Security Proof of Beetle and SpoC

Bishwajit Chakraborty and Ashwin Jha and Mridul Nandi

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

Abstract. In this paper, we generalize the duplexing interface of the duplex mode, which encompasses Beetle as well as a new sponge-like mode SpoC (round 1 submission to NIST LwC). We call this wide class of constructions, Transform-then-Permute. We show a tight security bound for Transform-then-Permute, which reduces to finding an exact estimation of the expected number of multi-chains (introduced in this paper). Further, we give an exact estimation of the expected number of multi-chains in case of Beetle and SpoC, which leads to an improved security bound. For both the constructions, we show that the dominating term is $rT/2^c + DT/2^{r+c}$, where D and T denote the data and time complexity, respectively, and r, c, and b denote the rate and capacity bits, respectively. In the context of NIST LwC requirement, SpoC based on 192-bit permutation achieves the desired security with as large as 64-bit rate, which is not achieved by either duplex or Beetle (as per the state-of-the-art bounds). As an extreme choice, our results imply that Beetle or SpoC mode with 160-bit permutation and 32-bit rate can also satisfy the NIST LwC requirements.

Keywords: Sponge, duplex, Beetle, SpoC, lightweight, AE, tight bound

1 Introduction

Sponge based authenticated encryption is mostly done via the duplex construction [3]. 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.

One of the dominating terms present in all of the existing analysis of duplex authenticated encryption is

$$DT/2^c.$$

A recent variant of duplex mode, called the Beetle mode of operation [4], modifies the duplexing phase by introducing a combined feedback based absorption/squeezing, similar to the feedback paradigm of CoFB [5]. In [4], 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 $\tau = r$. This means that for $c = r = b/2$, the beetle mode achieves close to $(c - \log_2 r)$-bit security.

Security of Sponge-based 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 based submission to NIST LwC standardization process uses 192-bit capacity, except CLX 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 close to 120-bit capacity and 120-bit rate to achieve NIST LwC requirements. In light of the ongoing NIST LwC standardization, it would be interesting to see whether we can get rid of the limitations in Beetle.

1.1 Our Contributions

In this paper, inspired by the NIST LwC requirements, we extend a long line of research on the security of Sponge-based AE schemes. Our contributions are threefold.

- First, we study Sponge-based AE 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 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.
- Second, we show that for a class of feedback function, containing the Beetle and SpoC modes, optimal AE security is achieved.
- Third, as a byproduct of the previous point, we give improved and tight bound for Beetle, and a security proof validating the security claims of SpoC [1]. Notably, we show that both Beetle and SpoC achieve NIST LwC requirements with just 128-bit capacity and $\geq$ 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.

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. $\lfloor x \rfloor_n$) denote 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 $(n)_k := n!/(n-k)! = n(n-1)\cdots(n-k+1)$. Note that $(n)_k \leq n^k$.

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 and random sampling of X from $\mathcal{X}$.

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:

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

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 $\mathcal{M} \times \mathcal{T}$, and $\Gamma \leftarrow_{\$} \mathsf{Func}$. Let $\perp$ denote the degenerate function from $(\mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T})$ to $\{\perp\}$. 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 2.1. *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 an adversary $\mathscr{A}$ against AE_Π is defined as,*

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

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, $\mathscr{A}^{\Gamma, \perp, \Pi^\pm}$ denotes $\mathscr{A}$'s response after its interaction with Γ, $\perp$, and $\Pi^\pm$.

In this paper, we 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 parametrize 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 sum of input (associated data and message) lengths across all encryption and decryption, respectively, queries. We sometime also write $q_c = q_e + q_d$ and $\sigma_c = \sigma_e + \sigma_d$ to denote the combined construction query resources.
- q_+ and q_- denote the number of queries to Π^+ and Π^-, respectively. We sometime also use $q_p = q_+ + q_-$, to denote the combined primitive query resources.

We remark here that q_c and σ_c 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 [7], known as the H-coefficient technique (or simply the H-technique).

Theorem 2.1 (H-coefficient technique [8]). *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:*
- $\Pr[\Theta_0 \in \Omega_{\mathsf{bad}}] \leq \epsilon_{\mathsf{bad}}$;
- *For any $\omega \notin \Omega_{\mathsf{bad}}$,*
$$\frac{\Pr[\Theta_1 = \omega]}{\Pr[\Theta_0 = \omega]} \geq 1 - \epsilon_{\mathsf{ratio}}.$$

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

A proof of this theorem is available in multiple papers including [8].

3 Transform-then-Permute Construction

In this section we describe Transform-then-Permute (or TtP in short), which generalizes dupleing method used in sponge AEAD. We first start of with a simple version of Transform-then-Permute. In particular, we make the following set of assumptions:
1. There is no associated data (so now it is AE rather than AEAD). Moreover, the message space $\mathcal{M} \subset (\{0,1\}^r)^+$ where r is called rate of Transform-then-Permute.
2. The key size κ is strictly less than b (input size of the underlying ideal permutation Π) and the nonce size is $b - \kappa$. We denote the tag size as τ which is also less than the block size b.

We make these assumptions to describe the general idea of our proof approach in a more simplified way. Towards the end of the section, we relax these assumptions to come up with a fairly generalized sponge construction.

3.1 Description of Basic Structure of Transform-then-Permute

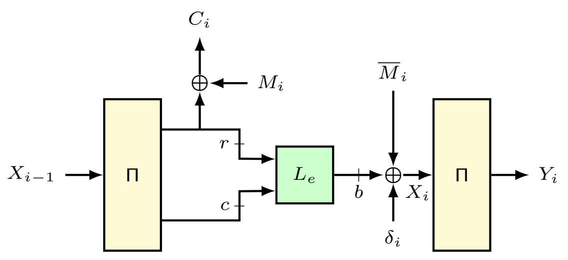

Fig. 3.1: Illustration of the feedback process for the i-th block of an m-block message $M = (M_1, \dots, M_m)$, where each $M_i \in \{0,1\}^r$. Here $\delta_i = 1$ if $i = m$, and 0 otherwise. Note that X_0 is defined as $N\|K$. See algorithm 3.1 for further details.

