Designing Full-Rate Sponge based AEAD modes

Bishwajit Chakraborty[1,2], Nilanjan Datta[3], and Mridul Nandi[1,3]

[1] Indian Statistical Institute, Kolkata, India
{bishu.math.ynwa,mridul.nandi}@gmail.com
[2] Nanyang Technological University, Singapore
bishwajit.chakrabort@ntu.edu.sg
[3] Institute for Advancing Intelligence, TCG CREST, Kolkata, India
nilanjan.datta@tcgcrest.org

Sponge based constructions have gained significant popularity for designing lightweight authenticated encryption modes. Most of the authenticated ciphers following the Sponge paradigm can be viewed as variations of the Transform-then-permute construction. It is known that a construction following the Transform-then-permute paradigm provides security against any adversary having data complexity D and time complexity T as long as $DT \ll 2^{b-r}$. Here, b represents the size of the underlying permutation, while r pertains to the rate at which the message is injected. The above result demonstrates that an increase in the rate leads to a degradation in the security of the constructions, with no security guaranteed to constructions operating at the full rate, where $r = b$. This present study delves into the exploration of whether adding some auxiliary states could potentially improve the security of the Transform-then-permute construction.

Our investigation yields an affirmative response, demonstrating that a special class of full rate Transform-then-permute with additional states, dubbed frTtP+, can indeed attain security when operated under a suitable feedback function and properly initialized additional state. To be precise, we prove that frTtP+ provides security as long as $D \ll 2^{s/2}$ and $T \ll 2^s$, where s denotes the size of the auxiliary state in terms of bits. To demonstrate the applicability of this result, we show that the construction ORANGE-ZEST$_{\text{mod}}$ belongs to this class, thereby obtaining the desired security. In addition, we propose a family of full rate Transform-then-permute construction with Beetle like feedback function, dubbed fr-Beetle, which also achieves the same level of security.

1 Introduction

Since the inception of the Sponge function [2] as a mode of operation for variable output length hash functions, it has received major attention in the symmetric key cryptography paradigm. With time, the Sponge mode found its application in a variety of cryptographic protocols such as message authentication [2,6], pseudorandom sequence generation [4], and the duplex mode [5] for authenticated encryption. This popularity of the Sponge mode is evident from the number of Sponge-based designs submitted in the CAESAR competition and the recently concluded NIST lightweight cryptography (LwC) standardization process. A Sponge duplex type scheme ASCON [13] turned out to be the winner of

the NIST lightweight competition and one of the joint winners in the category of lightweight applications (resource-constrained environments) in CAESAR competition.

At a high level, Sponge-type constructions consist of a b bit state, which is split into a c bit inner state, called the capacity, and an r bit outer state, called the rate, where $b = c + r$. Traditionally, in Sponge-like modes, r bits of data absorption and squeezing are done via the rate part at a time. However, there are a few exceptions, e.g., SpoC [1], where the absorption is done via the capacity part while the squeezing is done from the rate part. In [3], Bertoni et al. proved that the Sponge construction is indifferentiable from a random oracle with a birthday-type bound in the capacity. While it is well-known that this bound is tight for hashing, for keyed applications of the Sponge, especially authenticated encryption schemes, such as duplex mode, the security could be significantly higher.

1.1 Existing Security Bounds for Sponge-type AEAD Schemes

Sponge-type authenticated encryption is mostly done via the duplex construction [5]. The duplex mode is a stateful construction that consists of an initialization interface and a duplexing interface. Initialization creates an initial state using the underlying permutation π, and each duplexing call to π absorbs and squeezes r bits of data. The security of Sponge-type AEAD modes can be represented and understood in terms of two parameters, namely the data complexity D (total number of initialization and duplexing calls to π), and the time complexity T (total number of direct calls to π).

Initially, Bertoni et al. [5] proved that duplex is as strong as Sponge and achieves security up to $DT \ll 2^c$. At Asiacrypt'14, Jovanovic et al. [15] proved that sponge duplex achieves beyond the birthday bound of the capacity. To be precise, they have shown that it achieves privacy up to $D \ll \min\{2^{b/2}, 2^{\kappa}\}$, $T \ll \min\{2^{b/2}, 2^{c-\log_2 r}, 2^{\kappa}\}$, and integrity up to $DT \ll 2^c$, $D \ll \min\{2^{c/2}, 2^{\kappa}, 2^{\tau}\}$, $T \ll \min\{2^{b/2}, 2^{c-\log_2 r}, 2^{\kappa}\}$, where τ denotes the tag size. Later, a tight privacy analysis [16] was also provided. At Asiacrypt'15, Mennink et al. [19] introduced the full-state duplex and proved that this variant is secure up to $DT \ll 2^{\kappa}$, $D \ll 2^{c/2}$, where κ is the key size. In CHES'18 [8], Chakraborti et al. came up with a variant of duplex mode, dubbed Beetle, that acheives privacy up to $DT \ll 2^b$, $D \ll 2^{b/2}$, $T \ll 2^c$, and integrity up to $D \ll \min\{2^{b/2}, 2^{c-\log_2 r}, 2^r\}$, $T \ll \min\{2^{c-\log_2 r}, 2^r, 2^{b/2}\}$, when set with $\kappa = c$ and $\tau = r$. Recently, Chakraborty et. al. [10] introduced the Transform-then-Permute construction which encompasses most of the popular Sponge-type constructions and showed that popular designs like Beetle can achieve security upto $D, T \ll 2^{b-r}$. All the existing analysis show that increasing r degrades the security of a Sponge type design. If $2^r \geq \min\{2^{b-\log T}, 2^{b-\log D}\}$ then the security of all these existing constructions becomes void.

Chakraborty et al. designed a new Sponge-based authenticated encryption ORANGE-Zest [11] where the designers introduced some extra-state in the

protocol to construct a full-rate Sponge type AEAD scheme. This construction was a round 2 submission to the recently concluded NIST LwC standardization process. However, Dobraunig et. al. [14] and Khairallah et al. [17] mounted forgery attacks on the original and a modified variant of ORANGE-Zest. It seems interesting to investigate whether these attacks can be avoided with some minor changes in the design or if there is some inherent flaw in the overall design strategy.

1.2 Our Contributions

In this paper, we revisit the Transform-then-Permute construction introduced by Chakraborty et al. [10] and investigate the security dependency on the capacity in Sponge-type modes. Our contribution is two-fold.

1. **Full Rate Transform-then-Permute Mode with Extra State:** We show that at the cost of some additional state, suitable initialization of the extra state, one can indeed achieve a full rate Transform-then-Permute type authenticated encryption mode with security up to $D \ll 2^{s/2}, T \ll 2^s$, where s denotes the bit-size of this extra state. To do that, we first introduce a generic class of full rate Transform-then-Permute authenticated encryption constructions inspired by the Transform-then-Permute construction with the extra state in Sect. 3.2. First, we describe the general structure of such construction. Then, we consider a special class of Transform-then-Permute construction, dub it frTtP, by imposing several restrictions in the underlying feedback function. In Sect. 3.3, we provide the necessary justification for the choice of these restrictions to achieve the desired security. Roughly, the restrictions take care of all the necessary conditions required for the correctness and desired security of such constructions along with simplifying the proof. We prove the generic security of the class of frTtP construction in Sect. 4 (see Proposition 4). In addition, we also consider a special sub-class of frTtP constructions, called frTtP+, with some additional restrictions that obtain a much-simplified security bound (see Theorem 2).
2. **Concrete Instantiations:** Finally, we demonstrate the applicability of our results. First, in Sect. 4.1, we show that the modified ORANGE-Zest [11] belongs to the frTtP+ class, and hence obtains the desired security. This essentially shows that the weakness in the original ORANGE-Zest [11] was only due to improper initialization, not a flaw in the underlying design strategy. Next, in Sect. 4.2, we demonstrate that simple duplex sponge-type designs, even when extended to full rate using some extra state, do not satisfy one of the necessary conditions for security, and hence, are inherently insecure. Next, in Sect. 4.3, we consider a family of Transform-then-Permute constructions following Beetle like feedback, dub fr-Beetle that belong to the frTtP+ class and hence, achieve the desired security. As a concrete instantiation from the class of fr-Beetle, we demonstrate the example of fr-COFB that uses combined feedback, as used in CoFB [9].

1.3 Significance of the Result

In this subsection, we highlight the significance of our result. We provide comparative results among the proposed construction fr-COFB, Orange with existing constructions such as Sponge-Duplex, Beetle in terms of rate, state, security, and linear operations. As depicted in Table 1, consider Beetle with $b = 256$, $r = 128$ and fr-COFB with $s = 128$. They both achieve similar security (upto $D \ll 2^{64}$, $T \ll 2^{128}$). However, at the cost of the additional 128 bit additional state (and necessary additional xor operations), fr-COFB achieves double throughput as compared to Beetle.

Mode	Rate	State	Linear operations / block	Security Bound
Sponge-Duplex [5]	r/b	0	r bit xor	$\mathcal{O}(\frac{q_p^2+\sigma^2}{2^{b-r}})$
Beetle [8]	r/b	0	r bit xor, r bit shift	$\mathcal{O}(\frac{q_p}{2^{b-r}} + \frac{q_p\sigma}{2^b})$ [10]
ORANGE-ZEST$_{\text{mod}}$ [11]	1	s	$2b$ bit xor, $(b-s)$ bit shift	$\mathcal{O}(\frac{\sigma^2+q_p}{2^s} + \frac{q_p\sigma}{2^b})$ [Sect. 4.1]
fr-COFB [This paper]	1	s	$(2b+s)$ bit xor	$\mathcal{O}(\frac{\sigma^2+q_p}{2^s} + \frac{q_p\sigma}{2^b})$ [Sect. 4.3]

Table 1: A Comparative Study of Sponge-based constructions. b and r denote the permutation size and message injection rate, respectively. By state, we mean the additional state required. The security bound only considers the major terms.

We believe our result is significant in designing high throughput, lightweight authenticated encryption designs as it provides a general guideline for constructing full-rate sponge-based constructions.

2 Preliminaries

In this paper, for any $n \in \mathbb{N}$, $(n]$ (res. $[n]$) signifies the set $\{1, 2, \ldots, n\}$ (res. $\{0, 1, \ldots, n\}$). $\{0,1\}^n$ denotes the set of bit strings of length n, $\{0,1\}^* := \bigcup_{n\geq 0}\{0,1\}^n$, and $\mathsf{Perm}(n)$ signifies the set of all permutations over $\{0,1\}^n$. We say that the two distinct strings $a = a_1 \ldots a_m$ and $b = b_1 \ldots b_{m'}$ have a common prefix of length $n \leq \min\{m, m'\}$ if $a_i = b_i$ for all $i \in (n]$, and $a_{n+1} \neq b_{n+1}$. $\lceil x \rceil_n$ (res. $\lfloor x \rfloor_n$) designates the most (res. least) significant n bits of any bit string x with $|x| \geq n$. We use the notation $\langle N \rangle_x$ to denote the binary representation of N represented in x bits. We define the falling factorial $(n)_k := n(n-1)\cdots(n-k+1)$. For any finite set $\mathcal{X}$, $(\mathcal{X})_q$ signifies the set of all q-tuples with distinct elements from $\mathcal{X}$. $\mathsf{X} \xleftarrow{\$} \mathcal{X}$ signifies the uniform sampling of X from $\mathcal{X}$, which is independent of all other previously sampled random variables. An uniform sampling of t random variables $\mathsf{X}_1, \ldots, \mathsf{X}_t$ from $\mathcal{X}$ without replacement is denoted by $(\mathsf{X}_1, \ldots, \mathsf{X}_t) \xleftarrow{\text{wor}} \mathcal{X}$. We use the symbol $\star$ to denote that it can take any possible values.

