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 Bristol,
Merchant Venturers Building, Woodland Road,
Bristol, 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 gives a clear performance advantage, yet it comes at a price. Since ciphertext blocks cannot depend on later plaintext blocks, diffusion and hence security is limited. We show how one can attain the best of both worlds by providing provably secure constructions, achieving full cipher security, based on 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

1 Introduction

Modern understanding of symmetric cryptology has come a long way from a straightforward adaptation (cf. [KL08, Def. 3.30]) of the seminal definitions of probabilistic [public key] encryption [GM84]. Both authenticated encryption and variable input length ciphers have emerged as noteworthy primitives. From an efficiency perspective, a scheme is ideally one-pass and online, outputting ciphertexts as plaintext becomes available. For nonce-based authenticated encryption, online schemes do not suffer in security, as long as nonces are indeed unique (and decryption of invalid ciphertexts only produces a single error message [BDPS14, ABL+14, HKR15, BPS15]). Once nonces *do* repeat, prefix patterns in the input show up in the output; the same is true for online ciphers. Two pass schemes become a necessity. This raises the question how easily one can boost the security of an online scheme. In this paper, we concentrate on turning online ciphers into fully fledged ciphers using only two or three passes (depending on the desired security level).

The original goal of cryptography was data confidentiality. From a modern perspective, this is addressed by authenticated encryption (AE), which provides both confidentiality and integrity (including of associated data [Rog02]). 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 [Jou06, KR11]. The security goal of *misuse resistant* AE [RS06] 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 [NRS14, Ber13], but the number of options reduces drastically when misuse resistance is required. One approach is the encode-then-encipher (or pad-then-encipher) paradigm [BR00, RS06, ST13], where (public) redundancy is added to the message before it is being enciphered using a variable input length strong pseudorandom permutation (±PRP cipher).

Variable input length (VIL) ciphers (either ±prp or prp secure) are interesting in their own right, especially in scenarios where encryption has to occur *in situ* [BR99]. 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* [BBKN01] is a variable input length keyed permutation based on a blockcipher that outputs a ciphertext block as soon as it receives a plaintext block (but still based on all preceding plaintext blocks). In other words, it allows instant processing of plaintext and outputting ciphertext on the fly. Since online ciphers cannot be prp secure, relaxed security notions exist that capture "best possible" security. Online ciphers play a key role in achieving a similarly relaxed notion of online authenticated encryption with graceful security degradation against nonce reuse [FFL12].

We believe there are many scenarios where an online cipher's security limitations are outweighed by their efficiency, but at the same time there will be situations where full cipher security is paramount. One could create tailor-made solutions for each of the primitives, but often it is more desirable to share components. This could be solved by using two modes of operation on say AES, but we imagine an environment where black-box use of an online cipher is already available (such as via an existing API), and we are tasked to create a true cipher based on the access to the online cipher only.[1]

[1] Obviously if one had direct access to whatever primitive underlies the online cipher, more efficient

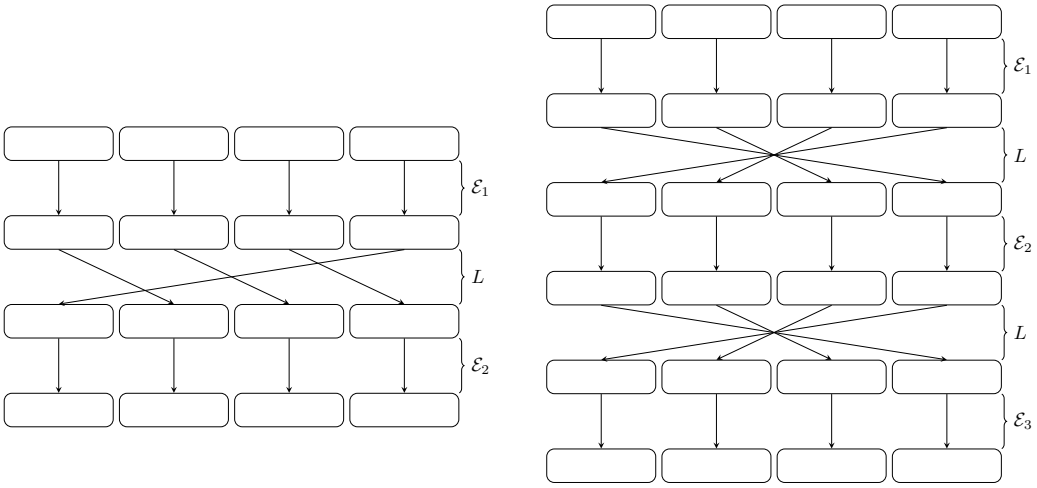


Figure 1: Examples of the construction. On the left is the two layer reversing scheme, and on the right the three layer right shift instantiated with independent ciphers.

1.0.1 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 2.0.1), 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 [BR99] or in the context of key-wrap schemes [Dwo04]. The novelty of our contribution resides in using an online cipher as underlying primitive, and what we are able to prove as a result. Table 1 provides a summary of our results. The security bounds are simplifications of those in the paper, compromising tightness in favour of clarity (for stricter bounds please refer to the relevant theorems). As a boon, we describe an explicit correspondence between tweakable blockciphers and online ciphers (Theorem 1), extending an observation by Rogaway and Zhang [RZ11].

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 7). 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. $\pm$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 8), 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 2). When three rounds are used, security is achieved beyond the birthday

(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 [BFSW13].

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 $n = \log N$. A bound is "tight" if there exists an attack that asymptotically (in q, N) matches the security bound.

Construction			Input	Security		
Linear Layer	Ciphers	Goal	Length	Advantage	Proof	Tight?
Right-shift	2	PRP	VIL	$1.5\ q^2/N$	Theorem 2	Lemma 7
Right-shift	3	PRP	VIL	$1.5(\sigma/N)^2$	Theorem 4	Lemma 10
Reversal	2	±PRP	VIL	Insecure	—	Lemma 8
Reversal	2	±PRP	AIL	q^2/N	Corollary 2	Lemma 7
Reversal	3	±PRP	AIL	$n\,q/N$	Theorem 5	

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.

1.0.2 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 [BR00, ST13]). Moreover, Hoang et al. demonstrate that with a tweakable cipher one may achieve the even stronger goal of Robust Authenticated Encryption [HKR15, Theorem 5] (itself implying full misuse-resistant security [RS06]). Incorporating our results plugs the gap to turn a secure online cipher into an Authenticated Encryption scheme meeting the strongest of security bounds.

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 [HKR15, 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 saves chip area 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.

1.0.3 Related work

The concept of an online cipher was first studied by Bellare et al. [BBKN01], 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 [JMV02, FMP03, FJMV04, BT04].

Later, Rogaway and Zhang exposed the close relationship between tweakable blockciphers and online ciphers [RZ11], an observation that has since been exploited by others, yielding several explicit constructions (e.g. McOE [FFL12]). There now exist a wide range of online cipher constructions, such as COPE [ABL+13], POE [AFF+15] and ELmE [DN14], the majority of which achieve birthday bound security. We are not aware of any online ciphers whose security might extend beyond the birthday bound.

As part of our study, we investigated some constructions similar to the CMC-core [HR03] (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 [ST13], but as discussed in Section 5.0.1 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 [Dwo04] 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 [Dwo04] uses the cyclic shift instead. Our results are a next 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 [Hal07] or AEZ [HKR15] do), or extend the domain of a tweakable blockcipher (e.g. Minematsu's construction [Min09]). 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.

1.0.4 Context and caveats

We will model the online cipher used as having ideal properties, leading to an information-theoretic proof. Instantiating the scheme with any specific online cipher construction incurs an extra term (expressing the online-cipher security of the specific construction). We 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 [Nan08, RZ11], and remark that existing domain completion techniques are not without issue.

2 Preliminaries

2.0.1 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. Both $|L|$ and $\#L$ will at times be used for 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, so a game sets bad if bad_i is triggered for any i. To ensure our results are verifiable and clear, longer proofs will be broken up into a

number of claims. The proof of a claim will end with ■, with □ denoting the end of the overall proof.

Finally, we will, at times, use the terms encryption and decryption (or variants thereof) to describe forward and backward queries to the construction. Later we will use $\mathcal{E}$ to refer both to an online cipher and its encryption routine, with $\mathcal{D}$ referring to the decryption routine. At times we will use notation like E^T to refer to 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 $S \in \Sigma^*$.

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 *prefix*).

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.

Triples as functions To concisely describe the partial state of a lazily sampled function, we will allow sets of triples to represent a tweakable functions, slightly overloading the notation. So for a set $\mathcal{P}$ that also represents a function, $\mathcal{P}^T(M) = C \iff (T,M,C) \in \mathcal{P}$ and similarly $\mathcal{P}^{-T}(C) = M \iff (T,M,C) \in \mathcal{P}$. When $\mathcal{P}$ is the internal table for a tweakable blockcipher (see Definition 2), we will also abuse notation slightly when extending it, allowing ourselves to add a string of blocks in one go. Formally this means that when $M, C \in \Sigma^k$ for some k and any $T \in \Sigma^*$, we let $\mathcal{P} \xleftarrow{\cup} (T,M,C)$ concisely represent "For every $i \in \{1,\dots,k\}$, $\mathcal{P} \xleftarrow{\cup} (T||\lceil M \rceil^{i-1}, M[i], C[i])$".

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 $\mathcal{A}$ interacting with $\mathcal{W}$ will terminate with output x, which we denote by $\mathcal{A}^{\mathcal{W}} \to x$.