We define the Transform-then-Permute construction on top of the ideal permutation $\Pi \leftarrow_{\$} \mathsf{Perm}$. To encrypt a message, we require two linear functions $L_e : \{0,1\}^b \to \{0,1\}^b$ (an invertible linear function) and $\mathsf{encode} : \{0,1\}^r \to \{0,1\}^b$. For simplicity we write $\overline{x}$ to denote $\mathsf{encode}(x)$. Mostly we choose zero padding as the encoding, i.e. $\overline{x} := x\|0^c$ where $c := b-r$ (called capacity of the construction).

The main steps are summarized in algorithm 3.1, and the i-th message block processing is illustrated in figure 3.1.

Algorithm 3.1 Encryption/Decryption algorithms for the Transform-then-Permute mode

```
1:  function Enc($N, K, M$)
2:      $(M_1, \dots, M_m) \xleftarrow{r} M$
3:      $X_0 \leftarrow N\|K$, $Y_0 \leftarrow \Pi(X_0)$
4:      for $i = 1$ to $m$ do
5:          $C_i \leftarrow M_i \oplus \lceil Y_{i-1} \rceil_r$
6:          if $i < m$ then
7:              $X_i = L_e(Y_{i-1}) \oplus \overline{M_i}$
8:          else
9:              $X_i = L_e(Y_{i-1}) \oplus \overline{M_i} \oplus 1$
10:         $Y_i \leftarrow \Pi(X_i)$
11:     $T \leftarrow \lceil Y_m \rceil_\tau$.
12:     return $(C := C_1\| \dots \|C_m, T)$
```

```
1:  function Dec($N, K, C, T$)
2:      $(C_1, \dots, C_m) \xleftarrow{r} C$
3:      $X_0 \leftarrow N\|K$, $Y_0 \leftarrow \Pi(X_0)$
4:      for $i = 1$ to $m$ do
5:          $M_i \leftarrow C_i \oplus \lceil Y_{i-1} \rceil_r$
6:          if $i < m$ then
7:              $X_i = L_d(Y_{i-1}) \oplus \overline{C_i}$
8:          else
9:              $X_i = L_d(Y_{i-1}) \oplus \overline{C_i} \oplus 1$
10:         $Y_i \leftarrow \Pi(X_i)$
11:     $T' \leftarrow \lceil Y_m \rceil_\tau$
12:     if $T' \neq T$ then
13:         return $M := \perp$
14:     else
15:         return $M := M_1\| \dots \|M_m$
```

For decryption, we use $L_d(x)$ to denote the linear function $L_e(x) \oplus \overline{\lceil x \rceil_r}$. It is easy to see why the decryption works correctly. During decryption X_i, for $i < m$, should be defined as $L_e(Y_{i-1}) \oplus \overline{M_i}$ where $M_i = \lceil Y_{i-1} \rceil_r \oplus C_i$. Hence, $X_i = L_e(Y_{i-1}) \oplus \overline{\lceil Y_{i-1} \rceil_r} \oplus \overline{C_i}$.

3.2 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 defined in some manner. We first need to describe some notations which would be required to define the security game.
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 $\mathcal{G}_{\mathcal{L}}$ 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.

The Multi-Chain Structure

Definition 3.1. *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 $1 \leq i \leq p$, $\mathcal{W}_i : v_0^i \xrightarrow{x} v_k^i$ and $\lceil u_0^1 \rceil_r = \cdots = \lceil u_0^p \rceil_r$ and $\lceil v_k^1 \rceil_\tau = \cdots = \lceil v_k^p \rceil_\tau$. We also call the multi-chain of length k.*

The maximum size of the set of multi-chain of length k (with some label x) is denoted as W_k (which is induced by $\mathcal{L}$).

Now consider an adversary $\mathscr{A}$ interacting at most t times 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. We write

$$\mu_{k,\mathscr{A}} := \mathsf{Ex}\,[\mathsf{W}_k].$$

Here W_k is defined for the labeled graph induced by the list θ' as defined above and expectation is defined over the randomness of the random permutation $\mathsf{\Pi}$ and the random coin of the adversary $\mathscr{A}$. Finally, we define $\mu_{k,t} = \max_{\mathscr{A}} \mu_{k,\mathscr{A}}$ where maximum is taken over all adversaries making at most t queries.

3.3 Security Analysis of Transform-then-Permute

We prove the following result on the AE security of Transform-then-Permute.

Theorem 3.1. *Let us assume that $\sigma := \sigma_e + \sigma_d \le q_p$. For any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, we have*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{TtP}}(\mathscr{A}) \le \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{5\sigma q_p}{2^b} + \frac{3q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c} + \sum_{i \in \mathcal{D}} \frac{\mu_{m_i^*, q_p}}{2^c} + \frac{q_p \sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}$$

where $\mathcal{D}$ denote the set of query indices for decryption queries.

By applying suitable bound for $\mathsf{mcoll}(\sigma_e, 2^r)$ we have the following bounds. When $r \ge 16$ and $\sigma_e \ge r2^r$ (which is reasonable when r is small like 16), we can simplify the above bound as

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{TtP}}(\mathscr{A}) \le \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{8\sigma q_p}{2^b} + \sum_{i \in \mathcal{D}} \frac{\mu_{q_p, m_i^*}}{2^c} + \frac{q_p \sigma^2}{2^{b+c}}.$$

In all other cases, we have

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{TtP}}(\mathscr{A}) \le \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{5\sigma q_p}{2^b} + \frac{r q_p}{2^c} + \sum_{i \in \mathcal{D}} \frac{\mu_{q_p, m_i^*}}{2^c} + \frac{r q_p \sigma}{2^{2c}}.$$

Proof. The proof employs H-technique tool of Theorem 2.1. 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).

Ideal World: The ideal world responds three oracles, namely encryption queries, decryption queries and primitive queries.

