Revisiting the Security of COMET Authenticated Encryption Scheme [⋆]

Shay Gueron[1,2], Ashwin Jha[3], and Mridul Nandi[4]

[1]University of Haifa, Haifa, Israel
[2]Amazon Web Services, Seattle, U.S.A.
[3]CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
[4]Indian Statistical Institute, Kolkata, India
shay.gueron@gmail.com,ashwin.jha@cispa.de,mridul.nandi@gmail.com

Abstract. COMETv1, by Gueron, Jha and Nandi, is a mode of operation for nonce-based authenticated encryption with associated data functionality. It was one of the second round candidates in the ongoing NIST Lightweight Cryptography Standardization Process. In this paper, we study a generalized version of COMETv1, that we call gCOMET, from provable security perspective. First, we present a comprehensive and complete security proof for gCOMET in the ideal cipher model. Second, we view COMET, the underlying mode of operation in COMETv1, as an instantiation of gCOMET, and derive its concrete security bounds. Finally, we propose another instantiation of gCOMET, dubbed COMETv2, and show that this version achieves better security guarantees as well as memory-efficient implementations as compared to COMETv1.

Keywords: COMET, ICM, provable security, rekeying, lightweight, AEAD

1 Introduction

Lightweight cryptography has seen a sudden surge in demand due to the recent advancements in the field of Internet of things (IoT). The NIST lightweight cryptography standardization project [1], henceforth referred as the NIST LwC project, intends to address this demand by standardizing lightweight authenticated encryption (AE) and cryptographic hash schemes.

The first round of NIST LwC project had 56 candidates, of which 32 were selected to continue to second round. Among these 32 candidates around 15 schemes were based on (tweakable) block ciphers. In this paper we focus on one particular block cipher based candidate, called COMET [2,3] by Gueron et al., that uses nonce and position based re-keying and a COFB [4] or Beetle [5] like feedback operation.

COMET can be viewed as an ideal cipher based alternative for Beetle [5] and COFB [4]. Indeed, the designers state that the mode of operation can be viewed as a mixture of CTR [6] and Beetle. COMET is parameterized by the block size of the underlying block cipher. Accordingly, COMET-n means COMET with block

[⋆] An abridged version of this paper appears in INDOCRYPT 2021. This is the full version.

size n. It has two versions, one with $n = \kappa$, and the other with $n = \kappa/2$, where κ denotes the key size of the block cipher. The concrete submissions using COMET mode are based on AES-128/128 [7], Speck-64/128 [8,9], CHAM-128/128 [10], and CHAM-64/128 [10]. Some of the standout features of COMET are as follows:

1. Design Simplicity: The design of COMET is extremely simple. Apart from the block cipher evaluations, it only requires simple shift and XOR operations.
2. Small State Size: Theoretically, COMET requires only $(n+\kappa)$-bit internal state, which makes it one of the smallest AEAD candidate in the ongoing NIST LwC project.
3. Efficiency: COMET is single-pass, which makes it quite efficient in both hardware and software. Apart from the block cipher call, only 1 shift and at most 2 XOR operations are required per block of input. This places COMET among the fastest candidates in the ongoing NIST LwC project. In fact, according to the publicly available software implementation and benchmarking by Weatherley [11], COMET outperforms all other candidates by a significant margin.

1.1 Motivations and Related Works

In this paper, we concentrate on the provable security of the COMET mode of operation. The designers made the following claims with respect to the security of COMET:

– COMET-128 is secure while the data complexity, denoted D, is at most 2^{64} bytes, and the time complexity, denoted T, is at most 2^{119}.
– COMET-64 is secure while $D < 2^{45}$ bytes, and $T < 2^{112}$.

Note that, the designers make a better claim with respect to the privacy of COMET-64. However, for the sake of uniformity, we mention the more conservative bound claimed for the integrity of COMET-64. In [12], Khairallah presented the first cryptanalytic work on COMET. Later, as noted by the designers [13], Bernstein, Henri and Turan [14] shared two observations on the security of COMET-64. While these works do not invalidate the security claims due to a breach in the data complexity limit, they do demonstrate a possible tightness of the security claims. Shortly after Khairallah's work, at NIST Lightweight Cryptography Workshop 2019, the designers presented a brief sketch of the security proof [13] for COMET-128. However, their proof approach was not applicable to COMET-64. In this paper, we aim to give a comprehensive proof of security for the COMET mode of operation.

1.2 Our Contributions

Our contributions are twofold:

1. We propose a generalization of COMET, dubbed as gCOMET (see section 3). We intend to employ the recently introduced proof strategy of Chakraborty et al. [15] to prove the security of gCOMET. Consequently, in section 4 and 5, we extend the tools and results used in [15]. We give a detailed security proof for gCOMET in section 6.

Table 1.1: Summary of security bounds for COMET and COMETv2 as per the results in this paper.

Submissions	Data (D)	Time (T)	Data-Time (DT) Trade-off
COMET-128	2^{63} bytes	$2^{125.19}$	$2^{184.24}$
COMET-64	2^{42} bytes	2^{112}	$2^{152.24}$
COMETv2-128	2^{64} bytes	$2^{125.19}$	$2^{184.24}$
COMETv2-64	2^{63} bytes	$2^{121.58}$	$2^{152.24}$

2. We view COMET as an instance of gCOMET and obtain concrete security bounds for both versions of COMET. Specifically, we show that
   - COMET-128 is secure while: $D < 2^{63}$ bytes and $T < 2^{125.19}$ and $DT < 2^{184.24}$.
   - COMET-64 is secure while: $D < 2^{42}$ bytes and $T < 2^{112}$ and $DT < 2^{152.24}$.