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* between worlds $\mathcal{W}_0$ and $\mathcal{W}_1$ within q queries $\Delta^{\mathcal{W}_1}_{\mathcal{W}_0}(q)$ corresponds to the maximum *distinguisher*. Formally,

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

Several of our proofs will demonstrate two worlds are indistinguishable until the adversary triggers some event "bad", known as identical-until-bad [BR06]. Having done this, we will bound the probability that an adversary communicating with either of the two (identical) worlds can cause a bad event, and in doing so bound the (in)security of the construction.

2.1 Primitives

We use a number of standard primitives, in particular the notions of a cipher (e.g. [BR99]), tweakable blockcipher [LRW02] and online cipher [BBKN01]. 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 *VIL cipher acting on blocks*.

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

Thus a tweakable blockcipher can be thought of as a collection of blockciphers, 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 $\mathcal{E}$ of permutations on Σ^+ indexed by some $k \in \mathcal{K}$, where for any $m > 0$ and $A \in \Sigma^m$, $\mathcal{E}_k(A||B)[1..m] = \mathcal{E}_k(A)$ for all $B \in \Sigma^*$.

This formalisation of an online cipher (due to Bellare et al. [BBKN01]) describes a construction that can output ciphertext blocks as soon as the corresponding message blocks arrive. 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. [ABL+13]).

Definition 4 (Online-but-last cipher)**.** An *online 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 $k \in \mathcal{K}$, where for any $m > 0$ and $A \in \Sigma^m$, $\mathcal{E}_k(A||B)[1..(m-1)] = \mathcal{E}_k(A)[1..(m-1)]$ for all $B \in \Sigma^*$.

2.1.1 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. Motivated by this, let $\mathrm{Perm}(l)$ be the set of all permutations π on l bits. Then the ideal cipher samples a permutation from $\mathrm{Perm}(l)$ uniformly, making every possible permutation is equally likely. The actual security of a cipher E can then be bounded by the likelihood an adversary can distinguish a randomly keyed instance of it from the ideal cipher (based on oracle access only).

Similarly, we define the ideal primitives for random functions, TBCs and online ciphers by first defining the set of all such objects, following standard terminology neatly collated by Halevi and Rogaway [HR03]. Define $\mathrm{Func}(\Sigma)$ to be the set of all functions $\$\colon \Sigma \to \Sigma$, $\mathrm{Func}(\mathcal{T}, \Sigma)$ the set of all functions $\$\colon \mathcal{T} \times \Sigma \to \Sigma$, $\mathrm{Perm}(\mathcal{T}, \Sigma)$ 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 permutation, and $\mathrm{OPerm}(N)$ the set of all online permutations $\mathring{\pi}$ with blocks Σ. 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

function $\$^T(M)$
  $C \leftarrow\!\!\$\ \Sigma^{|M|}$
  if $\exists x$ s.t. $(M, x) \in \mathcal{F}_T$ **then**
    $C \leftarrow x$
  $\mathcal{F}_T \leftarrow\!\!\cup\ (M, C)$
  return C

function $\$^{-T}(C)$
  $M \leftarrow\!\!\$\ \Sigma^{|C|}$
  if $\exists x$ s.t. $(x, C) \in \mathcal{F}_T$ **then**
    $M \leftarrow x$
  $\mathcal{F}_T \leftarrow\!\!\cup\ (M, C)$
  return M

Figure 2: (Tweakable) Random Function with inverse.

is returned, and if not a value is uniformly sampled and recorded (see Figure 2 for a code-based definition).

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 cipher, ideal tweakable blockcipher and ideal online cipher respectively (when provided with oracle access to just the encryption interface). Analogously, we define ±PRF, ±PRP, ±TPRF, ±TPRP, ±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 3.3. The complete list is provided in Appendix B, but as an example,

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

The goals above are defined for primitives of just a single length, termed the Arbitrary Input Length (AIL) setting. The Variable Input Length (VIL) setting (e.g. [BR99]) generalises this to allow variable length constructions, by providing an equivalent interface for every requested length. For some goal xxx, $\mathbf{Adv}^{\mathrm{xxx}}_{P(q)}$ is defined as the maximum across all adversaries making q queries. We say a scheme P is a *secure* xxx if $\mathbf{Adv}^{\mathrm{xxx}}_{P(q)}$ is sufficiently small. At times we use the term ±PRP to refer to a secure ±prp, and similarly for any other goals.

2.2 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). Restricting to linear mixing layers leads us to the following definition.

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

So, for example, $\Pi^L_2(M) = \mathcal{E}_2 \circ L \circ \mathcal{E}_1(M)$. We will consider various combinations of (L, i) and observe that some combinations lead to schemes with PRP or ±PRP security. When clear, we omit the linear layer or number of rounds from the notation.

Candidate linear layers The first and most obvious candidates for linear layers are blockwise permutations: maps that simply reorder the blocks. In this paper we will focus on the blockwise reversal map rev as the simplest map that my reasonably lead to a ±PRP. Inspired by the choices of AESKW [Dwo04], 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 will be important in later proofs. 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 Ideal primitives

For conciseness, let $\mathcal{E}$ be the encryption routine of the ideal online cipher, with inverse $\mathcal{D}$: our proofs will be constructed around this ideal primitive. Since $\mathcal{E}$ can be thought of as a randomly sampled online permutation, we do not notate a key, and so free up the subscript to denote separate instantiations of the primitive. Application of our results to real constructions requires swapping the real online cipher for the ideal primitive. The cost of this switch depends on the overall objective, since the online cipher need only be secure under the equivalent notion to the overall construction. For example, for $\pm$prp security the online cipher must be a secure $\pm$oprp, but for prp security the online cipher need just be an oprp.

3.1.2 Classical bounds

To bound final collision events, we use the well known birthday bound (Lemma 1). Lemma 2 reproduces $\pm$PRP–$\pm$PRF switching lemma by Halevi and Rogaway [HR03, Appendix C], which we use in several proofs to hop (in the ideal world) from a random permutation to a random function with inverse.

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

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

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

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

3.1.3 Lazy Sampling and bad events

At times, the oracles may themselves call out to internal oracles, to which the adversary does not have access. They will model standard primitives through "Lazy Sampling", where random elements are only sampled when required. Internal variables are shared between corresponding forward and backward interfaces, but not between different instantiations or primitives. As an example, Figure 3 defines an oracle that implements a tweakable block cipher through lazy sampling, building up the table $\mathcal{P}$ of known values as queries are made.

A direct result of using lazy sampling is that it becomes apparent how similar some of these ideal primitives are to one-another. Our next lemma states this, providing an alternative description of the afore-mentioned $\pm$PRP–$\pm$PRF switch in terms of actual code-based games. and will be used to perform the actual switch within our later proofs. The associated code, given in Figure 4, defines a $\pm$PRP, $\pm$PRF or pair of random samplers depending on which sections of optional code are included. Moreover, since the oracles

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

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

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

function $\tilde{\mathrm{E}}^T(M)$
  $C \leftarrow_{\$} \Sigma$
  if $(T, M, \star) \in \mathcal{P}$ **then**
    $\mathsf{bad}_1 \leftarrow \mathtt{true}$
    $C \leftarrow \mathcal{P}^T(M)$
  if $(T, \star, C) \in \mathcal{P}$ **then**
    $\mathsf{bad}_2 \leftarrow \mathtt{true}$
    $C \leftarrow_{\$} \Sigma \setminus \mathrm{Im}(\pi^T_{\mathcal{P}})$
  $\mathcal{P} \leftarrow_{\cup} (T, M, C)$
  return C

function $\tilde{\mathrm{D}}^T(C)$
  $M \leftarrow_{\$} \Sigma$
  if $(T, \star, C) \in \mathcal{P}$ **then**
    $\mathsf{bad}_3 \leftarrow \mathtt{true}$
    $M \leftarrow \mathcal{P}^{-T}(C)$
  if $(T, M, \star) \in \mathcal{P}$ **then**
    $\mathsf{bad}_4 \leftarrow \mathtt{true}$
    $M \leftarrow_{\$} \Sigma \setminus \mathrm{Dom}(\pi^T_{\mathcal{P}})$
  $\mathcal{P} \leftarrow_{\cup} (T, M, C)$
  return M

Figure 4: Comparing tweakable random functions. The list $\mathcal{P}$ is initialised empty. To be a tweakable random function with unique inputs (also known as a uniform sampler), we include none of the boxed code. A $\pm$tprf includes the unboxed code and also the code in dashed boxes. Finally, a $\pm$tprp uses all of the code. Thus we see the three constructions (and the various hybrids of them) are all identical-until-bad.

given are all identical-until-bad, any combination of the four bad events can be used for switching. Thus one may switch a $\pm$PRP for something that is a PRP in the forward direction, and a random sampler in the inverse direction.

Obviously there are significant consistency concerns that must be addressed when making such switches, and when used in hybrid arguments this technique often leads to excessively large adversarial advantages. However, when carefully combined such that bad events align between switches it can be an effective and efficient tool.

Lemma 3 (Code-based $\pm$PRP switches)**.** *Until either tweak–input or tweak–output pairs repeat, a tweakable* $\pm$prp *is indistinguishable from a pair of random samplers. Moreover, a partial switch can also be performed to exchange the two oracles of an tweakable* $\pm$*PRP for a tweakable random permutation in one direction and a random sampler in the other.*

Proof. This result is immediate, and demonstrated by 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[RZ11]. 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 the setting a result of Bellare et al. into modern terminology [BBKN01, Proposition 1], while the converse is an extension of the result of Rogaway and Zhang. So, the result is not particularly surprising, but is none the less important, since it yields neater terminology within which to study online ciphers.

