OleF: an Inverse-Free Online Cipher

An Online SPRP with an Optimal Inverse-Free Construction

Ritam Bhaumik and Mridul Nandi

Indian Statistical Institute, Kolkata
bhaumik.ritam@gmail.com,mridul.nandi@gmail.com

Abstract. Online ciphers, in spite of being insecure against an sprp adversary, can be desirable at places because of their ease of implementation and speed. Here we propose a single-keyed inverse-free construction that achieves online sprp security with an optimal number of blockcipher calls. We also include a partial block construction, without requiring any extra key.

Keywords: OleF · online cipher · blockcipher · inverse-free · diblock.

1 Introduction

Real Time Length-Preserving Encryption. We call an encryption scheme an enciphering scheme when it is length-preserving, i.e., when the length of the input matches the length of the output. Preserving length has uses in applications such as disk-sector encryption (as addressed by the IEEE SISWG P1619), where a length-preserving encryption preserves the file size after encryption. Other applications of enciphering schemes include bandwidth-efficient network protocols and security-retrofitting of old communication protocols.

Real time applications of enciphering schemes often find it convenient to use a low buffer size and process the data in one pass, or what we call online. A popular security notion for enciphering schemes is that of prp or pseudorandom permutation. However, prp security cannot be achieved in a single pass, and hence is not suitable to online ciphers, which easily admit prp distinguishing attacks.

Optimal Security for Online Ciphers. Bellare et al. [BBKN01] introduced the notion of online prp, the highest possible randomness in online ciphers. This notion tries to minimise the computational distance with the uniform distribution over all online ciphers. Certain popular online ciphers like CBC with fixed IV [BBKN01], ABC by Knudsen [KK00], etc. failed to be secure under this notion.

Two secure online ciphers have been are HCBC [BBKN01] (online prp) and HPCBC [Nan07] (online sprp), based on the AXU hash family [Rog02]. In these, the input lenghth of messages are variable multiples of n, blocksize of the underlying prp. More efficient constructions have been proposed in [Nan08] and later generalized, in the form of TC3 [RZ11], through a tweakable blockcipher.

Inverse-Free Designs. Another important design aspect relevant today in designing blockcipher-based encryption modes are inverse-free encryption modes, i.e., designs that rely solely on the encryption circuit of the blockcipher both while encrypting and decrypting, thus never needing to call its decryption circuit. These designs have numerous advantages, like requiring just a prf-secure blockcipher and a low footprint in a combined implementation. Encryption based on inverse-free primitive had been used earlier. Two-block or diblock

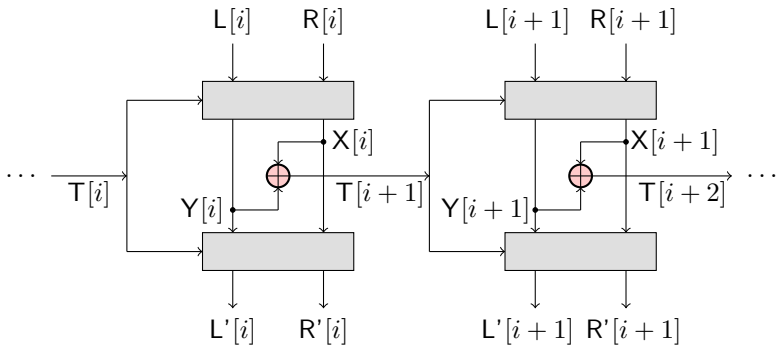

Figure 1: OleF: A Schematic View

Feistel design was used in early block ciphers like Lucifer [Fei74][Sor84] and DES [Sta77]. Luby and Rackoff gave a security proof of Feistel ciphers [LR88], and later the design was generalised to obtain inverse-free enciphering of longer messages [Nyb96].

1.1 Our Contributions

We propose an optimal inverse-free construction that achieves online sprp security (in a slightly modified form). In other words, in analogy to the notion of sprp security, an adversary with access to both encryption and decryption oracles cannot effectively distinguish it from a random online permutation.

Schematically, OleF consists of a sequential encryption layer, followed by a mixing layer, and another sequential encryption layer, so it follows the encrypt-mix-encrypt paradigm.

Advantages. Our construction has several advantages over its predecessors:

- Being inverse-free, this construction has two advantages:
  - A combined implementation of encryption and decryption keeps the footprint low;
  - Some blockciphers (like AES [Pub01]) have faster encryption than decryption, so and even individual implementation of OleF decryption can be faster than it would be if it needed AES decryption;
  - Underlying block cipher only needs prf-security, instead of sprp-security. This is believed to be achievable in a smaller number of rounds in some standard constructions like AES.

- Being online, this is easier to implement (due to a low buffer size) and also performs better.

- We believe this is an optimal inverse-free online sprp construction, in terms of the number of calls to the underlying prf. Earlier constructions like TC3 [RZ11] and MCBC [Nan08], when implemented as inverse-free, require more calls per diblock, as shown in the table. (An earlier version of McOE [FFLW11] had a variant called McOE-D which was similar to TC3.) For comparison, we also include the inverse-free full (offline) ciphers AEZ [HKR15] and FMix [BN].

- Partial blocks are handled with the same key, instead of an independent key, as seen in previous designs.

Table 1: f-calls per diblock for constructions in inverse-free implementations.

Construction	f-calls	Online?
MCBC	7	Yes
TC3	6	Yes
AEZ	5	No
FMix	4	No
OleF	4	Yes

Table 1 compares calls to f per diblock for various constructions in inverse-free implementations. For blockcipher based construction requiring the inverse, we count the number of calls after replacing the blockcipher (over a dibock) by a four-round Luby-Rackoff construction. This generic conversion, however, costs additional keys as four round Luby-Rackoff construction requires at least two keys.

2 Preliminaries

2.1 Basic Notions

Deterministic Encryption Schemes. Formally, with a deterministic encryption scheme $\mathfrak{E}$, we associate three finite spaces: the message space $\mathcal{M}$, the ciphertext space $\mathcal{C}$, and the key space $\mathcal{K}$. $\mathfrak{E}$ consists of two deterministic functions: $\mathfrak{e} : \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{C}$ and $\mathfrak{d} : \mathcal{K} \times \mathcal{C} \longrightarrow \mathcal{M}$, such that for any $k \in \mathcal{K}, m \in \mathcal{M}$, we have

$$\mathfrak{d}(k, \mathfrak{e}(k, m)) = m.$$

Since any finite set can be encoded in binary, we'll assume $\mathcal{K}, \mathcal{M}$ and $\mathcal{C}$ to be sets of binary strings. In addition, there is a key distribution μ_k (usually uniform over the key-space). For a set S and a distribution μ over S, we'll write $x \xleftarrow{\mu} S$ to denote x is sampled from S according to distribution μ. When μ is uniform, we also write $x \xleftarrow{\$} S$.

Blockcipher-based Inverse-Free Constructions. Formally, a blockcipher $e_K : \{0,1\}^n \longrightarrow \{0,1\}^n$ is a small-domain fixed-length deterministic encryption scheme. n is called the block-size, and is usually 128 or 256. Elements of $\{0,1\}^n$ are called blocks. In a wide class of encryption schemes and other constructions, the only non-linear components are blockciphers. We'll call such a construction *inverse-free* if both the encryption and the decryption use computations of e_K (for possibly different values of K) but never need to compute its inverse e_K^{-1}. In this case, one can also replace the blockcipher by a length-preserving keyed function, not necessarily invertible.

Random Functions. Let Func_n be the set of all block functions, i.e., functions from $\{0,1\}^n$ to $\{0,1\}^n$. When we call f a random function, we assume

$$f \xleftarrow{\$} \mathsf{Func}_n.$$

In our security proof, we will replace the blockcipher by a random function, using the hybrid argument. This will let us work in a purely information-theoretic scenario. Thus, the only source of randomness in our construction will be the random functions, and *all probabilities will be calculated based on this randomness.*

2.2 Security Games

Simple Distinguishing Game. Consider two classes of functions $\mathscr{C}_0$ and $\mathscr{C}_1$, with distributions μ_0 and μ_1 defined over them. In a *simple distinguishing game*, the oracles $\mathcal{O}_0$ and $\mathcal{O}_1$ behave as follows: for $b \in \{0,1\}$, $\mathcal{O}_b$ samples $f_b \xleftarrow{\mu_b} \mathscr{C}_b$ and simulates it. The challenger chooses a $b \in \{0,1\}$, and the adversary $\mathcal{A}$ is allowed to make q queries $x^1, \ldots, x^q$ to $\mathcal{O}_b$, which simply returns $f_b(x^1), \ldots, f_b(x^q)$, respectively. Note that $\mathcal{A}$ is allowed to make these queries adaptively, i.e., for $2 \leq i \leq q$, he can choose x^i after observing $f_b(x^1), ..., f_b(x^{i-1})$. Additionally, at the end of the queries, $\mathcal{A}$ may receive some additional information from $\mathcal{O}_b$, which can be some hidden part of the computation or something sampled randomly according to a known distribution. $\mathcal{A}$ finally returns a $b' \in \{0,1\}$, and wins if $b' = b$.

Strong Distinguishing Game. Suppose $\mathscr{C}_0$ and $\mathscr{C}_1$ are classes of permutations, i.e., invertible functions. As before, for $b \in \{0,1\}$, $\mathcal{O}_b$ samples a permutation $\pi_b \xleftarrow{\mu_b} \mathscr{C}_b$, and challenger chooses a $b \in \{0,1\}$. Now, however, each query of $\mathcal{A}$ is of the form (x^i, δ^i), where $\delta^i \in \{e, d\}$ represents the direction of the query. When $\delta^i = e$, i.e., in an encryption query, $\mathcal{O}_b$ returns $\pi_b(x^i)$ as before; when $\delta^i = d$, i.e., in a decryption query, $\mathcal{O}_b$ returns $\pi_b^{-1}(x^i)$. Again, at the end, $\mathcal{A}$ may receive additional information. As before, $\mathcal{A}$ returns $b' \in \{0,1\}$, and wins if $b = b'$. We also denote these oracles by $\mathcal{O}_b^{\pm}$ to mean that it can response both forward and inverse queries.

We'll use the notation $\Pr_b[\cdot]$ to denote probability under oracle $\mathcal{O}_b$.