Primitive: The ideal world simulates $\mathsf{\Pi}^{\pm}$ query honestly and maintains a list ω_p of the query-responses of $\mathsf{\Pi}$ as a partial injective function. More precisely,

$$\omega_p = ((\mathsf{U}_1, \mathsf{V}_1, \mathsf{dir}_1), (\mathsf{U}_2, \mathsf{V}_2, \mathsf{dir}_2), \ldots)$$

where dir_i is either 1 or -1 depending on forward or backward queries. ω_p represents forward only transcript and so $\mathsf{\Pi}(\mathsf{U}_i) = \mathsf{V}_i$ for all i. $\omega_p' := ((\mathsf{U}_1, \mathsf{V}_1), (\mathsf{U}_2, \mathsf{V}_2), \ldots)$ represents the transcript after removing the direction information.

Encryption: When the ith query is an encryption query $(\mathsf{N}_i, \mathsf{M}_i)$ with $(\mathsf{M}_{i,1}, \ldots, \mathsf{M}_{i,m_i}) \overset{r}{\leftarrow} \mathsf{M}_i$, it first samples $\mathsf{Y}_{i,0}, \ldots, \mathsf{Y}_{i,m_i} \leftarrow_{\$} \{0,1\}^b$ and then returns

$$\mathsf{C}_{i,j} = \lceil \mathsf{Y}_{i,j-1} \rceil_r \oplus \mathsf{M}_{i,j}, \ j \in [m_i], \quad \mathsf{T}_i \leftarrow \lceil \mathsf{Y}_{i,m_i} \rceil_\tau.$$

For all i, we define intermediate inputs (X-values) as follows

$$\mathsf{X}_{i,j} = \begin{cases} \mathsf{N}_i \| K & \textit{if } j = 0 \\ L_e(\mathsf{Y}_{i,j-1}) \oplus \overline{\mathsf{M}}_{i,j} & \textit{if } 1 \leq j < m_i \\ L_e(\mathsf{Y}_{i,j-1}) \oplus \overline{\mathsf{M}}_{i,j} \oplus 1 & \textit{if } j = m_i. \end{cases}$$

<u>DECRYPTION:</u> When the ith query is a decryption query $(\mathsf{N}_i^*, \mathsf{C}_i^*, \mathsf{T}_i^*)$, it always returns abort symbol $\mathsf{M}_i^* := \perp$. We write $(\mathsf{C}_{i,1}^*, \ldots, \mathsf{C}_{i,m_i^*}^*) \xleftarrow{r} \mathsf{C}_i^*$. The decryption transcript $\omega_d := (\mathsf{M}_i^*)_{i \in \mathcal{D}}$ where M_i^* is always $\perp$ in ideal world.

<u>OFFLINE:</u> Let $\mathcal{E}$ and $\mathcal{D}$ denote the set of all query indices corresponding to encryption and decryption queries respectively.

After completion of oracle interaction (the above three types of queries possibly in an interleaved manner), the ideal oracle returns all X-values (as computed above) and Y-values. So we define the encryption transcript $\omega_e = (\mathsf{X}_{i,j}\mathsf{Y}_{i,j})_{i \in \mathcal{E}, j \in (m_i]}$. So, the transcript of the adversary consists of $\omega := (\omega_p, \omega_e, \omega_d)$.

INTERNAL VALUES FOR DECRYPTION QUERIES: For each decryption query $(\mathsf{N}_i^*, \mathsf{C}_i^*, \mathsf{T}_i^*)$, (i.e. $i \in \mathcal{D}$), we define p_i as -1 if for all $j \in \mathcal{E}$, $\mathsf{N}_j \neq \mathsf{N}_i^*$. In other words, the i-th decryption query has been queried with a *fresh* nonce. 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 ℓ_i denote the length of the longest common prefix of $\mathsf{C}_{i'}$ and C_i^*. We define p_i as ℓ_i neither C_i^* not $\mathsf{C}_{i'}$ is a prefix to other. Otherwise, we define p_i as $\ell_i - 1$. For every $i \in \mathcal{D}$ with $p_i \geq 0$, we define

1. $\mathsf{Y}_{i,0..p_i}^* = \mathsf{Y}_{i',0..p_i}$,
2. $\mathsf{X}_{i,0..p_i}^* = \mathsf{X}_{i',0..p_i}$, and
3. $\mathsf{X}_{i,p_i+1}^* = L_d(\mathsf{Y}_{i',p_i}) \oplus \overline{\mathsf{C}^*}_{p_i+1} \oplus \chi(p_i + 1 =_? m_i)$.

By definition of longest common-prefix, we have $\mathsf{X}_{i,p_i+1}^* \neq \mathsf{X}_{i',p_i+1}$. Now, we further extend X^*-values and Y^*-values whenever possible based on the primitive transcript ω_p. Fix $i \in \mathcal{D}$ with $p_i \geq 0$. For the notational simplicity, let $x_{i,j}$ denote $\overline{\mathsf{C}^*}_{i,j} \oplus \chi(j =_? m_i)$ where $\chi(P)$ is 1 if the statement P is true, otherwise it sets zero. If there is a labeled walk (in the labeled directed graph induced by ω_p as described in section 3.2) from Y_{i,p_i}^* with label $(x_{i,p_i+1}, \ldots, x_{i,j})$ then we denote the end node as $\mathsf{Y}_{i,j}^*$. In notation we have

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

For each $i \in \mathcal{D}$ let p_i' denote the muximum possible value of j such that $\mathsf{Y}_{i,j}^*$ has been defined as described above. We define $\mathsf{X}_{i,j+1}^* := L_d(\mathsf{Y}_{i,j}^*) \oplus x_{j+1}$ for all $i \in \mathcal{D}$ and $p_i + 1 \leq j \leq p_i'$.