To this end, we first define an additional piece of notation emphasising the relationship, and borrow terminology from the tweakable setting to describe it. Let $\mathcal{E}(\cdot)$ be an online cipher and $A, B \in \Sigma^*$ with $|A| = a$ and $|B| = b$. Then, define $\mathcal{E}^A(B) := \mathcal{E}(A||B)[(a+1)..(a+b)]$. So, $\mathcal{E}^A(B)$ returns the output blocks corresponding to B when processed with a *prefix* of A. Then, by the online property, $\mathcal{E}(A||B) = \mathcal{E}^\epsilon(A)||\mathcal{E}^A(B)$. If we think of $\mathcal{E}^\cdot(\cdot)$ 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 [RWZ12], except that prefixes may be arbitrarily large, whereas states have fixed length.

Similar to the encryption case, we define $\hat{\mathcal{D}}^A(B) := \mathcal{D}(A||B)[(a+1)..(a+b)]$. By setting $\mathcal{D}^A(B) := \hat{\mathcal{D}}^{\mathcal{E}(A)}(B)$, we obtain the inverse of $\mathcal{E}^A$, as $\mathcal{D}^A(\mathcal{E}^A(B)) = B$. This is not a problem to compute, since to calculate $\mathcal{D}(A||B)$ one first calculates $M = \mathcal{D}(A)$, then $\mathcal{D}(A||B) = M||\mathcal{D}^M(B)$, something the online cipher does internally anyway. Our notation emphasises the correspondence with 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 throwing away all but the final block, effectively using the bulk of the cipher call preprocessing the tweak. So, if $\mathcal{E}$ is an online cipher then $f(\mathcal{E})$ is the tweakable blockcipher $f(\mathcal{E})_k^T(M) := \mathcal{E}_k^T(M)$ 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

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

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

3.3 Properties of Online Ciphers

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

Corollary 1. *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 $|B|$.*

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

function $\mathcal{O}_1(X_1||A_1||B_1)$
$X_2 \leftarrow \mathcal{E}^{\epsilon}(X_1)$
$A_2 \leftarrow \mathcal{E}^{X_1}(A_1)$
$B_2 \leftarrow \mathcal{E}^{X_1||A_1}(B_1)$
return $X_2||A_2||B_2$

function $\mathcal{O}_3(X_x||A_1||B_1)$
$X_1 \leftarrow \Sigma^x$
$C \leftarrow \mathcal{E}(X_1||A_1||B_1)$
return C

function $\mathcal{O}_2(X_1||A_1||B_1)$
$X_2 \leftarrow \mathcal{E}^{\epsilon}(X_1)$
$A_2 \leftarrow_{\$} \Sigma$
$B_2 \leftarrow_{\$} \Sigma^{k-1}$
if $(X_1, A_1, \star) \in \mathcal{P}$ **then**
$\mathsf{bad}_A \leftarrow \mathtt{true}$
if $(X_1, \star, A_2) \in \mathcal{P}$ **then**
$\mathsf{bad}_{A'} \leftarrow \mathtt{true}$
$\mathcal{P} \leftarrow_{\cup} (X_1, A_1||B_1, A_2||B_2)$
return $X_2||A_2||B_2$

Figure 5: The two routines $\mathcal{O}_1$ and $\mathcal{O}_2$ are identical until bad. In each case they have access to the online cipher $\mathcal{E}$, and its internal table $\mathcal{P}$. $\mathcal{O}_1$ is simply the result of expanding out the call to an online cipher via the online property, while $\mathcal{O}_2$ inlines the second to calls before simplifying to something that is identical-until-bad. Also shown is oracle $\mathcal{O}_3$, whose outputs are indistinguishable from those of a random permutation.

Lemma 4. *Let $\mathcal{R} = (R_1, \ldots, R_q)$ be a list of q blocks, where each $R_i = \mathcal{E}^{s_i}(t_i)$ is the output of the encryption of a unique input, meaning $s_i||t_i \neq s_j||t_j$ for any $i \neq j$. Then, the probability of a collision in the list (that $R_i = R_j$ for $i \neq j$) is bounded, with $\mathbb{P}[\exists i \neq j \text{ s.t. } R_i, R_j \in \mathcal{R}] \leq \frac{1}{2}q(q-1)/N$.*

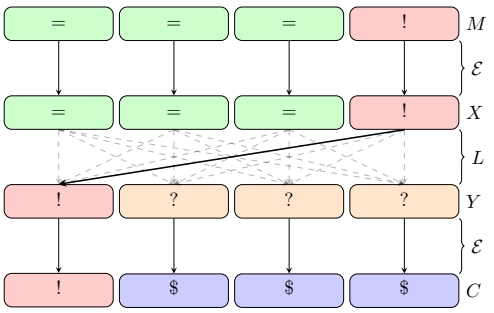
Proof. Let $i \neq j$. Then, by construction, $R_i = R_j \iff \mathcal{E}^{t_i}(s_i) = \mathcal{E}^{t_j}(s_j)$. If $t_i = t_j$, then $R_i = R_j$ implies that $s_i = s_j$, which contradicts the assumption that all inputs were unique, and so cannot happen. If $t_i \neq t_j$, the tweakable cipher has different tweaks in instance, and so the two distributions are independent. Thus R_i and R_j are both sampled uniformly at random and independently, and so collide with probability N^{-1}. So, taking the maximum of these probabilities, $\mathbb{P}[R_i = R_j \mid i \neq j] \leq N^{-1}$. Applying the union bound, we get the required result. □

The final tools we will provide are slightly more cumbersome to define, but their utility will soon be clear. We will use it frequently to replace the cipher call of our constructions with an alternative routine that is easier to reason about, essentially by extending Lemma 3 into the online context. Moreover, we show that a slight variant of this routine yields outputs are indistinguishable from those of a random permutation. Together these will provide the main structure for our later results, since, provided the inputs to the final cipher call are of the appropriate form, they will imply security of the scheme until bad.

Lemma 5. *The routines $\mathcal{O}_1$ and $\mathcal{O}_2$ in Figure 5 are identical-until-bad.*

Proof. Oracle $\mathcal{O}_2$ differs from $\mathcal{O}_1$ in that it expands out the online cipher calls used to define the output blocks A_2 and B_2. Under the assumption that X_1, A_1 is fresh (since otherwise bad_A is set), there is no need to look this value up within previous queries. Moreover, rather than sampling from the set of undefined values, A_2 is sampled uniformly, which the consistency check removed (since $\mathsf{bad}_{A'}$ is set first). This then corresponds to performing a PRP-PRF switch on just the forward direction of the tweakable blockcipher $\mathcal{E}^{X_1}$. Since X_1, A_1 was fresh, B_1 is encrypted under a new prefix and thus sampled uniformly at random. Finally, these values are added to the table $\mathcal{P}$ to ensure any future calls are consistent. □

Lemma 6. *Fix $x \geq 0, a \in \{0, 1\}$ and set $k = x + a$. Let $\mathcal{E}$ be an online cipher to which the adversary does not have access. Then the oracle $\mathcal{O}_3$ (Figure 5) is identical-until-bad to a random permutation.*



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 6: An attack against PRP security of Π_2^L (Theorem 7)

Proof. As shown in Lemma 5, $\mathcal{O}_2$ is identical-until-bad to the online cipher $\mathcal{E}$, so we first make this substitution. As X_1 is sampled uniformly, $\mathcal{O}_3$ is identical to a random function until bad, since the three output variables are independently uniformly sampled: A_2, B_2 directly and X_2 as the image of X_1 under a permutation. Moreover, any output collisions imply repeated values of $X_2||A_2$, which implies that (X_1, A_2) must have repeated because X_2 is the image of X_1 under a permutation. This would have set $\mathsf{bad}_{A'}$, and so it is identical to a random permutation until bad. □

Online-but-last ciphers A secure 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 block tweakable cipher for the final block. So, an online-but-last cipher $\mathcal{O}_{obl}$ is the concatenation of an online cipher $\mathcal{E}$, and a separate tweakable block cipher $\widetilde{\mathcal{E}}$. That is, for a query $M \in \Sigma^m$,

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

4 Two layer constructions

Any scheme with a single layer (and thus one call to the online cipher), can be trivially distinguished from a PRP with two queries. Explicitly, after querying $\mathcal{O}_{\mathrm{Enc}}(\langle 0\rangle_1||\langle i\rangle_1)$ for $i \in \{0,1\}$, the two ciphertexts always agree on the first block of output for an online cipher, yet rarely for a true prp. In this section we ask the natural question: what can be achieved using two layers?

In Lemma 7 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).

The most intuitive candidate for a ±PRP is Π_2^{rev}, but in Lemma 8 we show that this construction is not secure in the most general sense. We remedy this by providing a similar (but different) construction that, at the cost of a small overhead, can achieve SPRP security (Theorem 3) even against variable length queries. This allows us to recover security of the original Π_2^{rev} construction (Corollary 2) against AIL adversaries (from whom all queries have the same arbitrary input length). We begin by proving the limitations of a two-round construction.

Lemma 7. *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,* $\mathbf{Adv}^{\mathsf{prp}}_{\Pi}(q) \geq \frac{q(q-1)}{8 \cdot N}$ *for all* $q \leq \sqrt{N}$.