2.1 Authenticated Encryption: Definition and Security Model

Given any *key space* $\mathcal{K}$, *nonce space* $\mathcal{N}$, *associated data space* $\mathcal{A}$, *message space* $\mathcal{M}$, *ciphertext space* $\mathcal{C}$, and *tag space* $\mathcal{T}$ an authenticated encryption scheme with associated data functionality (or AEAD in short), is a tuple of algorithms $\mathsf{AE} = (\mathsf{E} : \mathcal{K}\times\mathcal{N}\times\mathcal{A}\times\mathcal{M} \to \mathcal{C}\times\mathcal{T}, \mathsf{D} : \mathcal{K}\times\mathcal{N}\times\mathcal{A}\times\mathcal{C}\times\mathcal{T} \to \mathcal{M}\cup\{\perp\})$ such that for all $(K, N, A, M) \in \mathcal{K}\times\mathcal{N}\times\mathcal{A}\times\mathcal{M}$ and $(C, T) \in \mathcal{C}\times\mathcal{T}$, $\mathsf{D}(K, N, A, C, T) = M$ if and only if $\mathsf{E}(K, N, A, M) = (C, T)$. We call E (res. D) the encryption (res. decryption) algorithm of AE. For any key $K \in \mathcal{K}$, let $\mathsf{E}_K(\cdot)$ (res. $\mathsf{D}_K(\cdot)$) denotes $\mathsf{E}(K, \cdot)$ (res. $\mathsf{D}(K, \cdot)$). In this paper, we assume $\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T} \subseteq \{0, 1\}^*$ and $\mathcal{C} = \mathcal{M}$.

For $b \in \mathbb{N}$, let $\Pi \xleftarrow{\$} \mathsf{Perm}(b)$, and $\Gamma \xleftarrow{\$} \mathsf{Func}(\mathcal{N}\times\mathcal{A}\times\mathcal{M}, \mathcal{M}\times\mathcal{T})$. Let $\perp$ denote the degenerate function from $(\mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T})$ to $\{\perp\}$. We use the superscript $\pm$ to denote bidirectional access to Π. By abuse of notation the oracle corresponding to a function (like E, Π etc.) is denoted by that function itself.

Definition 1. *Consider any AEAD scheme* AE_Π *defined over* $(\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T})$ *with the random permutation* Π *as it's underlying primitive. The AEAD advantage of an adversary* $\mathscr{A}$ *against* AE_Π *is defined as*

$$\mathbf{Adv}^{\mathrm{AEAD}}_{\mathsf{AE}_\Pi}(\mathscr{A}) := \left| \Pr_{\substack{\mathsf{K} \xleftarrow{\$} \mathcal{K} \\ \Pi^\pm}} \left[\mathscr{A}^{\mathsf{E}_\mathsf{K}, \mathsf{D}_\mathsf{K}, \Pi^\pm} = 1\right] - \Pr_{\Gamma, \Pi^\pm} \left[\mathscr{A}^{\Gamma, \perp, \Pi^\pm} = 1\right] \right|,$$

where $\mathscr{A}$'s response after its interaction with E_K, D_K, and $\Pi^\pm$ is deonoted by $\mathscr{A}^{\mathsf{E}_\mathsf{K}, \mathsf{D}_\mathsf{K}, \Pi^\pm}$. Similarly, $\mathscr{A}^{\Gamma, \perp, \Pi^\pm}$ denotes $\mathscr{A}$'s response after its interaction with Γ, $\perp$, and $\Pi^\pm$.

In this paper, we only consider adversaries which do not make any repetative or redundant queries. Let q_e and q_d denote the number of queries to E_K and D_K respectively. Let σ_e and σ_d denote the sum of input (associated data and message) lengths across all encryption and decryption queries respectively. Any adversary making q_p primitive calls, q_e encryption queries, q_d decryption queries with a total of at most σ_e and σ_d blocks of encryption and decryption queries is called a $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary or simply (q_p, σ)-adversary, where $\sigma := \sigma_e + \sigma_d$.

2.2 Coefficients H Technique

Consider any deterministic yet computationally bounded adversary $\mathscr{A}$ using a black box type interaction with one of two oracles $\mathcal{O}_0$ and $\mathcal{O}_1$ and trying to differentiate between them. The query-response tuple associated with $\mathscr{A}$'s interaction with its oracle is called its transcript. A transcript ω may also contain any other information that the oracle decides to reveal to the distinguisher at the end of the game's query-response phase. This expanded definition of transcript will be taken into consideration. Suppose Θ_1 (res. Θ_0) denotes the random transcript variable for $\mathscr{A}$'s interaction with $\mathcal{O}_1$ (res. $\mathcal{O}_0$) . The *interpolation probability* of

ω with regard to $\mathcal{O}$ is the probability of obtaining a given transcript ω in the security game with an oracle $\mathcal{O}$. Since $\mathscr{A}$ is deterministic, this probability only depends on the transcript ω and the oracle $\mathcal{O}$. A transcript ω is said to be *attainable* if $\Pr[\Theta_0 = \omega] > 0$. In this paper, $\mathcal{O}_1 = (\mathsf{E_K}, \mathsf{D_K}, \Pi^{\pm})$ and $\mathcal{O}_0 = (\Gamma, \mathsf{bot}, \Pi^{\pm})$ and the adversary is trying to distinguish $\mathcal{O}_1$ from $\mathcal{O}_0$ in the AEAD sense. We now state the coefficient H technique(or simply the H-technique), a simple yet powerful tool developed by Patarin [20]. A proof of this theorem can be found in a number of papers including [21,12,18].

Theorem 1 (H-technique [20,21]). *Let Ω be the set of all transcripts. For some $\epsilon_{\mathsf{bad}}, \epsilon_{\mathsf{ratio}} > 0$, suppose there is a set $\Omega_{\mathsf{bad}} \subseteq \Omega$ satisfying the following:*

- $\Pr[\Theta_0 \in \Omega_{\mathsf{bad}}] \leq \epsilon_{\mathsf{bad}}$;
- *For any $\omega \notin \Omega_{\mathsf{bad}}$, ω is attainable and*

$$\frac{\Pr[\Theta_1 = \omega]}{\Pr[\Theta_0 = \omega]} \geq 1 - \epsilon_{\mathsf{ratio}}.$$

Then the distinguishing advantage for any adversary $\mathscr{A}$ can be bounded as

$$\mathbf{Adv}^{\mathrm{dist}}_{\mathcal{O}_1}(\mathscr{A}) \leq \epsilon_{\mathsf{bad}} + \epsilon_{\mathsf{ratio}}.$$

2.3 Multi-chain Graph

In this section, we revisit the multi-chain graph structure and an important result that bounds the number multi-chains as discussed in Chakraborty et al. [10].

Multi-chain Graph. Let $\Theta = \{(U_1, V_1), \ldots, (U_t, V_t)\}$ be a list of pairs of b-bit elements such that $U_1, \ldots, U_t$ are distinct and $V_1, \ldots, V_t$ are distinct. For any such list of pairs, we write $\mathsf{domain}(\Theta) = \{U_1, \ldots, U_t\}$ and $\mathsf{range}(\Theta) = \{V_1, \ldots, V_t\}$. Let $\mathcal{L}$ be a linear function over b bits with the transformation matrix L. Given such a list Θ and a linear transformation matrix L, we define a labeled directed graph G^L_Θ (call it a multi-chain graph) over the set of vertices $\mathsf{range}(\Theta)$. Given $V_i, V_j \in G^L_\Theta$ and $X \in \{0,1\}^b$, we draw an X labeled directed edge $V_i \xrightarrow{X} V_j$ in the graph iff

$$L \cdot V_i \oplus X = U_j.$$

We can similarly extend this to a label walk $\mathcal{W}$ from a node W_0 to W_k as

$$\mathcal{W} : W_0 \xrightarrow{X_1} W_1 \xrightarrow{X_2} W_2 \cdots \xrightarrow{X_k} W_k$$

and simply denote it as $W_0 \xrightarrow{X} W_k$ where $X = (X_1, \ldots, X_k)$. Here k is the length of the walk.

Multi-chain. Let G^L_Θ be any multi-chain graph as defined above. Given any fixed levels $(X_1, \ldots, X_k)$, we say the set of k length walks $\{\mathcal{W}_i : W^i_0 \xrightarrow{(X_1,\ldots,X_k)} W^i_k\}$ form an *multi-chain* if and only if $W^i_k = W^j_k$ for all i, j. Note that a multi-chain is a set of walks and if $\mathcal{W}$ is a multi-chain then so is any subset of $\mathcal{W}$. The following lemma bounds the number of multi-chains of any length.

Lemma 1. *Consider the set of all multi-chains in G_{Θ}^{L} of length k. Let Γ_k denote the size of the largest of all such multi-chains of length k. If L is invertible, then*

$$\mu_t := \max_{k>0} \mathsf{Ex}\left[\frac{\Gamma_k}{k}\right] \leq 1.$$

The proof of this lemma can be found in [10].

3 Full-Rate-Transform-then-Permute AEAD

3.1 Revisiting Transform-then-Permute Paradigm

Let us first revisit the Transform-then-Permute construction introduced by Chakraborty et al. [10]. We assume that the underlying primitive of the construction is a b bit public permutation and r is the rate of message/associated data injection. Let κ, ν denote the size of the key and the nonce respectively. For simplicity, we assume $\kappa < b, \nu = b - \kappa, r \leq b$.
The construction takes a nonce N, an associated data A and a message M as input. We define a formatting function Fmt that maps any (A, M) to $(B_1, \ldots, B_{a+m}) \in (\{0,1\}^b)^{a+m}$ where $a := \lceil |A|/r \rceil$ and $m := \lceil |M|/r \rceil$, such that given any two tuples $(A, M) \neq (A', M')$ and $\mathsf{Fmt}(A, M) = (B_1, \ldots, B_{a+m})$ and $\mathsf{Fmt}(A', M') = (B'_1, \ldots, B'_{a'+m'})$, we have

1. $(B'_1, \ldots, B'_a) \neq (B_1, \ldots, B_a)$ whenever $A \neq A'$ and $a \leq a'$.
2. $(B'_{a+1}, \ldots, B'_{a+m}) \neq (B_{a+1}, \ldots, B_{a+m})$, whenever $A = A'$ and $m \leq m'$.

We consider the Sponge-type construction which takes state output Y_i and data input B_i and generate next state input X_{i+1} and the data output C_i using a linear feedback function $\mathcal{E} : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^b \times \{0,1\}^r$. This can alternatively be represented using a transformation matrix E as follows:

$$\begin{bmatrix} X_{i+1} \\ C_i \end{bmatrix} = E \cdot \begin{bmatrix} Y_i \\ B_i \end{bmatrix}.$$

Chakraborty et al. [10] considered a special type of Sponge based construction, dub them as Transform-then-Permute, where the transformation matrix is of the form

$$E = \begin{bmatrix} \star & \star \\ [I_r \ 0_{r\times(b-r)}] & [I_r \ 0_{r\times(b-r)}] \end{bmatrix}.$$

It is easy to see that accordingly the decryption transformation matrix D would also have the same form as E. A pictorial description of the Transform-then-Permute construction is depicted in Fig. 1., most of the Sponge-based AEAD designs such as Ascon, and Beetle belongs to this category.

We say that a Transform-then-Permute AEAD has Full-rate if $r = b$. It is well known that a Full-Rate-Transform-then-Permute construction is not secure:

Proposition 1. *Any full-rate Transform-then-Permute AEAD is insecure.*

For completeness, we provide the proof of Proposition 1 in Appendix.

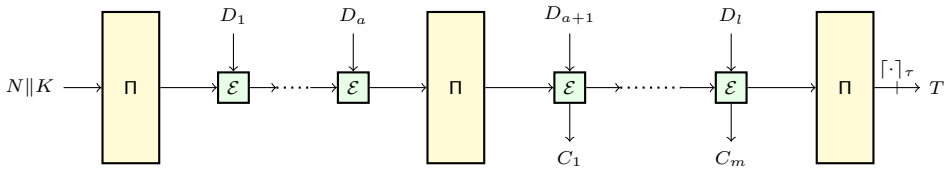

Fig. 1: Schematic of the Transform-then-Permute AEAD mode. $\mathsf{Fmt}(A, M) = (D_1, \ldots, D_l)$. The data outputs during the associated data processing are ignored.

3.2 Full-Rate-Transform-then-Permute AEAD with extra-state

We now define a Full-Rate-Transform-then-Permute (frTtP in short) AEAD mode which uses an s-bit extra secret state. The necessity of this extra state is evident from proposition 1.

General Structure of AEAD with extra-state As before considering a frTtP encryption protocol with a permutation Π of state size b bits, key size κ, nonce size $b - \kappa$ and tag size τ.