   Further, we observe that two simple changes in the design of COMET, improves the performance and increases the security (by avoiding the attacks in [12,14]). We call this new version, COMETv2. In terms of security, we show that
   - COMETv2-128 is secure while: $D < 2^{64}$ bytes and $T < 2^{125.19}$ and $DT < 2^{184.24}$.
   - COMETv2-64 is secure while: $D < 2^{63}$ bytes and $T < 2^{121.58}$ and $DT < 2^{152.24}$.

   We summarize the concrete security bounds for different variants of COMET and COMETv2 in Table 1.1. Our security bounds validate the security claims for COMET-128, as given in [3]. For COMET-64, our bounds are slightly lower than the ones claimed by the designers. However, we note that we could not find any matching attacks. So, the exact security of COMET-64 is still an open problem.

2 Preliminaries

Notational Setup: Let $\mathbb{N}$ denote the set of all natural numbers and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. Fix some $n \in \mathbb{N}$. We write $(n]$ to denote the set $\{0, \ldots, n-1\}$. For $m, k \in \mathbb{N}_0$, such that $m \geq k$, we define the falling factorial $(m)_k := m!/(m-k)!$. Note that, $(m)_k \leq m^k$. For $m, n \in \mathbb{N}$, $A_{m\times n}$ denotes an $m \times n$ binary matrix (or simply A_n, when $m = n$). The identity matrix of dimension n is denoted I_n and the null matrix of dimension $m \times n$ is denoted $\mathsf{0}_{m\times n}$. We write $\mathsf{rank}(A_n)$ to denote the rank of A_n. For any square matrix A_n, we define the period of A_n, denoted $\mathsf{cycle}(A_n)$, as the smallest integer k such that $A_n^k = \mathsf{I}_n$. We drop the dimensions of the matrix, whenever they are understood from the context.

We use $\{0,1\}^n$ and $\{0,1\}^+$ to denote the set of all n-bit strings, and non-empty binary strings, respectively. ε denotes the empty string and $\{0,1\}^* := \{0,1\}^+ \cup \{\varepsilon\}$. For any string $B \in \{0,1\}^+$, $|B|$ denotes the number of bits in B, also referred as the length or size of B. We use little-endian format of indexing, i.e., for any $B \in \{0,1\}^+$, we write and view B as a $|B|$-bit binary string $b_{|B|-1} \cdots b_0$, i.e., the most significant bit $b_{|B|-1}$ lies on the left. For $B \in \{0,1\}^+$, $(B_{\ell-1}, \ldots, B_0) \xleftarrow{n} B$, denotes the n-bit *block parsing* of B into $(B_{\ell-1}, \ldots, B_0)$, where $|B_i| = n$ for $0 \leq i \leq \ell - 2$, and $1 \leq |B_{\ell-1}| \leq n$. For $A, B \in \{0,1\}^+$, and $|A| = |B|$, $A \oplus B$ denotes the "bitwise XOR" operation on A and B. For $A, B \in \{0,1\}^*$, $A \| B$ denotes the "string concatenation" operation on A and B. For $A, B \in \{0,1\}^*$ and $X = A \| B$, A and B are called the *prefix* and *suffix* of X, respectively.

For $q \in \mathbb{N}$, X^q denotes the q-tuple $(X_0, \ldots, X_{q-1})$. For $q \in \mathbb{N}$ and any set $\mathcal{X}$ such that $|\mathcal{X}| \geq q$, we write $(\mathcal{X})_q$ to denote the set of all q-tuples with pairwise distinct elements from $\mathcal{X}$, i.e., $|(\mathcal{X})_q| = (|\mathcal{X}|)_q$. For a finite set $\mathcal{X}$, $\mathsf{X}^q \leftarrow_{\$} \mathcal{X}$ denotes the uniform at random sampling of q variables $\mathsf{X}_0, \ldots, \mathsf{X}_{q-1}$ from $\mathcal{X}$ in with replacement fashion.

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* $\mathcal{A}$, *plaintext space* $\mathcal{P}$, *ciphertext space* $\mathcal{C}$, and *tag space* $\mathcal{T}$, where:

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

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 $(K, N, A, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{P}$. 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.

Ideal Block Cipher: For $n \in \mathbb{N}$, let $\mathsf{Perm}(n)$ denote the set of all permutations of $\{0,1\}^n$. For $n, \kappa \in \mathbb{N}$, $\mathsf{ICPerm}(\kappa, n)$ denotes the set of all families of permutations $\pi_K := \pi(K, \cdot) \in \mathsf{Perm}(n)$ over $\{0,1\}^n$, indexed by $K \in \{0,1\}^\kappa$. A block cipher with key size κ and block size n is a family of permutations $\mathsf{IC} \in \mathsf{ICPerm}(\kappa, n)$. For $K \in \{0,1\}^\kappa$, we denote $\mathsf{IC}_K(\cdot) = \mathsf{IC}^+_K(\cdot) := \mathsf{IC}(K, \cdot)$, and $\mathsf{IC}^-_K(\cdot) := \mathsf{IC}^{-1}(K, \cdot)$. Throughout this paper, we denote the *key size* and *block size* of the block cipher by κ and n, respectively. In this context, a binary string X, with $|X| \leq n$, is called a *full* block if $|X| = n$, and *partial* block otherwise. *A block cipher is said to be an ideal cipher if for all $K \in \{0,1\}^\kappa$, $\mathsf{IC}_K \leftarrow_{\$} \mathsf{Perm}(n)$.*