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

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

We begin by calculating $\mathbb{P}\left[\mathcal{A}^{\Pi} \to 1\right]$, following the logic shown in Figure 6, and label the internal variables as M, X, Y, C as per the diagram. By the online property, $X^t = \mathcal{E}(M^t)$ begins with a blocks that are the same across all queries. Since the final block is encrypted under the same prefix each time, the values of $X^t[m]$ are distinct between queries. By assumption on L, $Y^t[1]$ is linearly dependent on $X^t[m]$. Since the other blocks of X^t are constant through all queries, we must have that $Y^t[1] = A \oplus X^t[m]$ for some A that is independent of t. 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 $\mathbb{P}\left[\mathcal{A}^{\Pi} \to 1\right] = 1$.

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

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

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

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

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 PRP, since the first output block cannot possibly depend on the final input block. Moreover, this means Π_i^{right} cannot be an $\pm$PRP, since its inverse is the Π_i^{left} scheme instantiated around $\mathcal{D}$. Indeed, for any linear layer L, Π_2^L cannot be a secure PRP if the linear layer's first output block is independent of the final input block.[2]

Combining this limitation with Lemma 7 (two layer constructions cannot be indistinguishable from a PRP 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 PRP up to the birthday bound. Explicitly,* $\mathbf{Adv}^{\mathsf{prp}}_{\Pi_2^L}(q) \leq \frac{q(q-1)}{N}$.

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

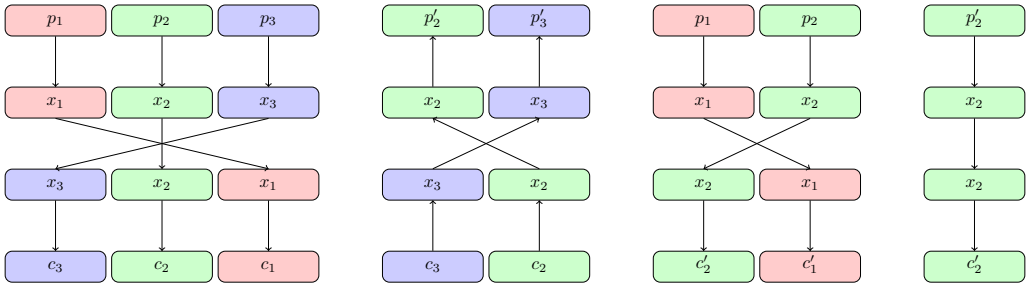


Figure 7: Attack against Π_2^{rev}, as described in Lemma 8. The adversary $\mathcal{A}$ makes the four queries shown in this diagram, and returns 1 if the two values labelled c'_2 agree. In the real case this will always occur, but almost never in the ideal case.

Proof. We will use a very simply identical-until-bad argument. Assume that the final block of output from the first online cipher call is always unique (at the cost of a collision term, by Lemma 4). Then, we can apply Lemma 6 in the case $(x, a) = (0, 1)$ to see that the construction is indistinguishable from a PRP until the first block of output (itself randomly and independently sampled on each query) repeats. Both of these events is bounded by the birthday bound, and their sum gives the claimed result. □

4.2 Two layers versus ±PRP security

We move on to consider ±PRP security. Such a scheme will still be susceptible to the birthday attack given in Lemma 7, 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 both $L(M)[1]$ and $L^{-1}(M)[1]$ must depend on $M[m]$.

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, in the VIL case, this turns out to be insecure, as shown by the following attack.

Lemma 8. Π_2^{rev} *is not a secure VIL–±PRP.*

Proof. We will provide an explicit adversary, whose queries are each a small number of blocks long. The attack still holds when each variable is considered a sequence of blocks, meaning one can generalise the attack to apply with most reasonable message-length requirements. Let Π and Π^{-1} be oracles corresponding to the construction, and let $\mathcal{E}$ be the internal online cipher.

Consider the adversary $\mathcal{A}$ who makes four queries of varying lengths to the construction, as visualised in Figure 7. First, $\mathcal{A}$ picks an arbitrary three-block string $p_1||p_2||p_3$, and queries $c_1||c_2||c_3 \leftarrow \mathrm{Enc}(p_1||p_2||p_3)$. Next, $\mathcal{A}$ queries $p'_2||p'_3 \leftarrow \mathrm{Enc}^{-1}(c_3||c_2)$ and then $c'_2||c'_1 \leftarrow \mathrm{Enc}(p_1||p_2)$. Finally, they query $\alpha \leftarrow \mathrm{Enc}(p'_2)$ and return 1 if and only if $\alpha = c'_2$.

As shown in the figure, we will always have $\mathcal{A}^{\Pi_2^{\mathsf{rev}}} \to 1$, Let $x_1x_2x_3 := \mathcal{E}(p_1||p_2||p_3)$. 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, this will almost certainly not occur, with $\mathbb{P}\left[\mathcal{A}^{\pi,\pi^{-1}} \to 1\right] = N^{-1}$. Taking the difference between these two terms, we see that $\mathcal{A}$ almost always distinguishes the scheme from a ±PRP. □

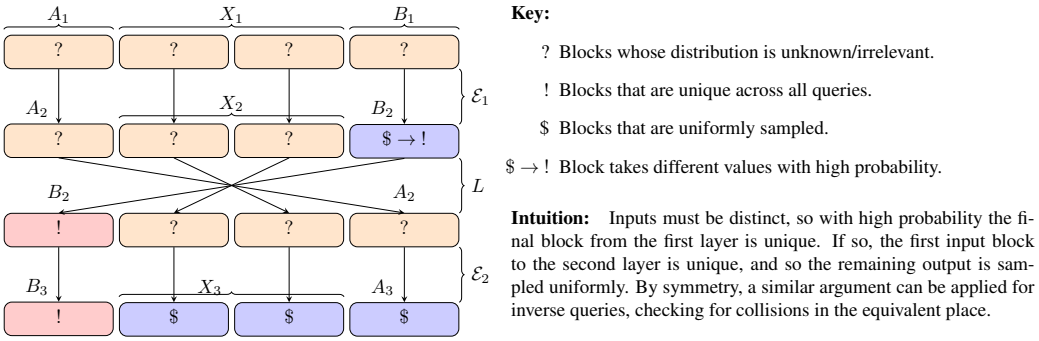


Figure 8: Intuition behind the AIL-±PRP security of Π_2^{rev}, as shown by the VIL security of an Online-but-last scheme (Theorem 3).

It is clear that this attack requires making variable input length queries, so a natural follow-up question is what minor modification are required to make the scheme secure. To answer this, we show that the scheme is secure if built around an "online-but-last" cipher, a cipher which are online for all but the final block, which is sampled independently for each query length. Since a secure online cipher is a secure online-but-last cipher when queried with messages of only 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 cipher around the linear layer that reverses the order of the blocks, is a secure VIL–±PRP up to the birthday bound. Explicitly,* $\mathbf{Adv}_{\Pi}^{\pm\mathsf{prp}}(q) \leq q(q-1)/N$.

Corollary 2. Π_2^{rev} *is a secure AIL-±PRP until the birthday bound:* $\mathbf{Adv}_{\Pi_2^{\mathsf{rev}}}^{\pm\mathsf{prp}}(q) \leq q(q-1)/N$.

Proof of Theorem 3. We will use a game-based proof, following the intuition shown in Figure 8 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 3.3, 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 this 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), 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 ±PRP, so it remains to bound this probability. 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 $q(q-1)/N$. □

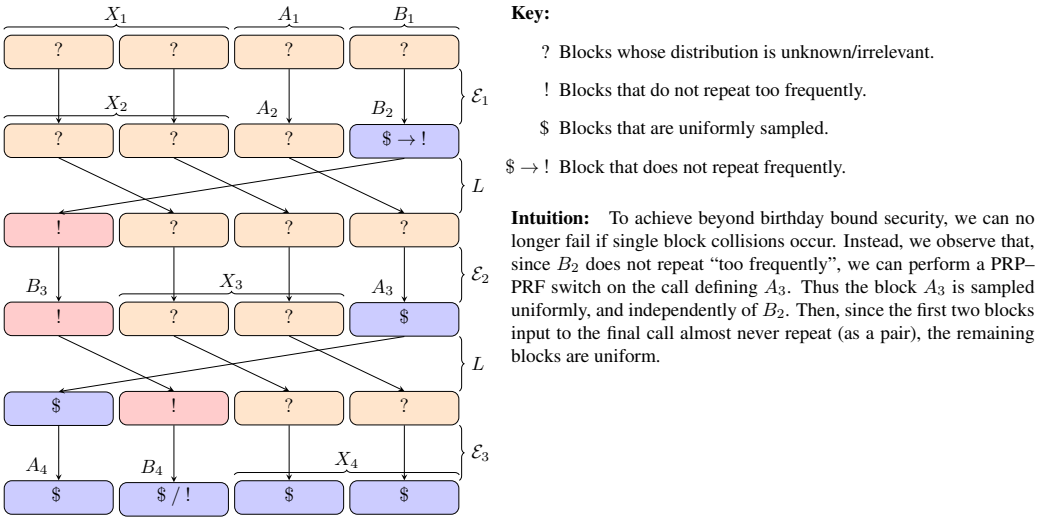


Figure 9: Intuition behind the PRP security of Π_3^{right} (Theorem 4)

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 ±PRP security up until almost the blocksize.