□ **Initialization:** Given any encryption query of the form (N, A, M), the encryption algorithm first applies a formatting function Fmt that maps any (A, M) to $(B_1, \ldots, B_{a+m}) \in (\{0,1\}^b)^{a+m}$, where the first a (≥ 1) blocks are generated from A. The format function should ensure that given any two tuples $(A, M) \neq (A', M')$ and $\mathsf{Fmt}(A, M) = (B_1, \ldots, B_{a+m})$ and $\mathsf{Fmt}(A', M') = (B'_1, \ldots, B'_{a'+m'})$, we have

(i) $(B_1, \ldots, B_a) \neq (B'_1, \ldots, B'_{a'})$, if and only if $A \neq A'$.
(ii) $(B_1, \ldots, B_{a+m}) \neq (B'_1, \ldots, B'_{a'+m'})$, if and only if $(A, M) \neq (A', M')$.

It is easy to see the following is a simple example of a format function satisfying the restrictions:

$$\mathsf{Fmt}(A, M) := \mathsf{ozs}(A) \parallel \mathsf{oozs}(M) \parallel \langle |A| \rangle_{b/2} \parallel \langle |M| \rangle_{b/2}.$$

Here ozs and oozs means $10^\star$ and optional $10^\star$ padding to make the blocks multiple of b bits.

In addition, we define $X_0 = K\|N$; $Y_0 = \Pi(\mathsf{X_0})$. The algorithm uses an extra-state initialization protocol to generate the initial extra-state $S_0 = \rho \circ \Pi(K\|N)$, where ρ can be any linear function from $\{0,1\}^b$ to $\{0,1\}^s$ with rank s. A trivial choice of ρ is $\rho(B) = \lfloor B \rfloor_s$.

□ **Associated Data and Message Processing:** For $i \in [1, a + m]$ and a linear feedback function $\mathcal{E} : \{0,1\}^{2b+s} \to \{0,1\}^{2b+s}$, the algorithm recursively calculates Y_i, S_i, C_i as follows:

$$(X_i, S_i, C_i) = \mathcal{E}(Y_{i-1}, S_{i-1}, B_i);\ Y_i = \Pi(X_i).$$

Alternatively, we can represent the feedback function via a transformation matrix as given below:

$$\begin{bmatrix} X_i \\ S_i \\ C_i \end{bmatrix} = \begin{bmatrix} E_1 & E_2 & E_3 \\ E_4 & E_5 & E_6 \\ E_7 & E_8 & E_9 \end{bmatrix} \begin{bmatrix} Y_{i-1} \\ S_{i-1} \\ B_i \end{bmatrix}.$$

Accordingly, there should exist a decryption transformation matrix D such that

$$\begin{bmatrix} X_i \\ S_i \\ B_i \end{bmatrix} = \begin{bmatrix} D_1 & D_2 & D_3 \\ D_4 & D_5 & D_6 \\ D_7 & D_8 & D_9 \end{bmatrix} \begin{bmatrix} Y_{i-1} \\ S_{i-1} \\ C_i \end{bmatrix} \Leftrightarrow \begin{bmatrix} X_i \\ S_i \\ C_i \end{bmatrix} = E \cdot \begin{bmatrix} Y_{i-1} \\ S_{i-1} \\ B_i \end{bmatrix}.$$

□ **Ciphertext and Tag generation:** Finally the protocol outputs $\lfloor C_{a+1} \| \cdots \| C_{a+l} \rfloor_{|M|}$ as the ciphertext and $\lfloor Y_{a+l} \rfloor_\tau$ as the tag.

We represent the generic structure in Figure 2.

Understanding the frTtP Class. Now we define a special class of full rate Transform-then-Permute, dub frTtp, where we impose the following four conditions on the feedback encryption and decryption matrices:

(C1) $E_9 = D_9 = I_b$, (C2) $E_6 = D_6 = 0$, (C3) $E_7 = D_7 = I_b$, (C4) $E_2 = E_3 \cdot E_8$ ($\neq 0$).

Note that the above restrictions ensure the following (via simple linear algebraic calculations):

$$D_1 = E_1 \oplus E_3,\ D_2 = 0,\ \text{and}\ D_i = E_i,\ \forall i = 3, \ldots, 9.$$

This simplified feedback function for frTtP is depicted in Fig. 3, where the initial extra state is calculated as $S_0 = \rho(Y_0)$, for some linear function $\rho : \{0,1\}^b \to \{0,1\}^s$ of rank s.

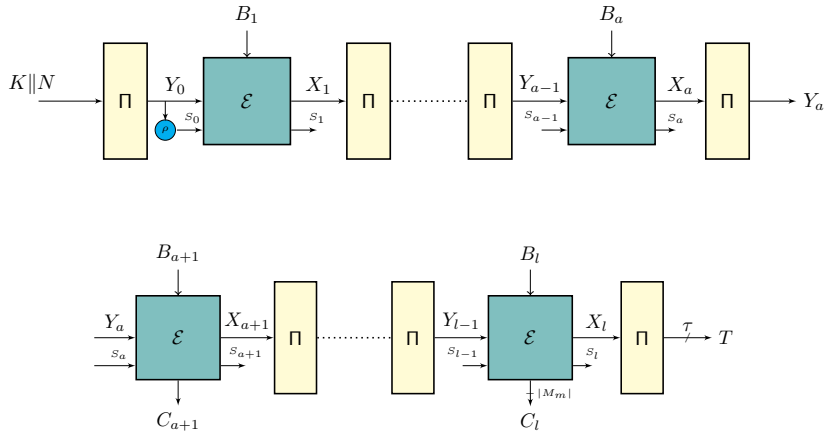

Fig. 2: A frTtP AEAD with extra state. Here $(B_1, \ldots, B_l) = \mathsf{Fmt}(A, M)$. $\rho : \{0,1\}^b \to \{0,1\}^s$ is a linear function of rank s.

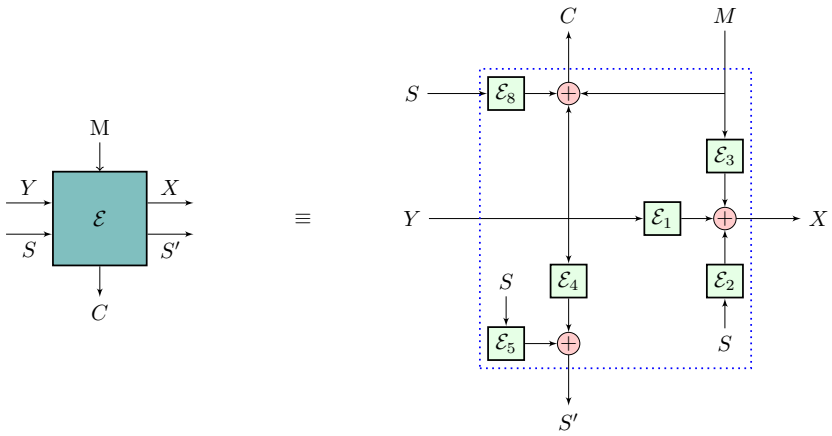

Fig. 3: Simplified Representation of an frTtP feedback function.

3.3 Rationale of the Assumptions on the Feedback Function

In this section, we justify our choices for the encryption and decryption submatrices.

□ **Choice on the Feedback Function.** To begin with, let us first state the following proposition that provides a few necessary conditions for the Encryption and Decryption Feedback Functions:

Proposition 2. *If $\mathcal{E}$ and $\mathcal{D}$ are the encryption and decryption feedback functions of a secured frTtP construction, then $\mathcal{E}, \mathcal{D}$ must satisfy the following conditions.*

(i) $\mathsf{rank}(E_9) = \mathsf{rank}(D_9) = b$,
(ii) $\mathsf{rank}(E_8) = \mathsf{rank}(D_8) \neq 0$,
(iii) $\mathsf{rank}(\begin{bmatrix} E_7 & E_8 \end{bmatrix})$, $\mathsf{rank}(\begin{bmatrix} D_7 & D_8 \end{bmatrix}) = b$.

Proof. Condition (i) follows from the correctness of the construction and the observation that if $\mathsf{rank}(E_9) \neq b$, then there exists $M \neq M'$ such that $E_9 \cdot M = E_9 \cdot M'$, and hence the decryption function will not be deterministic. $\mathsf{rank}(D_9) = b$ follows from a similar argument. Conditions (ii) and (iii) are necessary from a security perspective. Condition (ii) follows from the fact that if $\mathsf{rank}(E_8) = 0$ or $\mathsf{rank}(D_8) = 0$ then the internal Y state values are completely determined and hence the adversary can forge the construction in the same way as the frTtP construction with no extra-state. For condition 3, suppose $\mathsf{rank}(\begin{bmatrix} E_7 & E_8 \end{bmatrix}) \neq b$. Then, there exists a non zero vector γ such that $\gamma \cdot (\begin{bmatrix} E_7 & E_8 \end{bmatrix}) = 0$. Hence, $\gamma \cdot C = \gamma \cdot M$ with probability 1.

The above necessary conditions are incorporated to define a new simplified sufficient assumptions on the feedback function $\mathcal{E}$:

(C1) $E_9 = D_9 = I_b$, (C2) $E_6 = D_6 = 0$, (C3) $E_7 = D_7 = I_b$, (C4) $E_2 = E_3 \cdot E_8$ $(\neq 0)$.

Now let us try to justify the above-mentioned assumptions. To justify condition (C1), observe that with since $\mathsf{rank}(E_9) = b$ one can simply define $M' = E_9 \cdot M$, and proceed with that. For (C2), observe that since M is known, it doesn't contribute to the randomness of the extra state and hence taking $E_6 = 0$ doesn't affect the security of the AEAD scheme. $D_6 = 0$ follows from $E_6 = 0$ and assumption (i). Note that (C3) and (C4) takes care of the necessary conditions (iii) and (ii) respectively. However, they are stronger assumptions than necessary condition (iii) and (ii) respectively, which are used in simplifying the overall calculations for the special class of general frTtP feedback functions. Note that condition (C4) essentially ensure that $D_2 = 0$, i.e., during decryption the permutation input does not depend on the extra state. This condition helps in achieving the desired bound. As a consequence, we do not have any matching attacks on frTtP to justify (C3), (C4). Nonetheless, as we will see in section 4, many full-rate feedback functions used in popular constructions such as Orange-Zest, the one used in COFB or Beetle satisfy both these conditions. Moreover, in the feedback functions used in Transform-then-Permute constructions without an extra state, (C3) is a necessary condition.

□ **Choice on the Initial Extra-state Generation.** Consider a frTtP construction with extra state size s and linear feedback function $\mathcal{E}$ as defined above. Note that during associated data processing no information is leaked. Hence, for an encryption query say (N, A, M), if a many blocks of associated data are processed via format function, then it is not necessary to generate the extra-state values $S_0, \ldots, S_{a-1}$. Infact, even if $S_i = 0$ for all $0 \leq i \leq a-1$, the adversary cannot compute Y_a. So, a possible choice of defining the extra state is to define it via Y_a or nonce N. The following proposition suggests that simply generating it through (i) a linear function on Y_a or (ii) a linear function on N does not suffice.

Proposition 3. *For any encryption query (N, A, M), let a many blocks be processed due to associated data A via the format function. If (i) S_a is independent or linearly dependent on N, or (ii) $S(a)$ is a linear function of Y_a, then there exists a forging adversary against the frTtP construction.*

Proof. For part (i), assume that S_a^i is independent of N^i. Now, suppose an adversary makes two encryption queries $(N^1, A, M) \neq (N^2, A, M)$, and corresponding responses are $(C^1, T^1), (C^2, T^2)$. Let $\mathsf{Fmt}(A, M) = (B_1, \ldots, B_a, B_{a+1})$. It is easy to see that $S_a^1 = S_a^2$, and hence, $Y_a^1 = Y_a^2 \oplus C^1 \oplus C^2$. This implies $X_{a+1}^1 = B_1 \cdot Y_a^1 \oplus B_2 \cdot S_a^1 \oplus B_3 \cdot C^1 = B_1 Y_a^2 \oplus B_2 \cdot S_a^1 \oplus B_1(C^1 \oplus C^2) \oplus B_3 \cdot C^1$. Hence, if an adversary choses C^* in such a way that $B_3 \cdot (C^* \oplus C^1) = B_1 \cdot (C^1 \oplus C^2)$ then (N^2, A^2, C^*, T^1) is a valid forgery. A similar analysis goes through if S_a is linearly dependent on N. In that case, we have $S_a^1 \oplus S_a^2 = F \cdot (N^1 \oplus N^2)$ where F is some $s \times \nu$ linear matrix. Here (N^2, A^2, C^*, T^1) would be a valid forgery, where $B_3 \cdot (C^* \oplus C^1) = B_1 \cdot (C^1 \oplus C^2 \oplus E_8 \cdot F \cdot (N^1 \oplus N^2)) \oplus B_2 \cdot F \cdot (N^1 \oplus N^2)$.

