On the Security of Sponge-type Authenticated Encryption Modes

Bishwajit Chakraborty, Ashwin Jha,Mridul Nandi

Indian Statistical Institute, Kolkata, India
bishu.math.ynwa@gmail.com,ashwin.jha1991@gmail.com,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 of the algorithm), 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 forced us 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 · AE · 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 authenticated encryption, the duplex mode [BDPA11a]. In particular, the recently concluded CAESAR competition for the development of authenticated encryption (AE) 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 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 1 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 AE 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 AE 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 security up to $DT \ll 2^b$, $D \ll \min\{2^{b/2}, 2^\kappa\}$, $T \ll \min\{2^{b/2}, 2^{c-\log_2 r}, 2^\kappa\}$, and integrity security up to $DT \ll 2^c$, $D \ll \min\{2^{c/2}, 2^{b/2}, 2^\kappa, 2^\tau\}$, $T \ll \min\{2^{b/2}, 2^{c-\log_2 r}, 2^\kappa\}$, where τ denotes the tag size. Note that the integrity security 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, ([JLM+19]) tight privacy analysis is provided. However, one of the dominating terms present in all of the existing integrity analysis of duplex authenticated encryption is

$$DT/2^c.$$

Moreover, no known matching forgery attack on it is not 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 term $DT/2^c$. They showed privacy security up to $DT \ll 2^b$, $D \ll 2^{b/2}$, $T \ll 2^c$, and integrity security up to $DT \ll 2^b$, $D \ll \min\{2^{b/2}, 2^{c-\log_2 r}, 2^r\}$, $T \ll \min\{2^{c-\log_2 r}, 2^r, 2^{b/2}\}$, with an assumption that $\kappa = c$ and $\tau = r$.

1.1.1 Security of Sponge-typed AE in Light of NIST LwC Requirement:

In NIST's LwC call for submissions, it is mentioned that the primary AE version 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 sponge-type submission to NIST LwC standardization process uses 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 requires at least 120-bit capacity and 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, it would be interesting to see whether we can get rid of the limitations in 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 AE 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 AE security is achieved. To be specific, we show that the AE 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. improved and tight bound for Beetle, and
2. a security proof validating the security claims of SpoC.

Notably, we show that both Beetle and SpoC achieve NIST LwC requirements with just 128-bit capacity and ≥ 32-bit rate. In other words, they achieve NIST LwC requirements with just 160-bit state, which to the best of our knowledge is the smallest possible state size among all known sponge like 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. In Section 3 we state some multicollision results with proofs which are used in the paper. In section 4 we define what we call the multi-chain structure and give an upper bound on the expected number of multichains that can be formed by an adversary in the case when L is invertible. In section 5 we study a Sponge-type AEAD construction namely Transform-then-Permute with a generalization of the feedback function used in the duplexing interface and give a security bound for the special case when the feedback function is invertible. We show that it encompasses the feedback functions used in duplex, Beetle, SpoC etc. We show that Beetle and SpoC modes fall under the class where the feedback function is invertible and hence for those mode optimal AEAD security is achieved. In section 6 Using the multi-chain security game from section 4 we give a complete security proof of the AEAD security bound given in 5. Finally, in section 7 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, 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. $\mathsf{X} \overset{\mathsf{wor}}{\leftarrow} \mathcal{X}$ denotes uniform sampling of X 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 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: For $b \in \mathbb{N}$, let $\mathsf{Perm}(b)$ denote the set of all permutations over $\{0,1\}^b$, and $\Pi \leftarrow_{\$} \mathsf{Perm}(b)$. Let Func denote the set of all functions from $\mathcal{N} \times \mathcal{A} \times \mathcal{M}$ to $\mathcal{M} \times \mathcal{T}$ which are length preserving with respect to $\mathcal{M}$ 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,

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

Here $\mathscr{A}^{\mathsf{E_K}, \mathsf{D_K}, \Pi^\pm}$ denotes $\mathscr{A}$'s response after its interaction with $\mathsf{E_K}$, $\mathsf{D_K}$, and $\Pi^\pm$, respectively. Similarly, $\mathscr{A}^{\Gamma, \bot, \Pi^\pm}$ denotes $\mathscr{A}$'s response after its interaction with Γ, $\bot$, and $\Pi^\pm$. Note that here we consider only those adversaries $\mathscr{A}$ which do not make any decryption query, which is the response of any previous encryption queries. In such case the adversary can trivially distinguish between the real oracle from the ideal permutation one.