AEAD Security in the Ideal Cipher Model (ICM): Let $\mathsf{AE}_{\mathsf{IC}}$ be an AEAD scheme, based on the ideal cipher IC, defined over $(\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{P}, \mathcal{C}, \mathcal{T})$. In this paper, we fix $\mathcal{K} = \{0,1\}^\kappa$, $\mathcal{N} = \{0,1\}^\eta$, $\mathcal{T} = \{0,1\}^\tau$, and $\mathcal{C} = \mathcal{P} = \mathcal{A} = \{0,1\}^*$, for some fixed $\kappa, \eta, \tau \in \mathbb{N}$. Accordingly, we denote the *key size*, *nonce size*, and *tag size* by κ, η, and τ, respectively. Let

$$\mathsf{Func} := \{f : \mathcal{N} \times \mathcal{A} \times \mathcal{P} \to \mathcal{C} \times \mathcal{T} \ : \ \forall (N, A, M) \in \mathcal{N} \times \mathcal{A} \times \mathcal{P},\ |f(N, A, M)| = |M| + \tau\},$$

and $\mathsf{\Gamma} \leftarrow\!\!\$\ \mathsf{Func}$. Let $\perp$ denote the degenerate function from $(\mathcal{N}, \mathcal{A}, \mathcal{P}, \mathcal{T})$ to $\{\perp\}$. For brevity, we denote the oracle corresponding to a function by the function itself, and bidirectional access to IC is denoted by the superscript $\pm$.

Definition 2.1. *The AEAD advantage of any adversary $\mathscr{A}$ against $\mathsf{AE}_{\mathsf{IC}}$ is defined as,*

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

where $\mathscr{A}^{\mathsf{E}_{\mathsf{K}}, \mathsf{D}_{\mathsf{K}}, \mathsf{IC}^{\pm}}$ and $\mathscr{A}^{\mathsf{\Gamma}, \perp, \mathsf{IC}^{\pm}}$ denote $\mathscr{A}$'s response after its interaction with $(\mathsf{E}_{\mathsf{K}}, \mathsf{D}_{\mathsf{K}}, \mathsf{IC}^{\pm})$ and $(\mathsf{\Gamma}, \perp, \mathsf{IC}^{\pm})$, respectively.

In this paper, we assume that the adversary is *non-trivial* and *nonce respecting*, i.e., it never makes a duplicate query, it never makes a query for which the response is already known due to some previous query, and it does not repeat nonce values in encryption queries. Throughout, we use the following notations to parametrize 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 plaintext/ciphertext) lengths across all encryption and decryption queries, respectively. We also write $q_c = q_e + q_d$ and $\sigma_c = \sigma_e + \sigma_d$ to denote the combined construction query resources.
- q_p denotes the number of primitive queries.

An adversary $\mathscr{A}$ that abides by the above resources is referred as a $(q_e, q_d, \sigma_e, \sigma_d, q_p)$-adversary. We remark here that q_c and σ_c correspond to the *online complexity* (grouped under data complexity $D = q_c + \sigma_c$), and q_p corresponds to the *offline complexity* (grouped under time complexity $T = q_p$) of the adversary.

2.2 Expectation Method

We discuss the expectation method by Hoang and Tessaro [16] in context of AEAD security in the ideal cipher model. Consider a computationally unbounded and deterministic adversary $\mathscr{A}$ that tries to distinguish the real oracle $\mathcal{R} := (\mathsf{E}_{\mathsf{K}}, \mathsf{D}_{\mathsf{K}}, \mathsf{IC}^{\pm})$ from the ideal oracle $\mathcal{I} := (\mathsf{\Gamma}, \perp, \mathsf{IC}^{\pm})$. We denote the query-response tuple of $\mathscr{A}$'s interaction with its oracle by a transcript ω. Sometime this may also include any additional information the oracle chooses to reveal to the adversary at the end of the query-response phase of the game. We will consider this extended definition of transcript.

Let $\mathtt{R}$ (res. $\mathtt{I}$) denote the random transcript variable when $\mathscr{A}$ interacts with $\mathcal{R}$ (res. $\mathcal{I}$). 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 $\Pr[\mathtt{I} = \omega] > 0$.

Theorem 2.1 (Expectation method [16]). *Let Ω be the set of all transcripts. For some $\epsilon_{\mathsf{bad}} \geq 0$ and a non-negative function $\epsilon_{\mathsf{ratio}} : \Omega \to [0, \infty)$, suppose there is a set $\Omega_{\mathsf{bad}} \subseteq \Omega$ satisfying the following:*

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

$$\frac{\Pr[\mathtt{R} = \omega]}{\Pr[\mathtt{I} = \omega]} \geq 1 - \epsilon_{\mathsf{ratio}}(\omega).$$

Then, for any adversary $\mathscr{A}$, we have

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{AE}_{\mathsf{IC}}}(\mathscr{A}) \leq \epsilon_{\mathsf{bad}} + \mathsf{Ex}[\epsilon_{\mathsf{ratio}}(\mathtt{I})].$$

A proof of this theorem is available in multiple papers including [16,17]. The H-coefficient technique due to Patarin [18,19] is a simple corollary of this result, where $\epsilon_{\mathsf{ratio}}$ is a constant function.