BAD TRANSCRIPT : We say that an ideal world transcript $\omega = (\omega_p, \omega_e, \omega_d)$ is bad if any one of the following conditions holds:

Bad events due to encryption and primitive transcript:

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

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

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

B4: For some $(i \in \mathcal{E}, j \in (m_i]) \neq (i' \in \mathcal{E}, j' \in (m_{i'}])$, $\mathsf{Y}_{i,j} = \mathsf{Y}_{i',j'}$,
B5: For some $(i \in \mathcal{E}, j \in (m_i]) \neq (i' \in \mathcal{E}, j' \in (m_{i'}])$, $\mathsf{X}_{i,j} = \mathsf{X}_{i',j'}$,

Bad events due to decryption transcript:
B6: For some $i \in \mathcal{D}$, $(i' \in \mathcal{E}, j' \in (m_{i'}])$, $\mathsf{X}^*_{i,p_i+1} = \mathsf{X}_{i',j'}$,
B7: For some $i \in \mathcal{D}$ with $p_i \geq 0$, $p'_i = m_i$ and $\lceil \mathsf{Y}^*_{i,m_i} \rceil_\tau = \mathsf{T}^*_i$,
B8: For some $i \in \mathcal{D}$ with $p_i \geq 0$ and for some j, $\mathsf{Y}^*_{i,j} \in \mathsf{range}(\omega_e)$ or $\mathsf{X}^*_{i,j} \in \mathsf{domain}(\omega_e)$.

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

$$\mathtt{BAD} = \bigcup_{i=1}^{8} \mathtt{Bi}. \tag{2}$$

Lemma 3.1.

$$\Pr[\mathtt{BAD}] \leq \frac{q_p}{2^\kappa} + \frac{5\sigma_e q_p}{2^b} + \frac{3q_p \mathsf{mcoll}(\sigma_e, 2^r)}{2^c} + \sum_{i \in \mathcal{D}} \frac{\mu_{m^*_i, q_p}}{2^c} + \frac{q_p \sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.$$

We postpone the proof of lemma 3.1, i.e. the upper bound on the probability of realizing a bad transcript in the ideal world, to Appendix C.

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

The motivation for all the bad events would be clear from the understanding of a good transcript (i.e., not a bad transcript). Let $\omega = (\omega_p, \omega_e, \omega_d)$ be a good transcript. Suppose for all $1 \leq j \leq p'_i$, $\mathsf{Y}^*_{i,j}$ (provided $p_i \geq 0$) and $\mathsf{X}^*_{i,j+1}$ has been defined from the transcript as described above. Then, we have the following observations:

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. For all $i \in \mathcal{D}$, either $p'_i = m_i$ with $\lceil \mathsf{Y}^*_{i,m_i} \rceil_\tau \neq \mathsf{T}^*_i$ (type-1 decryption query) or $p'_i < m_i$ but $\mathsf{X}^*_{i,p'_i+1} \notin \mathsf{domain}(\omega_e \cup \omega_p)$ (type-2 decryption query). Type-1 decryption queries would be straightaway rejected. Type-2 decryption query can be computed based on $\omega_e \cup \omega_p$ until X^*_{i,p'_i+1}, which is fresh. So $\Pi(\mathsf{X}^*_{i,p'_i+1})$ is random over a large set. This would ensure with high probability we reject those decryption query also.

These two information would be used for good transcript analysis.

Good Transcript Analysis: Now fix a good transcript ω. 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 $\sigma_e = q_e + \sum_i m_i$. In the real world, for ω we denote the encryption query, decryption query, and primitive query tuples by ω_e, ω_d and ω_p, respectively. Then, we have

$$\begin{aligned}
\Pr[\Theta_1 = \omega] &= \Pr[\Theta_1 = (\omega_e, \omega_p, \omega_d)] \\
&= \Pr[\omega_e, \omega_p] \cdot \Pr[\omega_d \mid \omega_e, \omega_p] \\
&= \Pr[\omega_e, \omega_p] \cdot (1 - \Pr[\neg\omega_d \mid \omega_e, \omega_p]) \\
&\leq \Pr[\omega_e, \omega_p] \cdot \left(1 - \sum_{i \in \mathcal{D}} \Pr[\neg\omega_{d,i} \mid \omega_e, \omega_p]\right) \qquad (3)
\end{aligned}$$

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 $\cup_{i\in\mathcal{D}}\neg\omega_{d,i}$ (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 $\Pr_{\Theta_1}(\neg\omega_{d,i} \mid \omega_e, \omega_p) \leq \frac{m_i(\sigma_e+q_p)}{2^b-\sigma_e-q_p} + \frac{1}{2^\tau}$ for every type-2 decryption query. Recall that X^*_{i,p'_i+1} is fresh. If $\mathsf{X}^*_{i,j}$ is the last input block then $\Pi(\mathsf{X}^*_{i,j}) = \mathsf{T}^*_i$ 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). Suppose $\mathsf{X}^*_{i,j}$ is not the last block, then the next input block may collide with some encryption or primitive input block with probability at most $\frac{\sigma_e+q_p}{2^b}$. Applying this same argument for all the successive blocks till the last one, we get the probability at most $\frac{m_i(\sigma_e+q_p)}{2^b-\sigma_e-q_p}$, the last block input would be fresh. Hence the probability that the tag matches with at most $2/2^\tau$. Now, by union bound we have

$$\begin{aligned}
\Pr[\neg\omega_d \mid \omega_e, \omega_p] &\leq \sum_{i\in\mathcal{D}} \frac{m_i(\sigma_e+q_p)}{2^b-\sigma_e-q_p} + \frac{2}{2^\tau} \\
&\leq \frac{2\sigma_d(\sigma_e+q_p)}{2^b} + \frac{2q_d}{2^\tau} \\
&\leq \frac{4\sigma_d q_p}{2^b} + \frac{2q_d}{2^\tau}.
\end{aligned}$$

The result follows from H-technique theorem 2.1, combined with lemma 3.1 and Eq. (3). □

3.4 Extending the Analysis to Transform-then-Permute AEAD

Due to constrain of space we move this section to Appendix D.