Pointless Adversary. An adversary $\mathcal{A}$ will be called pointless if he makes a *pointless query*, i.e., a query whose response he already knows. In a simple distinguishing game, a pointless query is simply a repeated query, i.e., a query x^i such that $x^i = x^{i'}$ for some $i' < i$. In a strong distinguishing game, a pointless query can take two additional forms:

1. (y^i, d), such that for some $i' < i$ the query $(x^{i'}, e)$ returned y^i.
2. (x^i, e), such that for some $i' < i$ the query $(y^{i'}, d)$ returned x^i.

In our analysis we'll exclude pointless adversaries, because they can be trivially superseded by non-pointless ones. Since we will work in an information-theoretic setup, replacing all pointless queries by arbitrary non-pointless ones always gives the adversary more information.

Distinguishing Advantage. For an adversary $\mathcal{A}$ interacting with oracles either $\mathcal{O}_0$ or $\mathcal{O}_1$, we define the distinguishing advantage of $\mathcal{A}$ as

$$\mathbf{Adv}_{\mathcal{O}_1}^{\mathcal{O}_0}(\mathcal{A}) = \left|\Pr_0\left[\mathcal{A}^{\mathcal{O}_0} \text{ returns } 1\right] - \Pr_1\left[\mathcal{A}^{\mathcal{O}_1} \text{ returns } 1\right]\right|.$$

If the distinguishing advantage is bounded above by ϵ for any adversary $\mathcal{A}$, we say $\mathcal{O}_0$ and $\mathcal{O}_1$ are $(1-\epsilon)$-indistinguishable. When ϵ is "negligible", we'll often simply call $\mathcal{O}_0$ and $\mathcal{O}_1$ indistinguishable. When $\mathcal{O}_0^{\pm}$ and $\mathcal{O}_1^{\pm}$ are indistinguishable, we say $\mathcal{O}_0$ and $\mathcal{O}_1$ are $(1-\epsilon)$-strong-indistinguishable.

Pseudorandomness. Consider a class $\mathscr{C}$ of functions from $\mathcal{D}$ to $\mathcal{R}$, and a key space $\mathcal{K}$. Consider the keyed family of functions $\mathscr{F} = \{f_k | k \in \mathcal{K}\} \subset \mathscr{C}$. We say $\mathscr{F}$ is pseudorandom (or strong pseudorandom) in $\mathscr{C}$ if for $f \xleftarrow{\$} \mathscr{C}$ and $k \xleftarrow{\$} \mathcal{K}$, oracles $\mathcal{O}_0$ simulating f (called the ideal oracle) and $\mathcal{O}_1$ simulating f_k (called the real oracle) are indistinguishable (or strong-indistinguishable respectively).

Blockciphers and the Switching Lemma. Suppose $\mathscr{F}$ is pseudorandom in $\mathscr{C}$. When $\mathscr{C} = \mathsf{Func}_n$, $\mathscr{F}$ is called an n-bit pseudorandom function (prf). When $\mathscr{C} = \mathsf{Perm}_n$, the set of all permutations from $\{0,1\}^n$ to $\{0,1\}^n$, $\mathscr{F}$ is called an n-bit pseudorandom permutation (prp). A blockcipher will be assumed to be an n-bit prp. It is a well-known result that for any family $\mathscr{F}$, the prf distinguishing advantage of any adversary $\mathcal{A}$ cannot exceed its prp distinguishing advantage by more than $\frac{\sigma^2}{2^n}$, σ being the total number of query blocks. This result, known as the *prp-prf switching lemma* [BR04], lets us replace each different blockcipher by a prf. The distinguishing advantage between a prf and a truly random function will be considered negligible, allowing us to model the blockcipher as truly random functions, as mentioned before.

2.3 Diblock-Online Security

The Diblock-Online Property. Fix a block-size n bits. For any binary string x, let $||x||$ denote the number of bits in x, and for $1 \leq r \leq ||x||$, let $x_{1..r}$ denote the r-bit prefix of x. A permutation $\pi : \{0,1\}^{\geq 2n} \longrightarrow \{0,1\}^{\geq 2n}$ is said to be length-preserving if for any x in the domain of π,

$$||\pi(x)|| = ||x||.$$

We will call a length-preserving permutation *diblock-online* if for any $x \in \{0,1\}^l, y \in \{0,1\}^{l'}$ with $l \leq l'$, for any r such that $1 \leq 2nr \leq l - 2n$, we have

$$x_{1..2nr} = y_{1..2nr} \implies \pi(x)_{1..2nr} = \pi(y)_{1..2nr}.$$

In other words, the first r diblocks of $\pi(m)$ depend only on the first r diblocks of m, provided m has at least $r+1$ complete diblocks. (When there's an incomplete diblock at the end, the last complete diblock may violate this property.)

Note that this property imposes no restrictions on the last $2n + s$ bits of $\pi(x)$, where s is the remainder when $||x||$ is divided by $2n$. Here our definition deviates slightly from the classical notions of online permutations, which insists that $\pi(x)$ is a prefix of $\pi(y)$ whenever x is a prefix of y.

An encryption scheme $\mathfrak{E} = (\mathfrak{e}, \mathfrak{d})$ is said to be (diblock-online) length-preserving if for any $k \in \mathcal{K}$ the permutation $\mathfrak{e}(k, \cdot)$ is (diblock-online) length-preserving.

Diblock-Online Strong Pseudorandom Permutations. Take a diblock-online encryption scheme $\mathfrak{E} = (\mathfrak{e}, \mathfrak{d})$. Let $\mathscr{F}$ be the family $\{\mathfrak{e}_k = \mathfrak{e}(k, \cdot) | k \in \mathcal{K}\}$. $\mathscr{F}$ is called a diblock-online strong pseudorandom permutation (dosprp) if it is strong pseudorandom in the class of all diblock-online permutations. (Note that the class of such permutations is actually infinite, making uniform sampling meaningless, so we assume the length of the inputs do not exceed $2n\mathfrak{l}$, i.e., the number of diblocks cannot exceed some fixed cap $\mathfrak{l}$.) In such cases, we shall call the encryption scheme $\mathfrak{E}$ dosprp-secure.

2.4 Patarin's Technique

Interpolation Probability Fix a set $\mathbf{Q} = \{x^1, ..., x^q\}$ of queries and a set $\mathbf{R} = \{y^1, ..., y^q\}$ of responses, obtained from an oracle $\mathcal{O}$. (Note that here x^i represents the entire query, so for example in a strong distinguishing game, x^i will include the query direction as well. This is a departure from the notation used before, where x^i was just the query string.) Let $\mathbf{S}$ be the additional information revealed to $\mathcal{A}$. We call $\tau = (\mathbf{Q}, \mathbf{R}, \mathbf{S})$ a transcript of the game. Based on the randomness in $\mathcal{O}$, there will be a certain probability of obtaining $\mathbf{R}$ on querying with $\mathbf{Q}$. We call this probability the interpolation probability $\Pr_{\mathcal{O}}[\tau]$.

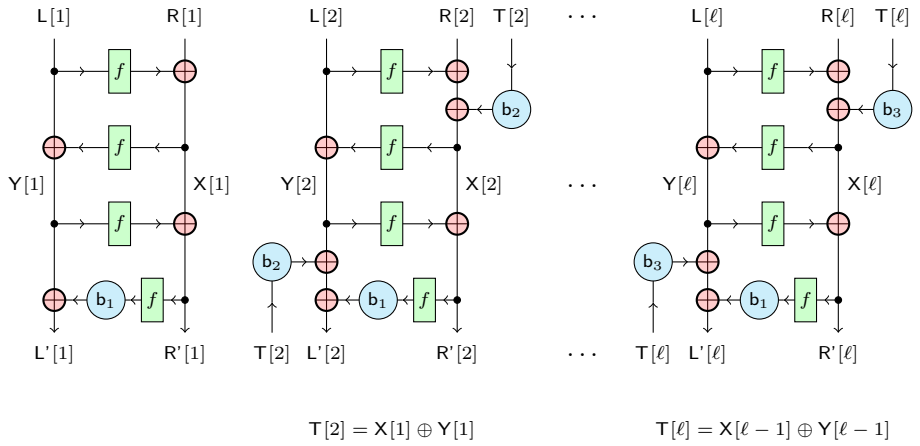

Figure 2: OleF for ℓ Complete Diblocks

Patarin's Coefficient H Technique Consider two oracles $\mathcal{O}_0$ and $\mathcal{O}_1$. Suppose for a set $\mathcal{V}_g$ of views (called the **good views**) the following hold:

1. For any adversary $\mathcal{A}$ playing against $\mathcal{O}_0$, the probability of getting a good transcript is at least $1 - \epsilon_1$;
2. For any good transcript τ, we have

$$\frac{\Pr_1[\tau]}{\Pr_0[\tau]} \geq 1 - \epsilon_2.$$

Theorem 1 (Coefficient H Technique)**.** *For oracles $\mathcal{O}_0$ and $\mathcal{O}_1$ satisfying (1) and (2) above, for any adversary $\mathcal{A}$, we have*

$$\mathbf{Adv}^{\mathcal{O}_0}_{\mathcal{O}_1}(\mathcal{A}) \leq \epsilon_1 + \epsilon_2.$$

This technique is due to Jacques Patarin [Pat09], and will be used in our security proofs.

3 The OleF Construction

3.1 Construction Details

Our construction takes as input one or more diblocks. When there are more than one diblocks, the last one maybe partial. (An incomplete diblock may consist of one complete block and one partial block, only one complete block or only one partial block.) Each plaintext diblock is processed using four Feistel rounds and one tweak that is obtained from processing the previous diblocks. After outputting the corresponding ciphertext diblock, a new tweak is passed on to the next diblock. The construction details are illustrated in the figure.

