On the Security of Sponge-type Authenticated Encryption Modes

Bishwajit Chakraborty, Ashwin Jha and Mridul Nandi

Indian Statistical Institute, Kolkata, India
{bishu.math.ynwa,ashwin.jha1991,mridul.nandi}@gmail.com

Abstract. The sponge duplex is a popular mode of operation for constructing authenticated encryption schemes. In fact, one can assess the popularity of this mode from the fact that around 25 out of the 56 round 1 submissions to the ongoing NIST lightweight cryptography (LwC) standardization process are based on this mode. Among these, 14 sponge-type constructions are selected for the second round consisting of 32 submissions. In this paper, we generalize the duplexing interface of the duplex mode, which we call Transform-then-Permute. It encompasses Beetle as well as a new sponge-type mode SpoC (both are round 2 submissions to NIST LwC). We show a tight security bound for Transform-then-Permute based on b-bit permutation, which reduces to finding an exact estimation of the expected number of multi-chains (defined in this paper). As a corollary of our general result, authenticated encryption advantage of Beetle and SpoC is about $\frac{T(D+r2^r)}{2^b}$ where T, D and r denotes the number of offline queries (related to time complexity of the attack), number of construction queries (related to data complexity) and rate of the construction (related to efficiency). Previously the same bound has been proved for Beetle under the limitation that $T \ll \min\{2^r, 2^{b/2}\}$ (that compels to choose larger permutation with higher rate). In the context of NIST LwC requirement, SpoC based on 192-bit permutation achieves the desired security with 64-bit rate, which is not achieved by either duplex or Beetle (as per the previous analysis).

Keywords: Sponge · duplex · Beetle · SpoC · lightweight · AEAD · tight bound

1 Introduction

The Sponge function was first proposed by Bertoni et al. at the ECRYPT Hash Workshop [BDPA07], as a mode of operation for variable output length hash functions. It received instant attention due to NIST's SHA-3 competition, which had several candidates based on the Sponge paradigm. Most notably, JH [Wu11] and Keccak [BDPA13] were among the five finalists, and Keccak became the eventual winner. In time, the Sponge mode found applications in message authentication [BDPA07, BDPA11b], pseudorandom sequence generation [BDPA10], and the duplex mode [BDPA11a] for authenticated encryption. In particular, the recently concluded CAESAR competition for the development of authenticated encryption with associated data (AEAD) schemes had received a dozen Sponge-based submissions. Ascon [DEMS16], a winner in lightweight applications (resource constrained environments) use-case of the CAESAR competition, also uses the duplex mode of authenticated encryption.

The Sponge construction is also one of the go-to mode of operation for designing lightweight cryptographic schemes. This is quite evident from the design of hash functions such as Quark [AHMN10], PHOTON [GPP11], and SPONGENT [BKL+13], and authenticated encryption schemes such as Ascon [DEMS16] and Beetle [CDNY18]. In fact, majority of the submissions to the ongoing NIST lightweight cryptography (LwC) standardization process are inspired by the Sponge paradigm.

At a very 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, data absorption and squeezing is done via the rate part, i.e. r bits at a time. SpoC [AGH+19], a round 2 submission to NIST LwC standardization process, is a notable exception, where the absorption is done via the capacity part and the squeezing is done via the rate part. In [BDPA08], 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, it seems that 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 [BDPA11a]. 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. [BDPA11a] proved that duplex is as strong as Sponge, i.e. secure up to $DT \ll 2^c$. Mennink et al. [MRV15] 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. Jovanovic et al. [JLM14] proved 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. Note that integrity has an additional restriction that $D \ll 2^{c/2}$, where D is dominated by the decryption data complexity. Daemen et al. [DMA17] gave a generalization of duplex that has built-in multi-user security. Very recently, a tight privacy analysis [JLM+19] is provided. However, one of the dominating restrictions present in all of the existing integrity analysis of duplex authenticated encryption is $DT \ll 2^c$. Moreover, no forgery attack with a matching bound is known. A recent variant of duplex mode, called the Beetle mode of operation [CDNY18], modifies the duplexing phase by introducing a combined feedback based absorption/squeezing, similar to the feedback paradigm of CoFB [CIMN17]. In [CDNY18], Chakraborti et al. showed that feedback based duplexing actually helps in improving the security bound, mainly to get rid of the condition $DT \ll 2^c$. They showed privacy up to $DT \ll 2^b$, $D \ll 2^{b/2}$, $T \ll 2^c$, and integrity up to $D \ll \min\{2^{b/2}, 2^{c-\log_2 r}, 2^r\}$, $T \ll \min\{2^{c-\log_2 r}, 2^r, 2^{b/2}\}$, with the assumptions that $\kappa = c$ and $\tau = r$.

1.1.1 Security of Sponge-type AEAD in Light of NIST LwC Requirement

In the call for submissions of NIST LwC standardization process, it is mentioned that the primary version of any AEAD submission should have at least 128-bit key, at least 96-bit nonce, at least 64-bit tag, data complexity $2^{50} - 1$ bytes, and time complexity 2^{112}. In order to satisfy these requirements, a traditional duplex-based scheme must have a capacity size of at least 160-bit. All duplex-based submissions to NIST LwC standardization process use at least 192-bit capacity, except CLX [WH19] for which no security proof is available.

On the other hand, the known bound for Beetle imposes certain limitations on the state size and rate. Specifically, Beetle-based schemes require approximately 120-bit capacity and approximately 120-bit rate to achieve NIST LwC requirements. This means that we need a permutation of size at least 240 bits. In light of the ongoing NIST LwC standardization process, it would be interesting to see whether these limitations can be relaxed for Beetle.

1.2 Our Contributions

In this paper, inspired by the NIST LwC requirements, we extend a long line of research on the security of Sponge-type AEAD schemes. We study Sponge-type AEAD construction with a generalization of the feedback function used in the duplexing interface, that encompasses the feedback used in duplex, Beetle, SpoC etc. We show that for a class of feedback function, containing the Beetle and SpoC modes, optimal AEAD security is achieved. To be specific, we show that the AEAD security of this generalized construction is bounded by adversary's ability of constructing a special data structure, called the *multi-chains*. We also show a matching attack exploiting the multi-chains. As a corollary of this we give

1. a security proof validating the security claims of SpoC, and
2. an improved and tight bound for Beetle.

We also derive improved security bounds for a variant of Beetle, called PHOTON-Beetle [BCD+19], which is a round 2 candidate in ongoing NIST LwC standardization process. In particular our analysis shows that both the primary and secondary recommendations of PHOTON-Beetle achieve security as per NIST requirements.

In fact, we show that both Beetle and SpoC achieve NIST LwC requirements with just 138-bit capacity and $\geq$ 46-bit rate. In other words, they achieve NIST LwC requirements with just 184-bit state, which to the best of our knowledge is the smallest possible state size among all known Sponge-type constructions which are proven to be secure.

1.3 Organization of the Paper

In section 2, we define different notations used in the paper. We give a brief description of the design and security models of AEAD. We also give a brief description of coefficient H technique [Pat91, Pat08]. In section 3, we propose a Sponge-type AEAD construction called Transform-then-Permute (or TtP) with a generalization of the feedback function used in the duplexing interface. In section 4, we state some multicollision results with proofs which are used in the paper. In section 5, we define what we call the multi-chain structure and give an upper bound on the expected number of multi-chains that can be formed by an adversary in a special case. In section 6, using the multi-chain security game from section 5 we give a complete AEAD security proof for TtP (see Theorem 4) in the special case when the feedback function is invertible. In section 7, we show that the TtP generalization encompasses the feedback functions used in Sponge AE, Beetle, SpoC etc. Particularly, Beetle and SpoC modes fall under the class where the feedback functions are invertible and hence for those modes optimal AEAD security is achieved. Finally in section 8, we give some attack strategies to justify the tightness of our bound.

2 Preliminaries

NOTATIONAL SETUP: For $n \in \mathbb{N}$, $(n]$ denotes the set $\{1, 2, \ldots, n\}$ and $[n]$ denotes the set $\{0\} \cup (n]$, $\{0,1\}^n$ denotes the set of bit strings of length n, $\{0,1\}^+ := \bigcup_{n \geq 0} \{0,1\}^n$ and $\mathsf{Perm}(n)$ denotes the set of all permutations over $\{0,1\}^n$.

For any bit string x with $|x| \geq n$, $\lceil x \rceil_n$ (res. $\lfloor x \rfloor_n$) denotes the most (res. least) significant n bits of x. For $n, k \in \mathbb{N}$, such that $n \geq k$, we define the falling factorial $(n)_k := n!/(n-k)! = n(n-1)\cdots(n-k+1)$.

For $q \in \mathbb{N}$, x^q denotes the q-tuple $(x_1, x_2, \ldots, x_q)$. For $q \in \mathbb{N}$, for any set $\mathcal{X}$, $(\mathcal{X})_q$ denotes the set of all q-tuples with distinct elements from $\mathcal{X}$. Two distinct strings $a = a_1 \ldots a_m$ and $b = b_1 \ldots b_{m'}$, are said to have a common prefix of length $n \leq \min\{m, m'\}$, if $a_i = b_i$ for all $i \in (n]$, and $a_{n+1} \neq b_{n+1}$. For a finite set $\mathcal{X}$, $\mathsf{X} \leftarrow_{\$} \mathcal{X}$ denotes the uniform sampling of X from $\mathcal{X}$ which is independent to all other previously sampled random variables. $(\mathsf{X}_1, \ldots, \mathsf{X}_t) \xleftarrow{\text{wor}} \mathcal{X}$ denotes uniform sampling of t random variables $\mathsf{X}_1, \ldots, \mathsf{X}_t$ from $\mathcal{X}$ without replacement.

2.1 Authenticated Encryption: Definition and Security Model

Authentication Encryption with Associated Data: An authenticated encryption scheme with associated data functionality, or AEAD in short, is a tuple of deterministic algorithms $\mathsf{AE} = (\mathsf{E}, \mathsf{D})$, defined over the *key space* $\mathcal{K}$, *nonce space* $\mathcal{N}$, *associated data space* $\mathcal{A}$, *message space* $\mathcal{M}$, *ciphertext space* $\mathcal{C}$, and *tag space* $\mathcal{T}$, where:

$$\mathsf{E} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T} \quad \text{and} \quad \mathsf{D} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \cup \{\bot\}.$$

Here, E and D are called the encryption and decryption algorithms, respectively, of AE. Further, it is required that $\mathsf{D}(K, N, A, \mathsf{E}(K, N, A, M)) = M$ for any $(K, N, A, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$. For all key $K \in \mathcal{K}$, we write $\mathsf{E}_K(\cdot)$ and $\mathsf{D}_K(\cdot)$ to denote $\mathsf{E}(K, \cdot)$ and $\mathsf{D}(K, \cdot)$, respectively. In this paper, we have $\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T} \subseteq \{0, 1\}^+$ and $\mathcal{C} = \mathcal{M}$, so we use $\mathcal{M}$ instead of $\mathcal{C}$ wherever necessary.

AEAD Security in the Random Permutation Model: Let $\Pi \leftarrow_\$ \mathsf{Perm}(b)$, Func denote the set of all functions from $\mathcal{N} \times \mathcal{A} \times \mathcal{M}$ to $\mathcal{M} \times \mathcal{T}$ such that for any input $(*, *, M)$ the output is of length $|M| + t$ for some predefined constant t and $\Gamma \leftarrow_\$ \mathsf{Func}$. Let $\bot$ denote the degenerate function from $(\mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T})$ to $\{\bot\}$. For brevity, we denote the oracle corresponding to a function (like E, Π etc.) by that function itself. A bidirectional access to Π is denoted by the superscript $\pm$.

Definition 1. Let AE_Π be an AEAD scheme, based on the random permutation Π, defined over $(\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T})$. The AEAD advantage of any nonce respecting adversary $\mathscr{A}$ against AE_Π is defined as,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{AE}_\Pi}(\mathscr{A}) := \left| \Pr_{\substack{\mathsf{K} \leftarrow_\$ \mathcal{K} \\ \Pi^\pm}} \left[\mathscr{A}^{\mathsf{E}_\mathsf{K}, \mathsf{D}_\mathsf{K}, \Pi^\pm} = 1 \right] - \Pr_{\Gamma, \Pi^\pm} \left[\mathscr{A}^{\Gamma, \bot, \Pi^\pm} = 1 \right] \right|. \tag{1}$$

Here $\mathscr{A}^{\mathsf{E}_\mathsf{K}, \mathsf{D}_\mathsf{K}, \Pi^\pm}$ denotes $\mathscr{A}$'s response after its interaction with E_K, D_K, and $\Pi^\pm$, respectively. Similarly, $\mathscr{A}^{\Gamma, \bot, \Pi^\pm}$ denotes $\mathscr{A}$'s response after its interaction with Γ, $\bot$, and $\Pi^\pm$.

In this paper, we assume that the adversary is nonce-respecting, i.e. it never makes more than one encryption queries with same nonce. We further assume that the adversary is non-trivial, i.e. it never makes a duplicate query, and it never makes a query for which the response is already known due to some previous query. We use the following notations to parameterize the adversary's resources:

- q_e and q_d denote the number of queries to E_K and D_K, respectively. σ_e and σ_d denote the total number of blocks of input (associated data and message) across all encryption and decryption (respectively) queries where (informally), number of blocks per query is determined by the total number of primitive calls required to process the input (see 3.1 for formal definition). We sometime also write $q = q_e + q_d$ and $\sigma = \sigma_e + \sigma_d$ to denote the combined construction query resources which can be interpreted as the *online or data complexity* D from section 1.

- q_f and q_b denote the number of queries to Π^+ and Π^-, respectively. We sometime also use $q_p = q_f + q_b$, to denote the combined primitive query resources which can be interpreted as the *offline or time complexity* T from section 1.

Any adversary that adheres to the above mentioned resource constraints is called a $(q_p, q_e, q_d, \sigma_e, \sigma_d)$*-adversary or simply* (q_p, σ)*-adversary.*

2.2 Coefficient H Technique

Consider a computationally unbounded and deterministic adversary $\mathscr{A}$ that tries to distinguish between two oracles $\mathcal{O}_0$ and $\mathcal{O}_1$ via black box interaction with one of them. We denote the query-response tuple of $\mathscr{A}$'s interaction with its oracle by a transcript ω. Sometimes, this may also include any additional information that the oracle chooses to reveal to the distinguisher at the end of the query-response phase of the game. We will consider this extended definition of transcript. We denote by Θ_1 (res. Θ_0) the random transcript variable when $\mathscr{A}$ interacts with $\mathcal{O}_1$ (res. $\mathcal{O}_0$). The probability of realizing a given transcript ω in the security game with an oracle $\mathcal{O}$ is known as the *interpolation probability* of ω with respect to $\mathcal{O}$. Since $\mathscr{A}$ is deterministic, this probability depends only on the oracle $\mathcal{O}$ and the transcript ω. A transcript ω is said to be *attainable* if $\Pr[\Theta_0 = \omega] > 0$. In this paper, $\mathcal{O}_1 = (\mathsf{E_K}, \mathsf{D_K}, \Pi^{\pm})$, $\mathcal{O}_0 = (\Gamma, \perp, \Pi^{\pm})$, and the adversary is trying to distinguish $\mathcal{O}_1$ from $\mathcal{O}_0$ in AEAD sense. Now we state a simple yet powerful tool due to Patarin [Pat91], known as the coefficient H technique (or simply the H-technique). A proof of this theorem is available in multiple papers including [Pat08, CS14, MN17].