4 Transform-then-Permute with Invertible L_d Function

In this section, we give concrete bounds for Transform-then-Permute under a special assumption that the underlying feedback function is invertible.

Theorem 4.1. *If the feedback function L is invertible, then we have*

$$\mu_{t,k} \leq \mathsf{mcoll}(t, 2^{\tau}) + \mathsf{mcoll}(t, 2^{r}) + k \cdot \mathsf{mcoll}'(t^2, 2^{b}).$$

Proof. Due to constrain of space the proof is moved to Appendix E.

It is easy to see that the decryption feedback function in Beetle [4] and SpoC [1] are invertible, whereas this is not the case for duplex. In the following subsections, we apply Eq. (9) and theorem 4.1 to give improved bounds for Beetle and SpoC.

4.1 Security of Beetle

In Beetle [4], the linear function L_e is defined as $L_e(x, y) \mapsto (x_2, x_2 \oplus x_1, y)$, where $(x_1, x_2, y) \in \{0,1\}^{r/2} \times \{0,1\}^{r/2} \times \{0,1\}^{c}$. The linear function L_d is defined by the mapping $L_d(x, y) \mapsto (x_2 \oplus x_1, x_1, y)$, where $(x_1, x_2, y) \in \{0,1\}^{r/2} \times \{0,1\}^{r/2} \times \{0,1\}^{c}$. Clearly the L_e and L_d functions are invertible. Further, they have full rank. Thus, from Eq. (9), theorem 4.1, Eq. (5), and Eq. (8), we have

Corollary 4.1. *For $r, \tau, b \geq 16$ and any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, we have*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{Beetle}}(\mathscr{A}) \leq \frac{q_p}{2^{\kappa}} + \frac{2q_d}{2^{\tau}} + \frac{5\sigma q_p}{2^{b}} + \frac{rq_p}{2^{c}} + \frac{rq_p\sigma}{2^{2c}} + \frac{rq_pq_d}{2^{b}} + \frac{\tau q_pq_d}{2^{c+\tau}} + \frac{2bq_p^2\sigma_d}{2^{b+c}}.$$

Further by assuming $r \leq \tau$, we get

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{Beetle}}(\mathscr{A}) \leq \frac{q_p}{2^{\kappa}} + \frac{2q_d}{2^{\tau}} + \frac{5\sigma q_p}{2^{b}} + \frac{rq_p}{2^{c}} + \frac{rq_p\sigma}{2^{2c}} + \frac{2\tau q_pq_d}{2^{b}} + \frac{2bq_p^2\sigma_d}{2^{b+c}}.$$

4.2 Security of SpoC

In SpoC [1], the linear function L_e is identity, and the linear function L_d is defined by the mapping $L(x, y) \mapsto (x, x\|0^c \oplus y)$, 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. (9), theorem 4.1, Eq. (5), and Eq. (8), we have

Corollary 4.2. *For $r, \tau, b \geq 16$ and any $(q_p, q_e, q_d, \sigma_e, \sigma_d)$-adversary $\mathscr{A}$, we have*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC}}(\mathscr{A}) \leq \frac{q_p}{2^{\kappa}} + \frac{2q_d}{2^{\tau}} + \frac{5\sigma q_p}{2^{b}} + \frac{rq_p}{2^{c}} + \frac{rq_p\sigma}{2^{2c}} + \frac{rq_pq_d}{2^{b}} + \frac{\tau q_pq_d}{2^{c+\tau}} + \frac{2bq_p^2\sigma_d}{2^{b+c}}.$$

Further by assuming $r \leq \tau$, we get

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{SpoC}}(\mathscr{A}) \leq \frac{q_p}{2^{\kappa}} + \frac{2q_d}{2^{\tau}} + \frac{5\sigma q_p}{2^{b}} + \frac{rq_p}{2^{c}} + \frac{rq_p\sigma}{2^{2c}} + \frac{2\tau q_pq_d}{2^{b}} + \frac{2bq_p^2\sigma_d}{2^{b+c}}.$$

Note that in the above corollaries we have utilized the crude bounds of Eq. (5) and (8) on $\mathsf{mcoll}(q_p, 2^{\tau})$, $\mathsf{mcoll}(q_p, 2^{r})$ and $\mathsf{mcoll}'(q_p^2, 2^{b})$. One can get an even tighter estimates depending upon the relationship between q_p and τ, r, and b.

5 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. Although we obtain tight expression for AE advantage, 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. We are able to provide tight estimation of μ when the feedback function for decryption is linear. This is the case for Beetle and SpoC, but not for Sponge duplex.

References

1. Riham AlTawy, Guang Gong, Morgan He, Ashwin Jha, Kalikinkar Mandal, Mridul Nandi, and Raghvendra Rohit. Spoc. Submission to NIST LwC Standardization Process (Round 1), 2019.
2. Ricardo A. Baeza-Yates and Gaston H. Gonnet. *Handbook of Algorithms and Data Structures in Pascal and C.* Addison-Wesley, 1991.
3. 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.
4. 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.
5. 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.
6. Gaston H. Gonnet. Expected length of the longest probe sequence in hash code searching. *J. ACM*, 28(2):289–304, 1981.
7. Jacques Patarin. *Etude des Générateurs de Permutations Pseudo-aléatoires Basés sur le Schéma du DES.* PhD thesis, Université de Paris, 1991.
8. Jacques Patarin. The "coefficients H" technique. In *Selected Areas in Cryptography - SAC 2008. Revised Selected Papers*, pages 328–345, 2008.
9. 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.
10. Robert Sedgewick and Philippe Flajolet. *An introduction to the analysis of algorithms.* Addison-Wesley-Longman, 1996.

A 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 [6,2,10,9]) in context of the complexity of hash table[1] probing.

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

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 as before. Moreover, we also extend these results for samples which, although are not uniform, have high entropy almost close to uniform.