For part (ii), let us assume $S_a = \rho(Y_a)$. Then, $(I_b \oplus E_8 \cdot \rho) \cdot Y_a = C_{a+1} \oplus B_{a+1}$. Now, if $\mathsf{rank}(I_b \oplus E_8 \cdot \rho) = b$, then Y_a can be calculated as $(I_b \oplus E_8 \cdot \rho)^{-1} \cdot (C_{a+1} \oplus B_{a+1})$. If $\mathsf{rank}((I_b \oplus E_8 \cdot \rho)) < b$, then there exists vector γ such that $\gamma \cdot (I_b \oplus E_8 \cdot \rho) = 0$ which implies $\gamma \cdot C_{a+1} = \gamma \cdot B_{a+1}$ with probability 1.

Now, assuming the underlying primitive Π is the only nonlinear component, a natural choice for the initial extra-state would be $\rho \circ \Pi \circ \rho'(N, K)$, where $\rho : \{0,1\}^b \to \{0,1\}^s$, $\rho' : \{0,1\}^\kappa \times \{0,1\}^\nu \to \{0,1\}^b$ are two linear functions. A straightforward choice for $\rho'(N, K)$ would be $K \| N$, which in fact is used in many popular AEAD protocols such as CoFB [9]. However, this doesn't work if no block is processed in the associated data (e.g., empty-associated data) due to the above Proposition. However, as mentioned in our format function, it always generates one associate data block to ensure that at least one block is processed during the associated data, and take $\rho'(N, K) = K \| N$.

4 Security of frTtP AEAD with Extra State

In this section, we bound the advantage of any AEAD adversary against the frTtP construction defined in section 3. Consider an frTtP construction with the encryption and decryption feedback functions $\mathcal{E}$ and $\mathcal{D}$ respectively. Alongside, we consider a linear function $\rho : \{0,1\}^b \to \{0,1\}^s$ of rank s for processing the initial extra-state. Before proceeding to the exact proposition statement, we define a notation for multi-collision as follows: Let $\mathsf{X}_1, \ldots, \mathsf{X}_\mu \xleftarrow{\mathsf{wor}} \mathcal{D}$ where $|\mathcal{D}| = \beta$ and $\beta \geq 4$. Let $\mathsf{mc}_{\mu,\beta}$ denote the maximum multicollision random variable for the sample i.e., $\mathsf{mc}_{\mu,\beta} = \max_a |\{i : \mathsf{X}_i = a\}|$. We define $\mathsf{mcoll}(\mu, \beta) := \mathsf{Ex}[\mathsf{mc}_{\mu,\beta}]$. Given this definition, we now state our main proposition as follows.

Proposition 4. *The AEAD advantage of all adversaries making q_p many primitive queries, a total of σ_e blocks in encryption queries, and a total of σ_d blocks in*

decryption queries against an frTtP construction with s bit extra-state as defined above, can be bounded as follows

$$\begin{aligned}\mathbf{Adv}^{\mathrm{AEAD}}_{\mathit{frTtP}}(q_p,\sigma_e,\sigma_d) \le \frac{q_p}{2^\kappa} + \frac{9\sigma_e q_p}{2^{r_{12}}} + \frac{2\sigma_e^2}{2^{r_{12}}} + \frac{2\sigma_e^2}{2^{r_{45}+r'_{45}-s}} + \frac{q_p\sigma_d}{2^{r_d}} + \frac{2\sigma_d}{2^\tau} + \frac{\sigma_d\sigma_e}{2^{r_d+r_3-b}} \\ + \frac{q_p\mathsf{mcoll}(\sigma_e,2^\tau)}{2^{b-\tau}} + \frac{2q_p\mathsf{mcoll}(\sigma_e,2^{b+s-r_{45}-r_8})}{2^{r_{45}+r_{12}+r_8-s-b}} \\ + \frac{3\sigma_d(\sigma_e+\sigma_d+q_p)}{2^{r_d}} + \frac{\sigma_d q_p\mathsf{mcoll}(\sigma_e,2^{b+s-r_{45}-r_8})}{2^{r_{12}+r_d+r_8-r_{45}-b-s}},\end{aligned}$$

where

$$r_3 := \mathsf{rank}(E_3);\quad r_8 = \mathsf{rank}(E_8);\quad r_{12} := \mathsf{rank}\left(E_1 \oplus E_2\cdot\rho\right);\quad r_d := \mathsf{rank}(D_1);$$

$$r_{45} := \mathsf{rank}\left((E_4\cdot E_8 \oplus E_5)^s\right);\quad r'_{45} = \min_{j\le\ell}\left\{\mathsf{rank}(I_s \oplus (E_4\cdot E_8\oplus E_5)^j)\right\}.$$

Here ℓ denotes the maximum allowed message length.

The above proposition gives a generic security bound on the security of frTtP. Now let us consider a special simplified class of frTtP, call it frTtP+, with the following additional restrictions:

$$\mathsf{rank}(E_3) = \mathsf{rank}(r_{12}) = \mathsf{rank}(D_1) = b,\quad rank(E_8) = s,\quad E_4\cdot E_8 \oplus E_5 = \alpha\cdot I_s,$$

where α is a primitive element in $GF(2^s)$. As we will see, we can construct several efficient authenticated cipher construction following frTtP+ paradigm. Now we state a simplified result on the security of this new class of frTtP constructions.

Theorem 2. *The AEAD advantage of any adversary against an authenticated encryption construction following frTtP+ paradigm making q_p many primitive queries, a total of σ blocks in encryption and decryption queries can be bounded by*

$$\mathbf{Adv}^{\mathrm{AEAD}}_{\mathit{frTtP}+}(q_p,\sigma) \le \frac{q_p}{2^\kappa} + \frac{2\sigma}{2^\tau} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{8\sigma q_p + 3\sigma^2}{2^b} + \frac{2\sigma^2 + 12(b-s)q_p}{2^s},$$

where $\sigma \le \mathsf{min}\{2^{b-s}, 2^\tau\},\ \ell < 2^s$.

Proof. First, observe that the restrictions on the encryption matrices, by definition, ensure $r_3 = r_{12} = r_d = b$, $r_8 = s$. In addition, note that $E_4\cdot E_8 \oplus E_5 = \alpha\cdot I_s$ implies that (i) $r_{45} := \mathsf{rank}\left((E_4\cdot E_8\oplus E_5)^s\right) = \mathsf{rank}(\alpha^s\cdot I_s) = s$ and (ii) $r'_{45} = \min_{j\le\ell}\left\{\mathsf{rank}(I_s\oplus(E_4\cdot E_8\oplus E_5)^j)\right\} = \min_{j\le\ell}\{\mathsf{rank}((\alpha^j\oplus 1)\cdot I_s)\} = s$. This follows from the fact that α is a primitive element in $GF(2^s)$, and $\ell < 2^s$. Next, we simplify all the terms involving mcoll by the following result: $\mathsf{mcoll}(\mu, 2^\beta) \le 4\beta$, for any $\mu,\ \beta$ with $\mu \le 2^\beta$ [10]. We can apply this result as we assume $\sigma \le \mathsf{min}\{2^{b-s}, 2^\tau\}$. Finally, the Theorem follows from Proposition 4 as we simplify all the terms and use $\sigma = \sigma_e + \sigma_d$.

4.1 Security of Modified ORANGE-Zest

In this section, we discuss the security of the modified variant of ORANGE-Zest, as proposed in [11]. Note that the construction uses the format function satisfying the definition and the initial extra secret state is generated by $\rho(\Pi(K\|N))$, where $\rho(X) = \lfloor X \rfloor_s$. Now let us look at the feedback function of the design. Note that the feedback function remains the original one, and it is given as follows.

$$E_{\text{ORANGE-Zest}} = \begin{bmatrix} \begin{bmatrix} I_{b-s} \oplus A^{-1} & 0_{(b-s)\times s} \\ 0_{s\times(b-s)} & 0_{s\times s} \end{bmatrix} & \begin{bmatrix} 0_{(b-s)\times s} \\ I_s \end{bmatrix} & I_b \\ \begin{bmatrix} 0_{s\times(b-s)} & \alpha \cdot I_s \end{bmatrix} & 0_s & 0_{s\times b} \\ I_b & \begin{bmatrix} 0_{(b-s)\times s} \\ I_s \end{bmatrix} & I_b \end{bmatrix},$$

where $A_{b-s} = \begin{bmatrix} 0_{(b-s-1)\times 1} & I_{b-s} \\ 1 & 0_{1\times(b-s)} \end{bmatrix}$.

It is easy to verify that the above feedback function along with the modified format function satisfies the definition of frTtP construction. Moreover, the feedback function satisfies (i) $\mathsf{rank}(r_{12}) = b$, (ii) $\mathsf{rank}(D_1) = b$, (iii) $E_4 \cdot E_8 \oplus E_5 = \alpha \cdot I_s$, and hence belongs to the $\mathsf{frTtP}+$ family. Hence, applying Theorem 2, we obtain the security of ORANGE-Zest$_{\text{mod}}$:

$$\mathbf{Adv}^{\text{AEAD}}_{\text{ORANGE-Zest}_{\text{mod}}}(\sigma, q_p) \leq \frac{q_p}{2^\kappa} + \frac{2\sigma}{2^\tau} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{8\sigma q_p + 3\sigma^2}{2^b} + \frac{2\sigma^2 + 12(b-s)q_p}{2^s}.$$

Remark 1. We would like to point out that Dobraunig et al. [14] mounted a forgery attack on the original construction, i.e., ORANGE-Zest, exploiting the property that the extra-state doesn't depend on the nonce under certain cases (to be precise, for the case of empty associated data), which is a necessary condition as discussed in Proposition 3. However, the attack is not applicable on the ORANGE-Zest$_{\text{mod}}$ as it follows the proper formatting, as mentioned. This result shows that the weakness was only due to the initial extra-state generation protocol, not the weakness of the underlying feedback function. Also, we would like to highlight that the security becomes void if $\tau = b$, as evident from the bound, justifying a matching attack as reported by Khairallah et al. [17].

4.2 (In)security of Full Rate Sponge-Duplex and Oribatida

In this subsection, we discuss the full-rate version of conventional Sponge-duplex which uses some extra-state. The corresponding feedback function can be represented by E_{duplex}.

$$E_{\mathsf{duplex}} = \begin{bmatrix} I_b \star I_b \\ \star \ \star \ \star \\ I_b \star I_b \end{bmatrix}, \quad E_{\mathsf{Oribatida}} = \begin{bmatrix} I_b & 0_{b\times b} & I_b \\ \alpha \cdot I_b & 0 & 0 \\ I_b & I_b & I_b \end{bmatrix}.$$

In [7], Bhattacharjee et al. designed Oribatida, a variant of Sponge-duplex with extra-state to achieve integrity security in the RUP setting, where plaintexts may be released before verification. Now let us look at the design of Oribatida when we make it full-rate. The feedback function of a full-rate Oribatida construction can be represented as $E_{\mathsf{Oribatida}}$.
Now let us consider a full rate Transform-then-Permute construction that uses an instantiation of E_{duplex} (and consequently $E_{\mathsf{Oribatida}}$) as the underlying feedback function. By simple linear algebra, one can show that for such a feedback function, we have $D_1 = 0$, which essentially says, during decryption, Y_{i+1} does not depend on X_i. This can be exploited by an adversary $\mathscr{A}$ to mount a forgery attack. Let us assume $\mathscr{A}$ makes an encryption query (N, A, M) such that $\mathsf{Fmt}(A, M) = (B_1, \ldots, B_a, B_{a+1})$, and the corresponding response is (C, T), where $|C| = b$. Now, $\mathscr{A}$ can choose an A' such that $\mathsf{Fmt}(A', M) = (B_1, \ldots, B_{a-1}, B'_a, B_{a+1})$, where $B'_a \neq B_a$ and makes a forging of the form (N, A', C, T). As S_a is generated using $N, K, B_1, \ldots, B_{a-1}$ which are the same in both the encryption and decryption queries. As $D_1 = 0$, for both the encryption and decryption queries, we have $X_{a+1} = D_2 \cdot S_a \oplus D_3 \cdot C$ validating (N, A', C, T) to be a valid forgery. This attack shows the insecurity of a full-rate variant of conventional Sponge-duplex (and consequently Oribatida), even when an additional extra state in incorporated.