Handling Partial Blocks. If the last diblock is partial, it may require up to six calls to f to handle it, as shown in the figure 3. pad is an injective pad (like 10^*), and chop simply chops bits off the end of a full block till its size matches that of the partial block. (In the figure, we use the conventional trapezium nodes to denote pad and chop).

input : ℓ plaintext diblocks $(\mathsf{L}[1], \mathsf{R}[1]), ..., (\mathsf{L}[\ell], \mathsf{R}[\ell])$
output : ℓ ciphertext diblocks $(\mathsf{L}'[1], \mathsf{R}'[1]), ..., (\mathsf{L}'[\ell], \mathsf{R}'[\ell])$

begin
  $\mathsf{T}[1] \leftarrow 0$
  if $|R[\ell]| = n$ **then**
    for $j \leftarrow 1$ **to** ℓ **do**
      $(\mathsf{T}[j+1], \mathsf{L}'[j], \mathsf{R}'[j]) \leftarrow \mathtt{process}(j, \mathsf{T}[j], \mathsf{L}[j], \mathsf{R}[j])$
    end for
  else
    for $j \leftarrow 1$ **to** $\ell - 2$ **do**
      $(\mathsf{T}[j+1], \mathsf{L}'[j], \mathsf{R}'[j]) \leftarrow \mathtt{process}(j, \mathsf{T}[j], \mathsf{L}[j], \mathsf{R}[j])$
    end for
    if $|\mathsf{L}[\ell]| < n$ **then**
      $\mathsf{Z} \leftarrow f(\mathtt{pad}\ (\mathsf{L}[\ell]))$
      $(\mathsf{T}[\ell], \mathsf{L}'[\ell-1], \mathsf{R}''[\ell-1]) \leftarrow \mathtt{process}(\mathsf{T}[\ell-1], \mathsf{L}[\ell-1] \oplus \mathsf{Z}, \mathsf{R}[\ell-1])$
      $\mathsf{L}'[\ell] \leftarrow \mathtt{chop}\ (f(\mathsf{T}[\ell]) \oplus \mathsf{L}[\ell])$
      $\mathsf{R}'[\ell-1] \leftarrow \mathsf{R}''[\ell-1] \oplus f(\mathtt{pad}\ (\mathsf{L}'[\ell]))$
    else
      $\mathsf{Z} \leftarrow f(\mathsf{L}[\ell]) \oplus f(\mathtt{pad}\ (\mathsf{R}[\ell])$
      $(\mathsf{T}[\ell], \mathsf{L}'[\ell-1], \mathsf{R}''[\ell-1]) \leftarrow \mathtt{process}(\mathsf{T}[\ell-1], \mathsf{L}[\ell-1] \oplus \mathsf{Z}, \mathsf{R}[\ell-1])$
      $\mathsf{L}'[\ell] \leftarrow f(\mathsf{T}[\ell]) \oplus \mathsf{L}[\ell]$
      $\mathsf{R}'[\ell] \leftarrow \mathtt{chop}\ (f(\mathsf{T}[\ell]) \oplus \mathsf{R}[\ell] \oplus 1)$
      $\mathsf{R}'[\ell-1] \leftarrow \mathsf{R}''[\ell-1] \oplus f(\mathsf{L}[\ell]) \oplus f(\mathtt{pad}\ (\mathsf{R}'[\ell])$
    end if
  end if
end

Module process
input : $j, \mathsf{T}, (\mathsf{L}, \mathsf{R})$
output : $\mathsf{T}^{\mathsf{NEXT}}, (\mathsf{L}', \mathsf{R}')$

begin
  if $j = \ell$ **then**
    $\mathsf{bT} \leftarrow \mathsf{b}_3(\mathsf{T})$
  else
    $\mathsf{bT} \leftarrow \mathsf{b}_2(\mathsf{T})$
  end if
  $\mathsf{X} \leftarrow f(\mathsf{L}) \oplus \mathsf{R} \oplus \mathsf{bT}$
  $\mathsf{Y} \leftarrow f(\mathsf{X}) \oplus \mathsf{L}$
  $\mathsf{R}' \leftarrow f(\mathsf{Y}) \oplus \mathsf{X}$
  $\mathsf{L}' \leftarrow \mathsf{b}_1(f(\mathsf{R}')) \oplus \mathsf{Y} \oplus \mathsf{bT}$
  $\mathsf{T}^{\mathsf{NEXT}} \leftarrow \mathsf{X} \oplus \mathsf{Y}$
end

Algorithm 1: The OIeF encryption algorithm for $\ell - 1$ complete diblocks and one possibly incomplete diblock, where a single block has n bits

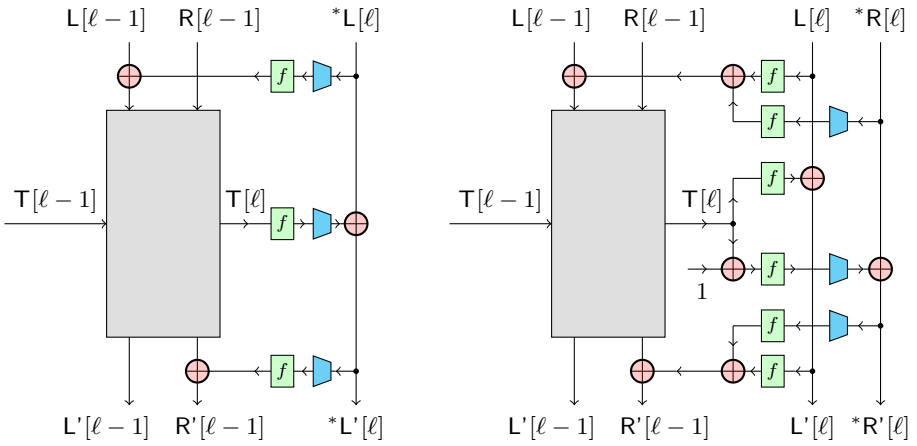

Figure 3: OleF for Partial Diblocks, where (a) $^*\mathsf{L}[\ell]$ has less than n bits, (b) $^*\mathsf{R}[\ell]$ has less than n bits

Balanced Linear Permutation. A permutation b over the set of n bits, is called a **balanced linear permutation**, if for $t \xleftarrow{\$} \{0,1\}^n$, each of $\mathsf{b}(t)$ and $t + \mathsf{b}(t)$ is uniformly distributed. A simple example of balanced linear permutation is $\mathsf{b}(t) = \alpha \cdot t$ where α is a non-zero non-1 constant and '$\cdot$' is the finite field multiplication. A software-friendly choice of b could be defined by $(y \| (x \oplus y)$ where $t := x \| y$ and x, y are $n/2$ bits. In this construction we use three balanced linear permutations b_1, b_2 and b_3.

3.2 Design Rationale

Injecting the Tweak. The first design question we consider is at which points in the processing of a diblock we add the tweak T obtained from its prefix. A tempting choice could be to add T to either strand midway, after two f calls have been completed, because this would make the top encrypt layer fully parallel. However, as shown in [HR04], sprp security can never be achieved in an ECB-linear mix-ECB mode, so this choice does not work. Instead we add T in the middle of the top layer (after one f call) and of the bottom layer (after three f calls). This allows some parallelisability of the top encrypt layer, without turning it into an ECB, thus avoiding the pitfall mentioned above.

Choice of T. A more involved question is how we generate a tweak T from a diblock. We could pick a public function of the view, such as $\mathsf{T}[i+1] = \mathsf{L}[i] + \mathsf{R}[i] + \mathsf{L}'[i] + \mathsf{R}'[i]$, so that the adversary can directly get T from the view. This, however, leads to a simple prp attack:

1. Make a query with first two diblocks $(\mathsf{L}[1], \mathsf{R}[1])$ and $(\mathsf{L}[2], \mathsf{R}[2])$. Find tweak T that comes from the first diblock.
2. Make a query with first diblock $(\mathsf{L}[1], \mathsf{R}[3])$. Find new tweak T' and calculate the difference $\Delta\mathsf{T} = \mathsf{T} + \mathsf{T}'$.
3. Make a query with first two diblocks $(\mathsf{L}[1], \mathsf{R}[3])$ and $(\mathsf{L}[2], \mathsf{R}[2] + \Delta\mathsf{T})$. Then the right half of the second output diblock matches the corresponding block from the first query.

This problem remains for a choice like $\mathsf{T}[i+1] = \mathsf{X}[i] + \mathsf{L}'[i] + \mathsf{R}'[i]$, because even though the adversary can no longer obtain T, he can still observe $\Delta\mathsf{T}$ after the first two queries above, by fixing the first input block which is the only source of randomness, and thus the attack still works. Hence, it becomes necessary that in the computation of T, randomness

is provided both by the plaintext diblock and the ciphertext diblock, because then the adversary cannot control both simultaneously, and differential attacks like the above no longer work. A simple and obvious choice satisfying this is $\mathsf{X} + \mathsf{Y}$.

Need for $\mathsf{b_1}$. We make a call b_1 after the last call to f in each diblock. This is necessary in order to break the symmetry, because otherwise each diblock computation becomes a 4-round Luby-Rackoff with a symmetric key sequence[Nan10], which is known to be insecure. A simple prp attack:

1. Make a query with first diblock (L, R). Note first diblock $(\mathsf{L}', \mathsf{R}')$ of output.
2. Make a query with first diblock $(\mathsf{R}', \mathsf{L}')$. The first diblock of the output will be (R, L).

Choice of $\mathsf{b_1}$. Suppose we choose a permutation b_1 that is not balanced. Assume for $t \xleftarrow{\$} \{0,1\}^n$, $t + \mathsf{b}_1(t)$ takes a value x with probability $p > \frac{1}{2^n}$. Then the following attack distinguishes it with advantage p:

1. Make a query with first diblock (L, R). Note first diblock $(\mathsf{L}', \mathsf{R}')$ of output.
2. Make a query with first diblock $(\mathsf{R}', \mathsf{L}' + x)$. The right half of the first diblock of the output will be L with probability at least p.

Thus, it is important to choose a balanced b_1.

Why Diblocks? Classically, when one talks of online ciphers, one considers a blockwise online property. However, since we're attempting an inverse-free construction, and there's no known inverse-free mode to encrypt a single block (which would be necessary in a blockwise online cipher), we settle for a diblock-online cipher instead.