Theorem 1 (H-technique [Pat91, Pat08])**.** *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 for any adversary $\mathscr{A}$, we have the following bound on its AEAD distinguishing advantage:
$$\mathbf{Adv}^{\mathsf{aead}}_{\mathcal{O}_1}(\mathscr{A}) \leq \epsilon_{\mathsf{bad}} + \epsilon_{\mathsf{ratio}}.$$

3 Transform-then-Permute Construction

In this section we describe $\mathsf{Transform\text{-}then\text{-}Permute}$ (or TtP in short), which generalizes duplexing method used in sponge AEAD encompassing many other constructions such as Beetle, SpoC etc.

3.1 Parameters and Components

We first describe some parameters of our wide family of AEAD algorithms.

1. State-size: The underlying primitive of the construction is a b-bit public permutation. We call b state size of the permutation.
2. Key-size: Let κ denote the key-size. Here we assume $\kappa < b$.
3. Nonce-size: In this paper we consider fixed size nonce. Let ν denote the size of nonce.
4. Rate: Let $r, r' \leq b$ denote the rate of processing message and associate data respectively. The capacity is defined as $c := b - r$.

Let $\mathbb{N}_0$ be the set of all non-negative integers and $\theta := b - \kappa - \nu$. For $x \in \mathbb{N}_0$, we define
$$a(x) := \begin{cases} 0 & \textit{if } x \leq \theta \\ \lceil \frac{x-\theta}{r'} \rceil & \textit{otherwise} \end{cases}$$

Parsing Function: Let $D = N \| A$ where $N \in \{0,1\}^{\nu}$ and $A \in \{0,1\}^*$ with $a := a(|A|)$.

– **Case** $|A| \le \theta$: $\mathsf{parse}(N, A) = D \parallel 0^{\theta-|A|} \in \{0,1\}^{b-\kappa}$.
– **Case** $|A| > \theta$: $\mathsf{parse}(N, A) := (IV, A_1, \ldots, A_a)$ where $D = IV \| D'$, $IV \in \{0,1\}^{b-\kappa}$ and $(A_1, \ldots, A_a) \xleftarrow{r'} D'$. Note that $|D'| = |A| - \theta$ and so when we parse D' to blocks of size r', we get $a(|A|) = \lceil \frac{|A|-\theta}{r'} \rceil$ many blocks.

In addition to parsing $N\|A$, we also parse a message or ciphertext Z as $(Z_1, \ldots, Z_m) \xleftarrow{r} Z$ into m blocks of size r where $m = \lceil |Z|/r \rceil$.

We define $t := a + m$ to be the total number of blocks corresponding to a input query of the form (N, A, Z).

Domain Separation: To every pair of non-negative integers $(|A|, |Z|)$ with $a = a(|A|)$, $m = \lceil |Z|/r \rceil$, and for every $0 \le i \le a + m$, we associate a small integer δ_i where

$$\delta_i = \begin{cases} 0 & \textit{if } i \notin \{a\} \cup \{t\} \\ 1 & \textit{if } (i = a \wedge r' \mid |A| - \theta) \vee (i = t \wedge r \mid |M|) \\ 2 & \textit{otherwise.} \end{cases}$$

We collect all these δ values through the following function $\mathsf{DS}(|A|, |Z|) = (\delta_0, \delta_1, \ldots, \delta_{a+m})$.

Encoding Function: Let $\mathcal{D}_{DS} := \{0,1\}^2 \times \{0,1,2\}$ and $r_{\max} = \max\{r, r'\}$. Let

$$\mathsf{encode} : \{0,1\}^{\le r_{\max}} \times \mathcal{D}_{DS} \to \{0,1\}^b$$

be an injective function such that for any $D, D' \in \{0,1\}^x$, $1 \le x \le r_{\max}$ and for all $\Delta \in \mathcal{D}_{DS}$, we have $\mathsf{encode}(D, \Delta) \oplus \mathsf{encode}(D', \Delta) = 0^{b-x} \| (D \oplus D')$. Actual description of this encode function is determined by the construction.

Format Function: We define a formatting function Fmt which maps a triple (N, A, M) to $(D_0, \ldots, D_{a+m}) \in (\{0,1\}^b)^{a+m+1}$ where $a := a(|A|)$ and $m = \lceil |Z|/r \rceil$. The exact description of format function is described in Algorithm 1.

Algorithm 1 Description of the format function (Fmt)

```
function FMT(N, A, Z)
    a ← a(|A|), m ← ⌈|Z|/r⌉
    (A_0, A_1, ..., A_a) ← Parse(N, A)
    (Z_1, ..., Z_m) ←r− Z
    (δ_0, ..., δ_t) ← DS(|A|, |Z|)
    for i = 0 to a do
        if i = a and m = 0 then
            D_i ← encode(A_i, (0, 1, δ_i))
        else
            D_i ← encode(A_i, (0, 0, δ_i))
    for i = 1 to m do
        D_{a+i} ← encode(Z_i, (1, 0, δ_{i+m}))
    return (D_0, ..., D_t)
```

Feedback functions: We also need some linear functions $L_{ad}, L_e : \{0,1\}^b \to \{0,1\}^b$ which are used to process associate data and message respectively in an encryption algorithm.

Now, given a linear function $L : \{0,1\}^b \to \{0,1\}^b$, $1 \le x \le r$, the following function $L' : \{0,1\}^b \times \{0,1\}^x \times \mathcal{D}_{DS} \to \{0,1\}^b \times \{0,1\}^x$, is used to process the j-th block Z (either

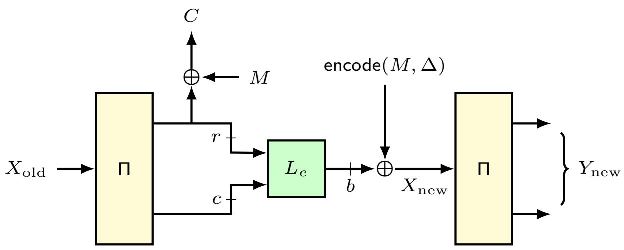

Figure 1: Illustration of the feedback process for a message block M of $|M|$ bits. Here $\mathsf{encode}(M, \Delta)$ represents some encoding of $|M|$ bits string to a b-bit string as described above and L_e is a linear transformation applied on b-bit strings.

a plaintext or a ciphertext) using the output Y of the previous invocation of the random permutation:

$$L'(Y, Z, \Delta) = (X := L(Y) \ \oplus \mathsf{encode}(Z, \Delta),\ Z' := \lceil Y \rceil_{|Z|} \ \oplus Z)$$

For $1 \le i \le r$, let $L_{d,i}(x)$ to denote the linear function $L_e(x) \oplus 0^{b-i} \| \lceil x \rceil_i$. Then, it is easy to see from the property of encoding function that $L'_{d,|C|}(Y, C, \Delta) = (X, M)$ if and only if $L'_e(Y, M, \Delta) = (X, C)$. Figure 1 provides an illustration how a message block is processed.

3.2 Description of the Transform-then-Permute AEAD

We describe the Transform-then-Permute construction in Algorithm 2 which generalizes duplexing method used in sponge-type AEADs. Figure 2 illustrates a simple case when $|N| = b - \kappa$.

Algorithm 2 Description of Encryption/Decryption algorithms of the Transform-then-Permute mode with Associated data. $X = (x =_? y : p, q)$ means $X = p$ if $x = y$ and $X = q$ otherwise.

function $\mathsf{Enc}(K, N, A, M)$
  $a \leftarrow a(|A|)$, $m \leftarrow \lceil |M|/r \rceil$
  $(D_0, D_1, \ldots, D_{a+m}) \leftarrow \mathsf{Fmt}(N, A, M)$
  $(M_1, \ldots, M_m) \xleftarrow{r} M$
  $X_0 \leftarrow K \| 0^{b-\kappa} \oplus D_0$
  $Y_0 \leftarrow \Pi(X_0)$
  for $i = 1$ to a **do**
    $X_i \leftarrow L_{ad}(Y_{i-1}) \oplus D_i$
    $Y_i \leftarrow \Pi(X_i)$
  for $j = 1$ to m **do**
    $i = a + j$
    $X_i \leftarrow L_e(Y_{i-1}) \oplus D_i$
    $C_j \leftarrow M_j \oplus \lceil Y_{i-1} \rceil_{|M_j|}$
    $Y_i \leftarrow \Pi(X_i)$
  $T \leftarrow \lceil Y_{a+m} \rceil_\tau$
  return $(C_1 \| \ldots \| C_m, T)$

function $\mathsf{Dec}(K, N, A, C, T)$
  $a \leftarrow a(|A|)$, $m \leftarrow \lceil |C|/r \rceil$
  $(D_0, D_1, \ldots, D_{a+m}) \leftarrow \mathsf{Fmt}(N, A, C)$
  $(C_1, \ldots, C_m) \xleftarrow{r} C$
  $X_0 \leftarrow K \| 0^{b-\kappa} \oplus D_0$
  $Y_0 \leftarrow \Pi(X_0)$
  for $i = 1$ to a **do**
    $X_i \leftarrow L_{ad}(Y_{i-1}) \oplus D_i$
    $Y_i \leftarrow \Pi(X_i)$
  for $j = 1$ to m **do**
    $i = a + j$
    $X_i \leftarrow L_{d,|C_i|}(Y_{i-1}) \oplus D_i$
    $M_j \leftarrow C_j \oplus \lceil Y_{i-1} \rceil_{|C_j|}$
    $Y_i \leftarrow \Pi(X_i)$
  $T \leftarrow \lceil Y_{a+m} \rceil_\tau$
  return $T' =_? T : M_1 \| \ldots \| M_m, \perp$

Lemma 1. *Given any two tuples $(N, A, Z) \neq (N', A', Z')$ and $\mathsf{Fmt}(N, A, Z) = (D_0, \ldots, D_t)$ and $\mathsf{Fmt}(N', A', Z') = (D'_0, \ldots, D'_{a'+m'})$, we have*

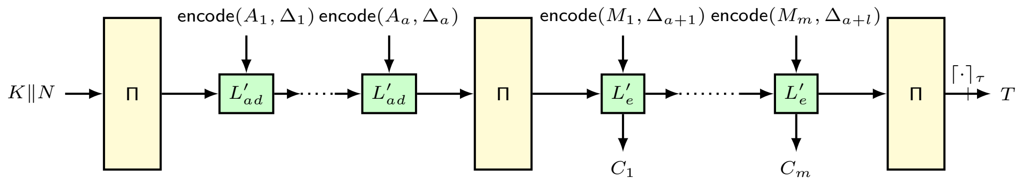

Figure 2: Schematic of the Transform-then-Permute AEAD mode. Here we assume $|N| = b - \kappa$, $L'_{ad}(Y, A) = L_{ad}(Y) \oplus A$. L_{ad}, L'_e, encode functions and Δ values are as described before.

1. $(D'_0, \ldots, D'_a) \neq (D_0, \ldots, D_a)$ *whenever* $(N, A) \neq (N', A')$ *and* $a \leq a'$.

2. $(D'_a, \ldots, D'_t) \neq (D_a, \ldots, D_t)$ *whenever* $(N, A) = (N', A')$ *and* $m \leq m'$.

Proof. We write $\mathsf{parse}(N, A) = (A_0, A_1, \ldots, A_a)$ and $\mathsf{parse}(N', A') = (A'_0, A'_1, \ldots, A'_{a'})$.

1. Let $(N, A) \neq (N', A')$. Then we have $(A_0, A_1, \ldots, A_a) \neq (A'_0, A'_1, \ldots, A'_{a'})$. Now if, $a < a'$ then we have $D_a = \mathsf{encode}(A_a, 0, \delta)$ where $\delta \in \{1, 2\}$ and $D'_a = \mathsf{encode}(A'_a, 0, 0)$. Hence by injectivity of encode we have $D_a \neq D'_a$. If $a = a'$ then there exists non-negative $i \leq a$ such that $A_i \neq A'_i$ and hence $D_i \neq D'_i$.

2. Let $(N, A) = (N, A')$. Then we have $(A_0, A_i, \ldots, A_a) = (A'_0, A'_i, \ldots, A'_a)$. Note that m, m' both cannot be 0. So if $m = 0$, then $m' > 0 \implies D_a = \mathsf{encode}(A_a, 0, \delta)$ for some $\delta \in \{1, 2\}$ and $D'_a = \mathsf{encode}(A_a, 0, 0)$. Hence $D_a \neq D'_a$. Let $m, m' > 0$ then if, $m < m'$ then we have $D_t = \mathsf{encode}(M_m, 1, \delta)$ where $\delta \in \{1, 2\}$ and $D'_a = \mathsf{encode}(M'_m, 1, 0)$. Else if $m = m'$, then there exists positive $i \leq m$ such that $M_i \neq M'_i$. Hence $D_{a+i} \neq D'_{a+i}$.

□

We analyze the security of Transform-then-Permute in section 6. In this paper, our main focus is the analysis of TtP with invertible L_e and L_d. However, we also present some results when the invertibility condition on L_d is relaxed.

4 Some Results on Multicollision

In this section, we briefly revisit some useful results on the expected value of maximum multicollision in a random sample. This problem has seen a lot of interest (see for instance [Gon81, BYG91, SF96, RS98]) in context of the complexity of hash table[1] probing. However, most of the results available in the literature are given in asymptotic forms. We state some relevant results in a more concrete form, following similar proof strategies and probability calculations as before. Moreover, we also extend these results for samples which, although are not uniform, have high entropy, almost close to uniform.

4.1 Expected Maximum Multicollision in a Uniform Random Sample

Let $\mathsf{X}_1, \ldots, \mathsf{X}_q \leftarrow\!\!\$\; \mathcal{D}$ where $|\mathcal{D}| = N$ and $N \geq 2$. We denote the maximum multicollision random variable for the sample as $\mathsf{mc}_{q,N}$. More precisely, $\mathsf{mc}_{q,N} = \max_a |\{i : \mathsf{X}_i = a\}|$.

[1] A popular data structure used for efficient searching applications.

For any integer $\rho \geq 2$,

$$\begin{aligned}\Pr[\mathsf{mc}_{q,N} \geq \rho] &\leq \sum_{a \in \mathcal{D}} \Pr[|\{i : \mathsf{X}_i = a\}| \geq \rho] \\ &\leq N \cdot \frac{\binom{q}{\rho}}{N^\rho} \\ &\leq N \cdot \frac{q^\rho}{N^\rho \rho!} \\ &\leq N \cdot \left(\frac{qe}{\rho N}\right)^\rho.\end{aligned}$$

We justify the inequalities in the following way: The first inequality is due to the union bound. If there are at least ρ indices for which X_i takes value a, we can choose the first ρ indices in $\binom{q}{\rho}$ ways. This justifies the second inequality. The last inequality follows from the simple observation that $e^\rho = \sum_{i \geq 0} \rho^i/i! \geq \rho^\rho/\rho!$. Thus, we have

$$\Pr[\mathsf{mc}_{q,N} \geq \rho] \leq N \cdot \left(\frac{qe}{\rho N}\right)^\rho. \tag{2}$$

For any positive integer valued random variable Y bounded above by q, we define another random variable Y' as

$$\mathsf{Y}' = \begin{cases} \rho - 1 & \text{if } \mathsf{Y} < \rho \\ q & \text{otherwise.} \end{cases}$$

Clearly, $\mathsf{Y} \leq \mathsf{Y}'$ and

$$\mathsf{Ex}[\mathsf{Y}] \leq (\rho - 1) + q \cdot \Pr[\mathsf{Y} \geq \rho].$$

Using Eq. (2), and the above relation we can prove the following results for the expected value of maximum multicollision. We write $\mathsf{mcoll}(q, N)$ to denote $\mathsf{Ex}[\mathsf{mc}_{q,N}]$. So from the above relation,

$$\mathsf{mcoll}(q, N) \leq (\rho - 1) + qN \cdot \left(\frac{qe}{\rho N}\right)^\rho \tag{3}$$

for all positive ρ. We use this relation to prove an upper bound of $\mathsf{mcoll}(q, N)$ by plugging in some suitable value for ρ.

