relative to an indistinguishability notion. A stronger notion is the indifferentiability framework [?], 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 [?] appear to extend to the prp case of online ciphers (curiously, the $\pm\mathsf{prp}$ 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. Similarly, 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). Similarly, the requirement of independence between the cipher calls is probably unnecessary, but removing it leads to a much more complicated setup, itself necessitating more complex security arguments.

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 $\pm\mathsf{prp}$ security.

8 Acknowledgments

The authors would like to thank Daniel Martin for thoughtful comments on early drafts of this document. We are also grateful to the reviews of Asiacrypt2015 for their comments on suggestions on a previous version.

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 Brlistol, supported by an EPSRC grant.

A Impracticality of Indifferentiability

The security definitions given in Section 2.2.1 are the standard indistinguishability notions for symmetric primitives, but are less strong than the *indifferentiability* notions of Maurer et al. [?]. 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 $\Delta\left(\begin{smallmatrix}\pi, S[\pi] \\ E[\mathcal{E}], \mathcal{E}\end{smallmatrix}\right)$, 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. [?] 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, 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 an ideal cipher (with respect to the online cipher). We assume $L(M)[1]$ is linearly dependant on $M[m]$ for all $M \in \Sigma^m$, 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. Let $E[\mathcal{E}]$ be the Π_i^L construction scheme instantiated with online cipher $\mathcal{E}$, and its inverse be $D[\mathcal{D}]$. Finally, let π be the ideal cipher, with inverse π^{-1}. Then, by the above attack, $\Delta\left(\begin{smallmatrix}\pi, S[\pi] \\ E[\mathcal{E}], \mathcal{E}\end{smallmatrix}\right)$ is large (in terms of number of simulator queries), and corresponds to insecurity under indifferentiability from an ideal cipher.

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 security of the notion

$$\Delta\begin{pmatrix}\pi & , \pi^{-1} & , S[\pi, \pi^{-1}] & , T[\pi, \pi^{-1}] \\ E[\mathcal{E}] & , D[\mathcal{D}] , & \mathcal{E} & , \quad \mathcal{D}\end{pmatrix}$$

(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 access. Other situations exist, such as a system providing interfaces for both directions of the online cipher, but only an interface for encryption queries of the true cipher. Whilst we find 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 2.2.1. Let E be a cipher on acting on Σ^t, a string of blocks of length t. Let $\tilde{E}$ a tweakable block cipher with blocks Σ and tweakspace $\mathcal{T}$ and $\mathring{\mathrm{E}}$ an online cipher acting on blocks Σ, all with keyspace $\mathcal{K}$. Then, the advantage of some adversary $\mathbb{A}$ against the security goals of the various objects are as follows:

$$\begin{aligned}
\mathbf{Adv}_E^{\mathsf{prp}}(\mathbb{A}) &:= \mathbb{P}\left[k \leftarrow\!\!\$\; \mathcal{K} \,:\, \mathbb{A}^{E_k} \to 1\right] - \mathbb{P}\left[\pi \leftarrow\!\!\$\; \mathrm{Perm}(\Sigma^t) \,:\, \mathbb{A}^{\pi} \to 1\right]\\
\mathbf{Adv}_E^{\pm\mathsf{prp}}(\mathbb{A}) &:= \mathbb{P}\left[k \leftarrow\!\!\$\; \mathcal{K} \,:\, \mathbb{A}^{E_k, E_k^{-1}} \to 1\right] - \mathbb{P}\left[\pi \leftarrow\!\!\$\; \mathrm{Perm}(\Sigma^t) \,:\, \mathbb{A}^{\pi,\pi^{-1}} \to 1\right]\\
\mathbf{Adv}_{\tilde{E}}^{\mathsf{tprp}}(\mathbb{A}) &:= \mathbb{P}\left[k \leftarrow\!\!\$\; \mathcal{K} \,:\, \mathbb{A}^{\tilde{E}_k} \to 1\right] - \mathbb{P}\left[\tilde{\pi} \leftarrow\!\!\$\; \mathrm{Perm}(\mathcal{T},\Sigma^t) \,:\, \mathbb{A}^{\tilde{\pi}} \to 1\right]\\
\mathbf{Adv}_{\tilde{E}}^{\pm\mathsf{tprp}}(\mathbb{A}) &:= \mathbb{P}\left[k \leftarrow\!\!\$\; \mathcal{K} \,:\, \mathbb{A}^{\tilde{E}_k, \tilde{E}_k^{-1}} \to 1\right] - \mathbb{P}\left[\tilde{\pi} \leftarrow\!\!\$\; \mathrm{Perm}(\mathcal{T},\Sigma^t) \,:\, \mathbb{A}^{\tilde{\pi},\tilde{\pi}^{-1}} \to 1\right]\\
\mathbf{Adv}_{\mathring{\mathrm{E}}}^{\mathsf{oprp}}(\mathbb{A}) &:= \mathbb{P}\left[k \leftarrow\!\!\$\; \mathcal{K} \,:\, \mathbb{A}^{\mathring{\mathrm{E}}_k} \to 1\right] - \mathbb{P}\left[\mathring{\pi} \leftarrow\!\!\$\; \mathrm{OPerm}(\Sigma) \,:\, \mathbb{A}^{\mathring{\pi}} \to 1\right]\\
\mathbf{Adv}_{\mathring{\mathrm{E}}}^{\pm\mathsf{oprp}}(\mathbb{A}) &:= \mathbb{P}\left[k \leftarrow\!\!\$\; \mathcal{K} \,:\, \mathbb{A}^{\mathring{\mathrm{E}}_k, \mathring{\mathrm{E}}_k^{-1}} \to 1\right] - \mathbb{P}\left[\mathring{\pi} \leftarrow\!\!\$\; \mathrm{OPerm}(\Sigma) \,:\, \mathbb{A}^{\mathring{\pi},\mathring{\pi}^{-1}} \to 1\right]\\
\mathbf{Adv}_F^{\mathsf{prf}}(\mathbb{A}) &:= \mathbb{P}\left[k \leftarrow\!\!\$\; \mathcal{K} \,:\, \mathbb{A}^{F_k} \to 1\right] - \mathbb{P}\left[\$ \leftarrow\!\!\$\; \mathrm{Func}(\Sigma^t) \,:\, \mathbb{A}^{\$} \to 1\right]\\
\mathbf{Adv}_F^{\pm\mathsf{prf}}(\mathbb{A}) &:= \mathbb{P}\left[k \leftarrow\!\!\$\; \mathcal{K} \,:\, \mathbb{A}^{F_k, F_k^{-1}} \to 1\right] - \mathbb{P}\left[\$, \$^{-1} \leftarrow\!\!\$\; \mathrm{Func}(\Sigma^t) \,:\, \mathbb{A}^{\$,\$^{-1}} \to 1\right]\\
\mathbf{Adv}_F^{\pm\mathsf{tprf}}(\mathbb{A}) &:= \mathbb{P}\left[k \leftarrow\!\!\$\; \mathcal{K} \,:\, \mathbb{A}^{F_k, F_k^{-1}} \to 1\right] - \mathbb{P}\left[\$, \$^{-1} \leftarrow\!\!\$\; \mathrm{Func}(\mathcal{T},\Sigma^t) \,:\, \mathbb{A}^{\$,\$^{-1}} \to 1\right]
\end{aligned}$$