4 Setup and Security Game

In this section we develop some notation we will use for our security analysis, and describe in detail the security game, in preparation of the security results we will present in Section 5

4.1 Some Notation and Definitions

Let $(\mathsf{L}^i[j], \mathsf{R}^i[j])$ and $(\mathsf{L}'^i[j], \mathsf{R}'^i[j])$ denote the j-th plaintext diblock and the j-th ciphertext diblock of the i-th query respectively. Let ℓ^i denote the number of diblocks in the i-th query, and let $\delta^i \in \{e, d\}$ denote the direction of the i-th query. Let $E = \{i \mid \delta^i = e\}$, and $D = \{i \mid \delta^i = d\}$. We assume the dictionary order on the set $\mathcal{I} = \{(i,j) \mid 1 \le i \le q, 1 \le j \le \ell^i\}$. $\sigma = |\mathcal{I}|$ will denote the total number of query diblocks. $N = 2^n$ will denote the number of possible values for each block, where n is the blocksize.

Internal Input and Output Blocks. The inputs and outputs of the internal f-calls will be called the internal input and output blocks. We introduce and summarise here some notation that'll be useful for describing these internal blocks. For $j < \ell^i$, we recall that

$$\mathsf{X}^i[j] = f(\mathsf{L}^i[j]) + \mathsf{R}^i[j] + \mathsf{b}_2(\mathsf{T}^i[j]),$$
$$\mathsf{Y}^i[j] = \mathsf{L}'^i[j] + \mathsf{b}_1(f(\mathsf{R}'^i[j])) + \mathsf{b}_2(\mathsf{T}^i[j]).$$

We further recall that

$$\mathsf{X}^{\ell^i}[j] = f(\mathsf{L}^{\ell^i}[j]) + \mathsf{R}^{\ell^i}[j] + \mathsf{b}_3(\mathsf{T}^{\ell^i}[j]),$$
$$\mathsf{Y}^{\ell^i}[j] = \mathsf{L}'^{\ell^i}[j] + \mathsf{b}_1(f(\mathsf{R}^{\ell^i}[j])) + \mathsf{b}_3(\mathsf{T}^i[j]).$$

Table 2: Summary of Internal Input and Output Blocks

f Call	Input Block	Output Block
First	$\mathsf{L}^i[j]$	$\mathsf{R}^i[j] + \mathsf{X}^i[j] + \mathsf{b}_2(\mathsf{T}^i[j])$
Second	$\mathsf{X}^i[j]$	$\mathsf{U}^i[j]$
Third	$\mathsf{Y}^i[j]$	$\mathsf{V}^i[j]$
Fourth	$\mathsf{R}'^i[j]$	$\mathsf{b}_1^{-1}(\mathsf{L}'^i[j] + \mathsf{Y}^i[j] + \mathsf{b}_2(\mathsf{T}^i[j]))$

Here $\mathsf{T}^i[1]$ is taken to be 0 and for $j \geq 2$,

$$\mathsf{T}^i[j] = \mathsf{X}^i[j-1] + \mathsf{Y}^i[j-1].$$

For any $(i, j) \in \mathcal{I}$, let $f(\mathsf{X}^i[j])$ be called $\mathsf{U}^i[j]$ and $f(\mathsf{Y}^i[j])$ be called $\mathsf{V}^i[j]$. We note that

$$\mathsf{Y}^i[j] = \mathsf{L}^i[j] + \mathsf{U}^i[j],$$
$$\mathsf{X}^i[j] = \mathsf{R}'^i[j] + \mathsf{V}^i[j].$$

The notation is summarised in Table 2. Note that all internal input and output blocks are linear functions $\mathsf{X}^i[j]$'s, $\mathsf{Y}^i[j]$'s and the plaintext and ciphertext blocks.

Prefixes. Let $\mathsf{P}^i[j]$ and $\mathsf{P}'^i[j]$ denote the j-diblock prefixes of the plaintext and ciphertext respectively in the i-th query. We shall call $i \in E$ **encryption j-fresh** if $j = \ell^i$, or if $j < \ell^i$ and for no $i' < i$ we have $\mathsf{P}^i[j] = \mathsf{P}^{i'}[j]$, i.e., the plaintext prefix $\mathsf{P}^i[j]$ is distinct from all previous plaintext prefixes. Similarly, we shall call $i \in D$ **decryption j-fresh** if $j = \ell^i$, or if $j < \ell^i$ and for no $i' < i$ we have $\mathsf{P}'^i[j] = \mathsf{P}'^{i'}[j]$ i.e., the ciphertext prefix $\mathsf{P}'^i[j]$ is distinct from all previous plaintext prefixes. (A plaintext or ciphertext is considered distinct as a prefix from proper prefixes of other plaintexts or ciphertexts, even if they match block-to-block. This is because we process the last diblock differently.) Note that for every distinct plaintext prefix, we have a corresponding (i, j) pair such that i is encryption j-fresh, and for every distinct ciphertext prefix, we have a corresponding (i, j) pair such that i is decryption j-fresh.

We call a value x (i, j)**-new** if

$$x \notin \left\{ \mathsf{L}^{i'}[j'], \mathsf{R}'^{i'}[j'] \mid (i', j') < (i, j) \right\}.$$

In other words, a value is (i, j)-new when it has not occurred before as an input to the outer-layer f-calls. Note that the set

$$\mathcal{D} := \left\{ \mathsf{L}^i[j] \mid \mathsf{L}^i[j] \text{ is } (i,j)\text{-new} \right\} \cup \left\{ \mathsf{R}'^i[j] \mid \mathsf{R}'^i[j] \text{ is } (i,j)\text{-new} \right\}$$

is precisely the set of distinct values among the $\mathsf{L}^i[j]$'s and the $\mathsf{R}'^i[j]$'s, i.e.,

$$\mathcal{D} = \left\{ \mathsf{L}^i[j], \mathsf{R}'^i[j] \mid (i, j) \in \mathcal{I} \right\}.$$

4.2 Oracle Behaviour and Bad Events

We're now ready to describe how the two oracles behave in our security game. We recall that the real oracle $\mathcal{O}_1$ interacts with the adversary using our construction with a random function f, while the ideal oracle simulates a random diblock-online permutation. In addition, at the end of the query phase, each oracle reveals some additional information for the adversary. We describe their behaviour next.

Real Oracle. The real oracle $\mathcal{O}_1$ samples a random function f. It uses f in the construction to answer each adversary query, according to Algorithm 1. In the end, it also reveals all the internal input and output blocks as described above.

Ideal Oracle. The ideal oracle $\mathcal{O}_0$ has two stages of sampling. It initially samples a random diblock-online permutation π. To each encryption query x it returns $\pi(x)$ and to each decryption query y it returns $\pi^{-1}(y)$. We say the event badA has occurred if any of the following happens:

- For some $i \in \{1, \ldots, q\}$, $j \leq \ell^i$,
$$\mathsf{L}^i[j] = \mathsf{R'}^i[j];$$
- For some $i \in E$, $j \leq \ell^i$ with i encryption j-fresh, $\mathsf{R'}^i[j]$ is not (i,j)-new;
- For some $i \in D$, $j \leq \ell^i$ with i decryption j-fresh, $\mathsf{L}^i[j]$ is not (i,j)-new.

If $\mathcal{O}_0$ does not encounter badA, it proceeds to the second stage, where it samples values for a set $\mathcal{B}$ described below, and uses these values to simulate the internal input and output blocks, which it then reveals to the adversary. Before making this formal, we describe $\mathcal{B}$ and its special property.

Basis. Recall the set
$$\mathcal{D} = \left\{\mathsf{L}^i[j], \mathsf{R'}^i[j] \mid (i,j) \in \mathcal{I}\right\}$$
of distinct values that occur as $\mathsf{L}^i[j]$ or $\mathsf{R'}^i[j]$. For $x \in \mathcal{D}$, let $(i(x), j(x))$ denote the first occurrence index of a shortest prefix where x occurs as $\mathsf{L}^i[j]$ or $\mathsf{R'}^i[j]$, i.e., $x = \mathsf{L}^{i(x)}[j(x)]$ or $x = \mathsf{R'}^{i(x)}[j(x)]$, and
$$x \notin \left\{\mathsf{L}^{i'}[j'], \mathsf{R'}^{i'}[j'] \mid (j',i') < (j(x), i(x))\right\}.$$
Note that this is similar to the definition of (i,j)-new, except we've reversed the order of j and i. We're now looking at the earliest among all shortest prefixes, instead of the shortest among all earliest ones.

We write $\mathcal{D}$ as $\mathcal{D}_1 \cup \mathcal{D}_2$, where
$$\mathcal{D}_1 := \left\{x \in \mathcal{D} \mid x = \mathsf{L}^{i(x)}[j(x)]\right\},$$
$$\mathcal{D}_2 := \left\{x \in \mathcal{D} \mid x = \mathsf{R'}^{i(x)}[j(x)]\right\}.$$
Note that unless badA has occurred,
$$\mathcal{D}_1 \cap \mathcal{D}_2 = \phi.$$
Next, we identify two sets of blocks $\mathcal{B}_1$ and $\mathcal{B}_2$ as
$$\mathcal{B}_1 := \left\{\mathsf{V}^{i(x)}[j(x)] \mid x \in \mathcal{D}_1\right\},$$
$$\mathcal{B}_2 := \left\{\mathsf{U}^{i(x)}[j(x)] \mid x \in \mathcal{D}_2\right\}.$$
Finally, we define $\mathcal{B}$ as
$$\mathcal{B} := \mathcal{B}_1 \cup \mathcal{B}_2.$$
We call $\mathcal{B}$ the **basis**, for reasons to be soon explained. It immediately follows that
$$|\mathcal{B}| = |\mathcal{D}|.$$

Extending a Basis. Suppose we have fixed values for all blocks in the basis $\mathcal{B}$. We'll now show that this, along with the plaintext and ciphertext blocks, uniquely determines all internal input and output blocks. For this it is enough to determine the $\mathsf{X}^i[j]$'s and the $\mathsf{Y}^i[j]$'s. We only consider the $\mathsf{X}^i[j]$'s; it is easy to see the same analysis also applies to the $\mathsf{Y}^i[j]$'s with little modification.