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 $\mathsf{E_K}$ and $\mathsf{D_K}$, respectively. σ_e and σ_d denote the total number of blocks of input (associated data and message) lengths across all encryption and decryption(respectively) queries. We sometime also write $q = q_e + q_d$ and $\sigma = \sigma_e + \sigma_d$ to denote the combined construction query resources.

- q_f and q_b denote the number of queries to Π^+ and Π^-, respectively. We sometime also use $q_p = qf + +q_b$, to denote the combined primitive query resources.

We remark here that q and σ correspond to the *online or data complexity*, and q_p corresponds to the *offline or time complexity* of the adversary. *Any adversary that adheres to the above mentioned resource constraints is called an* $(q_p, q_e, q_d, \sigma_e, \sigma_d)$*-adversary.*

2.2 H-coefficient Technique

Consider a computationally unbounded and deterministic adversary $\mathscr{A}$ that tries to distinguish the real oracle, say $\mathcal{O}_1$, from the ideal oracle, say $\mathcal{O}_0$. 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 H-coefficient technique (or simply the H-technique).

Theorem 1 (H-coefficient technique [Pat91, Pat08])**.** *Let Ω be the set of all realizable transcripts. For some $\epsilon_{\mathsf{bad}}, \epsilon_{\mathsf{ratio}} > 0$, suppose there is a set $\Omega_{\mathsf{bad}} \subseteq \Omega$ satisfying the following:*

- $\Pr[\Theta_0 \in \Omega_{\mathsf{bad}}] \le \epsilon_{\mathsf{bad}}$;
- *For any $\omega \notin \Omega_{\mathsf{bad}}$,*
$$\frac{\Pr[\Theta_1 = \omega]}{\Pr[\Theta_0 = \omega]} \ge 1 - \epsilon_{\mathsf{ratio}}.$$

Then for any adversary $\mathscr{A}$, we have the following bound on its AEAD distinguishing advantage:
$$\mathbf{Adv}^{\mathsf{aead}}_{\mathcal{O}_1}(\mathscr{A}) \le \epsilon_{\mathsf{bad}} + \epsilon_{\mathsf{ratio}}.$$

A proof of this theorem is available in multiple papers including [Pat08, CS14, MN17].

3 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.

3.1 Expected Maximum Multicollision in a Uniform Random Sample

Let $\mathsf{X}_1, \ldots, \mathsf{X}_q \leftarrow_{\$} \mathcal{D}$ where $|\mathcal{D}| = N$ and $N \ge 2$. For the simplicity of notations, we write $\log_2 N$ as n. We denote the maximum multicollision random variable for the sample as

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

$\mathsf{mc}_{q,N}$. More precisely, $\mathsf{mc}_{q,N} = \max_a |\{i : \mathsf{X}_i = a\}|$. 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 hence

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

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 $\mathsf{mcoll}(q, N)$.

Proposition 1. *For $n \geq 2$,*

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

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

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

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

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 $\mathsf{X}_1, \ldots, \mathsf{X}_q \overset{\text{wor}}{\leftarrow} \{0,1\}^b$ and then we define $\mathsf{Y}_i = \lceil \mathsf{X}_i \rceil_r$ for some $r < b$. In this case, we have

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

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.

3.2 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. 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 $\Pr\left[\mathsf{Z}_{i_1,j_1} = a, \cdots, \mathsf{Z}_{i_\rho,j_\rho} = a\right]$. Without loss of generality we can assume that $i_k = j_k = k$. Let $N := 2^b$. We also 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, $\forall 1 \leq i \leq \rho$. 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, $|V_{2\rho-k}| \leq \rho^{2k}(N-k)_{\rho-k}$.

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

We denote the maximum multicollision random variable for the sample as $\mathsf{mc}'_{q,N}$. Then we have as before

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

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

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

Proposition 2. *For $N > 2^{16}, n = \log N$*

$$\mathsf{mcoll}'(q, N) \leq \begin{cases} \frac{4n}{\log n} & \text{if } n^2 q \leq N \\ \frac{4n}{\log n}\lceil \frac{n^2 q}{N} \rceil & \text{if } n^2 q \geq N. \end{cases}$$

Proof. Let $q \leq \frac{N}{n^2}$. Since $N > 2^{16}$, if $\rho = \frac{4n}{\log n}$ then, we have $q \leq \frac{N-2\rho}{\rho^2}$. Hence, $2qN \cdot \left(\frac{qe\rho}{N-2\rho}\right)^{\rho} \leq N^2 \cdot \left(\frac{e}{\rho}\right)^{\rho}$. Now, $\left(\frac{e}{\rho}\right)^{\rho} \leq \left(\frac{e}{4}\right)^{4n} \leq \frac{1}{N^2} \implies N^2 \cdot \left(\frac{e}{\rho}\right)^{\rho} \leq 1$.