A.1 Expected Maximum Multicollision in a Uniform Random Sample

Let $\mathsf{X}_1, \ldots, \mathsf{X}_q \leftarrow\!\!\$\, \mathcal{D}$ where $|\mathcal{D}| = N$. For notational simplicity, we write $\log_2 N$ as n. We denote the maximum multicollision random variable for the sample as $\mathsf{mc}_{q,N}$. More precisely, $\mathsf{mc}_{q,N} = \max_a |\{i : \mathsf{X}_i = a\}|$. 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 $e^\rho \geq \rho^\rho/\rho!$. Thus, we have

$$\Pr[\mathsf{mc}_{q,N} \geq \rho] \leq N \cdot \left(\frac{qe}{\rho N}\right)^\rho. \tag{4}$$

For any positive integer valued random variable Y bounded by N:

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

Using Eq. (4), 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 $\mathsf{Ex}\left[\mathsf{mc}_{q,N}\right]$.

Proposition A.1. $\mathsf{mcoll}(q, N) \leq \begin{cases} \frac{4n}{\log n} & \textit{if } q = N, n \geq 16 \\ 4n & \textit{if } q = nN \\ 4n\lceil \frac{q}{nN} \rceil & \textit{if } q \geq nN \\ 4 \log q & \textit{if } q < N \end{cases}$

Proof. When $q = N$ we have, $\Pr[\mathsf{mc}_{N,N} \geq \rho] \leq N \cdot \left(\frac{e}{\rho}\right)^\rho$. So, we can write $\mathsf{Ex}\left[\mathsf{mc}_{N,N}\right] \leq \rho - 1 + N^2 \cdot \left(\frac{e}{\rho}\right)^\rho$. Let us take $\rho = 4n/\log n$. It can be easily shown after simplification that $\left(\frac{e}{\rho}\right)^{\rho} \leq N^{-2}$ (take logarithm on both sides and simplify) assuming $n \geq 16$. Thus, for all $n \geq 16$ we have

$$\mathsf{Ex}\left[\mathsf{mc}_{N,N}\right] \leq 4n/\log n.$$

Similarly, for $q = Nn$, we choose $\rho = 4n$. After simplification, we once again have $\mathsf{Ex}\left[\mathsf{mc}_{nN,N}\right] \leq 4n - 1 + N^2 \times (\frac{e}{4n})^{4n}$. Note that $(4/e)^4 > 4$ and hence for all positive n, $N^2 \times (\frac{e}{4n})^{4n} < 1$.

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 $q \geq nN$.

Finally, when $q < N$, we can simply bound $\mathsf{Ex}\left[\mathsf{mc}_{q,N}\right] \leq 4\log q$. □

When $n \geq 16$, for all q, we can write the bounds into one single form:

$$\mathsf{mcoll}(q, N) \leq nq/N \tag{5}$$

B Expected Maximum Multicollision in a Non-uniform Random Sample

Now we bound expectation of maximum multicollision in a sample $\mathsf{X}_1, \ldots, \mathsf{X}_q$ (can be arbitrarily dependent) which is not completely uniform random. However, it satisfies the following property for all distinct $i_1, \ldots, i_\rho$ for any integer $\rho \geq 2$:

$$\Pr(\mathsf{X}_{i_1} = a, \cdots \mathsf{X}_{i_\rho} = a) \leq \frac{1}{N'^r} \tag{6}$$

Then, we can actually perform the same analysis as before. For any integer $\rho \geq 2$, it can be shown that

$$\Pr[\mathsf{mc}_{q,N} \geq \rho] \leq N \cdot \left(\frac{qe}{\rho N'}\right)^{\rho} \tag{7}$$

Using it, we can prove the following results for expected value of maximum multicollision.

Proposition B.1. $\mathsf{Ex}\left[\mathsf{mc}_{q,N}\right] \leq \begin{cases} 4\log q & \textit{if } q < N' \\ \frac{4n}{\log n} & \textit{if } N' \leq q < N'n \\ \frac{4q}{N'} & \textit{if } q \geq N'n \end{cases}$

In the non-random case, we denote $\mathsf{Ex}\left[\mathsf{mc}_{q,N}\right]$ by $\mathsf{mcoll}'(q, N)$ As before, when $n \geq 16$, we have

$$\mathsf{mcoll}'(q, N) \leq nq/N' \tag{8}$$

B.1 Some Examples of Non-uniform Random Samples

In this paper apart from the uniform random sample over $\{0,1\}^b$, we also consider the following non-uniform random samples:

1. We sample $\mathsf{X}_1, \ldots, \mathsf{X}_q \xleftarrow{\mathsf{wor}} \{0,1\}^b$ and then we define $\mathsf{Y}_i = \lceil \mathsf{X}_i \rceil_r$ for some $r < b$. In this case, we have $\Pr(\mathsf{Y}_{i_1} = a, \cdots \mathsf{Y}_{i_\rho} = a) \leq \frac{(2^{(b-r)})_\rho}{(2^b)_\rho}$. 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). Now, $\frac{(2^{(b-r)})_\rho}{(2^b)_\rho} \leq \frac{1}{2^{r\rho}}$. So, $\mathsf{Ex}\left[\mathsf{mc}_{q,N}\right] \leq \mathsf{mcoll}(q, N)$ (same bound as random sample), where $N = 2^r$.

2. 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, it is easy to see that for any $(i_1, j_1), \ldots (i_\rho, j_\rho)$,

$$\Pr(\mathsf{Z}_{i_1,j_1} = a, \cdots \mathsf{Z}_{i_\rho,j_\rho} = a) \leq \frac{1}{(N-2\rho)^\rho}.$$

   So Proposition B.1 can be applied with $N' = N - 2\rho$.

C Proof of Lemma 3.1 (Bad Transcript Analysis)

Proof. From equation 2, we have

$$\Pr[\mathtt{BAD}] = \Pr\left[\bigcup_{i=1}^{8} \mathtt{Bi}\right] \leq \sum_{i=1}^{8} \Pr[\mathtt{Bi}].$$

It is sufficient to upper bound the probabilities of $\mathtt{Bi}$. 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. 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. Thus 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 $1/(2^b - a + 1)$. We have at most σ_e many (i,j) pairs and q_f many a indices. Thus this can be bounded by at most $\sigma_e q_f / (2^b - a + 1)$.