Proposition 1. *For $N \geq 4$, $n = \log_2 N$,*

$$\mathsf{mcoll}(q, N) \leq \begin{cases} \frac{4 \log_2 q}{\log_2 \log_2 q} & \textit{if } 4 \leq q \leq N \\ 5n \left\lceil \frac{q}{nN} \right\rceil & \textit{if } N < q \end{cases}$$

Proof. We first prove the result when $q = N$. A simple algebra shows that for $n \geq 2$, $\left(\frac{e \log_2 n}{4n}\right) \leq n^{-\frac{1}{2}}$. In other words, $\left(\frac{e}{\rho}\right)^\rho \leq N^{-2}$ where $\rho = 4n/\log_2 n$. So

$$\mathsf{mcoll}(q, N) \leq \rho - 1 + N^2 \cdot \left(\frac{e}{\rho}\right)^\rho \leq \rho.$$

When $q < N$, we can simply bound $\mathsf{Ex}[\mathsf{mc}_{q,N}] \leq \mathsf{Ex}[\mathsf{mc}_{q,q}] \leq \frac{4 \log_2 q}{\log_2 \log_2 q}$.

For $N < q \leq Nn$, we choose $\rho = 4n$. Now,

$$\begin{aligned}\mathsf{mcoll}(q, N) &\leq 4n - 1 + nN^2 \times \left(\frac{e}{4}\right)^{4n} \\ &\leq 4n - 1 + nN^2/4^n \leq 5n.\end{aligned}$$

When $q \geq nN$, we can group them into $\lceil q/nN \rceil$ samples each of size exactly nN (we can add more samples if required). This would prove the result when $q \geq nN$. □

Remark 1. Note that, similar bound as in Proposition 1 can be achieved in the case of non-uniform sampling. For example, when we sample $\mathsf{X}_1, \ldots, \mathsf{X}_q \xleftarrow{\text{wor}} \{0,1\}^b$ and then define $\mathsf{Y}_i = \lceil \mathsf{X}_i \rceil_r$ for some $r < b$. In this case, we have

$$\Pr(\mathsf{Y}_{i_1} = a, \cdots, \mathsf{Y}_{i_\rho} = a) \le \frac{(2^{b-r})_\rho}{(2^b)_\rho} \le \frac{1}{2^{r\rho}}.$$

This can be easily justified as we have to choose the remaining $b - r$ bits distinct (as $\mathsf{X}_1, \ldots, \mathsf{X}_q$ must be distinct). So, same bound as given in Proposition 1 can be applied for this distribution.

4.2 A Special Example of Non-Uniform Random Sample

In this paper we consider the following non-uniform random samples. Let $x_1, \ldots x_q$ be distinct and $y_1, \ldots, y_q$ be distinct b bits. Let Π denote the random permutation over b bits, $\Pi^2 := \Pi \circ \Pi$ denotes the composition of Π with itself. We define $\mathsf{Z}_{i,j} = \Pi(x_i) \oplus \Pi^{-1}(y_j)$. Now, for all distinct $i_1, \ldots, i_\rho$, distinct $j_1, \ldots, j_\rho$ and $a \in \{0,1\}^b$, we want to bound $\Pr\left[\mathsf{Z}_{i_1,j_1} = a, \cdots, \mathsf{Z}_{i_\rho,j_\rho} = a\right]$. By abuse of notations we write both i_k and j_k as k.

Let $N := 2^b$. We can assume $a = 0^b$. Since otherwise, we consider $\Pi'(x) = \Pi(x) \oplus a$ which is also a random permutation and consider $y'_i = y_i \oplus a$ instead of y_i, $\forall 1 \le i \le \rho$. Note that y'_i's are clearly distinct. So the problem reduces to bounding

$$\begin{aligned}\theta &:= \Pr\left[\Pi^2(x_1) = y_1, \cdots, \Pi^2(x_\rho) = y_\rho\right] \\ &= \sum_{c^\rho} \Pr\left[\Pi(x_1) = c_1, \Pi(c_1) = y_1, \cdots, \Pi(x_\rho) = c_\rho, \Pi(c_\rho) = y_\rho\right]\end{aligned}$$

We say that c^ρ valid if $c_i = x_j$ if and only if $c_j = y_i$. The set of all such valid tuples is denoted as V. For any valid c^ρ, define $S := \{x_1, \ldots, x_\rho\} \cup \{c_1, \ldots, c_\rho\}$. Then, $\Pr\left[\Pi(x_1) = c_1, \Pi(c_1) = y_1, \cdots, \Pi(x_\rho) = c_\rho, \Pi(c_\rho) = y_\rho\right] = \frac{1}{(N)_{|S|}}$. On the other hand, if c^ρ is not valid then the above probability is zero. Let V_s be the set of all valid tuples for which $|S| = s$.

If $|S| = 2\rho - k$, then we must have exactly k many pairs $(i_1, j_1), \ldots, (i_k, j_k)$ such that $c_i = x_j$. Now the number of ways this k-many pairs can be chosen is bounded by ρ^{2k}. The remaining $\rho - k$ many c_i's can be chosen in $(N-k)_{\rho-k}$ ways. Hence, $|V_{2\rho-k}| \le \rho^{2k}(N-k)_{\rho-k}$.

$$\begin{aligned}\Pr\left[\Pi^2(x_i) = y_i \ \forall 1 \le i \le \rho\right] &= \sum_{s=\rho}^{2\rho} \sum_{c^\rho \in V_s} \Pr\left[\Pi(x_i) = c_i, \Pi(c_i) = y_i \ \forall 1 \le i \le \rho\right] \\ &\le \sum_{s=\rho}^{2\rho} \frac{|V_s|}{(N)_s} \le \sum_{k=0}^{\rho} \frac{|V_{2\rho-k}|}{(N)_{2\rho-k}} \\ &\le \sum_{k=0}^{\rho} \frac{\rho^{2k}(N-k)_{\rho-k}}{(N)_{2\rho-k}} \le \sum_{k=0}^{\rho} \frac{\rho^{2(\rho-k)}}{(N-2\rho)^\rho} \\ &\le \left(\sum_{k=0}^{\rho} \frac{1}{\rho^{2k}}\right) \cdot \left(\frac{\rho^2}{N-2\rho}\right)^\rho \le 2 \cdot \left(\frac{\rho^2}{N-2\rho}\right)^\rho\end{aligned}$$

Since the sample space $\{(x_i, y_j)\}_{i,j \in [q]}$ is of size q^2, we denote the maximum multicollision random variable for the sample as $\mathsf{mc}'_{q^2,N}$. Then we have by a similar analysis as in the previous section,

$$\Pr\left[\mathsf{mc}'_{q^2,N} \ge \rho\right] \le 2N \cdot \binom{q^2}{\rho} \cdot \left(\frac{\rho^2}{N-2\rho}\right)^\rho \le 2N\left(\frac{q^2 e\rho}{N-2\rho}\right)^\rho.$$

We write $\mathsf{mcoll}'(q^2, N)$ to denote $\mathsf{Ex}\left[\mathsf{mc}'_{q^2,N}\right]$. So from the above relation,

$$\mathsf{mcoll}'(q^2, N) \leq (\rho - 1) + 2q^2 N \cdot \left(\frac{q^2 e\rho}{N - 2\rho}\right)^{\rho}$$

Proposition 2. *For $b > 16$,*

$$\mathsf{mcoll}'(q^2, N) \leq \begin{cases} \frac{4b}{\log_2 b} & \text{if } b^2 q^2 \leq N, \\ \frac{4b}{\log_2 b}\left\lceil \frac{b^2 q^2}{N} \right\rceil & \text{if } b^2 q^2 > N. \end{cases}$$

Proof. Let $b^2 q^2 \leq N$. Since $N > 2^{16}$, $\rho = \frac{4b}{\log_2 b} \implies q^2 \leq \frac{N-2\rho}{\rho^2}$. Hence, $2q^2 N \cdot \left(\frac{q^2 e\rho}{N-2\rho}\right)^{\rho} \leq N^2 \cdot \left(\frac{e}{\rho}\right)^{\rho}$. Now, $\left(\frac{e}{\rho}\right)^{\rho} \leq \left(\frac{e}{4}\right)^{4b} \leq \frac{1}{N^2} \implies N^2 \cdot \left(\frac{e}{\rho}\right)^{\rho} \leq 1$.

Now for $q^2 \geq \frac{N}{b^2}$ we can group the q^2 samples into $\left\lceil \frac{b^2 q^2}{N} \right\rceil$ groups each of size exactly $\frac{N}{b^2}$ (we can add more samples if required). This would prove the bounds. □

4.2.1 A Generalization of the Non-Uniform Random Sample

Here, we study a generalization of of the above problem, which will be useful when a non-invertible linear function is sandwiched between two random permutation calls. For example, this happens in case of PHOTON-Beetle [BCD+19] and duplex [BDPA11a].

Let $x_1, \ldots x_q$ be distinct and $y_1, \ldots, y_q$ be distinct b-bit strings. Let Π denote the random permutation over b bits, $L : \{0,1\}^b \to \{0,1\}^b$ be a linear function with rank $\mathsf{rank}(L)$, and $\varphi := \Pi \circ L \circ \Pi$. We define $\mathsf{Z}_{i,j} = L(\Pi(x_i)) \oplus \Pi^{-1}(y_j)$. Now, for all distinct $i_1, \ldots, i_\rho$, distinct $j_1, \ldots, j_\rho$ and $a \in \{0,1\}^b$, we want to bound $\Pr\left[\mathsf{Z}_{i_1,j_1} = a, \cdots, \mathsf{Z}_{i_\rho,j_\rho} = a\right]$. By a slight abuse of notations we write both i_k and j_k as k.

Let $N := 2^b$. We can assume $a = 0^b$. Since otherwise, we consider $\Pi'(x) = \Pi(x \oplus a)$ which is also a random permutation and consider $x'_i = x_i \oplus a$ instead of x_i, $\forall 1 \leq i \leq \rho$. Note that x'_i's are clearly distinct. So the problem reduces to bounding

$$\begin{aligned} \theta &:= \Pr\left[\varphi(x_1) = y_1, \cdots, \varphi(x_\rho) = y_\rho\right] \\ &= \sum_{c^\rho} \Pr\left[\Pi(x_1) = c_1, \Pi(L(c_1)) = y_1, \cdots, \Pi(x_\rho) = c_\rho, \Pi(L(c_\rho)) = y_\rho\right] \end{aligned}$$

For all $1 \leq i \leq \rho$, let $d_i = L(c_i)$. We say that c^ρ *valid* if $d_i = x_j$ if and only if $c_j = y_i$. The set of all such valid tuples is denoted as V. For any valid c^ρ, define $S := \{x_1, \ldots, x_\rho\} \cup \{d_1, \ldots, d_\rho\}$. Then, $\Pr\left[\Pi(x_1) = c_1, \Pi(d_1) = y_1, \cdots, \Pi(x_\rho) = c_\rho, \Pi(d_\rho) = y_\rho\right] = \frac{1}{(N)_{|S|}}$. On the other hand, if c^ρ is not valid then the above probability is zero. Let V_s be the set of all valid tuples for which $|S| = s$.

We say that d_i is *old* if $d_i = x_j$ for $1 \leq i, j \leq \rho$. If $|S| = 2\rho - k$, then we must have exactly k many pairs $(i_1, j_1), \ldots, (i_k, j_k)$ such that $d_i = x_j$. Now, the number of ways these k-many pairs can be chosen is bounded by ρ^{2k}. This fixes all old d_i values. Then, the number of c_i values corresponding to old d_i values is bounded by at most $\rho^{2k}\tilde{N}^k$ where $\tilde{N} = 2^{b-\mathsf{rank}(L)}$, as once we fix the $b - \mathsf{rank}(L)$ bits there is a unique solution for $L(c_i) = d_i$. Once we fix the c_i values corresponding to the old d_i values, then the remaining $\rho - k$

many c_i's can be chosen in $(N-k)_{\rho-k}\tilde{N}^{\rho-k}$ ways. Hence, $|V_{2\rho-k}| \leq \rho^{2k}(N-k)_{\rho-k}\tilde{N}^{\rho}$.

$$\begin{aligned}\Pr[\varphi(x_i) = y_i \ \forall 1 \leq i \leq \rho] &= \sum_{s=\rho}^{2\rho} \sum_{c^\rho \in V_s} \Pr[\Pi(x_i) = c_i, \Pi(d_i) = y_i, \ \forall 1 \leq i \leq \rho] \\ &\leq \sum_{s=\rho}^{2\rho} \frac{|V_s|}{(N)_s} \leq \sum_{k=0}^{\rho} \frac{|V_{2\rho-k}|}{(N)_{2\rho-k}} \\ &\leq \sum_{k=0}^{\rho} \frac{\rho^{2k}(N-k)_{\rho-k}\tilde{N}^\rho}{(N)_{2\rho-k}} \leq \sum_{k=0}^{\rho} \frac{\rho^{2(\rho-k)}\tilde{N}^\rho}{(N-2\rho)^\rho} \\ &\leq \left(\sum_{k=0}^{\rho} \frac{1}{\rho^{2k}}\right) \cdot \left(\frac{\rho^2\tilde{N}}{N-2\rho}\right)^\rho \leq 2 \cdot \left(\frac{\rho^2\tilde{N}}{N-2\rho}\right)^\rho\end{aligned}$$

Since the sample space $\{(x_i, y_j)\}_{i,j\in[q]}$ is of size q^2, we denote the maximum multicollision random variable for the sample as $\widetilde{\mathsf{mc}}_{q^2,N,\mathsf{rank}(L)}$. Then, we have by a similar analysis as in the previous section,

$$\Pr\left[\widetilde{\mathsf{mc}}_{q^2,N,\mathsf{rank}(L)} \geq \rho\right] \leq 2N \cdot \binom{q^2}{\rho} \cdot \left(\frac{\rho^2\widetilde{N}}{N-2\rho}\right)^\rho \leq 2N\left(\frac{q^2e\rho\widetilde{N}}{N-2\rho}\right)^\rho.$$

We write $\widetilde{\mathsf{mcoll}}(q^2, N, \mathsf{rank}(L))$ to denote $\mathsf{Ex}\left[\widetilde{\mathsf{mc}}_{q^2,N,\mathsf{rank}(L)}\right]$. So from the above relation,

$$\widetilde{\mathsf{mcoll}}(q^2, N, \mathsf{rank}(L)) \leq (\rho-1) + 2q^2N \cdot \left(\frac{q^2e\rho\widetilde{N}}{N-2\rho}\right)^\rho.$$

Finally, we have the following upper bound on $\widetilde{\mathsf{mcoll}}(q^2, N, \mathsf{rank}(L))$.

Proposition 3. *For $b > 16$,*

$$\widetilde{\mathsf{mcoll}}(q^2, N, \mathsf{rank}(L)) \leq \begin{cases} \frac{4b}{\log_2 b} & \text{if } b^2q^2 \leq 2^{\mathsf{rank}(L)}, \\ \frac{4b}{\log_2 b}\left\lceil \frac{b^2q^2}{2^{\mathsf{rank}(L)}} \right\rceil & \text{if } b^2q^2 > 2^{\mathsf{rank}(L)}. \end{cases}$$

Proof. Let $b^2q^2 \leq 2^{\mathsf{rank}(L)}$. Since $N > 2^{16}$, $\rho = \frac{4b}{\log_2 b} \implies q^2 \leq \frac{N-2\rho}{\rho^2\tilde{N}}$. Hence, $2q^2N \cdot \left(\frac{q^2e\rho}{N-2\rho}\right)^\rho \leq N^2 \cdot \left(\frac{e}{\rho}\right)^\rho$. Now, $\left(\frac{e}{\rho}\right)^\rho \leq \left(\frac{e}{4}\right)^{4b} \leq \frac{1}{N^2} \implies N^2 \cdot \left(\frac{e}{\rho}\right)^\rho \leq 1$.