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

4 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.

4.1 The 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 $v_i \overset{x}{\to} v_j$) 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

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

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.

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 $\mathcal{G}^L_{\mathcal{L}}$ 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 maximum size of the set of multi-chain of length k (with some label x) is denoted as W_k. Thus, 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.

4.2 The 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 many 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 many 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 and consider the directed graph $\mathcal{G}^L_{\theta'}$. For any k, we have already defined W_k. Now we define the maximum multi-chain advantage as

$$\mu_{d,t} = \max_{\mathscr{A}} \max_{k \le d} \mathsf{Ex}\left[\frac{\mathsf{W}_k}{k}\right].$$

4.3 Bounding $\mu_{d,t}$ for Invertible L Functions

In this section, we derive concrete bounds for $\mu_{d,t}$ under a special assumption that the underlying feedback function is invertible.

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

$$\mu_{d,t} \le \mathsf{mcoll}(t, 2^{\tau}) + \mathsf{mcoll}(t, 2^{r}) + \mathsf{mcoll}'(t^2, 2^{b}).$$

4.3.1 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 notations related to multi-chain W_k.

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 $\{i : \mathsf{dir}_i = -, \lceil v_i \rceil_{r} = a\}$ 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 $\{(i,j) : \mathsf{dir}_i = +,\ \mathsf{dir}_j = -,\ v_i \oplus u_j = a\}$ and $\max_a \mathsf{W}^{\mathsf{mitm},a}$ is denoted as $\mathsf{W}^{\mathsf{mitm}}$.

Lemma 1. *For all possible interactions, we have*

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

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: The multi-chain consists of at least one chain that uses 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 $\mathsf{W}_k^{\mathsf{fwd\text{-}bck}} = n$. Then, it is sufficient to show that there exist an index $j \in [k]$, such that the size of the set $\{i : (\mathsf{dir}_{j-1}^i, \mathsf{dir}_j^i) \in \{(+,-),(-,+)\},\ v_{j-1}^i \oplus u_j^i = x_j\} \geq \lceil n/k \rceil$. 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 many buckets and n many chains, by pigeonhole principle, we must have one bucket $j \in [k]$, such that it holds at least $\lceil n/k \rceil$ many chain indices. □ □

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, i.e., distribution 1 of subsection 3.2. Further, $\mathsf{W}^{\mathsf{mitm}}$ is the random variable corresponding to the maximum multicollision in a sum of random permutation sample of size t, i.e., distribution 2 of sub section 3.2. Now, using linearity of expectation, we have

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

4.4 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 $\mathit{yield}_{c,k}(\mathcal{L})$ 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 $\mathcal{G}_{\mathcal{L}}$.

Let $Q = (u_i, v_i)_{i \in (q]}$ be the query transcript of the underlying permutation in any Sponge. 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 $\mathit{yield}_{c,k}(\mathcal{L})$ 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 a different problem from $yield_{c,k}(\mathcal{L})$. A notable change in our work is to deal multicollision of sum of two permutation calls (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.

5 Transform-then-Permute Construction

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

5.1 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 nonnegative 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 $a := a(|A|, \nu)$.

– **Case** $|A| \leq \theta$: $\mathsf{parse}(N, A) = D \parallel 0^{\theta - |A|} \in \{0,1\}^{b-\kappa}$.

– **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 $(Z_1, \ldots, Z_m) \xleftarrow{r} Z$ where $m = \lceil |Z|/r \rceil$.

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

$$\delta_i = \begin{cases} 0 & \textit{if } i \notin \{a\} \cup \{a+m\} \\ 1 & \textit{if } (i = a \wedge r' \mid |A| - \theta) \vee (i = a + m \wedge r \mid |M|) \\ 2 & \textit{otherwise.} \end{cases}$$