5.0.1 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 [ST13], 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}). This is then used as a unique tweak to encrypt the remaining message blocks, before the first blocks are 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 ±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 ±PRP security because its inverse is trivially distinguishable. There exists a generic attack against right-shift based schemes described later (Lemma 10), and substituting in the appropriate parameters we see that for $N \geq 4$ and messages of length at least 4 blocks, $\mathbf{Adv}_{\Pi}^{\mathsf{prp}}(q) \geq \frac{q(q-1)}{8N^2} \approx (\frac{q}{N})^2$ for $q \leq N$. Thus the best we can reasonably expect is PRP security up to the blocksize, something we achieve, asymptotically matching the attack. Again, we present the logic behind our proof in a diagram (Figure 9).

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

Proof. To prove this, we split the internal variables into three sections. In this case, we do so such that for an m block message M, we have $|A_i| = |B_i| = 1$ and $|X_i| = m - 2$, for all $i \in \{1, \ldots, 4\}$. We set $X_1||A_1||B_1 \leftarrow M$, and "track" the ordering of these sections through the linear layers. Each time the cipher is called, the appropriate blocks of its output will be labelled by the same letter (and incremented index), meaning the ciphertext $C = A_4||B_4||X_4$. This labelling, and the logic described below, are represented in Figure 9, which shows how we convert from the Π_3^{right} scheme to a random function, a standard switch away from a random permutation.

The key observation behind the proof will be that B_2 does not repeat "too frequently", allowing us to perform a (tweakable) PRP–PRF 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 PRP, and measure the appropriate events.

To formalise this process, we will use an identical-until-bad argument, and split our proof into a series of claims. First we justify that single block queries do not assist the adversary (Claim 1). Then, we define a modified cipher (Figure 10) and demonstrate it identical-until-bad to one in the construction (Claim 2). Finally, we show that substituting this and applying Lemma 6 leads to a construction perfectly secure until bad is set (Claim 3) and measure the probability of bad (Claim 4). We will assume that when the adversary queries on a long message, he first queries on all prefixes of this message. This allows us to make statements about the freshness of inputs/outputs of the final block of a query, but leads to a loss of tightness when considering adversaries who do not make queries of many (if any) different lengths.

Claim 1. *Single Block queries do not assist the adversary.*

Proof. As the restriction of an online-cipher to a single block is a PRP and the three ciphers are independent, the randomness from the first block of $\mathcal{E}_1$ suffices to prove that the output from single block queries are perfectly indistinguishable from the outputs of a PRP. Since none of the following reasoning will make assertions about this distribution, the adversary learns nothing by making such queries. ■

Claim 2. $\widetilde{\mathcal{E}}_2$ *is identical to* $\mathcal{E}_2$ *until* bad*. Similarly,* $\widetilde{\mathcal{E}}_3$ *and* $\mathcal{E}_3$ *are identical until* bad*.*

Proof. The substitutions made in these two ciphers are very similar to those of Lemma 5, except we choose to track bad events using explicit lists rather than checking in the function tables. $\widetilde{\mathcal{E}}_2$ expands the online cipher call slightly, separating the final block and then applying a tweakable PRP-PRF switch. Until (B_2, A_2) repeats either the input or tweak is fresh, and the second check confirms that the output has not been chosen to contradict the permutation property. Thus until bad is set, this is identical to the online cipher $\mathcal{E}_2$.

That $\widetilde{\mathcal{E}}_3$ and $\mathcal{E}_3$ are identical-until-bad is given by Lemma 6, and repeated here for completeness.

The switch expands all but the first block of the online cipher call. We assume that (A_3, B_3) is new or trigger bad. If it is, then either the input or tweak is new for the encryption $B_4 \leftarrow \mathcal{E}^{A_3}(B_3)$, and we can switch this for a uniform sampler, as long as we confirm that the output (A_3, B_4) does not repeat. Since collisions on A_3 imply collisions on A_4, and so this second event is captured by bad_{out}. Moreover, the encryption of X_3 occurs under a new prefix, and is thus uniformly sampled. We do not require any bookkeeping for correctness since the bad events would be triggered before any previously defined value need be looked up in the permutation, other than the first block, which we pass through to the online cipher. ■

Claim 3. *Until* bad *is set,* Π *is identical to both the PRF defined by* $\tilde{\Pi}$ *and a PRP.*

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 $A_4||B_2||X_4$

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

function $\widetilde{\mathcal{E}}_3(A_3||B_3||X_3)$
$A_4 \leftarrow \mathcal{E}_3(A_3)$
if $(A_3, B_3) \in \mathcal{L}_C$ **then**
$\mathsf{bad}_C \leftarrow \mathtt{true}$
$B_4 \leftarrow\!\!\$\ \Sigma$
$X_4 \leftarrow\!\!\$\ \Sigma^{m-2}$
if $(A_4, B_4) \in \mathcal{L}_{out}$ **then**
$\mathsf{bad}_{out} \leftarrow \mathtt{true}$
return $A_4||B_4||X_4$

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

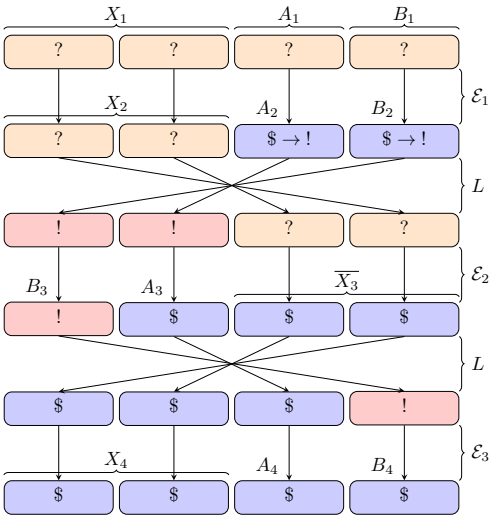
procedure CheckForBad(*)
if $(B_2, A_2) \in \mathcal{L}_A$ **then**
$\mathsf{bad}_A \leftarrow \mathtt{true}$
if $(B_2, A_3) \in \mathcal{L}_B$ **then**
$\mathsf{bad}_B \leftarrow \mathtt{true}$
if $(A_3, B_3) \in \mathcal{L}_C$ **then**
$\mathsf{bad}_C \leftarrow \mathtt{true}$
if $(A_4, B_4) \in \mathcal{L}_{out}$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

Figure 10: The various routines relevant to the proof of security for $\Pi = \Pi_3^{\mathsf{right}}$. Π is the code for the original scheme, and is identical to $\widetilde{\Pi}$ until procedure CheckForBad triggers bad. The two functions $\widetilde{\mathcal{E}}_2$ and $\widetilde{\mathcal{E}}_3$ are identical-until-bad to the online ciphers $\mathcal{E}_2$ and $\mathcal{E}_3$ respectively.

Proof. To move from Π to $\tilde{\Pi}$ one substitutes $\widetilde{\mathcal{E}}_2$ and $\widetilde{\mathcal{E}}_3$ in for $\mathcal{E}_2$ and $\mathcal{E}_3$ respectively, collecting all the bad events in the CheckForBad procedure. Thus until bad is set, these two routines are identical. Now, $\tilde{\Pi}$ is a PRF: it directly samples variables B_4 and X_4 uniformly, and output A_4 is the image of A_3 (itself sampled uniformly at random) under an independent permutation and thus also uniformly sampled. Moreover, as a PRF, it is indistinguishable from a PRP 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 $A_4||B_4$. ■

Claim 4. *The probability an adversary interacting with* $\tilde{\Pi}$ *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 cipher. By the same argument as Lemma 4, the probability of this 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. 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



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.

$ →! 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 11: Intuition behind the ±PRP security of Π_3^{rev} (Theorem 5)

first occurring. Finally then, 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,

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

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

5.2 Three layer reverse: ±PRP beyond the birthday bound

So, there exist three layer constructions that pass the birthday bound, and in this section we investigate whether the Π_3^{rev} construction is a ±PRP beyond 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 8), 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: all but the last of the the internal claims are true in both settings, leaving a the smaller problem for future work. Given this limitation has been overcome, 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 ciphers, around two calls of* rev*. Then the adversarial advantage in distinguishing Π_3 from a AIL ±PRP is bounded. For any adversary making $q \leq N/8$ queries,*

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

function $\widetilde{\mathcal{E}}_2^{\epsilon}(B_2||A_2||\overline{X}_2)$
$B_3 \leftarrow \mathcal{E}_2^{\epsilon}(B_2)$
$A_3 \leftarrow\!\!\$\ \Sigma$
$\overline{X}_3 \leftarrow\!\!\$\ \Sigma^{|M|-2}$
if $(B_2, A_2, \star) \in \mathcal{P}_2$ **then**
$\mathsf{bad}_B \leftarrow \mathtt{true}$
if $(B_2, \star, A_3) \in \mathcal{P}_2$ **then**
$\mathsf{bad}_{B'} \leftarrow \mathtt{true}$
$\mathcal{P}_2 \leftarrow\!\!\cup\ (B_2, A_2, A_3)$
return $B_3||A_3||\overline{X}_3$

function $\widetilde{\mathcal{D}}_2^{\epsilon}(B_3||A_3||\overline{X}_3)$
$B_2 \leftarrow \mathcal{D}_2^{\epsilon}(B_3)$
$A_2 \leftarrow\!\!\$\ \Sigma$
$\overline{X}_2 \leftarrow\!\!\$\ \Sigma^{|M|-2}$
if $(B_2, A_2, \star) \in \mathcal{P}_2$ **then**
$\mathsf{bad}_B \leftarrow \mathtt{true}$
if $(B_2, \star, A_3) \in \mathcal{P}_2$ **then**
$\mathsf{bad}_{B'} \leftarrow \mathtt{true}$
$\mathcal{P}_2 \leftarrow\!\!\cup\ (B_2, A_2, A_3)$
return $B_2||A_2||\overline{X}_2$

Figure 12: The online cipher $\widetilde{\mathcal{E}}_2$.