Now, for $b^2q^2 > 2^{\mathsf{rank}(L)}$ we can group the q^2 samples into $\left\lceil \frac{b^2q^2}{2^{\mathsf{rank}(L)}} \right\rceil$ groups each of size exactly $\frac{2^{\mathsf{rank}(L)}}{b^2}$ (we can add more samples if required). This would prove the bounds. □

Remark 2. Note that, for $\mathsf{rank}(L) = b$, $\widetilde{\mathsf{mcoll}}(q^2, N, \mathsf{rank}(L)) = \mathsf{mcoll}'(q^2, N)$.

4.3 Multicollisions in Context of the Analysis of Sponge-type AEAD

In later sections, we will use the bound on the expected number of multicollisions to give a tight security bound for Transform-then-Permute and some of its instantiations.

Here, we note that multicollisions have been previously studied in context with the duplex mode, most notably in [DMA17] and [JLM+19]. However, there is a fundamental difference between our approach and the previously used strategies in [DMA17, JLM+19]. In the following, r, c and b have their usual meaning in context of Sponge, i.e., $b = r + c$.

In [DMA17], the authors try to upper bound a parameter called the multicollision limiting function $\nu_{r,c}^q$. Assume we distribute q balls into 2^r bins, one at a time, where the bin for each ball is selected uniformly at random and independent of other choices. Then, $\nu^q_{r,c}$ is defined as the smallest natural number x such that $\Pr[\mathsf{mc}_{q,2^r} > x] < x/2^c$. On a closer inspection of the proof, one can see that the $\nu^q_{r,c}$ is dependent upon b and $\lambda = q2^{-r}$. The authors derive bounds for $\nu^q_{r,c}$, for three cases, viz. $\lambda < 1$, $\lambda = 1$, and $\lambda > 1$.

In [JLM+19], the authors upper bound $\Pr[\mathsf{mc}_{q,2^r} > \rho]$ to q/S, where $S = \min\{2^{b/2}, 2^c\}$ and ρ is viewed as a function of r and c. Basically, based on the value of r and c, they derive choices for ρ, such that the desired probability is bounded by q/S. To derive sharp bounds on ρ for various choices of r and c, they employ a detailed analysis involving Sterling's approximation and Lambert W function.

In contrast to the above strategies, we are interested in good estimates for the expectation of $\mathsf{mc}_{q,2^r}$ depending upon the relationship between q and 2^r. Further, our analysis is much more straightforward.

5 Multi-chain Security Game

In this section we consider a new security game which we call multi-chain security game. In this game, adversary $\mathscr{A}$ interacts with a random permutation and its inverse. It's goal is to construct multiple walks having same labels. We first need to describe some notations which would be required to define the security game.

5.1 Multi-Chain Structure

Labeled Walk: Let $\mathcal{L} = ((u_1, v_1), \ldots, (u_t, v_t))$ 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 $\mathsf{domain}(\mathcal{L}) = \{u_1, \ldots, u_t\}$ and $\mathsf{range}(\mathcal{L}) = \{v_1, \ldots, v_t\}$.

Let L be a linear function over b bits. Given such a list we define a labeled directed graph $\mathcal{G}^L_{\mathcal{L}}$ over the set of vertices $\mathsf{range}(\mathcal{L}) \subseteq \{0,1\}^b$ as follows: A directed edge $v_i \to v_j$ with label x (also denoted as $v_i \xrightarrow{x} v_j$) is in the graph if $L(v_i) \oplus x = u_j$. We can similarly extend this to a label walk $\mathcal{W}$ from a node w_0 to w_k as

$$\mathcal{W} : w_0 \xrightarrow{x_1} w_1 \xrightarrow{x_2} w_2 \cdots \xrightarrow{x_k} w_k.$$

We 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. We simply denote the directed graph $\mathcal{G}^L_{\mathcal{L}}$ by $\mathcal{G}_{\mathcal{L}}$ wherever the linear function L is understood from the context.

Definition 2. Let L be a fixed linear function over b bits. Let $r, \tau \leq b$ be some parameters. We say that a set of labeled walks $\{\mathcal{W}_1, \ldots, \mathcal{W}_p\}$ forms a *multi-chain with a label* $x := (x_1, \ldots, x_k)$ in the graph $\mathcal{G}_{\mathcal{L}}$ if for all $1 \leq i \leq p$, $\mathcal{W}_i : v^i_0 \xrightarrow{x} v^i_k$ and $\lceil u^1_0 \rceil_r = \cdots = \lceil u^p_0 \rceil_r$ and $\lceil v^1_k \rceil_\tau = \cdots = \lceil v^p_k \rceil_\tau$. We also say that the multi-chain is of length k. The labeled walks $\mathcal{W}_i$ are also called chains in this context.

Note that if $\{\mathcal{W}_1, \ldots, \mathcal{W}_p\}$ is a multi-chain then so is any subset of it. Also there can be different set of multi-chains depending on the starting and ending vertices and different $x = (x_1, \ldots, x_k)$. Let W_k denote the maximum order of all such multi-chains of length k. For a fixed linear function L, W_k is completely determined by $\mathcal{L}$. Now we describe how the list $\mathcal{L}$ is being generated through an interaction of an adversary $\mathcal{A}$ and a random permutation.

5.2 Multi-Chain Advantage

Consider an adversary $\mathscr{A}$ interacting with a b-bit random permutation $\Pi^{\pm}$. Suppose, the adversary $\mathscr{A}$ makes at most t interactions with $\Pi^{\pm}$. Let (x_i, dir_i) denote ith query where $x_i \in \{0,1\}^b$ and dir_i is either $+$ or $-$ (representing forward or inverse query). If $\mathsf{dir}_i = +$, it gets response y_i as $\Pi(x_i)$, else the response y_i is set as $\Pi^{-1}(x_i)$. After t interactions, we define a list $\mathcal{L}$ of pairs $(u_i, v_i)_i$ where $(u_i, v_i) = (x_i, y_i)$ if $\mathsf{dir}_i = +$, and $(u_i, v_i) = (y_i, x_i)$ otherwise. So we have $\Pi(u_i) = v_i$ for all i. We call the tuple of triples $\theta := ((u_1, v_1, \mathsf{dir}_1), \ldots, (u_t, v_t, \mathsf{dir}_t))$ the transcript of the adversary $\mathscr{A}$ interacting with $\Pi^{\pm}$. We also write $\theta' = ((u_1, v_1), \ldots, (u_t, v_t))$ which only stores the information about the random permutation. For the sake of simplicity we assume that adversary makes no redundant queries and so all $u_1, \ldots u_t$ are distinct and $v_1, \ldots, v_t$ are distinct. For a linear function L consider the directed graph $\mathcal{G}_{\theta'}$. For any k, we have already defined W_k. Now we define the maximum multi-chain advantage as

$$\mu_t = \max_{\mathscr{A}} \max_k \mathsf{Ex}\left[\frac{\mathsf{W}_k}{k}\right].$$

5.2.1 Bounding μ_t for Invertible L Functions

In this section, we derive concrete bounds for μ_t under a special assumption that the underlying linear function is invertible.

Theorem 2. *If the linear function L is invertible, then we have*

$$\mu_t \leq \mathsf{mcoll}(t, 2^\tau) + \mathsf{mcoll}(t, 2^r) + \mathsf{mcoll}'(t^2, 2^b). \quad (4)$$

Proof of Theorem 2: We first make the following observation which is straightforward as L is invertible.

Observation 1: If $v_i \xrightarrow{x} v_k$ and $v_j \xrightarrow{x} v_k$ then $v_i = v_j$.

We now describe some more notations related to multi-chains:

1. Let $\mathsf{W}^{\mathsf{fwd},a}$ denote the size of the set $\{i : \mathsf{dir}_i = +, \lceil v_i \rceil_\tau = a\}$ and $\max_a \mathsf{W}^{\mathsf{fwd},a}$ is denoted as $\mathsf{W}^{\mathsf{fwd}}$. This denotes the maximum multi-collision among τ most significant bits of forward query responses.

2. Similarly, we define the multi-collision for backward query responses as follows: Let $\mathsf{W}^{\mathsf{bck},a}$ denote the size of the set $\{i : \mathsf{dir}_i = -, \lceil u_i \rceil_r = a\}$ and $\max_a \mathsf{W}^{\mathsf{bck},a}$ is denoted as $\mathsf{W}^{\mathsf{bck}}$.

3. In addition to the multicollisions in forward only and backward only queries, we consider multicollisions due to both forward and backward queries. Let $\mathsf{W}^{\mathsf{mitm},a}$ denote size of the set $\{(i,j) : \mathsf{dir}_i = +,\ \mathsf{dir}_j = -,\ L(v_i) \oplus u_j = a\}$ and $\max_a \mathsf{W}^{\mathsf{mitm},a}$ is denoted as $\mathsf{W}^{\mathsf{mitm}}$.

Now, we state an intermediate result which is the main step of the proof.

Lemma 2. *For all possible interactions, we have*

$$\mathsf{W}_k \leq \mathsf{W}^{\mathsf{fwd}} + \mathsf{W}^{\mathsf{bck}} + k \cdot \mathsf{W}^{\mathsf{mitm}}.$$

Proof. We can divide the set of multi-chains into three sets:

Forward-only chains: Each chain is constructed by Π queries only. By definition, the size of such multi-chain is at most $\mathsf{W}^{\mathsf{fwd}}$.

Backward-only chains: Each chain is constructed by Π^- queries only. By definition, the size of such multi-chain is at most $\mathsf{W}^{\mathsf{bck}}$.

Forward-backward chains: Each chain is constructed by using both Π and Π^- queries. Let us denote the size of such multi-chain by $\mathsf{W}_k^{\mathsf{fwd\text{-}bck}}$.

Then, we must have

$$\mathsf{W}_k \leq \mathsf{W}^{\mathsf{fwd}} + \mathsf{W}^{\mathsf{bck}} + \mathsf{W}_k^{\mathsf{fwd\text{-}bck}}.$$

Now, we claim that $\mathsf{W}_k^{\mathsf{fwd\text{-}bck}} \leq k \cdot \mathsf{W}^{\mathsf{mitm}}$. Suppose $\mathsf{W}_k^{\mathsf{fwd\text{-}bck}} = w$. Then, it is sufficient to show that there exist an index $j \in [k]$, such that the size of the set $\{i : (\mathsf{dir}_{j-1}^i, \mathsf{dir}_j^i) \in \{(+,-),(-,+)\},\ L(v_{j-1}^i) \oplus u_j^i = x_j\} \geq \lceil w/k \rceil$. This can be easily argued by pigeonhole principle, given Observation 1. The argument works as follows:

For each of the individual chain W_i, we have at least one index $j \in [k]$ such that $(\mathsf{dir}_{j-1}^i, \mathsf{dir}_j^i) \in \{(+,-),(-,+)\}$. We put the i-th chain in a bucket labeled j, if $(\mathsf{dir}_{j-1}^i, \mathsf{dir}_j^i) \in \{(+,-),(-,+)\}$. Note that, it is possible that the i-th chain can co-exist in multiple buckets. But more importantly, it will exist in at least one bucket. As there are k buckets and w chains, by pigeonhole principle, we must have one bucket $j \in [k]$, such that it holds at least $\lceil w/k \rceil$ chains. □

Now we complete the proof of Theorem 2. Observe that $\mathsf{W}^{\mathsf{fwd}}$ and $\mathsf{W}^{\mathsf{bck}}$ are the random variables corresponding to the maximum multicollision in a truncated random permutation sample of size t, and corresponds to Remark 1 of subsection 4.1. Further, if we denote $x_i := u_i$ and $y_i := L(v_i)\ \forall i \in [t]$ then using Observation 1, $\mathsf{W}^{\mathsf{mitm}}$ is the random variable corresponding to the maximum multicollision in a sum of random permutation sample of size t^2, i.e., the distribution of subsection 4.2. Now, using linearity of expectation, we have

$$\begin{aligned}\mu_t &\leq \mathsf{Ex}\left[\mathsf{W}^{\mathsf{fwd}}\right] + \mathsf{Ex}\left[\mathsf{W}^{\mathsf{bck}}\right] + \mathsf{Ex}\left[\mathsf{W}^{\mathsf{mitm}}\right] \\ &\leq \mathsf{mcoll}(t, 2^{\tau}) + \mathsf{mcoll}(t, 2^{r}) + \mathsf{mcoll}'(t^2, 2^{b}).\end{aligned}$$

5.2.2 Bounding μ_t for Non-invertible L Functions

In case of invertible functions, Observation 1 facilitates a fairly simple argument in favor of an upper bound on μ_t in terms of some multicollision sizes. However, the same observation is not applicable to non-invertible functions. Specifically, Lemma 2 is not guaranteed to hold. For example, now the adversary can try to create a binary tree like structure using forward queries only. Clearly, we have to accommodate such attack strategies in order to upper bound μ_t.

Let `Collapse` denote the event that there exists distinct i and j, such that $\mathsf{dir}_i = \mathsf{dir}_j = +$, and $L(y_i) = L(y_j)$. We say that a transcript $\mathcal{L}$ is *collapse-free* if the event $\neg$`Collapse` holds. The following result is a variant of lemma 2 for collapse-free transcripts.

Lemma 3. *For all possible collapse-free transcripts, we have*

$$\mathsf{W}_k \leq \mathsf{W}^{\mathsf{fwd}} + \mathsf{W}^{\mathsf{bck}} + k \cdot \mathsf{W}^{\mathsf{mitm}}.$$

Proof. We can again divide the set of multi-chains into three sets:

Forward-only chains: Each chain is constructed by Π queries only. We collect all such chains into list FWD.

Backward-only chains: Each chain is constructed by Π^- queries only. We collect all such chains into list BCK.

Forward-backward chains: Each chain is constructed by using both Π and Π^- queries.

For the set of forward-backward chains, consider the smallest index j such that for two[2] distinct chains $\mathcal{W}_i$ and $\mathcal{W}_1 i'$ we have $v_j^i = v_j^{i'}$, i.e. the two chains merge. Since, the transcript is collapse-free, we must have $\mathsf{dir}_{j-1}^i = -$ or $\mathsf{dir}_{j-1}^{i'} = -$, or both. Now, we may have two cases:

[2]We may have more than two distinct chains merging at the same index. For brevity we consider only two. The general case can be handled in exactly the same manner.

1. Without loss of generality assume that only $\mathsf{dir}^i_{j-1} = -$. Now, if we traverse back along the walk $\mathcal{W}_i$ from vertex v^i_j, then either we get all backward edges (i.e. $\mathsf{dir}^i_{j'} = -$ for all $j' < j$), or there exists a $j' < j$ such that $\mathsf{dir}^i_{j'} = +$ and $\mathsf{dir}^i_{j'+1} = -$. In the first case we insert $\mathcal{W}_i$ in BCK, and in the second case we collect $\mathcal{W}_i$ in list MITM.

2. Suppose $\mathsf{dir}^i_{j-1} = -$ and $\mathsf{dir}^{i'}_{j-1} = -$. In this case, we traverse both $\mathcal{W}_i$ and $\mathcal{W}_{i'}$ and collect them in either BCK or MITM using the preceding argumentation.