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($N, A, Z$)
    $a \leftarrow a(|A|, |N|)$, $m \leftarrow \lceil |Z|/r \rceil$
    $(A_0, A_1, \ldots, A_a) \leftarrow \mathsf{Parse}(N, A)$
    $(Z_1, \ldots, Z_m) \xleftarrow{r} Z$
    $(\delta_0, \ldots, \delta_{a+m}) \leftarrow \mathsf{DS}(|A|, |Z|)$
    for $i = 0$ to $a$ do
        if $i = a$ and $m = 0$ then
            $D_i \leftarrow \mathsf{encode}(A_i, (0, 1, \delta_i))$
        else
            $D_i \leftarrow \mathsf{encode}(A_i, (0, 0, \delta_i))$
    for $i = 1$ to $m$ do
        $D_{a+i} \leftarrow \mathsf{encode}(Z_i, (1, 0, \delta_{i+m}))$
    return $(D_0, \ldots, D_{a+m})$
```

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 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 $L'_{d,|C|}(Y, C, \Delta) = (X, C)$ if and only if $L'_e(Y, M, \Delta) = (X, C)$. Fig 1 provides an illustration how a message block is processed.

5.2 The Description of Transform-then-Permute AEAD

We describe the $\mathsf{Transform\text{-}then\text{-}Permute}$ algorithm in Algorithm 2 which generalizes duplexing method used in sponge AEAD. Figure 2 illustrates a simple case when $|N| = b - \kappa$.

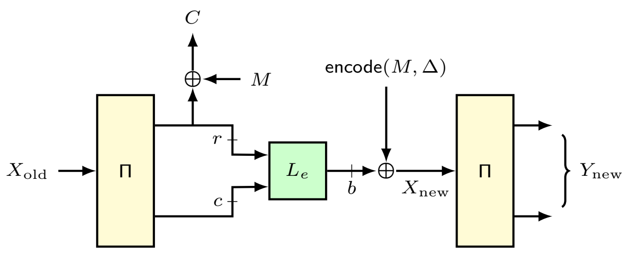

Figure 1: Illustration of the feedback process for a message block M of $|M|$ bits. Here encode(M, Δ) represents some encoding of $|M|$ bits string to a b-bit string as described above and L_e is a linear transformation applied on b-bit strings.

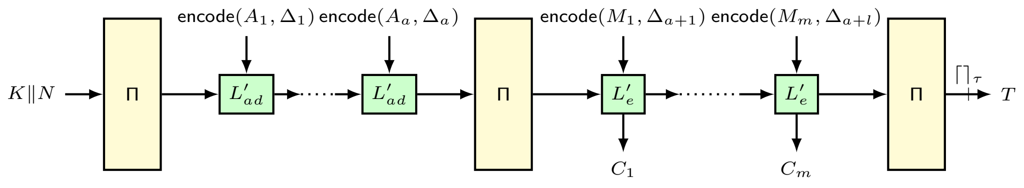

Figure 2: A Complete Diagram of the Transform-then-Permute AEAD mode. Here we assume $|N| = b - \kappa$, $L'_{ad}(Y, A) = L_{ad} \oplus A$. L_{ad}, L'_e, encode functions and Δ values are as described before.

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

```
1: function Enc$(K, N, A, M)$
2:     $a \leftarrow a(|A|, |N|)$, $m \leftarrow \lceil |M|/r \rceil$
3:     $(D_0, D_1, \ldots, D_{a+m}) \leftarrow$ Fmt$(N, A, M)$
4:     $(M_1, \ldots, M_m) \xleftarrow{r} M$
5:     $X_0 \leftarrow K\|0^{b-\kappa} \oplus D_0$
6:     $Y_0 \leftarrow \Pi(X_0)$
7:     for $i = 1$ to $a$ do
8:         $X_i \leftarrow L_{ad}(Y_{i-1}) \oplus D_i$
9:         $Y_i \leftarrow \Pi(X_i)$
10:    for $j = 1$ to $m$ do
11:        $i = a + j$
12:        $X_i \leftarrow L_e(Y_{i-1}) \oplus D_i$
13:        $C_j \leftarrow M_j \oplus \lceil Y_{i-1} \rceil_{|M_j|}$
14:        $Y_i \leftarrow \Pi(X_i)$
15:    $T \leftarrow \lceil Y_{a+m} \rceil_\tau$
16:    return $(C_1\| \ldots \|C_m, T)$
```