3 Generalized COMET Mode of Operation

COunter Mode Encryption with authentication Tag, or COMET in abbreviation, is a block cipher mode of operation by Gueron, Jha and Nandi [2,3] that provides authenticated encryption with associated data functionality. At a very high level, it can be viewed as a mixture of CTR [6], Beetle [5], and COFB [4] modes of operation. In this section, we provide a slightly generalized description of the COMET mode of operation, that we call gCOMET.

3.1 Parameters and Building Blocks

The gCOMET mode of operation is based on a block cipher IC with n-bit block and κ-bit key size.

Parameters: In the following, we describe various parameters used in gCOMET along with their limits:

1. *Block size:* The block size n of IC also denotes the block size of gCOMET. It is analogous to the *rate* parameter used in Sponge-based schemes [20,5].
2. *Key size:* The key size κ is simply the key size of the underlying block cipher IC, that follows $\kappa \geq n$.
3. *State size:* The $(n+\kappa)$-bit input size of the underlying block cipher IC denotes the state size $\mathfrak{s}$ of gCOMET.
4. *Control and Invariant-prefix size:* gCOMET uses a small number of bits, called *control bits* (or, control) for separating the various phases of execution, such as associated data (AD) processing and plaintext processing, and identifying full and partial block data. We denote the control size by $\mathfrak{c}$ and it follows $\mathfrak{c} \ll \kappa$. In fact, the control bits can be described in very few bits. For instance, COMET [2,3] uses $\mathfrak{c} = 5$.
   On a related note, we also use an auxiliary parameter $\mathfrak{c}'$, called the *invariant-prefix size*, following the relation $\mathfrak{c}' \geq \mathfrak{c}$. For example, COMET uses $\mathfrak{c}' = \kappa/2$.
5. *Nonce size:* The nonce size η follows the relation:

$$\begin{aligned} \eta &\leq n && \text{if } n = \kappa, \\ \eta &\leq \kappa - \mathfrak{c} && \text{if } n < \kappa. \end{aligned} \tag{2}$$

6. *Tag size:* The tag size τ follows the relation $\tau \leq n$.

From the above discussion, one can see that gCOMET is primarily parameterized by the block size n and the key size κ, and all other parameters are bounded in terms of these two. Accordingly, we write $\mathsf{fatCOMET}$ and $\mathsf{tinyCOMET}$ to denote gCOMET with $n = \kappa$ and $n < \kappa$, respectively. In each case, the nonce size η is a fixed number that follows the condition given in Eq. (2). For the sake of simplicity, we assume $\eta = n$ for $\mathsf{fatCOMET}$ and $\eta = \kappa - \mathfrak{c}$ for $\mathsf{tinyCOMET}$.

Building Blocks: Apart from the block cipher IC, gCOMET has three more components that are described below:

Control sequence generator: We define the control sequence generator as the function $\Delta : \mathbb{N}_0 \times \mathbb{N}_0 \rightarrow (\{0,1\}^{\mathfrak{c}})^+$ such that $|\Delta(a,m)| = (a+m+2)\mathfrak{c}$ for all $a, m \in \mathbb{N}_0$.

Feedback functions: Let Φ be an invertible linear map over $\{0,1\}^n$ and $\Phi' := \Phi \oplus \mathsf{I}$, the pointwise sum of Φ and I, where I denotes the identity map over $\{0,1\}^n$. We define the feedback functions as follows:

– $\mathsf{L}_{\mathrm{ad}} : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ is defined by the mapping

$$(X, A) \longmapsto X \oplus A.$$

– $\mathsf{L}_{\mathrm{pt}} : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n \times \{0,1\}^n$ is defined by the mapping

$$(X, M) \longmapsto (X \oplus M, \Phi(X) \oplus M).$$

– $\mathsf{L}_{\mathrm{ct}} : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n \times \{0,1\}^n$ is defined by the mapping

$$(X, C) \longmapsto (\Phi'(X) \oplus C, \Phi(X) \oplus C).$$

Key-update function: Let Ψ be an invertible linear map over $\{0,1\}^{\kappa-\mathfrak{c}'}$. We define the update function $\mathsf{U} : \{0,1\}^\kappa \rightarrow \{0,1\}^\kappa$ by the binary matrix

$$\mathsf{U} := \begin{bmatrix} \mathsf{I}_{\mathfrak{c}'} & 0_{\mathfrak{c}' \times \kappa - \mathfrak{c}'} \\ 0_{\kappa - \mathfrak{c}' \times \mathfrak{c}'} & \Psi \end{bmatrix},$$

where Ψ is viewed as a $(\kappa - \mathfrak{c}')$ square matrix with elements from $\{0,1\}$. The above definition implies that $\mathfrak{c}'$ controls the prefix size of the key that remains unchanged in the key updation. This motivates our nomenclature for $\mathfrak{c}'$ as the invariant-prefix size parameter.