4.3 frTtP with Combined and Beetle Feedback

Now let us look at what happened if we use a combined feedback function (as in CoFB) [9], or use a full-rate version of Beetle feedback [8] incorporating extra state. The combined and the full-rate Beetle feedback function (dubbed as Beetle-fb) can be represented as below:

$$E_{\mathsf{combined}} = \begin{bmatrix} G & \begin{bmatrix} 0_s \\ I_s \end{bmatrix} & I_b \\ 0_{s\times b} & \alpha \cdot I_s & 0_{s\times b} \\ I_b & 0_{b\times s} & I_b \end{bmatrix}, \quad E_{\mathsf{Beetle\text{-}fb}} = \begin{bmatrix} \rho_1 \star I_b \\ \star \ \star \ \star \\ I_b \star I_b \end{bmatrix}.$$

For combined feedback, G is a square matrix of size b, such that both G and $G \oplus I_b$ are non-singular. On the other hand, Beetle-fb considers a family of feedback functions with ρ_1 is a square matrix of size b such that both ρ_1 and $\rho_1 \oplus I_b$ are non-singular. Hence, we can visualize that combined feedback is essentially one instantiation from the more generalized Beetle-fb family of feedback functions.

Observe that $E_{\mathsf{Beetle\text{-}fb}}$ satisfies the conditions $\mathsf{rank}(D_1) = \mathsf{rank}(\rho_1 \oplus I) = b$. Next, we consider a sub-family of Beetle-fb where the sub matrices E_2, E_4, E_5, E_6, E_8 satisfies the condition $E_8 = E_2$, $\mathsf{rank}(E_2) = s$, $E_6 = 0$, and $E_4 \cdot E_8 \oplus E_5 = \alpha \cdot I_s$. Let us consider the family of frTtP constructions which uses a feedback function from this new sub-family of Beetle-fb, and call them fr-Beetle family of constructions. It is easy to see that, by the above definition, any fr-Beetle construction

belongs to frTtP+ class. Thus, applying Theorem 2, we obtain

$$\mathbf{Adv}^{\mathrm{AEAD}}_{\mathsf{fr\text{-}Beetle}}(\sigma, q_p) \leq \frac{q_p}{2^\kappa} + \frac{2\sigma}{2^\tau} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{8\sigma q_p + 3\sigma^2}{2^b} + \frac{2\sigma^2 + 12(b-s)q_p}{2^s}.$$

Now let us look at some efficient instantiation of Beetle-fb. More precisely, we look for the choices of E_2, E_4, E_5, E_6, E_8 satisfying the above properties. Interestingly, the choices for E_2, E_4, E_5, E_6 in E_{combined} satisfy the last three properties. If we modify the combined feedback function by defining $E_8 = E_2$, we will immediately obtain an efficient instantiation of Beetle-fb. So, let us consider this modified feedback matrix:

$$E_{\mathsf{combined+}} = \begin{bmatrix} G & \begin{bmatrix} 0_{\frac{b}{2}} \\ I_{\frac{b}{2}} \end{bmatrix} & I_b \\ 0_{\frac{b}{2}\times b} & \alpha \cdot I_{\frac{b}{2}} & 0_{\frac{b}{2}\times b} \\ I_b & \begin{bmatrix} 0_{\frac{b}{2}} \\ I_{\frac{b}{2}} \end{bmatrix} & I_b \end{bmatrix}.$$

We dub an frTtP the construction with $E_{\mathsf{combined+}}$ feedback function as fr-COFB. As mentioned already, fr-COFB belongs to the fr-Beetle family of frTtP constructions, and hence obtain the same security bound. The construction fr-COFB is depicted in Fig. 4.3.

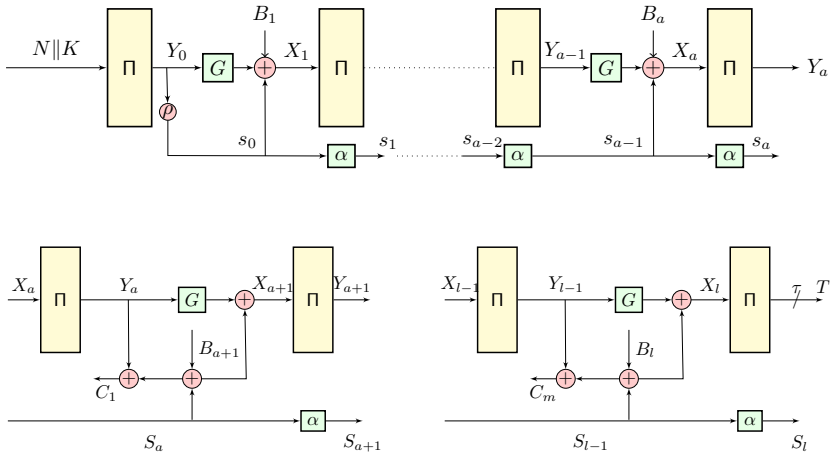

Fig. 4: fr-COFB. Here $(B_1, \ldots, B_l) = \mathsf{Fmt}(A, M)$ and $\rho : \{0,1\}^b \to \{0,1\}^s$ is a linear function of rank s. In the diagram, $B_i \oplus S_{i-1}$ represents the bit-wise xor of B_i and $0^{|B_i|-|S_{i-1}|} \| S_{i-1}$.

Discussion: Let us look at the original Beetle [8] contruction. Assuming message injection rate of r bits and state size of b bits, Chakraborty et. al. [10] showed that

Beetle achieves a security of $\mathcal{O}\left(\frac{\sigma+q_p}{2^{b-r}} + \frac{\sigma q_p}{2^{2b-2r}} + \frac{\sigma q_p^2}{2^{2b-r}}\right)$. Thus, assuming $q_p\sigma \ll 2^b$, Beetle construction is secure with a data absorption rate $r \ll b - \log(\sigma + q_p)$. Now with our new proposal fr-Beetle, we obtain the full rate with the same level of security at the cost of an extra state of size $s \gg \log(\sigma^2 + q_p)$ bits.

5 Proof of Theorem 2

5.1 Description of the Ideal World

The ideal world responds to the encryption queries, decryption queries, and primitive queries in the online phase as follows:

(1) On Primitive Query $(\mathsf{W}_i, \mathsf{dir}_i)$: The ideal world simulates $\Pi^{\pm}$ query honestly. In particular, if $\mathsf{dir}_i = 1$, it sets $\mathsf{U}_i \leftarrow \mathsf{W}_i$ and returns $\mathsf{V}_i = \Pi(\mathsf{U}_i)$. Similarly, when $\mathsf{dir}_i = -1$, it sets $\mathsf{V}_i \leftarrow \mathsf{W}_i$ and returns $\mathsf{U}_i = \Pi^{-1}(\mathsf{V}_i)$.

(2) On Encryption Query $Q_i := (\mathsf{N}_i, \mathsf{A}_i, \mathsf{M}_i)$: It first defines

$$(B_{i,1} \dots B_{i,a_i}, B_{i,a_i+1}, \dots, B_{i,l_i}) := \mathsf{Fmt}(\mathsf{A}_i, \mathsf{M}_i)$$

where a_i represents the number of blocks of size b bits generated using associated data A_i. It then, samples $Y_{i,0}, \dots, Y_{i,l_i} \xleftarrow{\$} \{0,1\}^b$. For all $1 \le j \le l_i$, it then calculates

$$S_{i,j} = \begin{cases} \rho \cdot Y_{i,0}, & \text{if } j = 0 \\ E_5^{j-1} \cdot (E_4 \oplus E_5 \cdot \rho) \cdot Y_{i,0} \oplus \bigoplus_{k=1}^{j-1} E_5^{j-1-k} \cdot E_4 \cdot Y_{i,k}, & \text{otherwise.} \end{cases}$$

Next, it computes

$$C_{i,j} = Y_{i,j-1} \oplus E_8 \cdot S_{i,j-1} \oplus B_{i,j},\ \forall\ a_i + 1 \le j \le l_i.$$

Finally, it returns $(\mathsf{C}_i, \mathsf{T}_i)$, where $\mathsf{C}_i = \lfloor C_{i,a_i+1} \| \cdots \| C_{i,l_i} \rfloor_{|\mathsf{M}_i|}$, $\mathsf{T}_i = \lfloor Y_{i,l_i} \rfloor_\tau$.

(3) On Decryption Query $Q_i := (\mathsf{N}_i^*, \mathsf{A}_i^*, \mathsf{C}_i^*, \mathsf{T}_i^*)$: We only consider non-trivial decryption queries, and the ideal world always aborts (returns the abort symbol $\perp$) for any such query.

Offline Phase of Ideal World. After completion of oracle interaction (the above three types of queries possibly in an interleaved manner), the ideal oracle sets $\mathbb{E}, \mathbb{D}, \mathbb{P}$ to denote the sets of all the query indices corresponding to the encryption, decryption, and primitive queries respectively. Let $|\mathbb{E}| = q_e, |\mathbb{D}| = q_d, |\mathbb{P}| = q_p$.

$\square$ Extended Transcripts (Encryption Queries). Now we describe the extended transcript for the encryption queries. It samples $K \xleftarrow{\$} \{0,1\}^\kappa$. For all $i \in \mathbb{E}$ and $j \in [0, l_i]$, we define

$$X_{i,j} = \begin{cases} K \| \mathsf{N}_i, & \text{if } j = 0 \\ E_1 \cdot Y_{i,j-1} \oplus E_2 \cdot S_{i,j-1} \oplus E_3 \cdot B_{i,j}, & \text{otherwise.} \end{cases}$$

□ Extended Transcripts (Decryption Queries). Now we describe an extended transcript for the decryption queries. Given any decryption query $(N_i^*, A_i^*, C_i^*, T_i^*)$, $i \in \mathbb{D}$, let $(B_{i,1}^*, \ldots, B_{i,a_i^*}^*)$ are the blocks generated corresponding to A^*, and $(C_{i,1}^* \ldots, C_{l_i^*-a_i^*}^*) \xleftarrow{b} C_i^*$. Now, we define an integer p_i as follows.

- If $\forall i' \in \mathbb{E},\ N_i^* \neq N_{i'}$, define $p_i = -1$.
- Else, consider $i' \in \mathbb{E}$, such that $N_i^* = N_{i'}$. Since the adversary is nonce respecting there exists a unique i'.
  - If $a_{i'} \leq a_i^*$, define p_i to be the length of the maximum blockwise common prefix of $(B_{i,1}^*, \ldots, B_{i,a_i^*}^*, C_{i,1}^* \ldots, C_{l_i^*-a_i^*}^*)$ and $(B_{i',1}, \ldots, B_{i',a_i^*}, C_{i',l_{i'}-a_i^*}, \ldots, C_{i',l_{i'}})$.
  - Else, $a_i^* < a_{i'}$. Using the extended encryption transcript, for all $j \in [a_i^* + 1, a_{i'}]$ define,

    $$C_{i,j} = Y_{i,j-1} \oplus E_8 \cdot S_{i,j-1} \oplus B_{i,j}.$$

    Finally define, p_i to be the length of the maximum block-wise common prefix of $(B_{i,1}^*, \ldots, B_{i,a_i^*}^*, C_{i,1}^* \ldots, C_{l_i^*-a_i^*}^*)$ and $(B_{i',1}, \ldots, B_{i',a_i^*}, C_{i',a_i^*+1}, \ldots, C_{l_{i'}})$.