$\mathsf{V}^1[1]$ is always in $\mathcal{B}$, so we get $\mathsf{X}^1[1] = \mathsf{R}^1[1] + \mathsf{V}^1[1]$. Suppose for some $i \geq 2$ we've determined the values of $\mathsf{X}^{i'}[1]$ and $\mathsf{Y}^{i'}[1]$ for all $i' < i$. (Note that this means we've also obtained $\mathsf{T}^{i'}[1]$ for all $i' \leq i$.) We consider three cases:

- if $i(\mathsf{L}^i[1]) = i$, $\mathsf{V}^i[1] \in \mathcal{B}$, so we get
$$\mathsf{X}^i[1] = \mathsf{R}'^i[1] + \mathsf{V}^i[1];$$
- if $\mathsf{L}^i[1] = \mathsf{L}^{i'}[1]$ for some $j' < j$, then
$$\mathsf{X}^i[1] = \mathsf{X}^{i'}[1] + \mathsf{R}^{i'}[1] + \mathsf{T}^{i'}[1] + \mathsf{R}^i[1] + \mathsf{T}^i[1];$$
- if $\mathsf{L}^i[1] = \mathsf{R}'^{i'}[1]$ for some $j' < j$, then
$$\mathsf{X}^i[1] = \mathsf{b}_1^{-1}(\mathsf{Y}^{i'}[1] + \mathsf{L}'^{i'}[1] + \mathsf{T}^{i'}[1]) + \mathsf{R}^i[1] + \mathsf{T}^i[1].$$

Let $\mathsf{b}[ij](x)$ denote $\mathsf{b}_3(x)$ when $j = \ell^i$ and $\mathsf{b}_2(x)$ otherwise. Thus, the (i,j)-th diblock gets as tweak $\mathsf{b}[ij](\mathsf{T}^i[j])$. Suppose for some $i \geq 2$ and for some $j \leq l^i$, we've already determined the values of $\mathsf{X}^{i'}[j']$ and $\mathsf{Y}^{i'}[j']$ for all $(j', i') < (j, i)$. We again consider three cases:

- if $i(\mathsf{L}^i[j]) = i$ and $j(\mathsf{L}^i[j]) = j$, $\mathsf{V}^i[j] \in \mathcal{B}$, so we get
$$\mathsf{X}^i[j] = \mathsf{R}'^i[j] + \mathsf{V}^i[j];$$
- if $\mathsf{L}^i[j] = \mathsf{L}^{i'}[j']$ for some $(j', i') < (j, i)$, then
$$\mathsf{X}^i[j] = \mathsf{X}^{i'}[j'] + \mathsf{R}^{i'}[j'] + \mathsf{b}[i'j'](\mathsf{T}^{i'}[j']) + \mathsf{R}^i[j] + \mathsf{b}[ij](\mathsf{T}^i[j]);$$
- if $\mathsf{L}^i[j] = \mathsf{R}'^{i'}[j']$ for some $(j', i') < (j, i)$, then
$$\mathsf{X}^i[j] = \mathsf{b}_1^{-1}(\mathsf{Y}^{i'}[j'] + \mathsf{L}'^{i'}[j'] + \mathsf{b}[i'j'](\mathsf{T}^{i'}[j'])) + \mathsf{R}^i[j] + \mathsf{b}[ij](\mathsf{T}^i[j]).$$

This completes the extension. The equations used for obtaining the value of an $\mathsf{X}^i[j]$ or a $\mathsf{Y}^i[j]$ from the basis or from previously determined values will be called its extension equations.

Sampling through the Basis. Now we formally describe the second stage of sampling by $\mathcal{O}_0$. For each $x \in \mathcal{B}$, $\mathcal{O}_0$ assigns a value through uniform with-replacement sampling. It then uses the basis extension equations above to determine the values of the internal input and output blocks, which it then reveals to the adversary. In doing so, it may encounter the event badB, which happens when we have an accidental collision on a pair of internal input blocks. We next describe what we mean by this.

Real Oracle $\mathcal{O}_1$	Ideal Oracle $\mathcal{O}_0$
1 : $f \xleftarrow{\$} \mathsf{Func}_n$	1 : $\pi \xleftarrow{\$} \mathsf{DOPerm}$
2 : use $\mathsf{OleF}[f]$ to answer queries	2 : use π to answer queries
3 : reveal internal input and output blocks	3 : if badA, classify transcript as bad
	4 : for each $x \in \mathcal{B}, x \xleftarrow{\$} \{0,1\}^n$
	5 : simulate internal input and output blocks
	6 : if badB, classify transcript as bad

Figure 4: Oracle Behaviour

Accidental Collisions. Let $\mathcal{F}$ be the first-occurrence indices of the distinct plaintext prefixes, and $\mathcal{F}'$ be the first-occurrence indices of the distinct ciphertext prefixes, i.e.,

$$\mathcal{F} = \Big\{(i,j) \mid (\nexists i' < i)(\mathsf{P}^{i'}[j'] = \mathsf{P}^{i}[j])\Big\} \cup \mathcal{L},$$
$$\mathcal{F}' = \Big\{(i,j) \mid (\nexists i' < i)(\mathsf{P'}^{i'}[j'] = \mathsf{P'}^{i}[j])\Big\} \cup \mathcal{L},$$

where $\mathcal{L} = \big\{(i, \ell^i) \mid i \in \{1, ..., q\}\big\}$. If any of the following input collisions is observed, it will be called accidental:

- For some $(i_1, j_1), (i_2, j_2) \in \mathcal{F}$ with $(i_1, j_1) \neq (i_2, j_2)$,
$$\mathsf{X}^{i_1}[j_1] = \mathsf{X}^{i_2}[j_2];$$
- For some $(i_1, j_1), (i_2, j_2) \in \mathcal{F}'$ with $(i_1, j_1) \neq (i_2, j_2)$,
$$\mathsf{Y}^{i_1}[j_1] = \mathsf{Y}^{i_2}[j_2];$$
- For some $(i_1, j_1) \in \mathcal{F}, (i_2, j_2) \in \mathcal{F}'$,
$$\mathsf{X}^{i_1}[j_1] = \mathsf{Y}^{i_2}[j_2];$$
- For some $(i_1, j_1) \in \mathcal{F}, (i_2, j_2) \in \mathcal{I}$,
$$\mathsf{X}^{i_1}[j_1] = \mathsf{L}^{i_2}[j_2];$$
- For some $(i_1, j_1) \in \mathcal{I}, (i_2, j_2) \in \mathcal{F}'$,
$$\mathsf{R'}^{i_1}[j_1] = \mathsf{Y}^{i_2}[j_2].$$

Good Transcript. The plaintexts and ciphertexts along with the internal input and output blocks revealed at the end constitute the transcript for this game. We say the transcript is bad if in sampling it the ideal oracle $\mathcal{O}_0$ encounters either of the events badA and badB. Note that whether a bad event has occurred can be seen from the transcript itself, so we don't really need to consider the temporal aspect of the game in order to classify a transcript. The oracle behaviour is summarised in Figure 4.

5 Security Results

In this section we provide the security analysis of our construction defined over all messages having ending in a complete diblock. The proof for the partial blocks case is similar.

Theorem 2. *Suppose an adversary $\mathcal{A}$ interacting against* OleF *instantiated with some block-function f in the real world, or an ideal random diblock-online permutation in the ideal world, makes at most q queries, each consisting only of complete diblocks, and having altogether σ diblocks. Then there is a prf-adversary $\mathcal{A}'$ against f making at most σ queries such that*

$$\mathbf{Adv}^{\mathbf{dosprp}}_{\mathit{OleF}[f]}(\mathcal{A}) \leq \mathbf{Adv}^{\mathbf{prf}}_{f}(\mathcal{A}') + \frac{7\sigma^2}{2^n}.$$

For a perfect blockcipher e_K, prp-prf switching lemma tells us that

$$\mathbf{Adv}^{\mathbf{prf}}_{e_K}(\mathcal{A}') \leq \frac{\sigma^2}{2^n},$$

so when we instantiate OleFwith e_K, we have the following security bound:

$$\mathbf{Adv}^{\mathbf{dosprp}}_{\mathsf{OleF}[e_K]}(\mathcal{A}) \leq \frac{8\sigma^2}{2^n}.$$

The rest of the section is to devote to a proof of Theorem 2. Using standard hybrid reduction, we can replace the keyed function f by a true random function at the cost of $\mathbf{Adv}^{\mathbf{prf}}_{f}(A')$. Then we apply coefficient H technique (as mentioned in Section 2) to obtain the rest of the bound. So from now onwards we assume f is a true random function.

In order to apply the Coefficient H Technique, we first examine the probability of a bad transcript in a game with the ideal oracle $\mathcal{O}_0$. We look at the two bad events one by one, beginning with badA.

5.1 Probability of badA

Suppose i is encryption j-fresh. Then $\mathsf{P}^i[j]$ is a fresh plaintext prefix. For a random diblock-online permutation π, the output will thus have almost full entropy in the final diblock, the only restriction being that if for some $i' < i$ we have $\mathsf{P}'^{i'}[j-1] = \mathsf{P}'^{i}[j-1]$, then $(\mathsf{L}'^{i}[j], \mathsf{R}'^{i}[j])$ cannot be equal to $(\mathsf{L}'^{i'}[j], \mathsf{R}'^{i'}[j])$. This rules out at most $\frac{q}{N^2}$ choices for the diblock, so for any value x,

$$\Pr\left[\mathsf{R}'^{i}[j] = x\right] \leq \frac{1}{N \cdot \left(1 - \frac{q}{N^2}\right)} = \frac{1}{N - \frac{q}{N}} \leq \frac{1}{N-1}.$$

Since there are at most 2σ distinct plaintext and ciphertext prefixes, badA corresponds to a collision on a set of size at most 2σ. Thus,

$$\Pr_0\left[\mathsf{badA}\right] \leq \frac{\binom{2\sigma}{2}}{N-1} \leq \frac{2\sigma^2}{N}.$$

5.2 Probability of badB

For this we assume that the first stage of sampling has been carried out, so the values of all plaintext blocks and ciphertext blocks are fixed, and the event badA has not been encountered. Suppose all elements of $\mathcal{B}$ have been uniformly sampled, and the internal input and output blocks generated by extending the basis as expounded in Section 4. A careful case-by-case analysis (see Appendix A) gives us the following bounds:

- For any $(i_1, j_1), (i_2, j_2) \in \mathcal{F}$ with $(i_1, j_1) \neq (i_2, j_2)$, $\Pr_0\left[\mathsf{X}^{i_1}[j_1] = \mathsf{X}^{i_2}[j_2]\right] \leq \frac{1}{N}$;
- For any $(i_1, j_1), (i_2, j_2) \in \mathcal{F}'$ with $(i_1, j_1) \neq (i_2, j_2)$, $\Pr_0\left[\mathsf{Y}^{i_1}[j_1] = \mathsf{Y}^{i_2}[j_2]\right] \leq \frac{1}{N}$;

- For any $(i_1, j_1) \in \mathcal{F}, (i_2, j_2) \in \mathcal{F}'$, $\Pr_0\left[\mathsf{X}^{i_1}[j_1] = \mathsf{Y}^{i_2}[j_2]\right] \leq \frac{1}{N}$;
- For any $(i_1, j_1) \in \mathcal{F}, (i_2, j_2) \in \mathcal{I}$, $\Pr_0\left[\mathsf{X}^{i_1}[j_1] = \mathsf{L}^{i_2}[j_2]\right] = \frac{1}{N}$;
- For any $(i_1, j_1) \in \mathcal{I}, (i_2, j_2) \in \mathcal{F}'$, $\Pr_0\left[\mathsf{R}'^{i_1}[j_1] = \mathsf{Y}^{i_2}[j_2]\right] = \frac{1}{N}$.