We follow similar approach for all indices (in increasing order) where two or more chains merge collecting chains in either BCK or MITM. Once, we have exhausted all merging indices, we are left with some uncollected chains. We claim that these chains are disjoint of each other. This is easy to argue as for any pair of merged chains the chains with backward edge are already collected before. So all that is remaining is a collection of disjoint chains. Further, each of these chains must contain an index j such that $(\mathsf{dir}_j, \mathsf{dir}_{j+1}) \in \{(+,-),(-,+)\}$. We collect all these remaining chains in the list MITM'. Thus, we have

$$\mathsf{W}_k = |\mathsf{FWD}| + |\mathsf{BCK}| + |\mathsf{MITM}| + |\mathsf{MITM}'|.$$

By using the collapse-free property of $\mathcal{L}$ we get $|\mathsf{FWD}| \le \mathsf{W}^{\mathsf{fwd}}$, and $|\mathsf{BCK}| \le \mathsf{W}^{\mathsf{bck}}$ by definition. Further, by using the Pigeonhole argument used in the proof of Lemma 2, we get $|\mathsf{MITM}| + |\mathsf{MITM}'| \le k \cdot \mathsf{W}^{\mathsf{mitm}}$. □

Finally, we get the following upper bound on μ_t for non-invertible L functions.

Theorem 3. *If the linear function L is non-invertible and $\mathcal{L}$ is collapse-free, then we have*

$$\mu_t \le \mathsf{mcoll}(t, 2^\tau) + \mathsf{mcoll}(t, 2^r) + \widetilde{\mathsf{mcoll}}(t^2, 2^b, \mathsf{rank}(L)). \quad (5)$$

Proof. As before $\mathsf{Ex}\left[\mathsf{W}^{\mathsf{fwd}}\right] \le \mathsf{mcoll}(t, 2^r)$, $\mathsf{Ex}\left[\mathsf{W}^{\mathsf{bck}}\right] \le \mathsf{mcoll}(t, 2^\tau)$. Further, $\mathsf{W}^{\mathsf{mitm}}$ is the multicollision random variable $\widetilde{mc}_{t^2,2^b,\mathsf{rank}(L)}$ defined in section 4.2.1. Thus, $\mathsf{Ex}\left[\mathsf{W}^{\mathsf{mitm}}\right] \le \widetilde{\mathsf{mcoll}}(t^2, 2^b, \mathsf{rank}(L))$. The result follows from linearity of expectation. □

Remark 3. Theorem 3 has a limited applicability. Specifically, it holds only when $\mathcal{L}$ is collapse-free. A straightforward upper bound on $\Pr[\mathtt{Collapse}]$ is $t^2/2^{\mathsf{rank}(L)}$, where t denotes the size of $\mathcal{L}$ and $\mathsf{rank}(L)$ denotes the rank of linear function L. At times this bound is weaker than the bound achievable from a more straightforward approach of using the loose upper bound of $\mu_t \le t$. See section 6.4.1 for further details.

5.3 Related Work

In [Men18] Mennink analyzed the Key-prediction security of Keyed Sponge using a special type of data structure which is close to but different from our multi-chain structure. Here we give a brief overview of Mennink's work in our notations and describe how our structure is different from the structure considered by him.

Let $\mathcal{L} = ((u_1, v_1), \ldots, (u_t, v_t))$ 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. Let $c < b$ be any positive integer. For any such list of pairs, we write $\mathsf{domain}(\mathcal{L}) = \{u_1, \ldots, u_t\}$ and $\mathsf{range}(\mathcal{L}) = \{v_1, \ldots, v_t\}$. Given such a list we define a labeled directed graph $\mathcal{G}_{\mathcal{L}}$ over the set of vertices $\mathsf{range}(\mathcal{L}) \subseteq \{0,1\}^b$ as follows: A directed edge $v_i \to v_j$ with label x (also denoted as $v_i \xrightarrow{x} v_j$) is in the graph if $v_i \oplus x \| 0^c = u_j$. We can similarly extend this to a label walk $\mathcal{W}$ from a node w_0 to w_k as

$$\mathcal{W} : w_0 \xrightarrow{x_1} w_1 \xrightarrow{x_2} w_2 \cdots \xrightarrow{x_k} w_k.$$

We 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. The set $yield_{c,k}(\mathcal{L})$ consists of all possible labels x such that there exists a k-length walk of the form $0^b \xrightarrow{x} w_k$ in the graph $\mathcal{G}_{\mathcal{L}}$.

Consider the graph, $\mathcal{G}_{\mathcal{L}}$. The configuration of a walk from w_0 to w_k is defined as a tuple $C = (C_1, \ldots, C_k) \in \{0,1\}^k$ where $C_i = 0$ if $w_{i-1} \xrightarrow{x_i} w_i$ comes from a forward primitive query and $C_i = 1$ if it corresponds to an inverse primitive query.

Mennink provided an upper bound of $yield_{c,k}(\mathcal{L})$ by bounding the maximum number of possible labeled walks from 0^b to any given $w_k \in \{0,1\}^b$ with a given configuration C.

The use of tools like multi-collision and the similarity in the data structure of [Men18] with our multi-chain structure can be misleading. Here we try to discuss the difference between them and show that the underlying motivation behind both the problems are philosophically as different as possible.

Note that using multi-chain structure, we try to bound the number of different walks with the same label and distinct starting points whereas $yield_{c,k}(\mathcal{L})$ is the number of different walks with same starting point namely 0^b and distinct labels. Hence the multi-chain structure deals with a different problem than $yield_{c,k}(\mathcal{L})$. A notable change in our work is to deal with multicollision of sum of two permutation calls (we call it meet in the middle multicollision, see definition of $\mathsf{W}^{\mathsf{mitm}}$). This computation is not straightforward like usual computation of expectation of multi-collision (see subsection 4.2).

6 Security Analysis of Transform-then-Permute

In Theorem 4, we prove the main technical result of this paper on the AEAD security of Transform-then-Permute when the linear functions L_e and $L_{d,i}$ are invertible for all $1 \leq i \leq r$. Towards the end of this section we also give a result when the invertibility condition on $L_{d,i}$ is relaxed.
Recall the notations q_p, q_e, q_d, σ_e and σ_d from section 2.1. Let

$$\epsilon_{\text{common}} := \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma(2\sigma+q_p)}{2^b} + \frac{6\sigma_e q_p}{2^b} + \frac{2q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}} + \frac{\sigma_e + q_p}{2^b} + \frac{q_p \sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}. \quad (6)$$

Theorem 4 (Main Theorem)**.** *Let L_e and $L_{d,i}$ be invertible for all $i \in [r]$. For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, we have*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{inv\text{-}TtP}}(\mathscr{A}) \leq \epsilon_{\text{inv-}\mathtt{mBAD}} + \epsilon_{\text{common}},$$

where $\epsilon_{\text{inv-}\mathtt{mBAD}} := \dfrac{\sigma_d \mathsf{mcoll}(q_p, 2^\tau)}{2^c} + \dfrac{\sigma_d \mathsf{mcoll}(q_p, 2^r)}{2^c} + \dfrac{\sigma_d \mathsf{mcoll}'(q_p^2, 2^b)}{2^c}$.

The proof employs coefficient H-technique of Theorem 1. To apply this method we need to first describe the ideal world. The real world behaves same as the construction and would be described later. For the sake of notational simplicity we assume size of the nonce is at most $b - \kappa$. Later we mention how one can extend the proof when nonce size is more than $b - \kappa$. We also assume that the adversary makes exactly q_p, q_e and q_d many primitive, encryption and decryption queries respectively.

6.1 Ideal World and Real World

Online Phase of Ideal World. The ideal world responds three oracles, namely encryption queries, decryption queries and primitive queries in the online phase.

(1) On Primitive Query $(\mathsf{W}_i, \mathsf{dir}_i)$:

The ideal world simulates $\Pi^{\pm}$ query honestly.[3] 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 samples $\mathsf{Y}_{i,0}, \ldots, \mathsf{Y}_{i,t_i} \leftarrow_\$ \{0,1\}^b$ where $t_i = a_i + m_i$, $a_i = a(|\mathsf{A}_i|)$ and $m_i = \lceil \frac{|\mathsf{M}_i|}{r} \rceil$. Then, it returns $(\mathsf{C}_{i,1} \| \cdots \| \mathsf{C}_{i,m_i}, \mathsf{T}_i)$ where $(\mathsf{M}_{i,1}, \ldots, \mathsf{M}_{i,m_i}) \xleftarrow{r} \mathsf{M}_i$, $\mathsf{C}_{i,j} = \lceil \mathsf{Y}_{i,a_i+j-1} \rceil_{|\mathsf{M}_{i,j}|} \oplus \mathsf{M}_{i,j}$ for all $j \in [m_i]$ and $\mathsf{T}_i \leftarrow \lceil \mathsf{Y}_{i,t_i} \rceil_\tau$.

(3) On Decryption Query $Q_i := (\mathsf{N}^*_i, \mathsf{A}^*_i, \mathsf{C}^*_i, \mathsf{T}^*_i)$:

According to our convention we assume that the decryption query is always non-trivial. So the ideal world returns abort symbol $\mathsf{M}^*_i := \perp$.

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 $\mathcal{E}, , \mathcal{D}, \mathcal{P}$ to denote the set of all query indices corresponding to encryption, decryption and primitive queries respectively. So $\mathcal{E} \sqcup \mathcal{D} \sqcup \mathcal{P} = [q_e + q_d + q_p]$ and $|\mathcal{E}| = q_e$, $|\mathcal{D}| = q_d$, $|\mathcal{P}| = q_p$. Let the primitive transcript $\omega_p = (\mathsf{U}_i, \mathsf{V}_i, \mathsf{dir}_i)_{i \in \mathcal{P}}$ and let $\omega'_p := (\mathsf{U}_i, \mathsf{V}_i)_{i \in \mathcal{P}}$. The decryption transcript $\omega_d := (\mathsf{M}^*_i)_{i \in \mathcal{D}}$ where M^*_i is always $\perp$.

Now we describe some extended transcript (releasing additional information) for encryption queries. It samples $K \leftarrow_\$ \{0,1\}^\kappa$. For all i, let $\mathsf{Fmt}(\mathsf{N}_i, \mathsf{A}_i, \mathsf{M}_i) = (D_{i,0}, \ldots, D_{i,t_i})$ and for every $0 \le j \le t_i$, the intermediate input (X-value) is defined as

$$\mathsf{X}_{i,j} = \begin{cases} D_{i,0} \oplus K \| 0^{b-\kappa} & \text{if } j = 0 \\ L_e(\mathsf{Y}_{i,j-1}) \oplus D_{i,j} & \text{if } 1 \le j \le t_i \end{cases}$$

The encryption transcript $\omega_e = (\mathsf{X}_{i,j} \mathsf{Y}_{i,j})_{i \in \mathcal{E}, j \in [0..t_i]}$. So, the transcript of the adversary consists of $\omega := (Q, \omega_p, \omega_e, \omega_d)$ where $Q := (Q_i)_{i \in \mathcal{E} \cup \mathcal{D}}$.

Real World. In the online phase, the AE encryption and decryption queries and direct primitive queries are faithfully responded based on $\Pi^{\pm}$. Like the ideal world, after completion of interaction, the real world returns all X-values and Y-values corresponding to the encryption queries only. Note that a decryption query may return M_i which is not $\perp$.

6.2 Bad Transcripts

We define the bad transcripts into two main parts. We first define bad events due to encryption and primitive transcript. The following bad events says that (i) there is a collision among inputs/outputs of ω_p and ω_e (ii) there is a collision among input/outputs of ω_e. So, given that there are no such collision, all inputs and outputs are distinct and hence $\omega_e \cup \omega_p$ is permutation compatible (can be realized by random permutation). More formally, we define the following bad events:

B1: For some $(\mathsf{U}, \mathsf{V}) \in \omega_p$, $\mathsf{K} = \lfloor \mathsf{U} \rfloor_\kappa$.

B2: For some $i \in \mathcal{E}$, $j \in [t_i]$, $\mathsf{Y}_{i,j} \in \mathsf{range}(\omega_p)$, (in other words, $\mathsf{range}(\omega_e) \cap \mathsf{range}(\omega_p) \neq \emptyset$)

B3: For some $i \in \mathcal{E}$, $j \in [t_i]$, $\mathsf{X}_{i,j} \in \mathsf{domain}(\omega_p)$, (in other words, $\mathsf{domain}(\omega_e) \cap \mathsf{domain}(\omega_p) \neq \emptyset$)

B4: For some $(i \in \mathcal{E}, j \in [t_i]) \neq (i' \in \mathcal{E}, j' \in [t_{i'}])$, $\mathsf{Y}_{i,j} = \mathsf{Y}_{i',j'}$,

[3] For example, one can use lazy sampling to simulate random permutation.

B5: For some $(i \in \mathcal{E}, j \in [t_i]) \neq (i' \in \mathcal{E}, j' \in [t_{i'}])$, $\mathsf{X}_{i,j} = \mathsf{X}_{i',j'}$,

Now we describe the bad event due to decryption queries. Suppose the bad events ($\mathtt{B1} \vee \cdots \vee \mathtt{B5}$) as defined above due to encryption queries and primitive don't occur i.e. we have $\omega_p \cup \omega_e$ is permutation compatible. Suppose Π' is the partially defined permutation defined over domain of $\omega_p \cup \omega_e$ and mapping the corresponding range elements. For each decryption query $Q_i = (\mathsf{N}_i^*, \mathsf{A}_i^*, \mathsf{C}_i^*, \mathsf{T}_i^*)$, we compute $a_i = a(|\mathsf{A}_i^*|)$, $m_i = \lceil |\mathsf{C}_i^*|/r \rceil$ and $\mathsf{Fmt}(\mathsf{N}_i^*, \mathsf{A}_i^*, \mathsf{C}_i^*) = (D_{i,0}^*, \ldots, D_{i,t_i}^*)$. We define p'_i is the largest index j for which the input X_j is in the domain of $\omega_e \cup \omega_p$ while we run the decryption algorithm using Π' for Q_i. Consider the case, $p'_i = t_i$ i.e. the complete decryption algorithm computation for the query is determined by the $\omega_e \cup \omega_p$ transcript. In such a case we define bad (called mBAD) if the corresponding tag also matches. Note that for this bad transcript the real world should not abort the decryption query. Now we define all bad events in a more formal way.

DEFINITION OF p_i. Before we define p'_i, we first define p_i which is the input index we can compute for the decryption query only using encryption queries transcript. Formally, p_i is defined as -1 if for all $i' \in \mathcal{E}$, $N_{i'} \neq N_i^*$. Otherwise, there exists a unique $i' \in \mathcal{E}$ such that $\mathsf{N}_{i'} = \mathsf{N}_i^*$ (as we consider nonce-respecting adversary only). Let $p_i + 1$ denote the length of the longest common prefix of $(D_{i',0}, \cdots, D_{i',t_{i'}})$ and $(D_{i,0}^*, \cdots, D_{i,t_i}^*)$. Note that $p_i = -1$ in case there is no common prefix.

We now define $\mathsf{Y}_{i,0..p_i}^* = \mathsf{Y}_{i',0..p_i}$, $\mathsf{X}_{i,0..p_i}^* = \mathsf{X}_{i',0..p_i}$ when $p_i \geq 0$ and

$$\mathsf{X}_{i,p_i+1}^* = \begin{cases} L_e(\mathsf{Y}_{i',p_i}) \ \oplus \ D_{i,p_i+1}^* & \textit{if } p_i \geq 0. \\ \mathsf{K} \| \mathsf{N}_i^* & \textit{if } p_i = -1. \end{cases}$$

By Lemma 1, $p_i < t_i$, $p_i < t_{i'}$. By definition of longest common-prefix, we have $\mathsf{X}_{i,p_i+1}^* \neq \mathsf{X}_{i',p_i+1}$.