Proof. First we will describe a routine $\widetilde{\mathcal{E}}_2$ (Figure 12) and show it is identical-until-bad to the online cipher $\mathcal{E}_2$ (Claim 1). Then, substituting this in for $\mathcal{E}_2$ and using the substitions used in Lemma 6 to replace $\mathcal{E}_1$ and $\mathcal{E}_3$, we derive a construction (Figure 13) that is identical-until-bad to both Π_3^{rev} and an $\pm$PRP (Claim 2). This leaves us to bound the probability of bad, which we do in two sections (Claims 3 and 4) to complete the result.

The variable naming scheme and parsing will follow those given in Figure 11. The parsing of M is done such that for a message of length $m \geq 2$, $|X| = m - 2$,$|A| = |B| = 1$. The case where $|M| = 1$ is trivially secure, since the restriction of a secure online cipher to a single block is a secure $\pm$PRP.

Claim 1. *$\widetilde{\mathcal{E}}_2$ is identical-until-bad to $\mathcal{E}_2$.*

Proof. This is the result of applying Lemma 5 to both encryption and decryption. So, they are equivalent to $\mathcal{E}_2, \mathcal{D}_2$ until the adversary repeats (B_2, A_2) or (B_2, A_3), events we capture by checking whether the appropriate elements of $\mathcal{P}_2$ have been defined.

First, let us expand the encryption into three sections based on the online property: $B_3 \leftarrow \mathcal{E}_2^{\epsilon}(B_2)$, $A_3 \leftarrow \mathcal{E}_2^{B_2}(A_2)$, and $\overline{X}_3 \leftarrow \mathcal{E}_2^{B_2||A_2}(\overline{X}_2)$. Until $B_2||A_2$ repeats, the prefix encrypting $\overline{X}_2$ is always unique, and thus its output is independently and uniformly sampled from all strings of the appropriate length. Moreover, until either (B_2, A_2) or (B_2, A_3) repeat, we may replace the single-block call $\mathcal{E}_2^{B_2}(\cdot)$ with a uniform sampler (Lemma 3).

These three events are all captured by checking whether (B_2, A_2) or (B_2, A_3) has repeated, which trigger events bad_B or $\mathsf{bad}_{B'}$ respectively. We update the partial permutation $\mathcal{P}$ with the newly defined entry (B_2, A_2, A_3), but do not need to record the entry $(B_2||A_2, \overline{X}_2, \overline{X}_3)$ because bad_B would be set before the cipher would ever need to look this up.

The switches to the decryption routine are equivalent, and so these alternative algorithms are identical to $\mathcal{E}_2$ until bad. ■

Claim 2. *The pair of routines (Enc, Dec) given in Figure 13 are identical-until-bad to both the forward/backward routines of Π_3^{rev}, and to the interfaces of a $\pm$PRP.*

Proof. To convert from Π_3^{rev} to the construction given in the figure, one first switches $\mathcal{E}_2$ for $\widetilde{\mathcal{E}}_2$ as described in the previous claim. Then, switching out the encryption of $\mathcal{E}_3$ and decryption $\mathcal{D}_1$ for the equivalent versions of the same code. This yields the routines (Enc, Dec), which are thus identical-until-bad.

Now, by Lemma 6, (Enc, Dec) is an $\pm$PRP until bad. To repeat the logic of that result somewhat, this is because their output variables are uniformly sampled, either directly (as B_4, Y_4 in Enc and B_1, Y_1 in Dec), or sampled prior to being passed through an independent permutation (as in the case of A_4, X_4 in Enc or A_1, X_1 under Dec). Thus it is a $\pm$PRF,

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 $(B_2, A_2, \star) \in \mathcal{P}_2$ **then**
$\mathsf{bad}_B \leftarrow \mathtt{true}$
if $(B_2, \star, A_3) \in \mathcal{P}_2$ **then**
$\mathsf{bad}_{B'} \leftarrow \mathtt{true}$
$\mathcal{P}_2 \leftarrow_{\cup} (B_2, A_2, A_3)$
- - - - - - - - - - - - - - -
$X_4||A_4 \leftarrow \mathcal{E}_3^{\epsilon}(X_3||A_3)$
$B_4 \leftarrow_{\$} \Sigma$
if $(X_3||A_3, B_3, \star) \in \mathcal{P}_3$ **then**
$\mathsf{bad}_C \leftarrow \mathtt{true}$
if $(X_3||A_3, \star, B_4) \in \mathcal{P}_3$ **then**
$\mathsf{bad}_{C'} \leftarrow \mathtt{true}$
$\mathcal{P}_3 \leftarrow_{\cup} (X_3||A_3, B_3, B_4)$
- - - - - - - - - - - - - - -
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 $(B_2, A_2, \star) \in \mathcal{P}_2$ **then**
$\mathsf{bad}_B \leftarrow \mathtt{true}$
if $(B_2, \star, A_3) \in \mathcal{P}_2$ **then**
$\mathsf{bad}_{B'} \leftarrow \mathtt{true}$
$\mathcal{P}_2 \leftarrow_{\cup} (B_2, A_2, A_3)$
- - - - - - - - - - - - - - -
$X_1||A_1 \leftarrow \mathcal{D}_1^{\epsilon}(X_2||A_2)$
$B_1 \leftarrow_{\$} \Sigma$
if $(X_1||A_1, \star, B_2) \in \mathcal{P}_1$ **then**
$\mathsf{bad}_A \leftarrow \mathtt{true}$
if $(X_1||A_1, B_1, \star) \in \mathcal{P}_1$ **then**
$\mathsf{bad}_{A'} \leftarrow \mathtt{true}$
$\mathcal{P}_1 \leftarrow_{\cup} (X_1||A_1, B_1, B_2)$
- - - - - - - - - - - - - - -
return $X_1||A_1||B_1$

Figure 13: A construction identical-until-bad to Π_3^{rev}. The dashed lines are for guidance only, and separate the elements of the routine from each of the three ciphers. The parsing of M is done such that for a message of length m, $|X| = m - 2$ and $|A| = |B| = 1$.

and so identical to a $\pm$PRP until the output repeats, which does not happen without first setting $\mathsf{bad}_{C'}$ (on Enc) or $\mathsf{bad}_{A'}$ (on Dec). ■

Claim 3. *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 $\mathsf{bad}_C, \mathsf{bad}_{C'}$, and only decryption $\mathsf{bad}_A, \mathsf{bad}_{A'}$. Let us bound the probability of the i^{th} query setting $\mathsf{bad}_C \vee \mathsf{bad}_{C'}$, and split into two cases based on $|M|$.

Firstly, consider the case $|M| > 2$, and thus $|X| \geq 1$. Then, the probability of setting either of the bad events is at most that of the uniformly sampled string $X_3||A_3$ taking a value that $X_3||A_3$ has taken previously. This is upper bounded by $(i-1)/N^2$.

Alternatively, suppose $|M| = 2$ and thus $|X| = 0$. Then, the probability of setting $\mathsf{bad}_{C'}$ is $(i-1)/N^2$ by the same reasoning as above. 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) is not new, meaning $\mathsf{bad}_{B'}$ had already been set. Given the assumption on $\mathsf{bad}_{B'}$, this does not occur.

Thus in either case, the probability of bad is at most $(i-1)/N^2$. Applying the same logic to decryption queries, we bound the probability of setting bad_A or $\mathsf{bad}_{A'}$ on a decryption query by the same value. Since every query is either encryption or decryption, but not

both, the probability of any query triggering any of the four events is at most this value also, and applying the union bound proves the claim. ■

Claim 4. *Assuming $q < N/8$ and letting $n = \log N$, $\mathbb{P}\left[\mathsf{bad}_B \vee \mathsf{bad}_{B'}\right] \leq \left(\frac{5n}{7} + 1\right) q/N$.*

Proof. Assume the adversary is making their i^{th} query, and that neither event has yet occurred. Let A_1^j be the value A_1 took on the j^{th} query, and similarly for all other variables. Let $\mathcal{Q}_i$ be the set of all (A_2^j, A_3^j) where $j < i$ and $B_2^j = B_2^i$. 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

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

We now bound these terms for an encryption query, as decryption is equivalent. Immediately, as A_3 is sampled uniformly at random on every query we have $\mathbb{P}\left[(\star, A_3) \in \mathcal{Q}_i | \#\mathcal{Q}_i \leq \alpha\right] \leq \alpha/N$.

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

The adversary cannot detect a collision $A_2 = A_2^j$ 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 $X_1 \neq X_1^j$, 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 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 $\mathbb{P}\left[\#\mathcal{Q}_i > \alpha\right]$. It is maximised if each query is made with a different prefix, and so $\mathbb{P}\left[\#\mathcal{Q}_i > \alpha\right] \leq \left(\frac{i-1}{N}\right)^{\alpha+1} / (\alpha + 1)!$ by a standard collision-counting argument.