3.2 Description of gCOMET

In the following, we describe the main phase of gCOMET's encryption/decryption algorithm for a tuple of input (K, N, A, I) where K, N, A, and I denote the key, nonce, associated data and plaintext (ciphertext in case of decryption), respectively:

Initialization phase: This phase computes the initial state for the algorithm. This is the only phase where the two gCOMET versions, namely $\mathsf{fatCOMET}$ and $\mathsf{tinyCOMET}$ differ. Specifically, In fatCOMET, we have

```
init$_{n,\kappa}(K, N, A, I)$ :
1: $a \leftarrow \left\lceil \frac{|A|}{n} \right\rceil$, $m \leftarrow \left\lceil \frac{|I|}{n} \right\rceil$, $\ell = a + m$
2: $\delta^{\ell+2} \leftarrow \Delta(a, m)$
3: $Y_0 \leftarrow K$
4: $Z'_0 \leftarrow \mathsf{IC}^{+}_{K}(N) \oplus \delta_0 \| 0^{\kappa-\mathfrak{c}}$
5: return $(Y_0, Z'_0, \delta^{\ell+2}, a, m, \ell)$
```

In tinyCOMET, we have

```
init$_{n,\kappa}(K, N, A, I)$ :
1: $a \leftarrow \left\lceil \frac{|A|}{n} \right\rceil$, $m \leftarrow \left\lceil \frac{|M|}{n} \right\rceil$, $\ell = a + m$
2: $\delta^{\ell+2} \leftarrow \Delta(a, m)$
3: $Y_0 \leftarrow \mathsf{IC}^{+}_{K}(0^n)$
4: $Z'_0 \leftarrow K \oplus \delta_0 \| N$
5: return $(Y_0, Z'_0, \delta^{\ell+2}, a, m, \ell)$
```

Data processing phase: This phase consists of two modules corresponding to associate data processing, denoted proc_ad, and plaintext/ciphertext processing, denoted proc_pt/proc_ct. Each of these modules only execute for non-empty data. The modules are identical except for the feedback functions. For non-empty data the processing is as follows:

```
proc_ad$(Y_0, Z'_0, A, \delta^{\ell+2})$:
1: $(A_{a-1}, \ldots, A_0) \xleftarrow{n} A$
2: for $i = 0$ to $a - 1$ do
3:     $Z_i \leftarrow \mathsf{U}(Z'_i)$
4:     $X_i \leftarrow \mathsf{IC}^{+}_{Z'_i}(Y_i)$
5:     $Y_{i+1} \leftarrow \mathsf{L}_{\mathrm{ad}}(X_i, A_i)$
6:     $Z'_{i+1} \leftarrow Z_i \oplus \delta_{i+1} \| 0^{\kappa-\mathfrak{c}}$
7:     return $(Y_a, Z'_a)$
```

```
proc_pt$(Y_a, Z'_a, I, \delta^{\ell+2})$:
1: $(I_{m-1}, \ldots, I_0) \xleftarrow{n} I$
2: for $j = 0$ to $m - 1$ do
3:     $k \leftarrow a + j$
4:     $Z_k \leftarrow \mathsf{U}(Z'_k)$
5:     $X_k \leftarrow \mathsf{IC}^{+}_{Z_k}(Y_k)$
6:     $(Y_{k+1}, O_j) \leftarrow \mathsf{L}_{\mathrm{pt}}(X_k, I_j)$
7:     $Z'_{k+1} \leftarrow Z_k \oplus \delta_{k+1} \| 0^{\kappa-\mathfrak{c}}$
8:     $O \leftarrow (O_{m-1}, \ldots, O_0)$
9:     return $(Y_\ell, Z'_\ell, O)$
```

```
proc_ct$(Y_a, Z'_a, I, \delta^{\ell+2})$:
1: $(I_{m-1}, \ldots, I_0) \xleftarrow{n} I$
2: for $j = 0$ to $m - 1$ do
3:     $k \leftarrow a + j$
4:     $Z_k \leftarrow \mathsf{U}(Z'_k)$
5:     $X_k \leftarrow \mathsf{IC}^{+}_{Z_k}(Y_k)$
6:     $(Y_{k+1}, O_j) \leftarrow \mathsf{L}_{\mathrm{ct}}(X_k, I_j)$
7:     $Z'_{k+1} \leftarrow Z_k \oplus \delta_{k+1} \| 0^{\kappa-\mathfrak{c}}$
8:     $O \leftarrow (O_{m-1}, \ldots, O_0)$
9:     return $(Y_\ell, Z'_\ell, O)$
```

Tag generation phase: This is the final step and generates the tag.

```
proc_tg$(Y_\ell, Z'_\ell, \delta_{\ell+1})$:
1: $Z'_\ell \leftarrow Z'_\ell \oplus \delta_{\ell+1} \| 0^{\kappa-\mathfrak{c}}$
2: $Z_\ell \leftarrow \mathsf{U}(Z'_\ell)$
3: $T := X_\ell \leftarrow \mathsf{IC}^{+}_{Z_\ell}(Y_\ell)$
4: return $T$
```

Algorithm 3.1 gives the complete algorithmic description of gCOMET, and figure 3.1 illustrates the major components of the encryption/decryption process.