Further, for all $i \in \mathbb{D}$ and $0 \leq j \leq p_i$, we define the internal states of the ith decryption query as follows: $X_{i,j}^* = X_{i',j},\ Y_{i,j}^* = Y_{i',j},\ S_{i,j}^* = S_{i,j}$. In addition, we compute

$$X_{i,p_i+1}^* = \begin{cases} E_1 \cdot Y_{i,p_i}^* \oplus E_2 \oplus S_{i,p_i}^* \oplus E_3 \cdot B_{i,p_i+1}^*, & \text{if } p_i < a_i^* \\ D_1 \cdot Y_{i,p_i}^* \oplus D_3 \cdot C_{i,l_i-p_i}^*, & \text{otherwise} \end{cases}$$

and $S_{i,p_i+1}^* = S_{i',p_i+1}$. Note that by property of Fmt function, $X_{i,p_i+1}^* \neq X_{i',p_i+1}^*$. However, it might collide with a permutation query, i.e., $(X_{i,p_i+1}^*, \star, \star) \in \omega_p$. To handle this case, we now consider multi-chain graph $G_{\omega_p}^L$, where $L = D_1$. Note that it is possible to apply the multi-chain graph as $D_2 = 0$ (justifying our choice of (C4) for frTtP). Let us assume $x_{i,j} := E_3 \cdot C_{i,j-a_i^*}^*$ for all $i \in \mathbb{D}$, $j \in [a_i^*, l_i^*]$. If $a_i^* \leq p_i$ using $Y_{p_i+1}^*$, we consider all possible labeled walks

$$Y_{p_i+1}^* \xrightarrow{(x_{i,p_i+2}, \ldots, x_{i,j})} Y_{i,k}^*.$$

Let $j_{\max}$ denote the maximum of all such j values. Now, we define a new integer p_i' in the following way:

$$p_i' = \begin{cases} p_i, & \text{if } p_i \leq a_i^* \text{ or } (X_{i,p_i+1}^*, \star, \star) \notin \omega_p \\ j_{\max}, & \text{otherwise.} \end{cases}$$

Finally, we define

$$X_{i,p_i'+1}^* = \begin{cases} E_1 \cdot Y_{i,p_i'}^* \oplus E_2 \oplus S_{i,p_i'}^* \oplus E_3 \cdot B_{i,p_i'+1}^* & \text{if } p_i < a_i^*. \\ D_1 \cdot Y_{i,p_i'}^* \oplus D_3 \cdot C_{i,l_i-p_i'}^* & \text{otherwise.} \end{cases}$$

□ Extended Adversarial Transcripts. The overall transcript of the adversary consists of $\omega = (\omega_e, \omega_d, \omega_p)$, where the primitive, encryption, and decryption transcripts are given as follows:

$$\begin{aligned}
\omega_p &= (U_i, V_i, \pm)_{i\in\mathbb{P}} \\
\omega_e &= (N_i, A_i, M_i, X_{i,j}, Y_{i,j}, S_{i,j}, C_i, T_i)_{i\in\mathbb{E}, j\in[l_i]} \\
\omega_d &= (N_i^*, A_i^*, C_i^*, T_i^*, X_{i,j}^*, Y_{i,j}^*, S_{i,j}^*, \perp)_{i\in\mathbb{D}, j\in[p_i'+1]}.
\end{aligned}$$

5.2 Defining and Bounding Bad Transcripts in Ideal World

We now consider some bad events that may occur due to the primitive, encryption and decryption transcript.

BAD1: $\exists\ (U, \star, \star) \in \omega_p :\ K = \lceil U \rceil_\kappa$.
BAD2: $\exists\ (i,j) \neq (i',j')$ such that $S_{i,j} = S_{i',j'}$, where $i \in \mathbb{E},\ j \in [l_i],\ i' \in \mathbb{E},\ j' \in [l_{i'}]$.
BAD3: $\exists\ i \in \mathbb{E}, j \in [l_i]$ such that $(\star, Y_{i,j}, \star) \in \omega_p$.
BAD4: $\exists\ i \in \mathbb{E},\ j \in [l_i]$ such that $(X_{i,j}, \star, \star) \in \omega_p$.
BAD5: $\exists\ (i,j) \neq (i',j')$ such that $Y_{i,j} = Y_{i',j'}$, where $i \in \mathbb{E},\ j \in [l_i],\ i' \in \mathbb{E},\ j' \in [l_{i'}]$.
BAD6: $\exists\ (i,j) \neq (i',j')$ such that $X_{i,j} = X_{i',j'}$, where $i \in \mathbb{E},\ j \in [l_i],\ i' \in \mathbb{E},\ j' \in [l_{i'}]$.
BAD7: $\exists\ i \in \mathbb{D}$, $p_i' = l_i^*$ and $\lfloor Y_{i,l_i}^* \rfloor_\tau = T_i^*$.
BAD8: $\exists\ i \in \mathbb{D}, i' \in \mathbb{E},\ j' \in [l_{i'}]$ such that $X_{i,p_i'+1}^* = X_{i',j}$, where $p_i' \leq l_i^* - 1$.

We would like to point out that the first six events broadly represent some collisions in the internal states during encryption and primitive queries. Such a collision essentially induces a collision in the permutation input or output and makes the transcript permutation incompatible which can be used to perform privacy attacks. Hence, we call them bad events. The last two bad events are due to the decryption queries and can lead to forgery attacks.

Now we use the following lemma to upper bound the probability of the bad events.

Lemma 2. *Let us define* $\mathtt{BAD} = \mathtt{BAD1} \cup \cdots \cup \mathtt{BAD8}$. *We can bound the probability of* BAD *as follows.*

$$\begin{aligned}
\Pr[\mathtt{BAD}] \leq\ & \frac{q_p}{2^\kappa} + \frac{9\sigma_e q_p}{2^{r_{12}}} + \frac{2\sigma_e^2}{2^{r_{12}}} + \frac{\sigma_e^2}{2^{r_{45}+r'_{45}-s}} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}} \\
&+ \frac{q_p \mu_{q_p} \sigma_d}{2^{r_d}} + \frac{\sigma_e + q_p}{2^{r_d}} + \frac{\sigma_d \sigma_e}{2^{r_d+r_3-b}} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{12}+r_{45}+r_8-b-s}} \\
&+ \frac{\sigma_d q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{12}+r_d+r_8+r_{45}-b-s}} + + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{45}+r_8-s}}.
\end{aligned}$$

The proof of the Lemma is given in the section B.2.

5.3 Good Transcript Analysis and Completion of the Proof

In the online phase, the AE encryption, decryption, and direct primitive queries are faithfully responded to based on $\Pi^{\pm}$. Like the ideal world, after the completion of interaction, the real world returns all X-values Y-values and S-values corresponding to the encryption queries only and all the derived X, Y, S values corresponding to the decryption queries.

Lemma 3. *Let Θ_0 and Θ_1 denote the random transcript variable obtained in the ideal and real worlds, respectively. For any good transcript $\omega = (\omega_p, \omega_e, \omega_d)$, we have*

$$\frac{\Pr\left[\Theta_1 = w\right]}{\Pr\left[\Theta_0 = w\right]} \geq 1 - \left(\frac{2q_d}{2^{\tau}} + \frac{2\sigma_d(\sigma + q_p)}{2^{r_d}}\right).$$

Here we briefly discuss an informal proof sketch of the lemma. The tuples ω_e is permutation compatible and disjoint from ω_p. So, the union of tuples $\omega_e \cup \omega_p$ also remains permutation compatible. Now, in the real world, a decryption query may return M_i which is not $\perp$. However, since a good transcript always aborts on a decryption query, we need to bound the probability of this event. Suppose for all $0 \leq j \leq p'_i$, $Y^*_{i,j}, S^*_{i,j}$ and $X^*_{i,j+1}$ are defined as before.Now, for all $i \in \mathbb{D}$, we have either $p'_i = l_i - 1$ and $(X^*_{i,m_i}, \star \| T^*_i) \in \omega_p \cup \omega_e$ (call it a Type-1 decryption query) or $p'_i < l_i - 1$ but $X^*_{i,p'_i+1} \notin \omega_p \cup \omega_e$(call it a Type-2 decryption query). Type-1 decryption queries are taken care of in bad events. To be precise, such queries are already rejected due to BAD6. For the Type-2 decryption query, observe that X^*_{i,p'_i+1} is fresh i.e. it has never been queried before by the adversary. So, $\Pi(X^*_{i,p'_i+1})$ would be random over a large set. This would ensure a high probability that such decryption queries will also be rejected. The formal proof is presented in section B.3.

Proof of Theorem 2: Finally the proof of the theorem is complete as we apply Lemma 2 and Lemma 3 in Theorem 1.

5.4 Conclusion and Future Direction

In this paper, we introduce a class of full-rate Sponge-type constructions called the frTtP by introducing an extra-state as compensation for increasing the size of the rate part. We further extend the result to show that a sub-class of the constructions, called frTtP+, achieves security up to $D \ll 2^{s/2}, T \ll 2^s$, where s is the size of the extra-state (in bits). Consequently, we have shown that ORANGE-ZEST$_{\text{mod}}$ and a family of constructions following Beetle-like feedback functions belongs to the frTtP+ class, and hence, achieve the desired security. Extending the result for a more general class of constructions (beyond the frTtP class) and designing a more efficient full-rate Transform-then-Permute than ORANGE-ZEST$_{\text{mod}}$ or fr-COFB can be considered as an interesting open problem. In fact, one can investigate whether a hybrid feedback function (as used in HyENA), can be used efficiently to construct a full rate Transform-then-Permute.

Finally, using an extra state may lead to the necessity of increased protection against a wide variety of side-channel attacks. A concrete side channel analysis of frTtP schemes is an important open problem and is left for future research.

References

1. Riham AlTawy, Guang Gong, Morgan He, Ashwin Jha, Kalikinkar Mandal, Mridul Nandi, and Raghvendra Rohit. Spoc. Submission to NIST LwC Standardization Process (Round 2), 2019.
2. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. In *ECRYPT Hash Workshop 2007. Proceedings*, 2007.
3. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In *Advances in Cryptology - EUROCRYPT 2008. Proceedings*, pages 181–197, 2008.
4. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge-based pseudo-random number generators. In *Cryptographic Hardware and Embedded Systems, CHES 2010. Proceedings*, pages 33–47, 2010.
5. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In *Selected Areas in Cryptography - 18th International Workshop, SAC 2011. Revised Selected Papers*, pages 320–337, 2011.
6. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the security of the keyed sponge construction. In *Symmetric Key Encryption Workshop 2011. Proceedings*, 2011.
7. Arghya Bhattacharjee, Eik List, Cuauhtemoc Mancillas López, and Mridul Nandi. The oribatida family of lightweight authenticated encryption schemes. Submission to NIST LwC Standardization Process (Round 2), 2019.
8. Avik Chakraborti, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. Beetle family of lightweight and secure authenticated encryption ciphers. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):218–241, 2018.
9. Avik Chakraborti, Tetsu Iwata, Kazuhiko Minematsu, and Mridul Nandi. Blockcipher-based authenticated encryption: How small can we go? In *Cryptographic Hardware and Embedded Systems - CHES 2017. Proceedings*, pages 277–298, 2017.
10. Bishwajit Chakraborty, Ashwin Jha, and Mridul Nandi. On the security of sponge-type authenticated encryption modes. *IACR Transactions on Symmetric Cryptology*, pages 93–119, 2020.
11. Bishwajit Chakraborty and Mridul Nandi. ORANGE. Submission to NIST LwC Standardization Process (Round 2), 2019.
12. Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In *Advances in Cryptology - EUROCRYPT 2014. Proceedings*, pages 327–350, 2014.
13. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1. 2: Lightweight authenticated encryption and hashing. *Journal of Cryptology*, 34:1–42, 2021.
14. Christoph Dobraunig, Florian Mendel, and Bart Mennink. Round 1 official comments : ORANGE. Submission to NIST LwC Standardization Process , 2019.
15. Philipp Jovanovic, Atul Luykx, and Bart Mennink. Beyond 2 c/2 security in sponge-based authenticated encryption modes. In *Advances in Cryptology - ASIACRYPT 2014. Proceedings, Part I*, pages 85–104, 2014.
16. Philipp Jovanovic, Atul Luykx, Bart Mennink, Yu Sasaki, and Kan Yasuda. Beyond conventional security in sponge-based authenticated encryption modes. *J. Cryptology*, 32(3):895–940, 2019.

17. Mustafa Mahmoud Mohammed Kairallah, Raghvendra Rohit, and Sumanta Sarkar. Round 2 official comments : ORANGE. Submission to NIST LwC Standardization Process, 2019.
18. Bart Mennink and Samuel Neves. Encrypted davies-meyer and its dual: Towards optimal security using mirror theory. In *Advances in Cryptology - CRYPTO 2017. Proceedings, Part III*, pages 556–583, 2017.
19. Bart Mennink, Reza Reyhanitabar, and Damian Vizár. Security of full-state keyed sponge and duplex: Applications to authenticated encryption. In *Advances in Cryptology - ASIACRYPT 2015. Proceedings, Part II*, pages 465–489, 2015.
20. Jacques Patarin. *Etude des Générateurs de Permutations Pseudo-aléatoires Basés sur le Schéma du DES*. PhD thesis, Université de Paris, 1991.
21. Jacques Patarin. The "coefficients H" technique. In *Selected Areas in Cryptography - SAC 2008. Revised Selected Papers*, pages 328–345, 2008.