Assuming $q \leq N/\beta$ for some β, we can collect the three terms to yield

$$\mathbb{P}\left[\mathsf{bad}_B^i \vee \mathsf{bad}_{B'}^i \,|\, \wedge_{j<i} \neg(\mathsf{bad}_B^j \vee \mathsf{bad}_{B'}^j)\right] \leq \frac{\alpha}{N} + \frac{\beta\alpha}{(\beta - 1)N} + \frac{1}{\beta^{\alpha+1}(\alpha + 1)!}.$$

Then, applying the union bound to sum across all q queries gives us a parametrised closed form of the bound:

$$\mathbb{P}\left[\mathsf{bad}_B \vee \mathsf{bad}_{B'}\right] \leq \frac{(2\beta - 1)\alpha}{\beta - 1}\frac{q}{N} + \frac{1}{\beta^{\alpha+1}(\alpha + 1)!}q.$$

To simplify it further, let us define $n = \log N$, and choose $(\alpha, \beta) = (\lfloor n/3 \rfloor, 8)$. Then $\beta^{\alpha+1} > N$ and $(\alpha + 1)! \geq 1$, and so $\mathbb{P}\left[\mathsf{bad}_B \vee \mathsf{bad}_{B'}\right] \leq \frac{q}{N}\left(\frac{5n}{7} + 1\right)$, which is the bound claimed. ■

Finally then, it remains to combine the claims. Π_3^{rev} is identical-until-bad to a $\pm$PRP and to the pair (Enc, Dec). Then, we bound bad by

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

[3] It is 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 every having collisions of two elements, meaning $\alpha = 2$.

and substitute in the calculated bounds to complete the proof. Requiring $q \leq N/8$, either $q = 0$ (in which case any non-negative bound holds) or $n = \log N \geq 3$, so assume the second of these. Then, using the parametrized version of Claim 4 with $(\alpha, \beta) = (\lfloor n/3 \rfloor, 8)$,

$$\mathbf{Adv}^{\pm\mathsf{prp}}_{\Pi_3}(q) \leq \mathbb{P}\left[\mathsf{bad}\right] \leq \frac{q}{N}\left(\frac{5n}{7} + \frac{1}{2}\right) + \frac{q(q-1)}{2N^2} \leq \frac{q}{N}\left(\frac{5n}{7} + \frac{1}{2} + \frac{1}{16}\right) \leq n\frac{q}{N},$$

which is the claimed bound. □

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^{rev}_3 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, at the cost of requiring the ciphers be independent.

Lemma 9. *The Π^L_i construction is no more distinguishable from a PRP or ±PRP than Π^L_{i-1}.*

Proof. Suppose there exists some adversary $\mathcal{A}$ who distinguishes Π^L_i from a PRP (or ±PRP). Let us construct an adversary $\mathcal{B}$ who distinguishes Π^L_{i-1} from a PRP (or ±PRP).

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

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

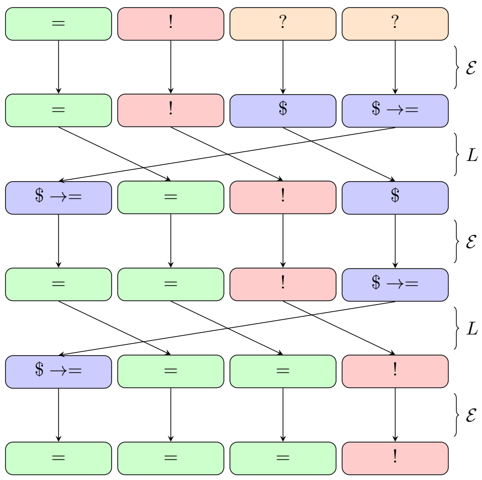
We observe that there exists an attack against the whole family of Π^{right}_i constructions. If an ideal online cipher is called with two messages that differ before the final block, the final ciphertext blocks are independently sampled. Now, if these independent random variables collide (which is likely to occur roughly every $\sqrt{N}$ queries) the right layer will simply add a common prefix to both messages. From this observation, we build a distinguisher against Π^{right}_i, following the logic shown in Figure 14.

Lemma 10. *Consider Π^{right}_i for some $i \geq 2$, and assume we may make queries of at least $a + c + i$ blocks, with c such that $N^c \geq 4$. Then for any $q \leq \min(N^a, N^{(i-1)/2})$, $\mathbf{Adv}^{\mathsf{prp}}_{\Pi}(q) \geq \frac{q(q-1)}{8N^{i-1}}$.*

Proof. Consider the adversary $\mathcal{A}$ 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 $\mathcal{A}$ successfully distinguishes Π from an ideal PRP.

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



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 14: An attack against Π_3^{right}.

the final block to the start, with this same probability there exist queries in which the first two blocks repeat. Since this output again consists of some repeated blocks, a unique section and then some arbitrary blocks, we may apply similar analysis. We do this for all but the final layer, albeit noting 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 $\mathcal{A} \to 1$. So, $\mathbb{P}\left[\mathcal{A}^{\Pi} \to 1\right]$ is at least this probability. Since the variables are independently sampled, this is equivalent to colliding a string of $(i - 1)$ independently sampled blocks, $\mathbb{P}\left[\mathcal{A}^{\Pi} \to 1\right] \geq \frac{q(q-1)}{4N^{i-1}}$ as long as $q \leq \sqrt{2N^{i-1}}$ (Lemma 1).

Alternatively, consider $\mathbb{P}\left[\mathcal{A}^{\pi} \to 1\right]$. 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 $\mathbb{P}\left[\mathcal{A}^{\pi} \to 1\right] \leq \frac{q(q-1)}{2N^{i+c}}$.

Combining these results,

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

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, $\epsilon \leq 2^{-127}$. □

7 Conclusion

We have shown how one can efficiently turn an online cipher in a fully fledged cipher, using two types of mixing layer. To build a PRP with birthday bound security, only two calls to the online cipher are required, with one suitable constructing using a linear layer that shifts blocks one step right. For close to blocksize security three calls to the online cipher are both necessary and sufficient, with right-shifting yielding a PRP and reversing achieving ±PRP 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.

7.0.1 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. [BBKN01]) is at its most general. The more specific definition of Rogaway et al. [RWZ12] 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. [BR00]), and even to achieve the recent, stronger goal of robust authenticated encryption [HKR15]. By instantiating these modes with our construction, one can build a very secure scheme from an online cipher.

7.0.2 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 [MRH04], 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 [DRST12] appear to extend to the PRP case of online ciphers (curiously, the ±PRP situation seems less straightforward). We provide a more detailed discussion in Appendix A.

From the CMC and EME constructions, it is clear that more involved mixing layers may reduce the security required of the cryptographic primitive. An interesting question is whether our work can be extended to show beyond birthday security of a 'CMCMC' or 'EMEME' like construction. 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 ±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 Bristol, supported by an EPSRC grant.

References

[ABL+13] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Elmar Tischhauser, and Kan Yasuda. Parallelizable and authenticated online ciphers. In Sako and Sarkar [SS13], pages 424–443.

[ABL+14] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. How to securely release unverified plaintext in authenticated encryption. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 105–125, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Springer, Heidelberg, Germany.

[ABPS15] Elena Andreeva, Guy Barwell, Dan Page, and Martijn Stam. Turning online ciphers off. Cryptology ePrint Archive, Report 2015/485, 2015. http://eprint.iacr.org/2015/485.

[AFF+15] Farzaneh Abed, Scott R. Fluhrer, Christian Forler, Eik List, Stefan Lucks, David A. McGrew, and Jakob Wenzel. Pipelineable on-line encryption. In Carlos Cid and Christian Rechberger, editors, *FSE 2014*, volume 8540 of *LNCS*, pages 205–223, London, UK, March 3–5, 2015. Springer, Heidelberg, Germany.

[BBKN01] Mihir Bellare, Alexandra Boldyreva, Lars R. Knudsen, and Chanathip Namprempre. Online ciphers and the hash-CBC construction. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 292–309, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany.

[BDPS14] Alexandra Boldyreva, Jean Paul Degabriele, Kenneth G. Paterson, and Martijn Stam. On symmetric encryption with distinguishable decryption failures. In Shiho Moriai, editor, *FSE 2013*, volume 8424 of *LNCS*, pages 367–390, Singapore, March 11–13, 2014. Springer, Heidelberg, Germany.

[Ber13] Daniel J. Bernstein. CAESAR competition call, 2013.

[BFSW13] Mike Bond, George French, Nigel P. Smart, and Gaven J. Watson. The low-call diet: Authenticated encryption for call counting HSM users. In Ed Dawson, editor, *CT-RSA 2013*, volume 7779 of *LNCS*, pages 359–374, San Francisco, CA, USA, February 25 – March 1, 2013. Springer, Heidelberg, Germany.

[BN15] Ritam Bhaumik and Mridul Nandi. Revisiting turning online cipher off. Cryptology ePrint Archive, Report 2015/813, 2015. http://eprint.iacr.org/2015/813.

[BPS15] Guy Barwell, Daniel Page, and Martijn Stam. Rogue decryption failures: Reconciling AE robustness notions. In Jens Groth, editor, *15th IMA International Conference on Cryptography and Coding*, volume 9496 of *LNCS*, pages 94–111, Oxford, UK, December 15–17, 2015. Springer, Heidelberg, Germany.

[BR99] Mihir Bellare and Phillip Rogaway. On the construction of variable-input-length ciphers. In Lars R. Knudsen, editor, *FSE'99*, volume 1636 of *LNCS*, pages 231–244, Rome, Italy, March 24–26, 1999. Springer, Heidelberg, Germany.

[BR00] Mihir Bellare and Phillip Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 317–330, Kyoto, Japan, December 3–7, 2000. Springer, Heidelberg, Germany.

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Vaudenay [Vau06], pages 409–426.