4 Expected Maximum Multicollision Sizes

We briefly revisit some results on the expectation of maximum multicollision size in a random sample. These results are largely based on the extensive analysis already given in [15]. We mostly reuse the strategy from [15] to derive some new

Algorithm 3.1 Encryption/Decryption algorithm in gCOMET.

```
1: function $\mathsf{gCOMET}_{[\mathsf{IC}]}.\mathsf{E}(K, N, A, M)$
2:     $C \leftarrow \perp$
3:     $(Y_0, Z'_0, \delta^{\ell+2}, a, m, \ell) \leftarrow \mathsf{init}_{n,\kappa}(K, N, A, M)$
4:     if $a \neq 0$ then
5:         $(Y_a, Z'_a) \leftarrow \mathsf{proc\_ad}(Y_0, Z'_0, A, \delta^{\ell+2})$
6:     if $m \neq 0$ then
7:         $(Y_\ell, Z'_\ell, C) \leftarrow \mathsf{proc\_pt}(Y_a, Z'_a, M, \delta^{\ell+2})$
8:     $T \leftarrow \mathsf{proc\_tg}(Y_\ell, Z'_\ell, \delta_{\ell+1})$
9:     return $(C, T)$
```

```
1: function $\mathsf{gCOMET}_{[\mathsf{IC}]}.\mathsf{D}(K, N, A, C, T)$
2:     $(Y_0, Z'_0, \delta^{\ell+2}, a, m, \ell) \leftarrow \mathsf{init}_{n,\kappa}(K, N, A, C)$
3:     if $a \neq 0$ then
4:         $(Y_a, Z'_a) \leftarrow \mathsf{proc\_ad}(Y_0, Z'_0, A, \delta^{\ell+2})$
5:     if $m \neq 0$ then
6:         $(Y_\ell, Z'_\ell, M) \leftarrow \mathsf{proc\_ct}(Y_a, Z'_a, C, \delta^{\ell+2})$
7:     $T' \leftarrow \mathsf{proc\_tg}(Y_\ell, Z'_\ell, \delta_{\ell+1})$
8:     if $T' = T$ then
9:         $\mathsf{is\_auth} \leftarrow 1$
10:    else
11:        $\mathsf{is\_auth} \leftarrow 0,\ M \leftarrow \perp$
12:    return $(\mathsf{is\_auth}, M)$
```

results required in case of COMET. For space limitation, we postpone the proofs of all the propositions in this section to supplementary material A.

Before delving into the results we state a simple observation (also given in [15]) that will be useful in bounding the expectation of any non-negative random variable. For any non-negative random variable Y bounded above by q, and $\rho \in \mathbb{N}$, we have

$$\mathsf{Ex}[\mathsf{Y}] \leq \rho - 1 + q \times \Pr[\mathsf{Y} \geq \rho]. \tag{3}$$

4.1 For Uniform Random Sample

For $n \geq 1$, let $\mathsf{X}^q \leftarrow_{\$} \{0,1\}^n$. We define the maximum multicollision size random variable, denoted $\Theta_{q,n}$, for the sample X^q as follows

$$\Theta_{q,n} := \max_{a \in \{0,1\}^n} |\{i \in (q] : \mathsf{X}_i = a\}|,$$

and write $\mu(q,n)$ to denote $\mathsf{Ex}[\Theta_{q,n}]$.

Proposition 4.1. *For $n \geq 2$,*

$$\mu(q,n) \leq \begin{cases} 3 & \text{if } q \leq 2^{\frac{n}{2}}, \\ \frac{4n}{\log_2 n} & \text{if } 2^{\frac{n}{2}} < q \leq 2^n, \\ 5n\left\lceil \frac{q}{n2^n} \right\rceil & \text{if } q > 2^n. \end{cases}$$

For Ideal Cipher Samples Let $(z_0, y_0), \ldots, (z_{q-1}, y_{q-1})$ be a q-tuple of distinct pairs of key and input to an ideal cipher IC with n-bit input block, such that $z_i \neq z_j$ for all $i \neq j$. For $i \in (q]$, let $\mathsf{X}_i = \mathsf{IC}_{z_i}(y_i)$. We define

$$\widehat{\Theta}_{q,n} := \max_{a \in \{0,1\}^n} |\{i \in (q] : \mathsf{X}_i = a\}|,$$

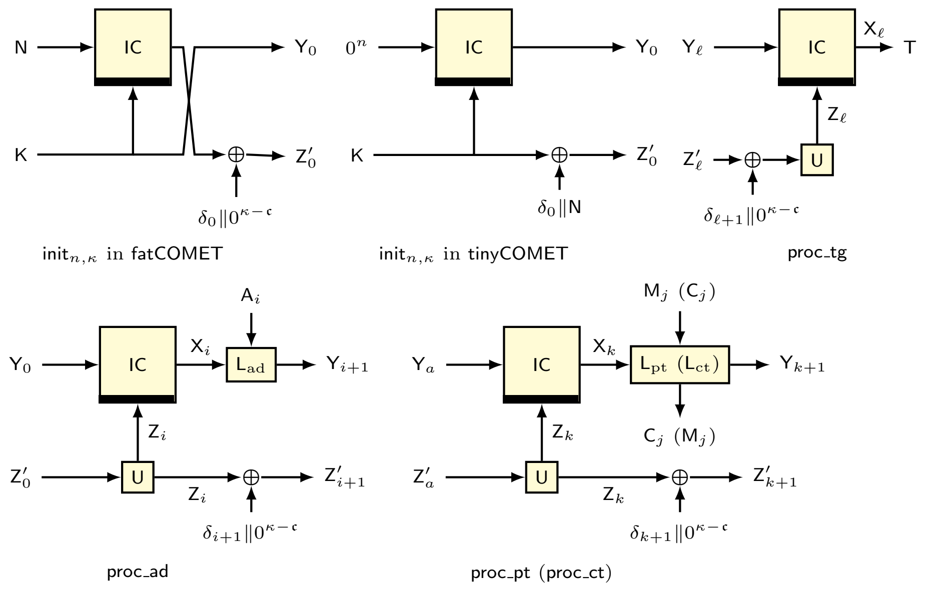

Fig. 3.1: Various phases in the encryption/decryption algorithm of gCOMET. Here, $i \in (a]$, $j \in (m]$ and $k = a + j$.