DEFINITION OF p'_i. If $p_i < a_i$ or if $\mathsf{X}_{i,p_i+1}^* \notin \mathsf{domain}(\omega_p)$ define $p'_i = p_i$. Else, we further extend X^*-values and Y^*-values based on the primitive transcript ω_p. Let $x_{i,j} := D_{i,j}^*$ for all $i \in \mathcal{D}, 1 \leq j \leq t_i$. If there is a labeled walk (in the labeled directed graph induced by ω_p as described in section 5 from Y_{i,p_i+1}^* with label $(x_{i,p_i+2}, \ldots, x_{i,j})$ then we denote the end node as $\mathsf{Y}_{i,j}^*$. In notation we have

$$\mathsf{Y}_{i,p_i+1}^* \overset{(x_{i,p_i+2},\ldots,x_{i,j})}{\longrightarrow} \mathsf{Y}_{i,j}^*.$$

Let p'_i denotes the maximum of all such possible j's. For all those i and j in which $\mathsf{Y}_{i,j}^*$ has been defined as described above, we define $\mathsf{X}_{i,j+1}^* := L_d(\mathsf{Y}_{i,j}^*) \oplus x_{i,j+1}$.

Bad events due to the decryption queries in the transcript:

mBAD: For some $i \in \mathcal{D}$ with $p'_i = t_i$ and $\lceil \mathsf{Y}_{i,t_i}^* \rceil_\tau = \mathsf{T}_i^*$.

B6: For some $i \in \mathcal{D}$, $p'_i < t_i$ and, $\mathsf{X}_{i,p'_i+1}^* \in \mathsf{domain}(\omega_e) \cup \mathsf{domain}(\omega_p)$.

We write BAD to denote the event that the ideal world transcript Θ_0 is bad. Then, with a slight abuse of notations and union bound, we have

$$\mathtt{BAD} = \mathtt{mBAD} \cup \left(\bigcup_{i=1}^{6} \mathtt{Bi} \right). \tag{7}$$

Lemma 4 upper bounds the probability of mBAD and Lemma 5 upper bounds the probability of $\bigcup_{i=1}^{6} \mathtt{Bi}$. The proofs of Lemma 4 and 5 are postponed to subsections 6.4 and 6.5, respectively.

Lemma 4. *Let μ_{q_p} be the maximum multi-chain advantage (see subsection 5.2) over q_p primitive queries. Then, we have*

$$\Pr[\mathtt{mBAD}] \le \frac{\sigma_d \mathsf{mcoll}(q_p, 2^\tau)}{2^c} + \frac{\sigma_d \mathsf{mcoll}(q_p, 2^r)}{2^c} + \frac{\sigma_d \mathsf{mcoll}'(q_p^2, 2^b)}{2^c}.$$

Lemma 5. *For $q_p < 2^{b-1}$, we have*

$$\Pr\left[\bigcup_{i=1}^{6} \mathtt{Bi}\right] \le \frac{q_p}{2^\kappa} + \frac{6\sigma_e q_p}{2^b} + \frac{2\sigma_e^2}{2^b} + \frac{\sigma_e + q_p}{2^b} + \frac{2q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^{b-\tau}} + \frac{q_p \sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.$$

6.3 Good Transcript Analysis

The motivation for all the bad events would be clear from the understanding of a good transcript (i.e., not a bad transcript). Let $\omega = (Q, \omega_p, \omega_e, \omega_d)$ be a good transcript. For the sake of notation simply we ignore the query transcript Q as it is not required to compute the probability of a transcript.

1. The tuples ω_e is permutation compatible and disjoint from ω_p. So union of tuples $\omega_e \cup \omega_p$ is also permutation compatible.

2. Let $\mathcal{D}_1$ (type-1 decryption query) be the set of all $i \in \mathcal{D}$, if $p'_i = t_i$ with $\lceil \mathsf{Y}^*_{i,t_i} \rceil_\tau \ne \mathsf{T}^*_i$. In this case, decryption algorithm should abort with probability one. Set of all other indices is denoted as $\mathcal{D}_2$ (type-2 decryption query). In this case, $p'_i < t_i$ but $\mathsf{X}^*_{i,p'_i+1} \notin \mathsf{domain}(\omega_e \cup \omega_p)$. So, Y^*_{i,p'_i+1} value and subsequent Y-values will have almost b-bit entropy. Thus, with a negligible probability we may not abort the query.

Ideal World Interpolation Probability. Let Θ_0 and Θ_1 denote the transcript random variable obtained in the ideal world and real world respectively. As noted before, all the input-output pairs for the underlying permutation are compatible. In the ideal world, all the Y values are sampled uniform at random; the list ω_p is just the partial representation of Π; and all the decryption queries are degenerately aborted; whence we get

$$\Pr[\Theta_0 = \omega] = \frac{1}{2^{b\sigma_e}(2^b)_{q_p}}.$$

Here σ_e denotes the total number of blocks present in all encryption queries including nonce. In notation $\sigma_e = q_e + \sum_i m_i$.

Real World Interpolation Probability. 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

$$\begin{aligned}
\Pr[\Theta_1 = \omega] &= \Pr[\Theta_1 = (\omega_e, \omega_p, \omega_d)] \\
&= \Pr[\omega_e, \omega_p] \cdot \Pr[\omega_d \mid \omega_e, \omega_p] \\
&= \Pr[\omega_e, \omega_p] \cdot (1 - \Pr[\neg\omega_d \mid \omega_e, \omega_p]) \\
&\le \Pr[\omega_e, \omega_p] \cdot \left(1 - \sum_{i \in \mathcal{D}_2} \Pr[\neg\omega_{d,i} \mid \omega_e, \omega_p]\right) \qquad (8)
\end{aligned}$$

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 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 encryption and primitive queries are mutually permutation compatible, so we have

$$\Pr_{\Theta_1}(\omega_e, \omega_p) = 1/(2^b)_{\sigma_e+q_p} \geq \Pr_{\Theta_0}(\omega_e, \omega_p).$$

Now we show an upper bound $\Pr_{\Theta_1}(\neg\omega_{d,i} \mid \omega_e, \omega_p) \leq \frac{2(\sigma+q_p)}{2^b} + \frac{2}{2^\tau}$ for every type-2 decryption query. We quickly recall that $\mathsf{Fmt}(\mathsf{N}_i^*, \mathsf{A}_i^*, \mathsf{C}_i^*) = (D_{i,0}^*, \ldots, D_{i,t_i}^*)$. So, $\neg\omega_{d,i}$ is same as $\lceil \Pi(\mathsf{X}_{i,t_i}^*) \rceil_\tau = \mathsf{T}_i^*$ where $\mathsf{X}_{i,j}^*$ values have been defined recursively as follows

$$\mathsf{X}_{i,j}^* = L_d\big(\Pi(\mathsf{X}_{i,j-1}^*)\big) \oplus D_{i,j}^*, \quad p_i' + 1 < j \leq t_i.$$

Let $\mathcal{I}$ and $\mathcal{O}$ denote the set of inputs and outputs for Π which are present in the transcript (ω_e, ω_p). Recall that $\mathsf{X}_{i,p_i'+1}^*$ is fresh, i.e., $\mathsf{X}_{i,p_i'+1}^* \notin \mathcal{I}$.

Claim 1. $\Pr(\mathsf{X}_{i,j}^* \text{ is fresh }) \geq (1 - \frac{2(\sigma_e+q_p+t_i)}{2^b}) \ \ \forall\ p_i' + 1 < j \leq t_i$.

Proof. Since $\mathsf{X}_{i,p_i'+1}^*$ is not the last block, then the next input block may collide with some encryption or primitive input block with probability at most $\frac{\sigma_e+q_p}{2^b-\sigma_e-q_p}$. 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 $\frac{(\sigma_e+q_p+t_i-p_i'+2)}{2^b-\sigma_e-q_p-t_i+p_i'+2} \leq \frac{2(\sigma_e+q_p+t_i)}{2^b}$. □

Claim 2. $\Pr(\neg\omega_{d,i} \mid \mathsf{X}_{i,j}^* \text{ are fresh }) \leq \frac{2}{2^\tau}$.

Proof. Since the last input block X_{i,t_i}^* is fresh, hence $\Pi(\mathsf{X}_{i,t_i}^*) = \mathsf{T}_i^*$ with probability at most $2/2^\tau$ (provided $\sigma_e + q_p \leq 2^{b-1}$ which can be assumed, since otherwise our bound is trivially true). □

Let $\mathtt{E}_j$ denote the event that $\mathsf{X}_{i,j}^*$ is fresh and $\mathtt{E} := \wedge_{j=p_i'+1}^{t_i} \mathtt{E}_j$
Using the claims, we have

$$\begin{aligned}\Pr_{\Theta_1}(\neg\omega_{d,i} \mid \omega_e, \omega_p) &\leq \Pr_{\Theta_1}(\neg\omega_{d,i} \wedge E \mid \omega_e, \omega_p) + \Pr(E^c).\\ &\leq \frac{2}{2^\tau} + \sum_{j=p_i'+1}^{t_i} \frac{\sigma_d + \sigma_e + q_p}{2^{b-1}}.\end{aligned}$$

The last inequality follows from the above claims. Now, we can proceed by using the union bound as follows.

$$\begin{aligned}\Pr[\neg\omega_d \mid \omega_e, \omega_p] &\leq \sum_{i\in\mathcal{D}} \frac{2t_i(\sigma_e + q_e + \sigma_d)}{2^b} + \frac{2}{2^\tau}\\ &\leq \frac{2\sigma_d(\sigma_e + \sigma_d + q_p)}{2^b} + \frac{2q_d}{2^\tau}\\ &= \frac{2\sigma_d(\sigma + q_p)}{2^b} + \frac{2q_d}{2^\tau}\end{aligned}$$

Finally, Theorem 4 follows from the H-technique (Theorem 1) combined with Lemma 4, 5 and Eq. (8).

Remark 4. As described in the algorithm, in the case where nonce size is greater than $b - \kappa$, we treat the excess length of the nonce as part of the associated data. For such a TtP construction the internal values of the encryption transcripts are chosen in a prefix respecting manner. Suppose the i, i'-th queries $(D_{i,0}, \ldots, D_{i,t_i})$ and $(D_{i',0}, \ldots, D_{i',t_j})$ have a maximum common prefix of length p_i and let without loss of generality $i < i'$. Then we set $Y_{i,j} = Y_{i',j}$ and $X_{i,j} = X_{i',j} \forall 0 \leq j \leq p_i$. The rest of the proof remains the same.

6.4 Proof of Lemma 4 (Multi-chain Bad Transcript Analysis)

Suppose the event holds for the i-th decryption query and $\mathsf{N}_i^* = \mathsf{N}_{i'}$. So, $\left(\mathsf{X}^*_{i,p_i+1}, \mathsf{Y}^*_{i,p_i+1}\right)$ must be the one of the starting node of the multi-chain. Hence as in definition 2, if (U, V) be any other starting node of the multi-chain, then we must have $\lceil U \rceil_r = \lceil \mathsf{X}^*_{i,p_i+1} \rceil_r$. Now as before , let $\mathsf{W}_{t_i - p_i}$ denote the maximum size of the set of multi-chain of length $t_i - p_i$, induced by L_d and ω_p. As $\lfloor \mathsf{Y}_{i',p_i} \rfloor_c$ is chosen at random (and independent of ω_p), and C^*_{i,p_i+1} is fixed, the probability to hold $\mathtt{mBAD}$ for i-th decryption query is at most $\mathsf{W}_{m_i}/2^c$ given the transcript ω_p. So by union bound, the conditional probability $\Pr[\mathtt{mBAD} \mid \omega_p] \leq \sum_{i \in \mathcal{D}} \frac{\mathsf{W}_{m_i}}{2^c}$.

Since the decryption query data complexity of the adversary is bounded by σ_d blocks we have $\sum_{i \in \mathcal{D}} m_i \leq \sigma_d$. Now,

$$\sum_{i \in \mathcal{D}} \mathsf{W}_{m_i} \leq \sum_{i \in \mathcal{D}} \left(\max_{k \leq m_i} \frac{\mathsf{W}_k}{k} \times m_i \right) \leq \max_k \frac{\mathsf{W}_k}{k} \times \sigma_d.$$

Hence,

$$\epsilon_{\text{inv-}\mathtt{mBAD}} := \Pr[\mathtt{mBAD}] \leq \sum_{i \in \mathcal{D}} \frac{\mathsf{Ex}[\mathsf{W}_{m_i}]}{2^c} \leq \max_k \mathsf{Ex}\left[\frac{\mathsf{W}_k}{k}\right] \times \frac{\sigma_d}{2^c} \leq \frac{\sigma_d \cdot \mu_{q_p}}{2^c}.$$

The result follows from application of Theorem 2.

6.4.1 Bounding $\Pr[\mathtt{mBAD}]$ when $L_{d,i}$ is Non-invertible

In Theorem 4, L_e and $L_{d,i}$ are invertible functions. So, we can apply Theorem 2 to tightly estimate μ_{q_p}. Here, we digress a little from the main proof to discuss the bound on $\Pr[\mathtt{mBAD}]$ when the invertibility condition is relaxed for $L_{d,i}$. In the following discussion, we compare two ways to upper bound $\Pr[\mathtt{mBAD}]$:

1. An obvious upper bound on μ_{q_p} is q_p. Using this crude bound we get

$$\Pr[\mathtt{mBAD}] \leq \epsilon_1 := \frac{\sigma_d q_p}{2^c}. \quad (9)$$

2. Yet another way is to employ Theorem 3 of section 5.2.2. In this case, we bound μ_{q_p} conditioned on a collapse-free (see section 5.2.2) transcript ω_p. More formally, we have

$$\begin{aligned}\Pr[\mathtt{mBAD}] &\leq \Pr[\mathtt{mBAD} \mid \text{collapse-free } \omega_p] + \Pr[\mathtt{Collapse}] \\ &\leq \frac{\sigma_d \mu_{q_p}}{2^c} + \frac{q_p^2}{2^{\mathsf{rank}(L_d)+1}} \\ &\leq \epsilon_2 := \frac{\sigma_d \mathsf{mcoll}(q_p, 2^\tau)}{2^c} + \frac{\sigma_d \mathsf{mcoll}(q_p, 2^r)}{2^c} \\ &\quad + \frac{\sigma_d \widetilde{\mathsf{mcoll}}(q_p^2, 2^b, \mathsf{rank}(L_d))}{2^c} + \frac{q_p^2}{2^{\mathsf{rank}(L_d)+1}}. \end{aligned} \quad (10)$$

The first inequality follows from a similar line of argument as used above in case of invertible L_d function, and the fact that $\Pr[\texttt{Collapse}]$ is upper bounded by $q_p^2/2^{\mathsf{rank}(L_d)+1}$ where $\mathsf{rank}(L_d)$ denotes the rank of L_d. The second inequality follows from Theorem 3.

In summary, we have

$$\epsilon_{\text{non-inv-}\texttt{mBAD}} := \Pr[\texttt{mBAD}] \leq \min\{\epsilon_1, \epsilon_2\}. \tag{11}$$

Under reasonable assumptions, one can see that the dominating term in ϵ_2 is $q_p^2/2^{\mathsf{rank}(L_d)+1}$, where $\mathsf{rank}(L_d) < b$. Now, depending upon the security requirement, the choice of parameters (r and c), and the $\mathsf{rank}(L_d)$, we can choose one of ϵ_1 or ϵ_2.

Consider the NIST LwC requirement where the AEAD should be secure while $q_p \leq 2^{112}$ and $r\sigma_d \leq 2^{53}$. For any Sponge-type AEAD scheme with $c \geq 165 - \log_2 r$, we can use $\Pr[\texttt{mBAD}] \leq \epsilon_1$, whereas if $b \geq 224$, $c \leq 165 - \log_2 r$ and $\mathsf{rank}(L_d)$ is close to b (say $b-1$) then we can use $\Pr[\texttt{mBAD}] \leq \epsilon_2$. In section 7.3.1, we use this methodology while deriving the security bounds for the two variants of PHOTON-Beetle [BCD+19], a round 2 candidate of NIST LwC standardization process.