[BT04] Alexandra Boldyreva and Nut Taesombut. Online encryption schemes: New security notions and constructions. In Tatsuaki Okamoto, editor, *CT-RSA 2004*, volume 2964 of *LNCS*, pages 1–14, San Francisco, CA, USA, February 23–27, 2004. Springer, Heidelberg, Germany.

[Can12] Anne Canteaut, editor. *FSE 2012*, volume 7549 of *LNCS*, Washington, DC, USA, March 19–21, 2012. Springer, Heidelberg, Germany.

[DN14] Nilanjan Datta and Mridul Nandi. ELmE: A misuse resistant parallel authenticated encryption. In Willy Susilo and Yi Mu, editors, *ACISP 14*, volume 8544 of *LNCS*, pages 306–321, Wollongong, NSW, Australia, July 7–9, 2014. Springer, Heidelberg, Germany.

[DRST12] Yevgeniy Dodis, Thomas Ristenpart, John P. Steinberger, and Stefano Tessaro. To hash or not to hash again? (In)differentiability results for h^2 and HMAC. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 348–366, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany.

[Dwo04] Morris Dworkin. Request for review of key wrap algorithms. Cryptology ePrint Archive, Report 2004/340, 2004. http://eprint.iacr.org/2004/340.

[FFL12] Ewan Fleischmann, Christian Forler, and Stefan Lucks. McOE: A family of almost foolproof on-line authenticated encryption schemes. In Canteaut [Can12], pages 196–215.

[FJMV04] Pierre-Alain Fouque, Antoine Joux, Gwenaëlle Martinet, and Frédéric Valette. Authenticated on-line encryption. In Mitsuru Matsui and Robert J. Zuccherato, editors, *SAC 2003*, volume 3006 of *LNCS*, pages 145–159, Ottawa, Ontario, Canada, August 14–15, 2004. Springer, Heidelberg, Germany.

[FMP03] Pierre-Alain Fouque, Gwenaëlle Martinet, and Guillaume Poupard. Practical symmetric on-line encryption. In Thomas Johansson, editor, *FSE 2003*, volume 2887 of *LNCS*, pages 362–375, Lund, Sweden, February 24–26, 2003. Springer, Heidelberg, Germany.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[Hal07] Shai Halevi. Invertible universal hashing and the TET encryption mode. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 412–429, Santa Barbara, CA, USA, August 19–23, 2007. Springer, Heidelberg, Germany.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 15–44, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.

[HR03] Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 482–499, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany.

[JMV02] Antoine Joux, Gwenaëlle Martinet, and Frédéric Valette. Blockwise-adaptive attackers: Revisiting the (in)security of some provably secure encryption models: CBC, GEM, IACBC. In Yung [Yun02], pages 17–30.

[Jou06] Antoine Joux. Authentication failures in nist version of gcm. NIST Comment, 2006.

[KL08] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2008.

[KR11] Ted Krovetz and Phillip Rogaway. The software performance of authenticated-encryption modes. In Antoine Joux, editor, *FSE 2011*, volume 6733 of *LNCS*, pages 306–327, Lyngby, Denmark, February 13–16, 2011. Springer, Heidelberg, Germany.

[LRW02] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. In Yung [Yun02], pages 31–46.

[Min09] Kazuhiko Minematsu. Beyond-birthday-bound security based on tweakable block cipher. In Orr Dunkelman, editor, *FSE 2009*, volume 5665 of *LNCS*, pages 308–326, Leuven, Belgium, February 22–25, 2009. Springer, Heidelberg, Germany.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39, Cambridge, MA, USA, February 19–21, 2004. Springer, Heidelberg, Germany.

[Nan08] Mridul Nandi. Two new efficient CCA-secure online ciphers: MHCBC and MCBC. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *INDOCRYPT 2008*, volume 5365 of *LNCS*, pages 350–362, Kharagpur, India, December 14–17, 2008. Springer, Heidelberg, Germany.

[NRS14] Chanathip Namprempre, Phillip Rogaway, and Thomas Shrimpton. Reconsidering generic composition. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 257–274, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.

[Rog02] Phillip Rogaway. Authenticated-encryption with associated-data. In Vijayalakshmi Atluri, editor, *ACM CCS 02*, pages 98–107, Washington D.C., USA, November 18–22, 2002. ACM Press.

[RS06] Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Vaudenay [Vau06], pages 373–390.

[RWZ12] Phillip Rogaway, Mark Wooding, and Haibin Zhang. The security of ciphertext stealing. In Canteaut [Can12], pages 180–195.

[RZ11] Phillip Rogaway and Haibin Zhang. Online ciphers from tweakable blockciphers. In Aggelos Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 237–249, San Francisco, CA, USA, February 14–18, 2011. Springer, Heidelberg, Germany.

[SS13] Kazue Sako and Palash Sarkar, editors. *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, Bengalore, India, December 1–5, 2013. Springer, Heidelberg, Germany.

[ST13] Thomas Shrimpton and R. Seth Terashima. A modular framework for building variable-input-length tweakable ciphers. In Sako and Sarkar [SS13], pages 405–423.

[Vau06] Serge Vaudenay, editor. *EUROCRYPT 2006*, volume 4004 of *LNCS*, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany.

[Yun02] Moti Yung, editor. *CRYPTO 2002*, volume 2442 of *LNCS*, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Heidelberg, Germany.

A Impracticality of Indifferentiability

The security definitions given in Section 2.1.1 are the standard indistinguishability notions for symmetric primitives, but are less strong than the *indifferentiability* notions of Maurer et al. [MRH04]. 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 $\Delta^{\pi,S[\pi]}_{E[\mathcal{E}],\mathcal{E}}$, 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. [DRST12] 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, $\Delta^{\pi,S[\pi]}_{E[\mathcal{E}],\mathcal{E}}$ 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,

$$\Delta^{\pi\ ,\pi^{-1},S[\pi,\pi^{-1}],T[\pi,\pi^{-1}]}_{E[\mathcal{E}],D[\mathcal{D}],\quad\mathcal{E}\quad,\quad\mathcal{D}}$$

(corresponding to indifferentiability from an ideal cipher under the ±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 2.1.1. 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 $\mathcal{E}$ an online cipher acting on blocks Σ, all with keyspace $\mathcal{K}$. Then, the advantage of some adversary $\mathcal{A}$ against the security goals of the various objects are as follows:

$$\mathbf{Adv}_{E}^{\mathsf{prp}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow\!\!\$\, \mathcal{K} \,:\, \mathcal{A}^{E_k} \to 1\right] - \mathbb{P}\left[\pi \leftarrow\!\!\$\, \mathrm{Perm}(\Sigma^t) \,:\, \mathcal{A}^{\pi} \to 1\right]$$

$$\mathbf{Adv}_{E}^{\pm\mathsf{prp}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow\!\!\$\, \mathcal{K} \,:\, \mathcal{A}^{E_k, E_k^{-1}} \to 1\right] - \mathbb{P}\left[\pi \leftarrow\!\!\$\, \mathrm{Perm}(\Sigma^t) \,:\, \mathcal{A}^{\pi, \pi^{-1}} \to 1\right]$$

$$\mathbf{Adv}_{\tilde{E}}^{\mathsf{tprp}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow\!\!\$\, \mathcal{K} \,:\, \mathcal{A}^{\tilde{E}_k} \to 1\right] - \mathbb{P}\left[\tilde{\pi} \leftarrow\!\!\$\, \mathrm{Perm}(\mathcal{T}, \Sigma^t) \,:\, \mathcal{A}^{\tilde{\pi}} \to 1\right]$$

$$\mathbf{Adv}_{\tilde{E}}^{\pm\mathsf{tprp}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow\!\!\$\, \mathcal{K} \,:\, \mathcal{A}^{\tilde{E}_k, \tilde{E}_k^{-1}} \to 1\right] - \mathbb{P}\left[\tilde{\pi} \leftarrow\!\!\$\, \mathrm{Perm}(\mathcal{T}, \Sigma^t) \,:\, \mathcal{A}^{\tilde{\pi}, \tilde{\pi}^{-1}} \to 1\right]$$

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

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

$$\mathbf{Adv}_{F}^{\mathsf{prf}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow\!\!\$\, \mathcal{K} \,:\, \mathcal{A}^{F_k} \to 1\right] - \mathbb{P}\left[\$ \leftarrow\!\!\$\, \mathrm{Func}(\Sigma^t) \,:\, \mathcal{A}^{\$} \to 1\right]$$

$$\mathbf{Adv}_{F}^{\pm\mathsf{prf}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow\!\!\$\, \mathcal{K} \,:\, \mathcal{A}^{F_k, F_k^{-1}} \to 1\right] - \mathbb{P}\left[\$, \$^{-1} \leftarrow\!\!\$\, \mathrm{Func}(\Sigma^t) \,:\, \mathcal{A}^{\$, \$^{-1}} \to 1\right]$$

$$\mathbf{Adv}_{F}^{\pm\mathsf{tprf}}(\mathcal{A}) := \mathbb{P}\left[k \leftarrow\!\!\$\, \mathcal{K} \,:\, \mathcal{A}^{F_k, F_k^{-1}} \to 1\right] - \mathbb{P}\left[\$, \$^{-1} \leftarrow\!\!\$\, \mathrm{Func}(\mathcal{T}, \Sigma^t) \,:\, \mathcal{A}^{\$, \$^{-1}} \to 1\right]$$

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

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

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 ([ABPS15], version 20150521:200909) contained a flaw that was pointed out by Bhaumik and Nandi [BN15]. They provided an attack against the Π_2^{rev} scheme, demonstrating it cannot achieve ±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.