Note that since the adversary is prevented from making queries to which he already knows the answer, this definition of an $\pm\mathsf{prf}$ 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

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

and where appropriate provision for variable input lengths by sampling an element for each length t. A primitive P is a secure xxx if $\mathbf{Adv}_P^{\mathrm{xxx}}(q)$ is sufficiently small.

C Changelog

C.1 Spring 2016

The original version of this paper ([?], version 20150521:200909) contained a flaw that was pointed out by Bhaumik and Nandi [?]. They provided an attack against the Π_2^{rev} scheme, demonstrating it cannot achieve $\pm\mathsf{prp}$ security as a VIL cipher, along with an alternative two-layer scheme, and H-coefficient security proofs for the later constructions. Although their attack is presented in terms of small queries, it can be generalised to longer minimum message lengths by replacing each “block” with an appropriately long “string of blocks”. This work has now been incorporated into Section 4.2.

Upon further investigation, we were able to generalise the Bhaumik–Nandi counterexample into a more general attack on the proof technique used throughout the earlier draft, and identified flaws in their alternative proofs. That is, while we did not construct attacks against the other schemes (and we expect their security still holds), we were able to demonstrate that the previous proofs were inaccurate due to poor handling of VIL attacks.

Motivated by these flaws and comments from the ASIACRYPT 2015 reviewers, the paper has been substantially rewritten using a different proof style, that better exposes the reasoning behind it. Our new results are highly modular, separated into a series of smaller claims.

The delay in providing this new draft stems predominantly from the extra research required, that only became apparent during the process of solving what initially appeared to be small proof bugs. That said, we apologise for the delay in posting this amended version and any misunderstandings this may have caused.

C.2 Winter 2017

The work was submitted for consideration for the IACR Transactions on Symmetric Cryptology 2017 Issue 2, in a form very similar to a recent ePrint version ([**?**], version 20160903:182510). The reviewers identified a number of areas that were not sufficiently clear, and encouraged us to remedy this and resubmit. Along with a number of typographical errors, two key points were highlighted for improvement, our responses to which are given below. In this updated document, we have incorporated the reviewers advice, and would like to thank them for their input.

Use cases and relevance Firstly, the reviewers suggested we explicitly incorporate the possible use-case of HSM-based setting, where a designer might not have direct access to the traditional toolkit of primitives. We have updated Section 1 to include the HSM example from our initial rebuttal.

Lemma 6 More significantly, Lemma 5 and Lemma 6 were unclear, both in their statement and use. We made significant modifications for the current revision to clarify both lemmas and added additional context.

Lemma 5 has been slightly modified, predominantly to remove poorly written sections. Lemma 6 has been completely rewritten. The previous version was heavily parameterised, this had been done in a somewhat cumbersome manner and made the result itself unclear. The new version is now defined through explicit code-based games, and is more specific in what it aims to achieve. An additional discussion paragraph has been added to explain the ideas behind the lemma, and how it will be used. As part of this, the variable names used for Lemmas 5 and 6 were changed to use P, Q, R. This allowed us to more directly state which variables in the theorems correspond to which variables in the lemma This also lead to renaming the internal tables of our ideal online permutations from $\mathcal{P}$ to $\mathring{\pi}$.

Clarification of identical-until-bad terminology While applying the reviewers comments and percolating the effects of this, it was identified that the work was at times a bit sloppy with use of terminology, using "online cipher" instead of "ideal online permutation" in the later sections. We believe this correction has been propogated through the work, but may have missed instances.

The second piece of particularly unclear terminology was the way in which identical-until-bad arguments were applied. We acknowledge that this was not the standard use of the term, and have added a section to describe our mechanism (Section 3.4). As part of this, we also clarified what we meant for two oracles to be identical-until-bad: that no adversary could distinguish between interacting with one from interactions with the other without first setting bad (Section 3.1.2).

Proofs The proofs were all modified slightly to use the updated versions of the lemmas. The most significant changes were made to Theorem 4, most of which was rewritten to improve readability.