```
1: function Dec$(K, N, A, C, T)$
2:     $a \leftarrow a(|A|, |N|)$, $m \leftarrow \lceil |C|/r \rceil$
3:     $(D_0, D_1, \ldots, D_{a+m}) \leftarrow$ Fmt$(N, A, C)$
4:     $(C_1, \ldots, C_m) \xleftarrow{r} C$
5:     $X_0 \leftarrow K\|0^{b-\kappa} \oplus D_0$
6:     $Y_0 \leftarrow \Pi(X_0)$
7:     for $i = 1$ to $a$ do
8:         $X_i \leftarrow L_{ad}(Y_{i-1}) \oplus D_i$
9:         $Y_i \leftarrow \Pi(X_i)$
10:    for $j = 1$ to $m$ do
11:        $i = a + j$
12:        $X_i \leftarrow L_{d,|C_i|}(Y_{i-1}) \oplus D_i$
13:        $M_j \leftarrow C_j \oplus \lceil Y_{i-1} \rceil_{|C_j|}$
14:        $Y_i \leftarrow \Pi(X_i)$
15:    $T \leftarrow \lceil Y_{a+m} \rceil_\tau$
16:    return $T' =_? T : M_1\| \ldots \|M_m, \perp$
```

Lemma 2. *Given any two tuples $(N, A, Z) \neq (N', A', Z')$ and Fmt$(N, A, Z) = (D_0, \ldots, D_{a+m})$ and Fmt$(N', A', Z') = (D'_0, \ldots, D'_{a'+m'})$, we have*

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

2. $(D'_a, \ldots, D'_{a+m}) \neq (D_a, \ldots, D_{a+m})$ *whenever* $(N, A) = (N', A')$ *and* $m \leq m'$.

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

1. Let $(N, A) \neq (N', A')$. Then we have $(A_0, A_1, \dots, A_a) \neq (A'_0, A'_1, \dots, A'_{a'})$. 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 nonnegative $i \leq a$ such that $A_i \neq A'_i$ and hence $D_i \neq D'_i$.

2. Let $(N, A) = (N, A')$. Then we have $(A_0, A_i, \dots, A_a) = (A'_0, A'_i, \dots, A'_a)$. 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 $D_{a+m} = \mathsf{encode}(M_m, 1, \delta)$ 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 $D_{a+i} \neq D'_{a+i}$.

□

.

5.3 Security Analysis of TtP

We prove the following result on the AE security of Transform-then-Permute when the linear functions $L_{d,i}$ and L_e are invertible for all $1 \leq i \leq r$. Let q_p, q_e and q_d define the number of primitive, encryption and decryption queries respectively by an adversary and let σ_e and σ_d define all the data blocks processed, including nonce, associated data and message, in those encryption and decryption queries, respectively,.

Theorem 3 (main theorem)**.** *Let TtP be a construction where $L_{d,i}$ for all $i \in [r]$ and L_e are invertible. For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, we have*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{inv\text{-}TtP}}(\mathscr{A}) \leq\ & \frac{\sigma_d \mathsf{mcoll}(q_p, 2^\tau)}{2^c} + \frac{\sigma_d \mathsf{mcoll}(q_p, 2^r)}{2^c} + \frac{\sigma_d \mathsf{mcoll}'(q_p^2, 2^b)}{2^c} \\ & + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma + q_p)}{2^b} + \frac{8\sigma_e q_p}{2^b} + \frac{3q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c} \\ & + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}} + \frac{\sigma_e + q_p}{2^b} + \frac{q_p \sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.\end{aligned}$$

5.4 How to Convert a Generalized Sponge-type Constructions to TtP

In this section we describe why Transform-then-Permute captures wide class of permutation based sequential construction in which only non-linear operation lie in the underlying permutation. Let $L : \{0,1\}^b \times \{0,1\}^r \to \{0,1\}^b \times \{0,1\}^r$ be any linear function. 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:

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

where $L = \begin{bmatrix} L_{1,1} & L_{1,2} \\ L_{2,1} & L_{2,2} \end{bmatrix}$ consists 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 $L_{2,2}$. As $L_{2,1} \cdot Y + L_{2,2} \cdot M = C$ we must have $rank(L_{2,2}) = r$, since 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 $rank(L_{2,1}) = r$. If not, then $\exists \gamma$, an $1 \times r$ matrix, such that $\gamma \cdot L_{2,1} = 0$. Hence, $\gamma \cdot M = \gamma \cdot C$ holds with probability 1. Since in case of ideal permutation this event occurs with probability $\frac{1}{2}$. Hence there will be no privacy security for such a construction. As $rank(L_{2,1}) = r$, 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 $Y_i' = Z^{-1} \cdot Y_i$. If we redefine the random permutation output as $Z^{-1} \cdot \Pi(X_i)$ then it is clearly again a random permutation. 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.

5.5 New 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 is defined by the mapping $L_d(y\|x_1\|x_2) \mapsto (y\|x_2 \oplus x_1\|x_1)$, 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 functions are invertible.Further, they have full rank.