and write $\widehat{\mu}(q, n)$ to denote $\mathsf{Ex}\left[\widehat{\Theta}_{q,n}\right]$. Since all the keys are pairwise distinct, the sample X^q is statistically indistinguishable from a sample following uniform distribution. Thus, using Proposition 4.1, we get the following proposition for ideal cipher generated samples.

Proposition 4.2. *For $n \geq 2$,*

$$\widehat{\mu}(q, n) \leq \begin{cases} \frac{4n}{\log_2 n} & \text{if } q \leq 2^n \\ 5n \left\lceil \frac{q}{n2^n} \right\rceil & \text{if } q > 2^n. \end{cases}$$

Note that, identical result holds for samples generated through inverse calls to the ideal cipher as well.

For Linear Post-processing: Consider a variant of the above given problem, where we are interested in multicollisions on $(\mathsf{L}(\mathsf{X}_i))_{i \in (q]}$ for some linear map L over $\{0,1\}^n$ with $\mathsf{rank}(\mathsf{L}) = r$. Obviously, $r \leq n$. We define

$$\widehat{\Theta}'_{q,n,r} := \max_{a \in \{0,1\}^n} \left|\{i \in (q] : \mathsf{L}(\mathsf{X}_i) = a\}\right|,$$

and write $\widehat{\mu}'(q, n, r)$ to denote $\mathsf{Ex}\left[\widehat{\Theta}'_{q,n,r}\right]$.

Proposition 4.3. *For $n \geq 2$,*

$$\widehat{\mu}'(q,n,r) \leq \begin{cases} \frac{4n}{\log_2 n} & \text{if } q \leq 2^r \\ 5n\left\lceil \frac{q}{n2^r} \right\rceil & \text{if } q > 2^r. \end{cases}$$

4.2 Sum of Ideal Cipher Sample

Let $(z_0, y_0, x'_0), \ldots, (z_{q-1}, y_{q-1}, x'_{q-1})$ be a q-tuple such that (z_i, y_i) are pairwise distinct and (z_i, x'_i) are pairwise distinct, where $z_i \in \{0,1\}^\kappa$ and $y_i, x'_i \in \{0,1\}^n$. Let L be a linear map over $\{0,1\}^n$ with $\mathsf{rank}(\mathsf{L}) = r$. For $i \in (q]$, let $z'_i = \mathsf{U}(z_i)$ and $\mathsf{C}_i = \mathsf{L}(\mathsf{IC}^{+}_{z_i}(y_i)) \oplus \mathsf{IC}^{-}_{z'_i}(x'_i)$. We define

$$\Theta'_{q,n,r} := \max_{a \in \{0,1\}^n} |\{i \in (q] : \mathsf{C}_i = a\}|,$$

and write $\mu'(q,n,r)$ to denote $\mathsf{Ex}\left[\Theta'_{q,n,r}\right]$. We want to bound $\mu'(q,n,r)$.

Proposition 4.4. *For $n \geq 4$, we have*

$$\mu'(q,n,r) \leq 2n\left\lceil \frac{22nq}{2^r} \right\rceil.$$

5 Super-Chain Structure

In [15], Chakraborty et al. proposed the *multi-chain* structure. They use this tool to give a tight security bound for Sponge-type AEAD constructions like Beetle [5] and SpoC [21]. In this section, we give an extension of the multi-chain structure in our notations. This extended tool will be used later in the security analysis of gCOMET.

Labeled Directed Graph: Let $\mathcal{L} = \{(z_i, y_i, x_i) : i \in (q]\}$ be a list of triples such that $(z_i, y_i) \neq (z_j, y_j)$ and $(z_i, x_i) \neq (z_j, x_j)$ for all $i \neq j \in (q]$, where $z_i \in \{0,1\}^\kappa$ and $x_i, y_i \in \{0,1\}^n$ for all $i \in (q]$. We write $\mathsf{range}(\mathcal{L}) = \{(z_i, x_i) : i \in (q]\}$. Let L be a linear map over $\{0,1\}^n$. To $\mathcal{L}$ and L, we associate a labeled directed graph $\mathcal{G}^{\mathsf{L}}_{\mathcal{L}} = (\mathsf{range}(\mathcal{L}), \mathcal{E})$ over the set of vertices $\mathsf{range}(\mathcal{L})$ with edge set $\mathcal{E}$. For all edge $((z,x),(z',x')) \in \mathcal{E}$ with label $c \in \{0,1\}^n$, denoted $(z,x) \xrightarrow{c} (z',x')$, we have $\mathsf{L}(x) \oplus c = y'$ and $\mathsf{U}(z) = z'$. By extending the notation, a labeled walk $\mathcal{W} = (w_0, \ldots, w_k)$ with label c^k is defined as $\mathcal{W} : w_0 \xrightarrow{c_0} w_1 \xrightarrow{c_1} w_2 \cdots w_{k-1} \xrightarrow{c_{k-1}} w_k$. We usually write it as $w_0 \xrightarrow{c^k} w_k$, where k is referred as the length of the walk. We simply write $\mathcal{G}$, dropping the list $\mathcal{L}$ and linear function L, whenever they are understood from the context.