Case 3: $\exists i, j, a$, $\mathsf{Y}_{i,j} = \mathsf{V}_a$, $\mathsf{dir}_a = -$, 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 on $\lceil \mathsf{Y}_{i,j} \rceil_r$. Now, we have

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

Since the three cases are mutually exclusive, we have

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

Bounding $\Pr[\mathsf{B3}|\neg\mathsf{B1}]$: We can have the following cases:
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 has full rank. 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 $1/(2^b - a + 1)$. We have at most σ_e many (i, j) pairs and q_b many a indices. Thus this can be bounded by at most $\sigma_e q_b/(2^b - a + 1)$.
Case 3: $\exists i, j, a$, $\mathsf{X}_{i,j} = \mathsf{U}_a$, $\mathsf{dir}_a = +$, encryption before primitive: Let Φ_{in} denote the number of multicollisions on $\lceil \mathsf{X}_{i,j} \rceil_r$. With a similar analysis on the multicollision of output values, we have $\Pr[\text{Case 3}] \leq \mathsf{Ex}\left[\Phi_{in}\right] \frac{q_b}{2^c}$. Since the three cases are mutually exclusive, we have

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

Bounding $\Pr[\mathsf{B4}]$: The probability of this event can be bounded in a straightforward manner by at most $\sigma_e(\sigma_e - 1)/2^{b+1}$.
Bounding $\Pr[\mathsf{B5}]$: This event is similar to $\mathsf{B4}$, and the probability is bounded by at most $\sigma_e(\sigma_e - 1)/2^{b+1}$.
Bounding $\Pr[\mathsf{B6}]$: Similar to $\mathsf{B3}$, $\Pr[\mathsf{B6}]$ bounded by at most $\mathsf{mcoll}(\sigma_e, 2^r)q_p/2^c$.
Bounding $\Pr[\mathsf{B7}]$: Let $\mathsf{W}_k(\omega'_p)$ denote the k-length multi-chain for the graph induced by ω_p. Suppose the event holds for the ith decryption query and $\mathsf{N}^*_i = \mathsf{N}_{i'}$. So, Y_{i',p_i} must be the one of the starting node of the multi-chain. As $\lfloor \mathsf{Y}_{i',p_i} \rfloor_c$ is chosen at random (and independent of ω_p), the probability to hold $\mathsf{B7}$ for ith

decryption query is at most $\mathsf{W}_{m_i^*}/2^c$ given the transcript ω_p. So by union bound, the conditional probability $\Pr[\mathsf{B7} \mid \omega_p] \leq \sum_{i\in\mathcal{D}} \frac{\mathsf{W}_{m_i^*}}{2^c}$. Hence,

$$\Pr[\mathsf{B7}] \leq \sum_{i\in\mathcal{D}} \frac{\mu_{m_i^*,q_p}}{2^c}.$$

Bounding $\Pr[\mathsf{B8}|\neg(\mathsf{B2}\vee\mathsf{B3}\vee\mathsf{B6}\vee\mathsf{B7})]$: This conditional event corresponds to the case when the first non-trivial decryption query matches with a primitive query, and follows a partial chain and then matches with some encryption query block. The probability that this happens for ith decryption is at most $q_p/2^c \times m_i^* \Phi_{in}/2^c$. Summing over all $i \in \mathcal{D}$, the conditional probability is at most $\frac{q_p\sigma_d\Phi_{in}}{2^{2c}}$. By taking expectation we obtain the following:

$$\Pr[\mathsf{B8}] \leq \frac{q_p\sigma_d \mathsf{mcoll}(\sigma_e, 2^r)}{2^{2c}}.$$

By adding all these probabilities we prove our result. $\square$

D Extending the Analysis to Transform-then-Permute AEAD

We now describe Transform-then-Permute, which generalizes duplexing method used in sponge AEAD. The details of the algorithm is given in Algorithm D.1. The security of the general construction is almost same as the simplified version. Here we need to consider more inputs X-values and outputs Y-values while processing associated data which was not present in the previous analysis. We define same set of bad events and use the same multi-chain argument. A notable change is the introduction of $\mathsf{DS} : \mathbb{N}\times\mathbb{N} \to \left(\{0,1\}^b\right)^*$ function, which maps $(x, y) \in \mathbb{N}\times\mathbb{N}$ to a sequence of binary strings $(\delta_1, \ldots, \delta_{x+y}) \in \left(\{0,1\}^b\right)^{x+y}$. The DS function, in combination with the encode function, acts as the domain separator.

The DS and encode functions must be such that for any $(A, M) \neq (A', M')$, $(\overline{A}_1 \ldots, \overline{A}_a, \overline{M}_1, \ldots, \overline{M}_m)$ is not a prefix of $(\overline{A'}_1 \ldots, \overline{A'}_{a'}, \overline{M'}_1, \ldots, \overline{M'}_{m'})$. We note that this prefix-free property is required to bound the modified B5. We need $\mathsf{X}^*_{i,p_i+1} \neq \mathsf{X}_{i',p_i+1}$ which can be argued from the prefix-free property of the combined DS and encode function. Now with simple extensions of the notations σ_e and σ_d to cover all the data blocks, i.e. nonce, associated data and message, in encryption and decryption, respectively, we get exactly the same security bound as before. Particularly, we use the crude bound form of theorem 3.1 to get the following AEAD security bound for Transform-then-Permute with associated data

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{TtP}}(\mathscr{A}) \leq \frac{q_p}{2^\kappa} + \frac{2q_d}{2^\tau} + \frac{5\sigma q_p}{2^b} + \frac{rq_p}{2^c} + \sum_{i\in\mathcal{D}} \frac{\mu_{m_i^*,q_p}}{2^c} + \frac{rq_p\sigma}{2^{2c}}. \quad (9)$$