We know that $|\mathcal{F}|$ and $|\mathcal{F}'|$ are both bounded by σ. Thus, number of pairs over $\mathcal{F}$ and $\mathcal{F}'$ is at most $2\sigma^2$. We further know that $|\mathcal{I}|$ is also bounded by σ. Thus we conclude that

$$\Pr_0\left[\mathsf{badB} \mid \text{not } \mathsf{badA}\right] \leq \frac{4\sigma^2}{N}.$$

Together with the result from the previous subsection, this implies that

$$\Pr_0\left[\mathsf{badB} \text{ or } \mathsf{badA}\right] \leq \frac{6\sigma^2}{N}.$$

5.3 Interpolation Probability of a Bad Transcript

In a good transcript τ, for the real oracle $\mathcal{O}_1$, there are exactly $|\mathcal{D}| + |\mathcal{F}| + |\mathcal{F}'|$ distinct inputs to f. Thus there are $N^{N-(|\mathcal{D}|+|\mathcal{F}|+|\mathcal{F}'|)}$ choices for f that lead to τ. Since $\mathcal{O}_1$ samples a random f and then works with $\mathsf{OIeF}[f]$, and the total number of choices for f is N^N, we have

$$\Pr_1\left[\tau\right] \geq N^{-(|\mathcal{D}|+|\mathcal{F}|+|\mathcal{F}'|)}.$$

For the ideal oracle $\mathcal{O}_0$, the $\mathcal{D}$ basis elements are chosen randomly, and the $|\mathcal{F}| + |\mathcal{F}'|$ plaintext and ciphertext blocks are chosen through a random diblock-online permutation π. Thus,

$$\Pr_0\left[\tau\right] \leq N^{-(|\mathcal{D}|+|\mathcal{F}|+|\mathcal{F}'|)} \cdot \frac{1}{1 - \frac{\sigma^2}{N}},$$

since $|\mathcal{F}| + |\mathcal{F}'| \leq \sigma$. Thus,

$$\frac{\Pr_1\left[\tau\right]}{\Pr_0\left[\tau\right]} \geq 1 - \frac{\sigma^2}{N}.$$

5.4 Wrapping up the proof

Let $\epsilon_1 := \frac{6\sigma^2}{N}$ and $\epsilon_2 := \frac{\sigma^2}{N}$. Then we have shown that

$$\Pr_0\left[\text{transcript is bad}\right] \leq \epsilon_1,$$

and when τ is a good transcript,

$$\frac{\Pr_1\left[\tau\right]}{\Pr_0\left[\tau\right]} \geq 1 - \epsilon_2.$$

From Theorem 1, the Coefficient H Technique, we have

$$\mathbf{Adv}^{\mathbf{doprp}^{\pm}}_{\mathsf{OIeF}[f]}(\mathcal{A}) - \mathbf{Adv}^{\mathbf{prf}}_{f}(\mathcal{A}') \leq \epsilon_1 + \epsilon_2.$$

This completes a proof of Theorem 2.

References

[BBKN01] Mihir Bellare, Alexandra Boldyreva, Lars Knudsen, and Chanathip Namprempre. Online ciphers and the hash-cbc construction. In Joe Kilian, editor, *Advances in Cryptology âĂŤ CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 292–309. Springer Berlin Heidelberg, 2001.

[BN] Ritam Bhaumik and Mridul Nandi. An inverse-free single keyed tweakable enciphering scheme. (due to appear in proceedings of *ASIACRYPT 2015*).

[BR04] Mihir Bellare and Phil Rogaway. The game-playing technique. 2004.

[Fei74] H. Feistel. Block cipher cryptographic system, March 19 1974. US Patent 3,798,359.

[FFLW11] Ewan Fleischmann, Christian Forler, Stefan Lucks, and Jakob Wenzel. Mcoe: A family of almost foolproof on-line authenticated encryption schemes. Cryptology ePrint Archive, Report 2011/644, 2011. http://eprint.iacr.org/.

[HKR15] VietTung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption aez and the problem that it solves. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 15–44. Springer Berlin Heidelberg, 2015.

[HR04] Shai Halevi and Phillip Rogaway. A parallelizable enciphering mode. In Tatsuaki Okamoto, editor, *Topics in Cryptology âĂŞ CT-RSA 2004*, volume 2964 of *Lecture Notes in Computer Science*, pages 292–304. Springer Berlin Heidelberg, 2004.

[KK00] Lars R. Knudsen and Lars R. Knudsen. Block chaining modes of operation, 2000.

[LR88] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, April 1988.

[Nan07] Mridul Nandi. A simple security analysis of hash-cbc and a new efficient one-key online cipher, 2007.

[Nan08] Mridul Nandi. Two new efficient cca-secure online ciphers: Mhcbc and mcbc. Cryptology ePrint Archive, Report 2008/401, 2008. http://eprint.iacr.org/.

[Nan10] Mridul Nandi. The characterization of luby-rackoff and its optimum single-key variants. In Guang Gong and KishanChand Gupta, editors, *Progress in Cryptology - INDOCRYPT 2010*, volume 6498 of *Lecture Notes in Computer Science*, pages 82–97. Springer Berlin Heidelberg, 2010.

[Nyb96] Kaisa Nyberg. Generalized feistel networks. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology âĂŤ ASIACRYPT '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 91–104. Springer Berlin Heidelberg, 1996.

[Pat09] Jacques Patarin. The âĂIJcoefficients hâĂİ technique. In RobertoMaria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography*, volume 5381 of *Lecture Notes in Computer Science*, pages 328–345. Springer Berlin Heidelberg, 2009.

[Pub01] NIST FIPS Pub. 197: Advanced encryption standard (aes). *Federal Information Processing Standards Publication*, 197:441–0311, 2001.

[Rog02] Phillip Rogaway. Authenticated-encryption with associated-data. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, pages 98–107, New York, NY, USA, 2002. ACM.

[RZ11] Phillip Rogaway and Haibin Zhang. Online ciphers from tweakable blockciphers. In Aggelos Kiayias, editor, *Topics in Cryptology âĂŞ CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 237–249. Springer Berlin Heidelberg, 2011.

[Sor84] Arthur Sorkin. Lucifer, a cryptographic algorithm. *Cryptologia*, 8(1):22–42, 1984.

[Sta77] Data Encryption Standard. Fips pub 46. *Appendix A, Federal Information Processing Standards Publication*, 1977.

Appendices

A A Detailed Analysis of badB

A.1 Notation

As mentioned in the previous section, we assume that the first stage of sampling has been carried out, so the values of all plaintext blocks and ciphertext blocks are fixed, and the event badA has not been encountered. Recall that in Section 4, we introduced the notation $(i(x), j(x))$ for some $x \in \mathcal{D}$ to denote its first-occurrence index. We introduce some further notation to simplify the subsequent analysis. For $i \in E$, let $\tilde{i}$ denote $i(\mathsf{L}^i[j])$ and $\tilde{j}$ denote $j(\mathsf{L}^i[j])$, i.e., $(\tilde{i}, \tilde{j})$ is the first-occurrence index of $\mathsf{L}^i[j]$. (Note that $(i(\mathsf{R'}^i[j]), j(\mathsf{R'}^i[j]))$ is simply (i, j), unless badA has occurred.) Simlarly, for $i \in D$, let $\tilde{i}$ denote $i(\mathsf{R'}^i[j])$ and $\tilde{j}$ denote $j(\mathsf{R'}^i[j])$, while $(i(\mathsf{L}^i[j]), j(\mathsf{L}^i[j]))$ is simply (i, j), unless badA has occurred. Note that using this notation, the basis can be redefined as

$$\mathcal{B}_1 := \left\{\mathsf{V}^i[j] \mid i \in E, (i, j) = (\tilde{i}, \tilde{j})\right\},$$
$$\mathcal{B}_2 := \left\{\mathsf{U}^i[j] \mid i \in D, (i, j) = (\tilde{i}, \tilde{j})\right\}.$$

Suppose all elements of $\mathcal{B}$ have been uniformly sampled, and the internal input and output blocks generated by extending the basis as expounded in Section 4. In the subsequent analysis, we shade basis elements to simplify notation.

A.2 badB: Type 1.