Remark 2. The PHOTON-Beetle [AGH+19] design which is currently in the round 2 of NIST LwC competition uses a feedback function which is a linear transformation of the feedback function of Beetle [CDNY18]. By applying the conversion method as described in subsection 5.4 the PHOTON-Beetle design can be viewed as a TtP design with the same linear function L_e as described above.

Thus, from theorem 3, we have

Corollary 1. *For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, we have the primary version of PHOTON-Beetle can be bounded by*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{PHOTON\text{-}Beetle}}(\mathscr{A}) \le \frac{4\tau\sigma_d}{2^c} + \frac{4r\sigma_d}{2^c} + \frac{4b\sigma_d}{2^c} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma+q_p)}{2^b} + \frac{8\sigma_e q_p}{2^b} \\ + \frac{12rq_p}{2^c} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{\sigma_e+q_p}{2^b} + \frac{4rq_p\sigma_d}{2^{2c}}.\end{aligned}$$

The secondary version of PHOTON-Beetle can be bounded by

$$\begin{aligned}\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{PHOTON\text{-}Beetle}}(\mathscr{A}) \le \frac{4\tau\sigma_d}{2^c} + \frac{4\sigma_d \cdot q_p}{2^b} + \frac{4b\sigma_d}{2^c} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma+q_p)}{2^b} + \frac{8\sigma_e q_p}{2^b} \\ + \frac{15q_p\sigma_e}{2^b} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{\sigma_e+q_p}{2^b} + \frac{5q_p\sigma_d\sigma_e}{2^{b+c}}.\end{aligned}$$

In [CIMN17], the authors proved that for any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$,

$$\mathbf{Adv}^{AE}_{\mathsf{Beetle}}(\mathscr{A}) \le \frac{2(\sigma_e+q_p)\sigma_d}{2^b} + \left(\frac{\sigma_e+q_p}{2^{r-1}} + \frac{q_p}{2^c}\right)^r + \frac{r\sigma_d}{2^c} + \frac{q_v}{2^r}. \quad (4)$$

The primary version of the PHOTON-Beetle[BCD+19] mode of AEAD has $r = \tau = c = 128$ and $b = 256$. Comparing with the σ and q_p values prescribed by NIST we have $2^r = 2^\tau \ge q_p \ge \sigma$ and $2^b \ge b^2 q_p^2$.

The secondary version of the PHOTON-Beetle[BCD+19] mode of AEAD has $r = 32, c = 224, \tau = 128$ and $b = 256$. Comparing with the σ and q_p values prescribed by NIST we have $2^\tau \ge q_p \ge \sigma$, $\sigma \ge 2^r$ and $2^b \ge b^2 q_p^2$.

By equation 4 the advantage of Beetle is bounded by $\left(\frac{q_p}{2^{r-1}}\right)^r$. Hence for Beetle to be secure, r has to be large.

It can be noticed that the primary version of PHOTON-Beetle has $r = 128 > 112$. Hence by equation 4, it is secure within the NIST requirements.

For secondary version of PHOTON-Beetle, we have $r = 32 < 112$ and hence equation 4 fails to prove the security for this version under NIST requirements.

The major difference between our analysis and the analysis of [CIMN17] is that, we use the expected number of multichains to bound the security of Beetle whereas in [CIMN17], it was only done using multicollision probability at the rate part. Hence our new bound is much tighter than that of the existing one.

Now Corollary 1 follows from, theorem 3,proposition 1 and proposition 2.

Further using the relation that $\sigma \leq q_p$ (as per NIST LwC requirements) we can bound the advantage for the primary version as,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{PHOTON\text{-}Beetle}}(\mathscr{A}) \leq \frac{q_p}{2^\kappa} + \mathcal{O}\left(\frac{rq_p}{2^c}\right).$$

and the secondary version as,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{PHOTON\text{-}Beetle}}(\mathscr{A}) \leq \frac{q_p}{2^\kappa} + \mathcal{O}\left(\frac{q_p\sigma}{2^b}\right)$$

Hence, by this new improved security bound, it is proved that both the primary and the secondary version of PHOTON-Beetle are secured under the NIST requirements.

5.6 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 the L_e and L_d functions are involutions, and hence invertible. Further, it is easy to check that they have full rank. Thus, from Eq. (**??**) and theorem 2, we have

Corollary 2. *For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, we have the primary version of SpoC can be bounded as,*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC}}(\mathscr{A}) \leq{} & \frac{5q_p\sigma_d}{2^{c+\tau}} + \frac{5q_p\sigma_d\cdot}{2^b} + \frac{4b^3q_p^2\sigma_d}{2^{b+c}} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma+q_p)}{2^b} \\ & + \frac{8\sigma_e q_p}{2^b} + \frac{12rq_p}{2^c} + \frac{4\tau q_p}{2^{b-\tau}} + \frac{\sigma_e + q_p}{2^b} + \frac{4rq_p\sigma_d}{2^{2c}}\end{aligned}$$