Decoding the Security Bound: The only variable in the security bound given in theorem 3.1 or Eq. (9) is the multi-chain term $\mu_{t,k}$. In section 4, we

Algorithm D.1 A complete Encryption/Decryption Algorithm for Transform-then-Permute mode with Associated data.

```
**function** $\mathsf{Enc}(N, K, A, M)$
    $(A_1, \ldots, A_a) \xleftarrow{r} A$
    $(M_1, \ldots, M_m) \xleftarrow{r} M$
    $(\delta_1, \ldots, \delta_{a+m}) \leftarrow \mathsf{DS}(|A|, |M|)$
    $Y_0 \leftarrow \Pi(N \| K)$
    **for** $i = 1$ to $a$ **do**
        $\overline{A}_i \leftarrow \mathsf{encode}(A_i) \oplus \delta_i$
        $X_i \leftarrow L_e(Y_{i-1}) \oplus \overline{A}_i$
        $Y_i \leftarrow \Pi(X_i)$
    **for** $i = 1$ to $m - 1$ **do**
        $C_i \leftarrow M_i \oplus \lceil Y_{a+i-1} \rceil_r$
        $\overline{M}_i \leftarrow \mathsf{encode}(M_i) \oplus \delta_{a+i}$
        $X_{a+i} \leftarrow L_e(Y_{a+i-1}) \oplus \overline{M}_i$
        $Y_{a+i} \leftarrow \Pi(X_{a+i})$
    $C_m \leftarrow M_m \oplus \lceil Y_{a+m-1} \rceil_{|M_m|}$
    $\overline{M}_m \leftarrow \mathsf{encode}(M_m) \oplus \delta_{a+m}$
    $X_{a+m} \leftarrow L_e(Y_{a+m-1}) \oplus \overline{M}_m$
    $Y_{a+m} \leftarrow \Pi(X_{a+m})$
    $T \leftarrow \lceil Y_{a+m} \rceil_\tau$
    **return** $(C_1 \| \ldots \| C_m, T)$

**function** $L_d(Y, i)$
    **return** $L_e(Y) \oplus \mathsf{encode}(\lceil Y \rceil_i)$

**function** $\mathsf{Dec}(N, K, A, C, T)$
    $(A_1, \ldots, A_a) \xleftarrow{r} A$
    $(C_1, \ldots, C_m) \xleftarrow{r} C$
    $(\delta_1, \ldots, \delta_{a+m}) \leftarrow \mathsf{DS}(|A|, |C|)$
    $Y_0 \leftarrow \Pi(N \| K)$
    **for** $i = 1$ to $a$ **do**
        $\overline{A}_i \leftarrow \mathsf{encode}(A_i) \oplus \delta_i$
        $X_i \leftarrow L_e(Y_{i-1}) \oplus \overline{A}_i$
        $Y_i \leftarrow \Pi(X_i)$
    **for** $i = 1$ to $m - 1$ **do**
        $M_i \leftarrow C_i \oplus \lceil Y_{a+i-1} \rceil_r$
        $\overline{C}_i \leftarrow \mathsf{encode}(C_i) \oplus \delta_{a+i}$
        $X_{a+i} \leftarrow L_d(Y_{a+i-1}, r) \oplus \overline{C}_i$
        $Y_{a+i} \leftarrow \Pi(X_{a+i})$
    $M_m \leftarrow C_m \oplus \lceil Y_{a+m-1} \rceil_{|C_m|}$
    $\overline{C}_m \leftarrow \mathsf{encode}(C_m) \oplus \delta_{a+m}$
    $X_{a+m} \leftarrow L_d(Y_{a+m-1}, |C_m|) \oplus \overline{C}_m$
    $Y_{a+m} \leftarrow \Pi(X_{a+m})$
    $T' \leftarrow \lceil Y_{a+m} \rceil_\tau$
    **if** $T' \neq T$ **then**
        **return** $M := \perp$
    **else**
        **return** $M := M_1 \| \ldots \| M_m$
```

show that for invertible feedback functions L_d of decryption function the value is roughly bounded by $\mathsf{mcoll}(q_p, 2^\tau) + \mathsf{mcoll}(q_p, 2^r) + k \cdot \mathsf{mcoll}'(q_p^2, 2^b)$. Using this simple fact, we derive improved bounds for Beetle and SpoC. The feedback function in duplex is not invertible, so we do not get any improvement in that case. But, in the following subsection we show that a tight bound on the multi-chain term will indeed give a tight security bound for duplex.

D.1 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. Clearly 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 $\mathcal{O}(q_d/2^\tau)$.
2. Similar 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.
3. Suppose $\sum_{i\in\mathcal{D}} \frac{\mu_{t,m_i}}{2^c}$ maximizes for some adversary $\mathcal{B}$ interacting with $\mathsf{\Pi}$. 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 $1 \le i \le q_d$, $\lceil Y_{i,0} \rceil_r$ 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 is u_i) to the nodes (whose τ most significant bits is T_i) and with label x_i. Now we choose the appropriate ciphertext C_i^* such that $Y_{i,1}^* = u_i$. 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. By taking expectation, we achieve the desired success probability.

E Proof of Theorem 4.1

Fix an invertible linear function L. Now, one can easily draw the following observations:

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

More Notations: We now describe some more notations related to multi-chain W_k defined on the interactive transcript of $\mathcal{A}^{\mathsf{\Pi}}$. Let θ be the transcript and θ' be the list containing input-output information (as described above).

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. 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 $\mathsf{W}^{\mathsf{bck}}$.

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 $\mathsf{W}^{\mathsf{mitm}}$. Note that $\mathsf{W}^{\mathsf{mitm},a}$ denotes the number of forward and backward query tuples that meet in the middle for a fixed value a, whence the notation.

Before proving theorem 4.1, we give an intermediate result in lemma E.1 which in combination with the multicollision result of proposition B.1 gives the proof of theorem 4.1.

Lemma E.1. *For any transcript, 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: 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. □

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 sub section B.1. 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 B.1. Now, using linearity of expectation, we have

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