Let $(i_1, j_1), (i_2, j_2) \in \mathcal{F}$ with $(j_1, i_1) > (j_2, i_2)$. We will show that

$$\Pr_0\left[\mathsf{X}^{i_1}[j_1] = \mathsf{X}^{i_2}[j_2]\right] \leq \frac{1}{N}.$$

We recall that

$$\mathsf{X}^{i_1}[j_1] = \mathsf{V}^{i_1}[j_1] + \mathsf{R'}^{i_1}[j_1].$$

Thus, when $\mathsf{X}^{i_1}[j_1] = \mathsf{V}^{i_1}[j_1] + \mathsf{R'}^{i_1}[j_1]$, since $\mathsf{V}^{i_1}[j_1]$ cannot affect $\mathsf{X}^{i_2}[j_2]$, we are done. Thus we assume $\mathsf{V}^{i_1}[j_1] \notin \mathcal{B}$. Next we recall that, when $\mathsf{L}^{i_1}[j_1] = \mathsf{L}^{\tilde{i_1}}[\tilde{j_1}]$, we have

$$\mathsf{X}^{i_1}[j_1] = \mathsf{X}^{\tilde{i_1}}[\tilde{j_1}] + \mathsf{R}^{\tilde{i_1}}[\tilde{j_1}] + \mathsf{b}[\tilde{i_1}\tilde{j_1}](\mathsf{T}^{\tilde{i_1}}[\tilde{j_1}]) + \mathsf{R}^{i_1}[j_1] + \mathsf{b}[i_1 j_1](\mathsf{T}^{i_1}[j_1]),$$

and when $\mathsf{L}^{i_1}[j_1] = \mathsf{R'}^{\tilde{i}_1}[\tilde{j}_1]$, we have

$$\mathsf{X}^{i_1}[j_1] = \mathsf{b}_1^{-1}(\mathsf{Y}^{\tilde{i}_1}[\tilde{j}_1] + \mathsf{L'}^{\tilde{i}_1}[\tilde{j}_1] + \mathsf{b}[\tilde{i}_1\tilde{j}_1](\mathsf{T}^{\tilde{i}_1}[\tilde{j}_1])) + \mathsf{R}^{i_1}[j_1] + \mathsf{b}[i_1j_1](\mathsf{T}^{i_1}[j_1]).$$

For any (i, j) such that $(i, j) \neq (\tilde{i}, \tilde{j})$ define

$$\begin{aligned} \alpha_{ij} &:= \mathsf{X}^{\tilde{i}}[\tilde{j}] + \mathsf{R}^{\tilde{i}}[\tilde{j}] + \mathsf{b}[\tilde{i}\tilde{j}](\mathsf{T}^{\tilde{i}}[\tilde{j}]) + \mathsf{R}^{i}[j] && \text{when } \mathsf{L}^{i}[j] = \mathsf{L}^{\tilde{i}}[\tilde{j}], \\ &:= \mathsf{b}_1^{-1}(\mathsf{Y}^{\tilde{i}}[\tilde{j}] + \mathsf{L'}^{\tilde{i}}[\tilde{j}] + \mathsf{b}[\tilde{i}\tilde{j}](\mathsf{T}^{\tilde{i}}[\tilde{j}])) + \mathsf{R}^{i}[j] && \text{when } \mathsf{L}^{i}[j] = \mathsf{R'}^{\tilde{i}}[\tilde{j}]. \end{aligned}$$

Note that α_{ij} is well-defined unless badA has occurred. Thus we can write

$$\begin{aligned} \mathsf{X}^{i_1}[j_1] &= \alpha_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{T}^{i_1}[j_1]) \\ &= \alpha_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{X}^{i_1}[j_1-1]) + \mathsf{b}[i_1j_1](\mathsf{Y}^{i_1}[j_1-1]) \\ &= \alpha_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{V}^{i_1}[j_1-1]) + \mathsf{b}[i_1j_1](\mathsf{U}^{i_1}[j_1-1]) \\ &\qquad + \mathsf{b}[i_1j_1](\mathsf{R'}^{i_1}[j_1-1] + \mathsf{L}^{i_1}[j_1-1]). \end{aligned}$$

As a final piece of notation, we define

$$\beta_{ij} := \alpha_{ij} + \mathsf{b}[ij](\mathsf{R'}^{i}[j-1] + \mathsf{L}^{i}[j-1]).$$

Thus,

$$\mathsf{X}^{i_1}[j_1] = \beta_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{V}^{i_1}[j_1-1]) + \mathsf{b}[i_1j_1](\mathsf{U}^{i_1}[j_1-1]).$$

We consider three cases, and some subcases under each :

- $\boxed{j_2 < j_1 - 1}$
  - When $\mathsf{X}^{i_1}[j_1] = \beta_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{V}^{i_1}[j_1-1]) + \mathsf{b}[i_1j_1](\mathsf{U}^{i_1}[j_1-1])$, since $\mathsf{b}[i_1j_1]$ is full-rank, and $\mathsf{V}^{i_1}[j_1-1]$ cannot affect $\mathsf{X}^{i_2}[j_2]$, we are done.
  - When $\mathsf{X}^{i_1}[j_1] = \beta_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{V}^{i_1}[j_1-1]) + \mathsf{b}[i_1j_1](\mathsf{U}^{i_1}[j_1-1])$, since $\mathsf{b}[i_1j_1]$ is full-rank, and $\mathsf{U}^{i_1}[j_1-1]$ cannot affect $\mathsf{X}^{i_2}[j_2]$, we are done.
- $\boxed{j_2 = j_1 - 1}$
  - When $\mathsf{X}^{i_1}[j_1] = \beta_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{V}^{i_1}[j_1-1]) + \mathsf{b}[i_1j_1](\mathsf{U}^{i_1}[j_1-1])$, if $\mathsf{V}^{i_1}[j_1-1]$ does not affect $\mathsf{X}^{i_2}[j_2]$, it's the same as before. Otherwise, since $\mathsf{V}^{i_1}[j_1-1]$ cannot appear in $\mathsf{X}^{i_2}[j_2]$ through a b-call, and $\mathsf{b}[i_1j_1] + \mathsf{id}$ is full-rank, we're done.
  - When $\mathsf{X}^{i_1}[j_1] = \beta_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{V}^{i_1}[j_1-1]) + \mathsf{b}[i_1j_1](\mathsf{U}^{i_1}[j_1-1])$, if $\mathsf{U}^{i_1}[j_1-1]$ does not affect $\mathsf{X}^{i_2}[j_2]$, it's the same as before. Otherwise, since $\mathsf{U}^{i_1}[j_1-1]$ cannot appear in $\mathsf{X}^{i_2}[j_2]$ through a b-call, and $\mathsf{b}[i_1j_1] + \mathsf{id}$ is full-rank, we're done.
- $\boxed{j_1 = j_2}$ These calculations are longer and omitted in this version of the paper.

A.3 badB: Type 2.

Let $(i_1, j_1), (i_2, j_2) \in \mathcal{F}'$ with $(j_1, i_1) > (j_2, i_2)$. We will show that

$$\Pr_0\left[\mathsf{Y}^{i_1}[j_1] = \mathsf{Y}^{i_2}[j_2]\right] \leq \frac{1}{N}.$$

We recall that

$$\mathsf{Y}^{i_1}[j_1] = \mathsf{U}^{i_1}[j_1] + \mathsf{L}^{i_1}[j_1].$$

When $\mathsf{Y}^{i_1}[j_1] = \mathsf{U}^{i_1}[j_1] + \mathsf{L}^{i_1}[j_1]$, since $\mathsf{U}^{i_1}[j_1]$ cannot affect $\mathsf{Y}^{i_2}[j_2]$, we are done. Thus we can assume $\mathsf{Y}^{i_1}[j_1] \notin \mathcal{B}$. Next we recall that, when $\mathsf{R}'^{i_1}[j_1] = \mathsf{R}'^{\tilde{i}_1}[\tilde{j}_1]$, we have

$$\mathsf{Y}^{i_1}[j_1] = \mathsf{Y}^{\tilde{i}_1}[\tilde{j}_1] + \mathsf{L}'^{\tilde{i}_1}[\tilde{j}_1] + \mathsf{b}[\tilde{i}_1\tilde{j}_1](\mathsf{T}^{\tilde{i}_1}[\tilde{j}_1]) + \mathsf{L}'^{i_1}[j_1] + \mathsf{b}[i_1j_1](\mathsf{T}^{i_1}[j_1]),$$

and when $\mathsf{R}'^{i_1}[j_1] = \mathsf{L}^{\tilde{i}_1}[\tilde{j}_1]$, we have

$$\mathsf{Y}^{i_1}[j_1] = \mathsf{b}_1(\mathsf{X}^{\tilde{i}_1}[\tilde{j}_1] + \mathsf{R}^{\tilde{i}_1}[\tilde{j}_1] + \mathsf{b}[\tilde{i}_1\tilde{j}_1](\mathsf{T}^{\tilde{i}_1}[\tilde{j}_1])) + \mathsf{L}'^{i_1}[j_1] + \mathsf{b}[i_1j_1](\mathsf{T}^{i_1}[j_1]).$$

As before, for (i, j) such that $(i, j) \neq (\tilde{i}, \tilde{j})$, we define

$$\begin{aligned}
\alpha'_{ij} &:= \mathsf{Y}^{\tilde{i}}[\tilde{j}] + \mathsf{L}'^{\tilde{i}}[\tilde{j}] + \mathsf{b}[\tilde{i}\tilde{j}](\mathsf{T}^{\tilde{i}}[\tilde{j}]) + \mathsf{L}'^{i}[j] && \text{when } \mathsf{R}'^{i}[j] = \mathsf{R}'^{\tilde{i}}[\tilde{j}],\\
&:= \mathsf{b}_1(\mathsf{X}^{\tilde{i}}[\tilde{j}] + \mathsf{R}^{\tilde{i}}[\tilde{j}] + \mathsf{b}[\tilde{i}\tilde{j}](\mathsf{T}^{\tilde{i}}[\tilde{j}])) + \mathsf{L}'^{i}[j] && \text{when } \mathsf{R}'^{i}[j] = \mathsf{L}^{\tilde{i}}[\tilde{j}].
\end{aligned}$$

Again, α'_{ij} is well-defined unless badA has occurred. Thus we can write

$$\begin{aligned}
\mathsf{Y}^{i_1}[j_1] &= \alpha'_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{T}^{i_1}[j_1])\\
&= \alpha'_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{X}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1j_1](\mathsf{Y}^{i_1}[j_1 - 1])\\
&= \alpha'_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1j_1](\mathsf{U}^{i_1}[j_1 - 1])\\
&\qquad + \mathsf{b}[i_1j_1](\mathsf{R}'^{i_1}[j_1 - 1] + \mathsf{L}^{i_1}[j_1 - 1]).
\end{aligned}$$

As before, we define

$$\beta'_{ij} := \alpha'_{ij} + \mathsf{b}[ij](\mathsf{R}'^{i}[j - 1] + \mathsf{L}^{i}[j - 1]).$$

Thus,

$$\mathsf{Y}^{i_1}[j_1] = \beta'_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1j_1](\mathsf{U}^{i_1}[j_1 - 1]).$$

Again we consider three cases, and some subcases under each:

- $\boxed{j_2 < j_1 - 1}$
  - When $\mathsf{Y}^{i_1}[j_1] = \beta'_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1j_1](\mathsf{U}^{i_1}[j_1 - 1])$, since $\mathsf{b}[i_1j_1]$ is full-rank, and $\mathsf{V}^{i_1}[j_1 - 1]$ cannot affect $\mathsf{Y}^{i_2}[j_2]$, we are done.
  - When $\mathsf{Y}^{i_1}[j_1] = \beta'_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1j_1](\mathsf{U}^{i_1}[j_1 - 1])$, since $\mathsf{b}[i_1j_1]$ is full-rank, and $\mathsf{U}^{i_1}[j_1 - 1]$ cannot affect $\mathsf{Y}^{i_2}[j_2]$, we are done.
- $\boxed{j_2 = j_1 - 1}$
  - When $\mathsf{Y}^{i_1}[j_1] = \beta_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1j_1](\mathsf{U}^{i_1}[j_1 - 1])$, if $\mathsf{V}^{i_1}[j_1 - 1]$ does not affect $\mathsf{Y}^{i_2}[j_2]$, it's the same as before. Otherwise, since $\mathsf{V}^{i_1}[j_1 - 1]$ cannot appear in $\mathsf{Y}^{i_2}[j_2]$ through a b-call, and $\mathsf{b}[i_1j_1] + \mathsf{id}$ is full-rank, we're done.
  - When $\mathsf{Y}^{i_1}[j_1] = \beta_{i_1j_1} + \mathsf{b}[i_1j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1j_1](\mathsf{U}^{i_1}[j_1 - 1])$, if $\mathsf{U}^{i_1}[j_1 - 1]$ does not affect $\mathsf{Y}^{i_2}[j_2]$, it's the same as before. Otherwise, since $\mathsf{U}^{i_1}[j_1 - 1]$ cannot appear in $\mathsf{Y}^{i_2}[j_2]$ through a b-call, and $\mathsf{b}[i_1j_1] + \mathsf{id}$ is full-rank, we're done.
- $\boxed{j_1 = j_2}$ These calculations are longer and omitted in this version of the paper.

A.4 badB: Type 3.

Let $(i_1, j_1) \in \mathcal{F}, (i_2, j_2) \in \mathcal{F}'$. We will show that

$$\Pr_0\left[\mathsf{X}^{i_1}[j_1] = \mathsf{Y}^{i_2}[j_2]\right] \leq \frac{1}{N}.$$

First we observe that if $(j_1, i_1) > (j_2, i_2)$ and $\mathsf{X}^{i_1}[j_1] = \mathsf{V}^{i_1}[j_1] + \mathsf{R'}^{i_1}[j_1]$, $\mathsf{V}^{i_1}[j_1]$ cannot affect $\mathsf{Y}^{i_2}[j_2]$, so we're done; and if $(j_2, i_2) > (j_1, i_1)$ and $\mathsf{Y}^{i_2}[j_2] = \mathsf{U}^{i_2}[j_2] + \mathsf{L}^{i_2}[j_2]$, $\mathsf{U}^{i_2}[j_2]$ cannot affect $\mathsf{X}^{i_1}[j_1]$, so we're done. Otherwise, we consider six cases, each with several subcases:

- $\boxed{j_2 < j_1 - 1}$
  - When $\mathsf{X}^{i_1}[j_1] = \beta_{i_1 j_1} + \mathsf{b}[i_1 j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1 j_1](\mathsf{U}^{i_1}[j_1 - 1])$, since $\mathsf{b}[i_1 j_1]$ is full-rank, and $\mathsf{V}^{i_1}[j_1 - 1]$ cannot affect $\mathsf{Y}^{i_2}[j_2]$, we are done.
  - When $\mathsf{X}^{i_1}[j_1] = \beta_{i_1 j_1} + \mathsf{b}[i_1 j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1 j_1](\mathsf{U}^{i_1}[j_1 - 1])$, since $\mathsf{b}[i_1 j_1]$ is full-rank, and $\mathsf{U}^{i_1}[j_1 - 1]$ cannot affect $\mathsf{Y}^{i_2}[j_2]$, we are done.
- $\boxed{j_1 < j_2 - 1}$
  - When $\mathsf{Y}^{i_2}[j_2] = \beta'_{i_2 j_2} + \mathsf{b}[i_2 j_2](\mathsf{V}^{i_2}[j_2 - 1]) + \mathsf{b}[i_2 j_2](\mathsf{U}^{i_2}[j_2 - 1])$, since $\mathsf{b}[i_2 j_2]$ is full-rank, and $\mathsf{V}^{i_2}[j_2 - 1]$ cannot affect $\mathsf{X}^{i_1}[j_1]$, we are done.
  - When $\mathsf{Y}^{i_2}[j_2] = \beta'_{i_2 j_2} + \mathsf{b}[i_2 j_2](\mathsf{V}^{i_2}[j_2 - 1]) + \mathsf{b}[i_2 j_2](\mathsf{U}^{i_2}[j_2 - 1])$, since $\mathsf{b}[i_2 j_2]$ is full-rank, and $\mathsf{U}^{i_2}[j_2 - 1]$ cannot affect $\mathsf{X}^{i_1}[j_1]$, we are done.
- $\boxed{j_2 = j_1 - 1}$
  - When $\mathsf{X}^{i_1}[j_1] = \beta_{i_1 j_1} + \mathsf{b}[i_1 j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1 j_1](\mathsf{U}^{i_1}[j_1 - 1])$, if $\mathsf{V}^{i_1}[j_1 - 1]$ does not affect $\mathsf{Y}^{i_2}[j_2]$, it's the same as before. Otherwise, since $\mathsf{V}^{i_1}[j_1 - 1]$ cannot appear in $\mathsf{Y}^{i_2}[j_2]$ through a b-call, and $\mathsf{b}[i_1 j_1] + \mathsf{id}$ is full-rank, we're done.
  - When $\mathsf{X}^{i_1}[j_1] = \beta_{i_1 j_1} + \mathsf{b}[i_1 j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1 j_1](\mathsf{U}^{i_1}[j_1 - 1])$, if $\mathsf{U}^{i_1}[j_1 - 1]$ does not affect $\mathsf{Y}^{i_2}[j_2]$, it's the same as before. Otherwise, since $\mathsf{U}^{i_1}[j_1 - 1]$ cannot appear in $\mathsf{Y}^{i_2}[j_2]$ through a b-call, and $\mathsf{b}[i_1 j_1] + \mathsf{id}$ is full-rank, we're done.
- $\boxed{j_1 = j_2 - 1}$
  - When $\mathsf{Y}^{i_2}[j_2] = \beta_{i_2 j_2} + \mathsf{b}[i_2 j_2](\mathsf{V}^{i_2}[j_2 - 1]) + \mathsf{b}[i_2 j_2](\mathsf{U}^{i_2}[j_2 - 1])$, if $\mathsf{V}^{i_2}[j_2 - 1]$ does not affect $\mathsf{X}^{i_1}[j_1]$, it's the same as before. Otherwise, since $\mathsf{V}^{i_2}[j_2 - 1]$ cannot appear in $\mathsf{X}^{i_1}[j_1]$ through a b-call, and $\mathsf{b}[i_2 j_2] + \mathsf{id}$ is full-rank, we're done.
  - When $\mathsf{Y}^{i_2}[j_2] = \beta_{i_2 j_2} + \mathsf{b}[i_2 j_2](\mathsf{V}^{i_2}[j_2 - 1]) + \mathsf{b}[i_2 j_2](\mathsf{U}^{i_2}[j_2 - 1])$, if $\mathsf{U}^{i_2}[j_2 - 1]$ does not affect $\mathsf{X}^{i_1}[j_1]$, it's the same as before. Otherwise, since $\mathsf{U}^{i_2}[j_2 - 1]$ cannot appear in $\mathsf{X}^{i_1}[j_1]$ through a b-call, and $\mathsf{b}[i_2 j_2] + \mathsf{id}$ is full-rank, we're done.
- $\boxed{j_1 = j_2}$ These calculations are longer and omitted in this version of the paper.

A.5 badB: Type 4.

Let $(i_1, j_1) \in \mathcal{F}, (i_2, j_2) \in \mathcal{I}$. We will show that

$$\Pr_0\left[\mathsf{X}^{i_1}[j_1] = \mathsf{L}^{i_2}[j_2]\right] = \frac{1}{N}.$$

This is simple to see. As seen before, one of the following holds:

- $\mathsf{X}^{i_1}[j_1] = \mathsf{V}^{i_1}[j_1] + \mathsf{R}'^{i_1}[j_1]$;
- $\mathsf{X}^{i_1}[j_1] = \beta_{i_1 j_1} + \mathsf{b}[i_1 j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1 j_1](\mathsf{U}^{i_1}[j_1 - 1])$;
- $\mathsf{X}^{i_1}[j_1] = \beta_{i_1 j_1} + \mathsf{b}[i_1 j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1 j_1](\mathsf{U}^{i_1}[j_1 - 1])$.

In all three cases, by the randomness of the basis element (since $\mathsf{b}[i_1 j_1]$ is full-rank), we are done.

A.6 badB: Type 5.

Let $(i_1, j_1) \in \mathcal{I}, (i_2, j_2) \in \mathcal{F}'$. We will show that

$$\Pr_0\left[\mathsf{R}'^{i_1}[j_1] = \mathsf{Y}^{i_2}[j_2]\right] = \frac{1}{N}.$$

This is simple to see. As seen before, one of the following holds:

- $\mathsf{Y}^{i_1}[j_1] = \mathsf{U}^{i_1}[j_1] + \mathsf{L}^{i_1}[j_1]$;
- $\mathsf{Y}^{i_1}[j_1] = \beta'_{i_1 j_1} + \mathsf{b}[i_1 j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1 j_1](\mathsf{U}^{i_1}[j_1 - 1])$;
- $\mathsf{Y}^{i_1}[j_1] = \beta'_{i_1 j_1} + \mathsf{b}[i_1 j_1](\mathsf{V}^{i_1}[j_1 - 1]) + \mathsf{b}[i_1 j_1](\mathsf{U}^{i_1}[j_1 - 1])$.

In all three cases, by the randomness of the basis element (since $\mathsf{b}[i_1 j_1]$ is full-rank), we are done.