The primary version of SpoC mode of AEAD has $r = \tau = 64$, $b = 192$. Using the NIST prescribed values of σ and q_p we have $\sigma < 2^r$ but $2^r = 2^\tau \leq q_p$ and $2^b \leq b^2q_p^2$.

Hence Corollary 2 follows from theorem 3, proposition 1 and proposition 2.

Further using the relation that $\sigma \leq q_p$ (as per NIST LwC requirements) we can bound the advantage as,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC}}(\mathscr{A}) \leq \frac{q_p}{2^\kappa} + \mathcal{O}\left(\frac{\sigma}{2^\tau}\right) + \mathcal{O}\left(\frac{q_p\sigma}{2^b}\right).$$

5.7 Interpretation of Corollary 1 and Corollary 2

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 the Beetle and SpoC modes can achieve security. For this discussion we ignore the constants appearing in bounding the advantage terms. we take $2^r \leq \sigma \leq q_p \leq 2^c$. We further assume that $\sigma \leq 2^\tau \leq q_p$ and $2^b \leq b^2 q_p^2$.

Then by applying proposition 1 and proposition 2 in Corollary 1 or Corollary 2 we have,

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC/Beetle}}(\mathscr{A}) \leq \frac{q_p}{2^\kappa} + \mathcal{O}\left(\frac{\sigma}{2^\tau}\right) + \mathcal{O}\left(\frac{\sigma q_p}{2^b}\right).$$

Hence ignoring the constant we conclude that, in case of Beetle and SpoC, with rate $r = 32$ and permutation size $b = 160$, we achieve security almost close to the NIST LwC requirements.

5.8 Security of Sponge

In case of the original Sponge 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 Theorem 2, we have a bound for μ_{σ_d, q_p} in the cases where L_d is invertible or more specifically in the cases where Observation 1 holds. Hence the results of Theorem 2 can not be applied in case of original Sponge. However Since L_e is invertible, with a similar analysis as in the case of TtP we get,

$$\begin{aligned}\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{Sponge}}(\mathscr{A}) \leq & \frac{\sigma_d \cdot \mu_{\sigma_d, q_p}}{2^c} + \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{2\sigma_d(\sigma + q_p)}{2^b} + \frac{8\sigma_e q_p}{2^b} + \frac{3q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c} \\ & + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^\tau)}{2^{b-\tau}} + \frac{\sigma_e + q_p}{2^b} + \frac{q_p \sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}. \end{aligned} \tag{5}$$

Bounding μ_{σ_d, q_p} in case of Sponge is an interesting problem which is open to further research. However, it seems very hard to have a tight estimate of μ_{σ_d, q_p} for duplex case. A straightforward estimate of μ_{σ_d, q_p} leads to the known bound of $\sigma_d q_p / 2^c$. So as of now the tight security bound of Sponge duplex 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_{σ_d, 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 duplex.

6 Proof of Theorem 3

The proof employs coefficient H-technique of Theorem 1. To apply this method we need to first describe the ideal world which basically tries to simulate the construction. 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.[2] 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 $(\mathsf{M}_{i,1}, \ldots, \mathsf{M}_{i,m_i}) \overset{r}{\leftarrow} \mathsf{M}_i$, $\mathsf{C}_{i,j} = \lceil \mathsf{Y}_{i,a_i+j-1} \rceil_{|\mathsf{M}_{i,j}|} \oplus \mathsf{M}_{i,j}$ 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

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

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, $\mathsf{domain}(\omega_e) \cap \mathsf{domain}(\omega_p) \neq \emptyset$)

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

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

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. Now, if $p_i' = t_i$ i.e. if the complete decryption algorithm computation for the query is determined by the $\omega_e \cup \omega_p$ transcript then 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. We also define some more auxiliary bad events. 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

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

By lemma 2, $p_i < t_i$, $p_i < t_{i'}$. 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 4 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 decryption 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

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

Lemma 3.