6.5 Proof of Lemma 5 (Bad Transcript Analysis)

From the union bound we have

$$\Pr\left[\bigcup_{i=1}^{6} \texttt{Bi}\right] \leq \Pr[\texttt{B1}] + \Pr[\texttt{B2}] + \Pr[\texttt{B3}|\neg\texttt{B1}] + \Pr[\texttt{B4}] + \Pr[\texttt{B5}] + \Pr[\texttt{B6}|\neg\texttt{B1}].$$

It is sufficient to upper bound each of these individual probabilities. We bound the probabilities of these events in the following:

Bounding $\Pr[\texttt{B1}]$: This is basically 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 $(\mathsf{U}, \mathsf{V}) \in \omega_p$, the probability that $\mathsf{K} = \lfloor \mathsf{U} \rfloor_\kappa$ is bounded by at most $2^{-\kappa}$, as K is chosen uniform at random from $\{0,1\}^\kappa$. Thus, we have

$$\Pr[\texttt{B1}] \leq \frac{q_p}{2^\kappa}.$$

Bounding $\Pr[\texttt{B2}]$: This event can be analyzed in several cases as below:

Case 1: $\exists i, j, a$, $\mathsf{Y}_{i,j} = \mathsf{V}_a$, 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: $\exists i, j, a$, $\mathsf{Y}_{i,j} = \mathsf{V}_a$, $\mathsf{dir}_a = +$, 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 $\sigma_e q_p/(2^b - q_p + 1) \leq 2\sigma_e q_p/2^b$ (as $q_p \leq 2^{b-1}$).

Case 3: $\exists i, j \neq t_i, a$, $\mathsf{Y}_{i,j} = \mathsf{V}_a$, $\mathsf{dir}_a = -$, encryption before primitive: Here the adversary has access to $\lceil \mathsf{Y}_{i,j} \rceil_r$, as this value has already been released. Let Φ_{out} denote the number of multicollisions among all $\lceil \mathsf{Y}_{i',j'} \rceil_r$ values. Now, we have

$$\begin{aligned}\Pr[\text{Case 3}] &= \sum_{\Phi_{out}} \Pr[\text{Case 3} \mid \Phi_{out}] \cdot \Pr[\Phi_{out}] \\ &\leq \sum_{\Phi_{out}} \frac{\Phi_{out} \times q_p}{2^c} \cdot \Pr[\Phi_{out}] \\ &\leq \frac{q_p}{2^c} \times \mathsf{Ex}\left[\Phi_{out}\right] \\ &\leq \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c}.\end{aligned}$$

Case 4: $\exists i, a$, $\mathsf{Y}_{i,t_i} = \mathsf{V}_a$, $\mathsf{dir}_a = -$, encryption before primitive: This case is same as case-3 plugging in r as τ and c as $b - \tau$. So, $\Pr[\text{Case 4}] \leq \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}}$.
By using the union bound, we have

$$\Pr[\mathtt{B2}] \leq \frac{3\sigma_e q_p}{2^b} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}}.$$

Bounding $\Pr[\mathtt{B3}|\neg\mathtt{B1}]$: This means $\exists i, j, a$, $\mathsf{X}_{i,j} = \mathsf{U}_a$ where $j > 0$ (as B1 does not hold). So, we can have the following cases with $j > 0$:

Case 1: $\exists i, j, a$, $\mathsf{X}_{i,j} = \mathsf{U}_a$, encryption after primitive: This case can be bounded by probability at most $1/2^b$, as $\mathsf{Y}_{i,j-1}$ is chosen uniform at random and L_e in invertible. We have at most σ_e many (i, j) pairs and q_p many a indices. Thus this can be bounded by at most $\sigma_e q_p / 2^b$.

Case 2: $\exists i, j, a$, $\mathsf{X}_{i,j} = \mathsf{U}_a$, $\mathsf{dir}_a = -$, 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 $2\sigma_e q_p / 2^b$.

Case 3: $\exists i, j, a$, $\mathsf{X}_{i,j} = \mathsf{U}_a$, $\mathsf{dir}_a = +$, encryption before primitive: Since L_e is invertible, we can define $\mathsf{V}' = L_e^{-1}(U_a \oplus D_j)$. Then using the invertibility of L_e we have this event is same as the event $\exists i, 0 < j$, $\mathsf{Y}_{i,j-1} = \mathsf{V}'$ for some $\mathsf{V}' \in \omega_p$. Since $j \leq t_i$ we have this event is the same as Case 3 of B2. Hence,

$$\Pr[\text{Case 3}] \leq \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c}.$$

$$\Pr[\mathtt{B3}|\neg\mathtt{B1}] \leq \frac{3\sigma_e q_p}{2^b} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c}.$$

Bounding $\Pr[\mathtt{B4}]$ and $\Pr[\mathtt{B5}]$: The probability of this event can be simply bounded by birthday paradox and so it is at most $\sigma_e(\sigma_e - 1)/2^b$.

Bounding $\Pr[\mathtt{B6}|\neg\mathtt{B1}]$: This event can be analyzed in several cases.

Case 1 $p'_i < a_i$: Since during associated data processing no information is leaked to the adversary and $Y^*_{i,j}$-s are sampled uniformly at random hence for $p'_i < a_i$, the distribution function of $X^*_{i,p'_i+1} = Y^*_{i,p'_i} \oplus D^*i, p'_i + 1$ is uniform. Hence

$$\Pr\left[\text{Case 1}\right] \leq \frac{\sigma_e + q_p}{2^b}.$$

Case 2 $a_i \leq p_i \leq p'_i$: This corresponds to the case when either the first non-trivial decryption query block doesn't match any primitive query or it matches a primitive query and follows a partial chain and then matches with some encryption query block. Doing similar analysis as in Case 3 of $\mathtt{B3}|\neg\mathtt{B1}$, The probability that this happens for i-th decryption is at most $q_p/2^c \times m_i\Phi_{out}/2^c$. Summing over all $i \in \mathcal{D}$, the conditional probability is at most $\frac{q_p\sigma_d\Phi_{out}}{2^{2c}}$. By taking expectation we obtain the following:

$$\Pr[\text{Case 3}] \leq \frac{q_p\sigma_d\mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.$$

$$\Pr\left[\mathtt{B6}|\neg\mathtt{B1}\right] \leq \frac{\sigma_e + q_p}{2^b} + \frac{q_p\sigma_d\mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.$$

By adding all these probabilities we prove our result.

Using Eq. (11) and the proof approach of Theorem 4, we get the following corollary for non-invertible feedback function.

Corollary 1. *Let TtP be a construction, where for all $i \in [r]$, $L_{d,i}$ is non-invertible, and L_e is invertible. For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, we have*

$$\mathbf{Adv}^{\mathsf{aead}}_{\text{non-inv-}\mathsf{TtP}}(\mathscr{A}) \leq \epsilon_{\text{non-inv-}\mathtt{mBAD}} + \epsilon_{\text{common}},$$

where $\epsilon_{\text{non-inv-}\mathtt{mBAD}} := \min\{\epsilon_1, \epsilon_2\}$, ϵ_{common} is defined in Eq. (6), and ϵ_1 and ϵ_2 are defined as before in Eq. (11).

Remark 5. Note that the bounds in Theorem 4 and Corollary 1 are exactly the same except for the probability of multi-chain bad event, i.e., $\epsilon_{\text{inv-}\mathtt{mBAD}}$ and $\epsilon_{\text{non-inv-}\mathtt{mBAD}}$, respectively.

7 Instantiating TtP and Application of Theorem 4

Now, we describe how Transform-then-Permute can capture a wide class of permutation based sequential constructions such as duplex (or Sponge AE), Beetle and SpoC, in which the only non-linear operation is the underlying permutation. We further show that SpoC and Beetle fall under a special class of TtP constructions where the feedback functions are invertible and hence we can apply Theorem 4 in those cases to get tight security bounds. A variant of Beetle, called PHOTON-Beetle, with non-invertible feedback function is a round 2 candidate in NIST LwC standardization process. In case of PHOTON-Beetle, we can use Corollary 1. Note that Sponge AE does not belong to the special class of invertible feedback functions. Towards the end we discuss the hardness in obtaining better security bounds for Sponge AE.

7.1 How to Convert a Generalized Sponge-type Construction to TtP

Let $L : \{0,1\}^b \times \{0,1\}^r \to \{0,1\}^b \times \{0,1\}^r$ be any linear function defined by the transformation matrix $L = \begin{bmatrix} L_{1,1} & L_{1,2} \\ L_{2,1} & L_{2,2} \end{bmatrix}$ consisting of $b \times b$ matrix $L_{1,1}$, $b \times r$ matrix $L_{1,2}$, $r \times b$ matrix $L_{2,1}$, $r \times r$ matrix $L_{2,2}$. Consider the Sponge-type construction which takes state input X_i and data input M_i and generate the data output C_i and next state input X_{i+1} as follows:

$$Y_i = \Pi(X_i); \quad \begin{bmatrix} X_{i+1} \\ C_i \end{bmatrix} = L \cdot \begin{bmatrix} Y_i \\ M_i \end{bmatrix}$$

As $L_{2,1} \cdot Y + L_{2,2} \cdot M = C$, the rank of $L_{2,2}$ must be r, otherwise encryption is not a bijective function from message space to ciphertext space. For the sake of simplicity we can assume that $L_{2,2} = I_r$ (the identity matrix of size r). Otherwise, we can redefine message block as $M' = L_{2,2} \cdot M$.

Now, we observe that rank of $L_{2,1}$ is r. If not, then there exists a non-zero vector γ such that $\gamma \cdot L_{2,1} = 0$. Hence, $\gamma \cdot M = \gamma \cdot C$ holds with probability 1. In case of ideal permutation as γ is non-zero and C is chosen uniformly independent of M, this event occurs with probability $\frac{1}{2}$. Hence the privacy advantage of any adversary for such a construction will be $\geq \frac{1}{2}$. As rank of $L_{2,1}$ is r, there exists an invertible matrix $Z_{b\times b}$ such that $L_{2,1} \cdot Z = I_r \| 0_{r\times(b-r)}$. Let $L_e = L_{1,1} \cdot Z$. Then by simple matrix algebra we have

$$\begin{bmatrix} X_{i+1} \\ C_i \end{bmatrix} = \begin{bmatrix} L_e & L_{1,2} \\ I_r \| 0_{r\times(b-r)} & I_r \end{bmatrix} \cdot \begin{bmatrix} Y_i' \\ M_i \end{bmatrix}$$

where $Y_i' = Z^{-1} \cdot Y_i$. Note that, multiplication by an invertible matrix is a permutation and composition of a random permutation with a public permutation is again a random permutation. Hence, we can redefine the random permutation output as $Z^{-1} \cdot \Pi(X_i)$. Let us denote $\mathsf{encode}(M) = L_{1,2} \cdot M$ and hence the the general linear function based Sponge-type construction boils down to the construction TtP.

7.2 Security of SpoC

In SpoC [AGH+19], the linear function L_e is identity, and the linear function L_d is defined by the mapping $L(x, y) \mapsto (x, x\|0^{c-r} \oplus y)$, where $(x, y) \in \{0,1\}^r \times \{0,1\}^c$. Clearly, L_e and L_d functions are involutions, and hence invertible.

Corollary 2. *For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, the AEAD advantage of $\mathscr{A}$ against* SpoC *is given by,*

1. *If $c > r$, then*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC}}(\mathscr{A}) \leq \frac{10q_p\sigma_d}{2^b} + \frac{4b^3q_p^2\sigma_d}{2^{b+c}} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^r} + \frac{2\sigma(2\sigma + q_p)}{2^b} + \frac{6\sigma_e q_p}{2^b} + \frac{12rq_p}{2^c} + \frac{\sigma_e + q_p}{2^b} + \frac{4rq_p\sigma_d}{2^{2c}}. \quad (12)$$

2. *If $c = r$, then*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC}}(\mathscr{A}) \leq \frac{8r\sigma_d}{2^c} + \frac{4b^3q_p^2\sigma_d}{2^{b+c}} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^r} + \frac{2\sigma(2\sigma + q_p)}{2^b} + \frac{6\sigma_e q_p}{2^b} + \frac{12rq_p}{2^c} + \frac{\sigma_e + q_p}{2^b} + \frac{4rq_p\sigma_d}{2^{2c}}. \quad (13)$$

Corollary 2 follows from Theorem 4, Proposition 1, 2, and the fact that $r = \tau$.

7.2.1 Further Simplifications

The primary version of the NIST submission SpoC has $b = 192$, $r = \tau = 64$, and $\kappa = c = 128$. Using the NIST prescribed values of σ and q_p we have $\sigma < 2^r = 2^\tau \leq q_p < 2^c$ and $4b^3q_p\sigma_d < 2^b$. We simplify Eq. (12) using these relations to obtain the following bound

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC\text{-}64}}(\mathscr{A}) \leq \frac{14rq_p}{2^c} + \frac{2\sigma}{2^r} + \frac{24q_p\sigma}{2^b}.$$

The secondary version of SpoC has $b = 256$, and $r = \tau = c = \kappa = 128$. Using the NIST prescribed values of σ and q_p we have $\sigma \leq q_p$ and $4b^3q_p\sigma_d < 2^b$. We simplify Eq. (13) using these relations to obtain the following bound

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC\text{-}128}}(\mathscr{A}) \leq \frac{14rq_p}{2^c} + \frac{9r\sigma}{2^r} + \frac{14q_p\sigma}{2^b}.$$

7.3 Improved Security of Beetle

In Beetle [CDNY18], the linear function L_e is defined as $L_e(y\|x_1\|x_2) \mapsto (y\|x_2\|x_2 \oplus x_1)$, where $(y, x_1, x_2) \in \{0,1\}^c \times \{0,1\}^{r/2} \times \{0,1\}^{r/2}$. The linear function $L_{d,i}$ is defined by

$$L_{d,i}(y\|x_1\|x_2) = \begin{cases} (y\|x_2\|\lfloor x_2 \oplus x_1 \rfloor_{r/2-i}\|\lceil x_1 \rceil_i) \text{ for } 0 \le i \le r/2 \\ (y\|\lfloor x_2 \rfloor_{r-i}\|\lceil x_2 \oplus x_1 \rceil_{i-r/2}\|x_1) \text{ for } r/2 \le i \le r \end{cases},$$

where $(y, x_1, x_2) \in \{0,1\}^c \times \{0,1\}^{r/2} \times \{0,1\}^{r/2}$. Clearly the L_e and $L_{d,i}$ functions are invertible for all $0 \le i \le r$.

Previous Bound: In [CDNY18], the authors proved that for any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{Beetle}}(\mathscr{A}) \le \frac{2(\sigma_e + q_p)\sigma_d}{2^b} + \left(\frac{\sigma_e + q_p}{2^{r-1}} + \frac{q_p}{2^c}\right)^r + \frac{r\sigma_d}{2^c} + \frac{q_v}{2^r}. \tag{14}$$

New Improved Bound: Since the feedback function of Beetle is invertible, we can apply Theorem 4. Specifically, we have

Corollary 3. *For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, the AEAD advantage of $\mathscr{A}$ against* Beetle *is given by,*

1. *If $c > r$, then*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{Beetle}}(\mathscr{A}) \le \frac{10q_p\sigma_d}{2^b} + \frac{4b^3q_p^2\sigma_d}{2^{b+c}} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^r} + \frac{2\sigma(2\sigma + q_p)}{2^b} + \frac{6\sigma_e q_p}{2^b} + \frac{12rq_p}{2^c} + \frac{\sigma_e + q_p}{2^b} + \frac{4rq_p\sigma_d}{2^{2c}}. \tag{15}$$

2. *If $c = r$, then*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{Beetle}}(\mathscr{A}) \le \frac{8r\sigma_d}{2^c} + \frac{4b^3q_p^2\sigma_d}{2^{b+c}} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^r} + \frac{2\sigma(2\sigma + q_p)}{2^b} + \frac{6\sigma_e q_p}{2^b} + \frac{12rq_p}{2^c} + \frac{\sigma_e + q_p}{2^b} + \frac{4rq_p\sigma_d}{2^{2c}}. \tag{16}$$

Corollary 3 follows from Theorem 4, and Proposition 1 and 2.