Appendix

A Proof of Proposition 1

Consider a TtP construction with full-rate i.e. $r = b$. Then the encryption feedback function $\mathcal{E}$ can be written as

$$E = \begin{bmatrix} \star & \star \\ I_b & I_b \end{bmatrix}.$$

The weakness of the construction comes from the fact that at each internal state during the encryption query the previous permutation output can be completely recovered. More formally an adversary $\mathscr{A}$ can forge as follows.

(i) $\mathscr{A}$ makes an encryption query (N, A, M) with $|M| = b$. Suppose the corresponding ciphertext is (C, T).
(ii) $\mathscr{A}$ computes $B_1 \| \dots \| B_{a+1} = \mathsf{Fmt}(A, M)$, $Y_a = C \oplus B_a$.
(iii) $\mathscr{A}$ choses any $C' \neq C$ of b bits and makes a forging query of the form (N, A, C', T'), where $T' = \lceil \Pi \left(Y_a \oplus C' \right) \rceil_\tau$.

It is easy to see that the adversary $\mathscr{A}$ succeeds in forging with probability 1.

B Security Proof

B.1 Some Important Mathematical Results

In this subsection, we list down a few important results from linear algebra which will be used in Appendix B.2.

Proposition 5. *Let A be an $m \times n$ matrix and B be an $n \times l$ matrix. Then,*

$$\mathsf{rank}(A \cdot B) \leq \min\{\mathsf{rank}(A), \mathsf{rank}(B)\}.$$

Corollary 1. *Let A be an $n \times n$ square matrix.*

$$\mathsf{rank}(A^i) \geq \mathsf{rank}(A^n) \ \forall i \in \mathbb{N}.$$

Corollary 2. *Let A be an $m \times m$ matrix B be an $m \times n$ matrix and C be an $n \times m$ matrix.*

$$\mathsf{rank}(A \oplus B \cdot C) \leq \mathsf{rank}(\begin{bmatrix} A & B \end{bmatrix}).$$

Proof. Note that $A \oplus B \cdot C = \begin{bmatrix} A & B \end{bmatrix} \cdot \begin{bmatrix} I_m \\ C \end{bmatrix}$

Proposition 6. *(Sylvester rank inequality) If A is an $m \times n$ matrix and B is an $n \times k$ matrix, then*

$$\mathsf{rank}(AB) \geq \mathsf{rank}(A) + \mathsf{rank}(B) - n.$$

B.2 Proof of Lemma 2

We start by bounding the individual bad events and then apply the union bound.

□ **Bounding** BAD1: This is the key recovery event, i.e., the event that the adversary recovers the master key K by direct queries to the internal random permutation (can be both forward or backward). For a fixed entry $(U, \star, \star) \in \omega_p$, the probability that $K = \lceil U \rceil_\kappa$ is bounded by at most $2^{-\kappa}$, as K is chosen uniform at random from $\{0,1\}^\kappa$. Thus, we have

$$\Pr\left[\texttt{BAD1}\right] \leq \frac{q_p}{2^\kappa}. \tag{1}$$

□ **Bounding** BAD2: Let us fix $i \in \mathbb{E}$, $j \in [l_i]$, $i' \in \mathbb{E}$, $j' \in [l_{i'}]$. We bound the probability of the event $S_{i,j} = S_{i',j'}$ in several cases as given below.

Case 1: $i \neq i'$, $j, j' = 0$. In this case $S_{i,0} = \rho(Y_{i,0})$ and $S_{i',0} = \rho(Y_{i',0})$. Since $Y_{i,0}$ and $Y_{i',0}$ are chosen uniformly at random, varying overall i, i', the probability of the event, in this case, is at most $\frac{q_e^2}{2^s}$.

Case 2: $i \neq i'$, $j' = 0, j > 0$. In this case $S_{i',0} = \rho(Y_{i',0})$. By simple algebra, we can write $S_{i,j} = (E_4 \cdot E_8 \oplus E_5)^j \cdot \rho(Y_{i,0}) \oplus \chi_{i,j}$, for some $\chi_{i,j}$ which is independent of $Y_{i,0}$. Again since $Y_{i',0}$ and $Y_{i,0}$ are chosen uniformly at random, varying overall i, i', the probability of the event, in this case, can be bounded by $\frac{q_e \sigma_e}{2^{r_{45}}}$.

Case 3: $i \neq i'$, $j', j > 0$. In this case we can write, $S_{i,j} = (E_4 \cdot E_8 \oplus E_5)^j \cdot \rho(Y_{i,0}) \oplus \chi_{i,j}$ and $S_{i',j'} = (E_4 \cdot E_8 \oplus E_5)^{j'} \cdot \rho(Y_{i',0}) \oplus \chi_{i',j'}$, for some $\chi_{i,j}, \chi_{i',j'}$ which are independent of $Y_{i,0}, Y_{i',0}$. Again, since $Y_{i',0}$ and $Y_{i,0}$ are chosen uniformly at random, varying over all i, i', the probability of this case can be bounded by $\frac{\sigma_e^2}{2^{r_{45}}}$.

Case 4: $i = i'$, $j \neq j'$. In this case, suppose $j > j'$, then $S_{i,j} = S_{i,j'}$ if and only if $(E_4 \cdot E_8 \oplus E_5)^{j'}(I_s \oplus (E_4 \cdot E_8 \oplus E_5)^{j-j'})\rho(Y_{i,0}) = A_{i,j}$ for some $A_{i,j}$ which is independent of $Y_{i,0}$. Since $Y_{i,0}$ is chosen at random this case is bounded by $\frac{\sigma_e^2}{2^{r_{45}+r'_{45}-s}}$.

Combining every case together, we have

$$\Pr[\mathtt{BAD2}] \leq \frac{\sigma_e^2}{2^{r_{45}+r'_{45}-s}}. \tag{2}$$

□ **Bounding** BAD3: This event is also analyzed in several cases as given below.

Case 1: $\exists i, j, a$, $Y_{i,j} = V_a$, encryption after primitive. Since $Y_{i,j}$ are chosen uniformly at random, this case can be bounded for fixed i, j, a with probability at most $1/2^b$. We have at most σ_e many (i, j) pairs and q_p many a indices. Hence, this case can be bounded by at most $\sigma_e q_p/2^b$.

Case 2: $\exists i, j, a$, $Y_{i,j} = V_a$, $\mathsf{dir}_a = +$, encryption before primitive. This case can be bounded by probability at most $1/(2^b - q_p + 1)$. We have at most σ_e many (i, j) pairs and q_p many a indices. Thus this can be bounded by at most $\sigma_e q_p/(2^b - q_p + 1) \leq 2\sigma_e q_p/2^b$ (assuming $q_p \leq 2^{b-1}$).

Case 3: $\exists i$, $j < a_i$, a, $Y_{i,j} = V_a$, $\mathsf{dir}_a = -$, encryption before primitive. For $j < a_i$, since no output is generated hence the probability is bounded by $\frac{q_p \sigma_e}{2^b}$.

Case 4: $\exists i$, $a_i \leq j < l_i$, a, $Y_{i,j} = V_a$, $\mathsf{dir}_a = -$, encryption before primitive. In this case, we can not argue as in the previous case as we release $C_{i,j}$, and that can leak information about $Y_{i,j}$. Note that we can rewrite $Y_{i,j}$ as

$$Y_{i,j} = E_8 \cdot (E_4 \cdot E_8 \oplus E_5)^j \cdot \rho(Y_{i,0}) \oplus \sum_{k=1}^{j-1} (E_4 \cdot E_8 \oplus E_5)^{k-1} \cdot E_4 \cdot (C_{i,k} \oplus B_{i,k}).$$

Since $Y_{i,0}$ is chosen at random, at least $\mathsf{rank}(E_8 \cdot (E_4 \cdot E_8 \oplus E_5)^j \cdot \rho)$ bits of $Y_{i,j}$ is random, i.e. the adversary can know at most $b - \mathsf{rank}(E_8 \cdot (E_4 \cdot E_8 \oplus E_5)^j \cdot \rho)$ bits of $Y_{i,j}$. Now, we claim that $\mathsf{rank}(E_8 \cdot (E_4 \cdot E_8 \oplus E_5)^j \cdot \rho) \geq r_{45} + r_8 - s$. This follows since, by Proposition 6, $\mathsf{rank}(E_8 \cdot (E_4 \cdot E_8 \oplus E_5)^j \cdot \rho) \geq \mathsf{rank}(E_4 \cdot E_8 \oplus E_5)^j + r_8 - s$ and by Corollary 1, $\mathsf{rank}(E_4 \cdot E_8 \oplus E_5)^j \geq \mathsf{rank}(E_4 \cdot E_8 \oplus E_5)^s = r_{45}$. Assuming mcoll_Y to be the number of multicollisions among all the $Y_{i,j}$ values, we bound the probability of this case by

$$\begin{aligned}
\sum_N \Pr[\text{Case 4} \mid \mathsf{mcoll}_Y = N] \times \Pr[\mathsf{mcoll}_Y = N] &\leq \sum_N \frac{N \times q_p}{2^{r_{45}+r_8-s}} \times \Pr[\mathsf{mcoll}_Y = N] \\
&\leq \frac{q_p}{2^{r_{45}+r_8-s}} \times \mathsf{Ex}[\mathsf{mcoll}_Y] \\
&\leq \frac{q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{45}+r_8-s}}.
\end{aligned}$$

Case 5: $\exists i, a$, $Y_{i,l_i} = V_a$, $\mathsf{dir}_a = -$, encryption before primitive. Note that $\lceil Y_{i,l_i} \rceil_\tau = T_i$ is known to the adversary. Hence this case is similar to Case 3. The only difference is that the adversary has access to $\lfloor Y_{i,l_i} \rfloor_\tau$. Hence doing a similar analysis

as in the previous case one can show that the probability of this case can be bounded by $\frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}}$.

Combining everything together, we have

$$\Pr[\texttt{BAD3}] \leq \frac{4\sigma_e q_p}{2^b} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{45}+r_8-s}} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}}. \tag{3}$$

□ **Bounding** BAD4: Here we actually bound the event that BAD4 occurs and BAD1 doesn't occur. Note that this event occurs if and only if there exists $i \in \mathbb{E}, j \in [1, l_i], k \in \mathbb{P}$ such that

$$\begin{bmatrix} E_1 & E_2 \end{bmatrix} \begin{bmatrix} Y_{i,j-1} \\ S_{i,j-1} \end{bmatrix} = U_k.$$

$$\begin{aligned}
\sum_k \sum_{i,j} \Pr[X_{i,j} = U_k] &= \sum_k \sum_{i,j} \Pr\left[\begin{bmatrix} E_1 & E_2 \end{bmatrix} \begin{bmatrix} Y_{i,j-1} \\ S_{i,j-1} \end{bmatrix} = U_k \right] \\
&\leq \sum_k \sum_i \Pr[(E_1 \oplus E_2 \cdot \rho) Y_{i,0} = U_k] \\
&\quad + \sum_k \sum_{\substack{i \\ j>1}} \Pr[E_1 \cdot Y_{i,j-1} = E_2 \cdot S_{i,j-1} \oplus U_k] \\
&\leq \sum_k \sum_i \Pr[(E_1 \oplus E_2 \cdot \rho) Y_{i,0} = U_k] \\
&\quad + \sum_k \sum_{U \in \{0,1\}^b} \sum_{\substack{i \\ j>1}} \Pr[E_1 \cdot Y_{i,j-1} = U] \\
&\qquad \times \Pr\left[\begin{bmatrix} E_1 & E_2 \end{bmatrix} \begin{bmatrix} U \\ S_{i,j-1} \end{bmatrix} = U_k \mid U \right]
\end{aligned}$$

Since $Y_{i,0}$ are chosen uniformly at random, $\sum_i \Pr[(E_1 \oplus E_2 \cdot \rho) Y_{i,0} = U_k]$ is bounded by $\frac{q_p q_e}{2^{r_{12}}}$.