$$\Pr[\mathtt{mBAD}] \le \frac{\sigma_d \cdot \mu_{\sigma_d,q_p}}{2^c}$$

Lemma 4.

$$\Pr\left[\bigcup_{i=1}^{6} \mathtt{Bi}\right] \le \frac{q_p}{2^\kappa} + \frac{8\sigma_e q_p}{2^b} + \frac{2\sigma_e^2}{2^b} + \frac{3q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^{b-\tau}} + \frac{\sigma_e + q_p}{2^b} + \frac{q_p \sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.$$

We postpone the proof of lemma 3 and 4 to subsection 6.5.

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

$$\begin{aligned}\Pr[\Theta_1 = \omega] &= \Pr[\Theta_1 = (\omega_e, \omega_p, \omega_d)] \\ &= \Pr[\omega_e, \omega_p] \cdot \Pr[\omega_d \mid \omega_e, \omega_p] \\ &= \Pr[\omega_e, \omega_p] \cdot (1 - \Pr[\neg\omega_d \mid \omega_e, \omega_p]) \\ &\le \Pr[\omega_e, \omega_p] \cdot \left(1 - \sum_{i \in \mathcal{D}_2} \Pr[\neg\omega_{d,i} \mid \omega_e, \omega_p]\right) \qquad (7)\end{aligned}$$

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. $\Pr(\mathsf{X}_{i,j}^*$ 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. $\Pr(\neg\omega_{d,i} \mid \mathsf{X}_{i,j}^*$ 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.

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

Theorem 3 follows from H-technique theorem 1 combined with theorem 2, lemma 3, lemma 4 and Eq. (7). □

Remark 3. 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 3 (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 multichain, 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,

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

Hence,

$$\Pr[\mathtt{mBAD}] \leq \sum_{i \in \mathcal{D}} \frac{\mathsf{Ex}[\mathsf{W}_{m_i}]}{2^c} \leq \max_{k \leq d} \mathsf{Ex}\left[\frac{\mathsf{W}_k}{k}\right] \times \frac{\sigma_d}{2^c} \leq \frac{\sigma_d \cdot \mu_{\sigma_d, q_p}}{2^c}.$$

6.5 Proof of Lemma 4 (Bad Transcript Analysis)

From the union bound we have

$$\Pr\left[\bigcup_{i=1}^{8} \mathtt{Bi}\right] \leq \Pr[\mathtt{B1}] + \Pr[\mathtt{B2}] + \Pr[\mathtt{B3}|\neg\mathtt{B1}] + \Pr[\mathtt{B4}] + \Pr[\mathtt{B5}]$$
$$+ \Pr[\mathtt{B6}|\neg\mathtt{B1}] + \Pr[\mathtt{B7}|\neg\mathtt{B1}] + \Pr[\mathtt{B8}].$$

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

Bounding $\Pr[\mathtt{B1}]$: 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

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

Bounding $\Pr[\mathtt{B2}]$: 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: This case can be bounded by probability at most $1/2^b$. We have at most σ_e many (i, j) pairs and q_p many a indices. Since L_e has rank b', this this 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[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 $\Pr[\mathtt{B3} \wedge \neg\mathtt{B1}]$: 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[B4] and Pr[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 $p_i = p'_i = a_i$: Similar to $\texttt{B3}|\neg\texttt{B1}$, Pr [Case 2] can be bounded by at most $\frac{2\sigma_e q_p}{2^b} + \frac{q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c}$.

Case 3 $a_i \leq p_i < p'_i$: This corresponds to the case when the first non-trivial decryption query block matches with a primitive query, and follows a partial chain and then matches with some encryption query block. Doing similar analysis as in Case 3 of $\texttt{B3}|\neg\texttt{B1}$, 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:

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

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

By adding all these probabilities we prove our result.

7 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 $\frac{\sigma_d \mu_{\sigma_d, q_p}}{2^c}$ 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 $q_p/2^c$ if we make primitive queries after making all encryption queries.

4. A similar attack strategy can be adapted to achieve $q_p/2^{b-\tau}$ 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 3 and equation (5) are tight.

8 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 duplex. We are able to provide tight estimation of μ 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 7, we obtain tight expression for AE advantage for Sponge duplex, the variable $q_d\mu_{q_p,m^*}/2^c$ (present in our upper bound assuming that all decryption queries are of length m^*) needs to be tightly estimated.

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.

[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. *Journal of 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.