Remark 6. The major difference between our analysis and the analysis of [CDNY18] is that, we use the expected number of multi-chains to bound the security of Beetle, whereas in [CDNY18], it was only done using multicollision probability at the rate part. This is the reason why our new bound is much tighter than the existing one.

7.3.1 Security Bounds for PHOTON-Beetle

PHOTON-Beetle [BCD+19], a round 2 candidate of NIST LwC standardization process, uses a simple variant of Beetle as the underlying mode of operation. PHOTON-Beetle uses a different feedback function as compared to Beetle, such that the ranks of resulting L_e and L_d functions are b and $b-1$, respectively. Thus, L_d is non-invertible. So, we cannot apply Theorem 4 to get the security bound for PHOTON-Beetle. However, we can still apply Corollary 1 to obtain the following security bound.

Corollary 4. *For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, the AEAD advantage of $\mathscr{A}$ against* PHOTON-Beetle *is given by,*

1. *If $c = r$, then*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{PHOTON\text{-}Beetle}}(\mathscr{A}) \leq \frac{8r\sigma_d}{2^c} + \frac{8b^3q_p^2\sigma_d}{2^{b+c}} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^r} + \frac{2\sigma(2\sigma+q_p)}{2^b} + \frac{q_p^2}{2^b} + \frac{6\sigma_e q_p}{2^b} + \frac{12rq_p}{2^c} + \frac{\sigma_e + q_p}{2^b} + \frac{4rq_p\sigma_d}{2^{2c}}. \quad (17)$$

2. *If $c > r$, then*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{PHOTON\text{-}Beetle}}(\mathscr{A}) \leq \frac{4\tau\sigma_d}{2^c} + \frac{5q_p\sigma_d}{2^b} + \frac{8b^3q_p^2\sigma_d}{2^{b+c}} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma(2\sigma+q_p)}{2^b} + \frac{q_p^2}{2^b} + \frac{6\sigma_e q_p}{2^b} + \frac{10\sigma_e q_p}{2^b} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{\sigma_e + q_p}{2^b} + \frac{5q_p\sigma_e\sigma_d}{2^{b+c}}. \quad (18)$$

Corollary 4 follows from Corollary 1, and Proposition 1 and 3. Further, using the relations that $\sigma \leq q_p$ (as per NIST LwC requirements) we can bound the advantage in case of primary version as,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{PHOTON\text{-}Beetle[128]}}(\mathscr{A}) \leq \frac{q_p}{2^\kappa} + \frac{2\sigma}{2^r} + \frac{10b^2q_p^2}{2^b} + \frac{24rq_p}{2^c} + \frac{12\sigma q_p}{2^b},$$

and the secondary version as,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{PHOTON\text{-}Beetle[32]}}(\mathscr{A}) \leq \frac{q_p}{2^\kappa} + \frac{3\sigma}{2^\tau} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{4q_p^2}{2^b} + \frac{28q_p\sigma}{2^b}.$$

Clearly, by this new improved security bound, it is proved that both the primary and the secondary version of PHOTON-Beetle achieve NIST LwC requirements, i.e., security upto $q_p \leq 2^{112}$ and $\sigma \leq 2^{53}$ bits.

7.4 Smallest Sponge-type AEAD Satisfying NIST LwC Requirements

Keeping in mind the NIST LwC requirement of time complexity $q_p = 2^{112}$ and data complexity $r\sigma = 2^{53}$ we try to find out the smallest possible permutation under which a Sponge-type AEAD mode, like Beetle and SpoC, can achieve security.

In this subsection, we try to derive a Sponge-type AEAD with smallest state size that can achieve security under NIST LwC requirement.Interpretation of Corollary 2 and 3.

First observe that it is wise to concentrate on AEAD modes with invertible feedback functions, as the non-invertible functions have an added requirement of $q_p^2 < 2^b$ or $q_p\sigma < 2^c$ (as per the current progress in security bounds). Consequently, we concentrate on Beetle and SpoC, the two candidates for Sponge-type AEADs with invertible feedback functions.

Corollary 2 and 3 are derived using the parameter values recommended in the original proposals [CDNY18, AGH+19]. Now, we derive a more general security bound under the assumptions: $\kappa < b$, $r < c$, $2^r \leq \sigma \leq 2^\tau \leq q_p \leq 2^c$ and $2^b \leq b^2 q_p^2$. Specifically, we have

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC/Beetle}}(\mathscr{A}) \leq \frac{3q_p}{2^\kappa} + \frac{2\sigma}{2^\tau} + \frac{32q_p\sigma}{2^b} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{4b^3q_p^2\sigma}{2^{b+c}} + \frac{5q_p\sigma^2}{2^{b+c}}.$$

Now, the primary version of AEAD must have $\tau = 64$. This immediately sets $b \geq \tau + \log_2(4\tau q_p) = 184$. Further, if we take $4b^3q_p \leq 2^c$, then the above bound simplifies to

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC/Beetle}}(\mathscr{A}) \leq \frac{3q_p}{2^\kappa} + \frac{2\sigma}{2^\tau} + \frac{34q_p\sigma}{2^b} + \frac{4\tau q_p}{2^{b-\tau}}.$$

By setting $4b^3q_p = 2^c$, we get $c \approxeq 138$ and $r = 46$. Thus, SpoC and Beetle achieve security under NIST LwC requirements, when instantiated with a 184-bit permutation with 46-bit rate. For instance, Beetle and SpoC instantiated with the 192-bit permutation sLiSCP-light [ARH+18] or the 196-bit permutation from the PHOTON family [GPP11], achieve NIST LwC security requirements.

7.5 Security of Sponge AEAD

In case of the original Sponge AEAD or duplex construction, the L_d function is defined by $L_d(x, y) \mapsto (0^r, y)$ where $(x, y) \in \{0,1\}^r \times \{0,1\}^c$. Note that the L_d function is not invertible. As described in section 5.2.2 and 6.4.1, bounding μ_{q_p} for non-invertible L_d is difficult.

The problem is further compounded in case of Sponge AEAD as $\mathsf{rank}(L_d) = c$, a substantial drop from b (for fairly efficient constructions). So, even the application of Corollary 1 gives a sub-optimal (and existing) bound with dominating term $q_p\sigma/2^c$. More generally, we can derive the following security bound using a similar analysis as in the case of TtP with invertible L_d,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{Sponge}}(\mathscr{A}) \leq \frac{\sigma_d \cdot \mu_{q_p}}{2^c} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma(2\sigma + q_p)}{2^b} + \frac{6\sigma_e q_p}{2^b} + \frac{2q_p\mathsf{mcoll}(\sigma_e, 2^r)}{2^c} + \frac{q_p\mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}} + \frac{\sigma_e + q_p}{2^b} + \frac{q_p\sigma_d\mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}, \tag{19}$$

where the main hardness lies in giving a tight estimate for μ_{q_p}. Bounding μ_{q_p} in case of Sponge AE is an interesting problem which is open to further research. However, it seems very hard to have a tight estimate of μ_{q_p} for Sponge AE. As mentioned above, a straightforward estimate of μ_{q_p} leads to the known security bound of $q_p\sigma/2^c$. So as of now the tight security bound of Sponge AE is still an open problem. However, our result helps in reducing the problem of finding tight bound to solving some functional graph problem (estimation of μ_{q_p}). The functional graph of random functions are well-studied in cryptanalysis of iterated hash functions and MACs [PW14, BWGG17, BGW18]. It is quite possible that similar approach may lead to a better understanding of the security of Sponge AE.

8 Matching Attack on Transform-then-Permute

Now we see some matching attacks for the bound. We explain the attacks for the simplified version (by considering empty associated data).

1. Suppose μ_{q_p} maximizes for some adversary $\mathcal{B}$ interacting with Π. Now, the AE algorithm $\mathcal{A}$ will run the algorithm $\mathcal{B}$ to get the primitive transcript ω_p. We first make q_d many encryption queries with single block messages with distinct nonces $N_1, \ldots, N_{q_d}$ and hence for all $1 \leq i \leq q_d$, $\lceil Y_{i,0} \rceil_r$, $\lceil X_{i,1} \rceil_r$ and $\lceil Y_{i,1} \rceil_\tau$ values are known. Suppose for length m_i, the multi-chain for the graph induced by ω_p start from the nodes (whose r most significant bits of the domain is u_i) to the nodes (whose τ most significant bits of the range is T_i) and with label x_i. Now we choose the appropriate ciphertext C^*_1 such that $\lceil X^*_{i,1} \rceil_r = u_i$. Moreover, we choose $C^*_{i,j}$ such that $\overline{C^*_{i,j}}$ is same as $x_{i,j}$ (here we assume that $\mathcal{B}$ makes queries so that the labels are compatible with encoding function).

   Now, we make decryption queries (N_i, C^*_i, T_i). With probability $\mathsf{W}_{m_i}/2^c$, the ith forgery attempt would be successful. Then maximizing $\frac{\mathsf{W}_{m_i}}{m_i}$ and by taking expectation, we achieve the desired success probability.

2. Guessing the key K through primitive query would lead a key-recovery and hence all other attacks. The correct guess of the key can be easily detected by making some more queries for each guess to compute an encryption query. This attack requires $q_p = \mathcal{O}(2^\kappa)$. Similarly random forging gives success probability of forging about $\mathcal{O}(q_d/2^\tau)$.

3. Another attack strategy can be adapted to achieve $\sigma_e q_p/2^b$ bound. We look for a collision among X-values and primitive-query inputs. This can be again detected by adding one or two queries to each guess. The same attack works with success probability $q_p\mathsf{mcoll}(\sigma_e, 2^r)/2^c$ if we make primitive queries after making all encryption queries.

4. A similar attack strategy can be adapted to achieve $q_p\mathsf{mcoll}(\sigma_e, 2^r)/2^{b-\tau}$ bound. We look for a collision among T-values and primitive-query inputs where primitive queries are done after the encryption queries to predict the unknown $b - \tau$ bits of the final output value.

These attacks show that the bounds in Theorem 4 and equation (19) are tight.

9 Conclusion

In this paper we have proved improved bound for Beetle and provided similar bound for newly proposed mode SpoC. Our bound resolves all limitations known for Beetle and Sponge AE. We are able to provide tight estimation of μ_{q_p} when the feedback function for decryption is invertible. This is the case for Beetle and SpoC, but not for Sponge duplex.

Although as discussed in section 8, we obtain tight expression for AE advantage for Sponge AE, the variable μ_{q_p} (present in our upper bound) needs to be tightly estimated.

Acknowledgements

The authors would like to thank Dr. Jooyoung Lee for his insightful comments and assistance in preparing the final draft of this paper. The authors would also like to thank all the anonymous reviewers of ToSC 2020 for their valuable comments. The authors are supported by the project "Study and Analysis of IoT Security" under Government of India at R. C. Bose Centre for Cryptology and Security, Indian Statistical Institute, Kolkata, India.

References

[AGH+19] 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.

[AHMN10] Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and María Naya-Plasencia. Quark: A lightweight hash. In *Cryptographic Hardware and Embedded Systems, CHES 2010. Proceedings*, pages 1–15, 2010.

[ARH+18] Riham AlTawy, Raghvendra Rohit, Morgan He, Kalikinkar Mandal, Gangqiang Yang, and Guang Gong. Sliscp-light: Towards hardware optimized sponge-specific cryptographic permutations. *ACM Trans. Embedded Comput. Syst.*, 17(4):81:1–81:26, 2018.

[BCD+19] Zhenzhen Bao, Avik Chakraborti, Nilanjan Datta, Jian Guo, Mridul Nandi, Thomas Peyrin, and Kan Yasuda. PHOTON-Beetle. Submission to NIST LwC Standardization Process (Round 2), 2019.

[BDPA07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. In *ECRYPT Hash Workshop 2007. Proceedings*, 2007.

[BDPA08] 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.

[BDPA10] 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.

[BDPA11a] 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.

[BDPA11b] 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.

[BDPA13] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In *Advances in Cryptology - EUROCRYPT 2013. Proceedings*, pages 313–314, 2013.

[BGW18] Zhenzhen Bao, Jian Guo, and Lei Wang. Functional graphs and their applications in generic attacks on iterated hash constructions. *IACR Trans. Symmetric Cryptol.*, 2018(1):201–253, 2018.

[BKL+13] Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. SPONGENT: the design space of lightweight cryptographic hashing. *IEEE Trans. Computers*, 62(10):2041–2053, 2013.

[BWGG17] Zhenzhen Bao, Lei Wang, Jian Guo, and Dawu Gu. Functional graph revisited: Updates on (second) preimage attacks on hash combiners. In *Advances in Cryptology - CRYPTO 2017. Proceedings, Part II*, pages 404–427, 2017.

[BYG91] Ricardo A. Baeza-Yates and Gaston H. Gonnet. *Handbook of Algorithms and Data Structures in Pascal and C.* Addison-Wesley, 1991.

[CDNY18] 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.

[CIMN17] 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.

[CS14] Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In *Advances in Cryptology - EUROCRYPT 2014. Proceedings*, pages 327–350, 2014.

[DEMS16] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon. CAESAR recommendation for lightweight applications, 2016.

[DMA17] Joan Daemen, Bart Mennink, and Gilles Van Assche. Full-state keyed duplex with built-in multi-user support. In *Advances in Cryptology - ASIACRYPT 2017. Proceedings, Part II*, pages 606–637, 2017.

[Gon81] Gaston H. Gonnet. Expected length of the longest probe sequence in hash code searching. *J. ACM*, 28(2):289–304, 1981.

[GPP11] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. In *Advances in Cryptology - CRYPTO 2011. Proceedings*, pages 222–239, 2011.

[JLM14] 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.

[JLM+19] 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.

[Men18] Bart Mennink. Key prediction security of keyed sponges. *IACR Transactions on Symmetric Cryptology*, 2018(4):128–149, Dec. 2018.

[MN17] 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.

[MRV15] 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.

[Pat91] Jacques Patarin. *Etude des Générateurs de Permutations Pseudo-aléatoires Basés sur le Schéma du DES.* PhD thesis, Université de Paris, 1991.

[Pat08] Jacques Patarin. The "coefficients H“ technique. In *Selected Areas in Cryptography - SAC 2008. Revised Selected Papers*, pages 328–345, 2008.

[PW14] Thomas Peyrin and Lei Wang. Generic universal forgery attack on iterative hash-based macs. In *Advances in Cryptology - EUROCRYPT 2014. Proceedings*, pages 147–164, 2014.

[RS98] Martin Raab and Angelika Steger. "balls into bins" - A simple and tight analysis. In *Randomization and Approximation Techniques in Computer Science, Second International Workshop, RANDOM'98. Proceedings*, pages 159–170, 1998.

[SF96] Robert Sedgewick and Philippe Flajolet. *An introduction to the analysis of algorithms.* Addison-Wesley-Longman, 1996.

[WH19] Hongjun Wu and Tao Huang. Clx. Submission to NIST LwC Standardization Process (Round 1), 2019.

[Wu11] Hongjun Wu. The hash function jh. SHA-3 candidate submitted to NIST, 2011.