Since $Y_{i,j-1}$ is chosen uniformly at random and $S_{i,j-1}$ is calculated independently of $Y_{i,j-1}$[4] for all $j > 1$, we get,

Given any fix $U \in \{0,1\}^b$ with a similar analysis as in BAD3,

$$\sum_{i,j} \Pr[Y_{i-1} = U] \leq \frac{4\sigma_e}{2^b} + \frac{\mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{45}+r_8-s}}.$$

[4] Note that $S_{i,j-1} = \bigoplus_{k=1}^{j-2} E_5^{j-2-k} E_4 Y_{i,k} \oplus E_5^{j-2}(E_4 \oplus E_5 \rho) Y_{i,0}$

Further, if $r_e := \mathsf{rank}(\begin{bmatrix} E_1 & E_2 \end{bmatrix})$ then using corollary 2,

$$\sum_{U \in \{0,1\}^b} \Pr\left[\begin{bmatrix} E_1 & E_2 \end{bmatrix} \begin{bmatrix} U \\ S_{i,j-1} \end{bmatrix} = U_k \right] \leq 2^{b-r_e} \leq 2^{b-r_{12}}.$$

Combining everything, we obtain

$$\Pr\left[\mathtt{BAD4} \wedge \neg \mathtt{BAD1}\right] \leq \frac{5q_p\sigma_e}{2^{r_{12}}} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{12}+r_{45}+r_8-b-s}}. \tag{4}$$

□ **Bounding** BAD5: Since all the $Y_{i,j}$'s are chosen uniformly at random,

$$\Pr\left[\mathtt{BAD5}\right] \leq \frac{\sigma_e(\sigma_e-1)}{2^b}. \tag{5}$$

□ **Bounding** BAD6: Suppose BAD2 doesn't occur. Since all $Y_{i,j}$'s are chosen uniformly at random and $S_{i,j}$ are all distinct and independent of $Y_{i,j}$,

$$\Pr\left[\mathtt{BAD6} \wedge \neg \mathtt{BAD2}\right] \leq \frac{\sigma_e(\sigma_e-1)}{2^{r_{12}}}. \tag{6}$$

□ **Bounding** BAD7: Suppose the event holds for the i-th decryption query and $N_i^* = N_{i'}$ for some $i' \in \mathbb{E}$. We use the multi-chain structure to bound the probability of this bad event. To be precise, this bad event implies that $Y^*_{i,p_i+1} \xrightarrow{(x_{i,p_i+2},\ldots,x_{i,l_i^*})} Y^*_{i,l_i^*}$ is an element of an $(l_i^*-p_i)$ length multi-chain in $G^{D_1}_{\omega_p}$ with label $(x_{i,p_i+2},\ldots,x_{i,l_i})$ terminating at some V with $(\star, V, \star) \in \omega_p$, such that $\lfloor V \rfloor_\tau = T_i^*$. Now since the adversary can make both forward/backward primitive queries, the number of such V is bounded by q_p. Hence, the probability that the above bad holds for the i-th decryption query is bounded by

$$\begin{aligned}
&q_p \times \sum_{Y' \xrightarrow{(x_{i,p_i+2},\cdots,x_{i,l_i^*})} V} \Pr\left[Y^*_{i,p_i+1} = Y'\right] \\
&\leq q_p \times \mu_{q_p}(l_i - p_i) \times \Pr\left[D_1 \cdot Y^*_{i,p_i} \oplus x_{i,p_i+1} = X' \mid (X', Y') \in \omega_p\right] \\
&\leq \frac{q_p \times \mu_{q_p}(l_i - p_i)}{2^{r_d}}.
\end{aligned}$$

Now varying overall $i \in \mathbb{D}$,

$$\Pr[\mathtt{BAD7} \wedge (\overline{\mathtt{BAD1}} \wedge \cdots \wedge \overline{\mathtt{BAD6}})] \leq \sum_{i \in \mathbb{D}} \frac{q_p \times \mu_{q_p}(l_i - p_i)}{2^{r_d}} \leq \frac{q_p\sigma_d}{2^{r_d}}. \tag{7}$$

Here the last inequality follows from the fact that $\sum_{i\in\mathbb{D}}(l_i - p_i) < \sigma_d$ and $\mu_{q_p} = 1$ (applying Lemma 1 as D_1 is invertible).

□ **Bounding** BAD8: To bound this bad event we consider several cases, and bound the probability for each of the cases.

Case 1: $p'_i < a_{i'}$. Since no information is leaked during the associated data processing of the encryption queries Y_{i',p'_i} is sampled uniformly at random. Now, BAD8 occurs if and only if

$$X^*_{i,p'_i+1} = E_1 \cdot Y_{i',p'_i} \oplus E_2 \cdot S_{i',p'_i} \oplus E_3 \cdot B^*_{i,p'_i+1}.$$

Hence, at least r_e bits of X^*_{i,p'_i+1} is random, and the probability of this case is bounded by $\frac{\sigma_e+q_p}{2^{r_e}}$.

Case 2: $\exists i' \in \mathbb{E}, j \in [l_{i'}]$ s.t. $a_{i'} \leq p'_i = p_i$ and $X^*_{i,p_i+1} = X_{i',j}$. Note that, by definition of p'_i there exists an $i'' \in \mathbb{E}$ such that $Y_{i'',p_i} = Y^*_{i,p_i}$. Hence, this event occurs if and only if

$$D_3 \cdot (C^*_{i,p_i+1-a^*_i} \oplus \cdot C_{i',j}) = D_1 \cdot (Y_{i',j-1} \oplus Y_{i'',p_i}).$$

Now, by definition of p_i, either $i' \neq i''$ or $i' = i''$ but $j > p_i + 1$. In each of the cases, we have $Y_{i',j-1}, Y_{i'',p_i}$ are independent and random. Hence, the probability that any of these happens in the i th query is bounded by at most $\frac{\sigma_e}{2^{r_d}}$. Further, given everything else fixed, there are at most 2^{b-r_3} many possible choices of B^*_{i,p_i+1}. Hence given any i, this event can be bounded by at most $\frac{\sigma_e}{2^{r_d+r_3-b}}$. Varying over all $i \in \mathbb{D}$, we bound the probability of this case by $\frac{\sigma_d\sigma_e}{2^{r_d+r_3-b}}$.

Case 3: $p'_i > p_i$ and $\exists i' \in \mathbb{E}, j \in [m_{i'}]$ s.t. $X^*_{i,p'_i+1} = X_{i',j}$. This corresponds to the case when the first nontrivial decryption query block matches a primitive query and follows a partial chain before and then matches an encryption query block. Hence, doing a similar analysis as in the event BAD3, the probability of this case occuring in the i th decryption query can be bounded by $\frac{q_p}{2^{r_d}} \times \frac{m_i \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{12}+r_{45}+r_8-s-b}}$ Summing over all $i \in \mathbb{D}$, we obtain the bound $\frac{\sigma_d q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{12}+r_d+r_8+r_{45}-b-s}}$.

Combining all the above three cases, we have

$$\Pr\left[\mathtt{BAD8} \wedge (\overline{\mathtt{BAD1}} \wedge \cdots \wedge \overline{\mathtt{BAD6}})\right] \leq \frac{\sigma_e + q_p}{2^{r_d}} + \frac{\sigma_d\sigma_e}{2^{r_d+r_3-b}} + \frac{\sigma_d q_p \mathsf{mcoll}(\sigma_e, 2^{b+s-r_{45}-r_8})}{2^{r_{12}+r_d+r_8+r_{45}-b-s}}. \tag{8}$$

Lemma 2 follows from Eqn. (1) - Eqn. (8) as we combine the probability of all the bad events and apply the union bound.

B.3 Proof of Lemma 3

Fix a good transcript ω. Note that all the input-output pairs for the underlying permutation are compatible. In the ideal world, all the Y values are sampled uniformly at random; the list ω_p is just the partial representation of Π, and all the decryption queries are degenerately aborted. Hence we get

$$\Pr\left[\Theta_0 = w\right] \leq \frac{1}{(2^b)^{\sigma_e}(2^b)_{q_p}}, \tag{9}$$

where σ_e denotes the total number of blocks present in all encryption queries including the nonce. In notation $\sigma_e = \sum_{i \in \mathbb{E}} l_i$.

Now let us look at the real world. In the real world, for ω we denote the encryption query, decryption query, and primitive query tuples by ω_e, ω_d, and ω_p, respectively. Then, we have

$$\begin{aligned}
\Pr[\Theta_1 = \omega] &= \Pr[\Theta_1 = (\omega_e, \omega_p, \omega_d)] \\
&= \Pr[\omega_e, \omega_p] \cdot \Pr[\omega_d \mid \omega_e, \omega_p] \\
&= \Pr[\omega_e, \omega_p] \cdot (1 - \Pr[\neg\omega_d \mid \omega_e, \omega_p]) \\
&\leq \Pr[\omega_e, \omega_p] \cdot \left(1 - \sum_{i \in \mathbb{D}} \Pr[\neg\omega_{d,i} \mid \omega_e, \omega_p]\right) \\
&\leq \frac{1}{(2^b)_{\sigma_e + q_p}} \cdot \left(1 - \sum_{i \in \mathbb{D}} \Pr[\neg\omega_{d,i} \mid \omega_e, \omega_p]\right) \qquad (10)
\end{aligned}$$

Here we have slightly abused the notation to use $\neg\omega_{d,i}$ to denote the event that the i-th decryption query successfully decrypts and $\neg\omega_d$ is the union $\cup_{i \in \mathcal{D}_2} \neg\omega_{d,i}$ (i.e. at least one decryption query successfully decrypts). The last inequality follows from the fact that the encryption and primitive queries are mutually permutation compatible.

From Eqn. 9 and Eqn. 10, we have

$$\frac{\Pr[\Theta_1 = w]}{\Pr[\Theta_0 = w]} \geq \left(1 - \sum_{i \in \mathbb{D}} \Pr[\neg\omega_{d,i} \mid \omega_e, \omega_p]\right). \qquad (11)$$

In the rest of this subsection, we will show

$$\sum_{i \in \mathbb{D}} \Pr[\neg\omega_{d,i} \mid \omega_e, \omega_p] \leq \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma + q_p)}{2^{r_d}},$$

which essentially completes the proof of the Lemma. To prove the above equation, recall that $\neg\omega_{d,i}$ occurs if and only if $\lfloor \Pi(X^*_{i,m_i}) \rfloor_\tau = T^*_i$, where X^*_{i,p'_i+1} is fresh. Note that, for all $p'_i + 1 < j \leq l_i$, $X^*_{i,j}$ values have been defined recursively as follows

$$X^*_{i,j} = D_1 \cdot \left(\Pi(X^*_{i,j-1})\right) \oplus D_3 \cdot B^*_{i,j}.$$

Now we make the following two important observations:

- If X^*_{i,p'_i+1} is not the last block, then the next input block may collide with some encryption or primitive input block with probability at most $\frac{\sigma_e + q_p}{2^{r_d} - \sigma_e - q_p}$. Applying this same argument for all the successive blocks till the last one, we get that if none of the previous block input collides then the probability that the last block input collides is at most $\frac{(\sigma_e + q_p + l^*_i - p'_i + 2)}{2^{r_d} - \sigma_e - q_p - l^*_i + p'_i + 2} \leq \frac{2(\sigma_e + q_p + m_i)}{2^{r_d}}$.
- If the last input block X^*_{i,l_i} is fresh, then $\Pi(X^*_{i,l^*_i}) = T^*_i$ with probability at most $2/2^\tau$, assuming $\sigma_e + q_p \leq 2^{b-1}$.

Let $\mathtt{E}_j$ denotes the event that $X^*_{i,j}$ is fresh and $\mathtt{E}$ denotes the event $\wedge_{j=p'_i+1}^{m_i}\mathtt{E}_j$ Applying the above two observations, we have

$$\begin{aligned}\sum_{i\in\mathbb{D}} \Pr_{\Theta_1}(\neg\omega_{d,i} \mid \omega_e,\omega_p) &\leq \Pr_{\Theta_1}(\neg\omega_{d,i}\wedge \mathtt{E} \mid \omega_e,\omega_p) + \Pr(\overline{\mathtt{E}}).\\ &\leq \sum_{i\in\mathbb{D}}\Big(\frac{2}{2^\tau} + \sum_{j=p'_i+1}^{l^*_i}\frac{\sigma_d+\sigma_e+q_p}{2^{r_d-1}}\Big).\\ &\leq \sum_{i\in\mathbb{D}}\frac{2}{2^\tau} + \frac{2m_i(\sigma_e+q_p+\sigma_d)}{2^{r_d}}\\ &\leq \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma+q_p)}{2^{r_d}}.\end{aligned}$$

This completes the proof of 3.