Definition 5.1 (Chain). *A chain, denoted $\mathcal{C}(c^{k+1})$, with label c^{k+1} in $\mathcal{G}^{\mathsf{L}}_{\mathcal{L}}$ is simply a labeled walk $(z_{i_0}, x_{i_0}) \xrightarrow{c^k} (z_{i_k}, x_{i_k})$ with an additional parameter called sink, denoted $\mathsf{sink}[\mathcal{C}(c^{k+1})]$, and defined as follows*

$$\mathsf{sink}[\mathcal{C}(c^{k+1})] := \begin{cases} x_{i_k} & \text{if } c_k = \varepsilon \\ \mathsf{L}(x_{i_k}) \oplus c_k & \text{if } c_k \neq \varepsilon. \end{cases}$$

We call $\mathcal{C}(c^{k+1})$ a complete (resp. partial) chain if $c_k = \varepsilon$ (resp. $c_k \neq \varepsilon$). We define the source and key of the chain as $\mathsf{src}[\mathcal{C}(c^{k+1})] := x_{i_0}$ and $\mathsf{key}[\mathcal{C}(c^{k+1})] := z_{i_0}$, respectively. Length of $\mathcal{C}(c^{k+1})$, denoted $\#\mathcal{C}(c^{k+1})$, is simply the length of the walk, i.e., k.

In context of this work, a chain is a graphical representation of (a part of) an execution of gCOMET encryption/decryption process, where the label of the chain plays the role of the input string, the key and source of the chain denote the starting point in the execution and the sink denotes the end point. Looking ahead momentarily, in our analysis we will need a special collection of chains starting from a common source and ending in (possibly) distinct sinks.

Definition 5.2 (Super-chain). *A t-sink super-chain, denoted $\mathcal{S}(c^{k+1})$, with label c^{k+1} in $\mathcal{G}^{\mathsf{L}}_{\mathcal{L}}$ is a set of chains $\{\mathcal{C}_0(d_0), \ldots, \mathcal{C}_{l-1}(d_{l-1})\}$ such that*
- *for $i \in (k]$, $c_i \in \{0,1\}^n$ and $c_k = \epsilon$.*
- *for $i \in (l]$, $d_i = c^{j+1}$ for some $j \in (k+1]$.*
- *for distinct $i, j \in (l]$, $\mathsf{src}[\mathcal{C}_i(d_i)] = \mathsf{src}[\mathcal{C}_j(d_j)]$ and $\mathsf{key}[\mathcal{C}_i(d_i)] \neq \mathsf{key}[\mathcal{C}_j(d_j)]$.*
- *$|\{(\mathsf{sink}[\mathcal{C}_i(d_i)], \#\mathcal{C}_i(d_i)) : i \in (l]\}| = t$.*

Size of $\mathcal{S}(c^{k+1})$, denoted $|\mathcal{S}(c^{k+1})|$, is simply the cardinality of $\mathcal{S}(c^{k+1})$, i.e., l.

A super-chain can be viewed as a collection of parallel chains starting at a common decryption query block (source of the super-chain), albeit with different keys, and ending at any one of the possible encryption query blocks or the committed tag value. If an adversary succeeds in generating a super-chain of significant size for a sequence of ciphertext blocks, then it can herd the corresponding decryption query to a desired tag value (or intermediate encryption query block) with significantly high probability. Simply put, a non-trivial[1] forgery would imply that the adversary succeeds in herding a decryption query to one of the chains in the super-chain. As a consequence, we aim to upper bound the size of the super-chain. Note that the multi-chain structure of [22,15] is a special case of super-chain structure, where $t = 1$ and for all $i \in (l]$, $d_i = c^{k+1}$. These extra conditions imply that all the chains are of length k, and they end in a common sink.

5.1 Maximum Size of t-Sink Super-Chain of Length k

Consider a non-trivial adversary $\mathscr{A}$ interacting with an ideal cipher oracle $\mathsf{IC}^{\pm}$. Suppose, $\mathscr{A}$ makes q queries to $\mathsf{IC}^{\pm}$. For $i \in (q]$, let $(\widehat{\mathsf{Z}}_i, \widehat{\mathsf{Y}}_i, \widehat{\mathsf{X}}_i, \widehat{\mathsf{d}}_i)$ denote the i-th query-response tuple, where $\widehat{\mathsf{Z}}_i \in \{0,1\}^\kappa$, $\widehat{\mathsf{Y}}_i, \widehat{\mathsf{X}}_i \in \{0,1\}^n$, and $\widehat{\mathsf{d}}_i \in \{0,1\}$. If $\widehat{\mathsf{d}}_i = 0$, $\mathscr{A}$ queries $(\widehat{\mathsf{Z}}_i, \widehat{\mathsf{Y}}_i)$ and gets response $\widehat{\mathsf{X}}_i := \mathsf{IC}^+(\widehat{\mathsf{Z}}_i, \widehat{\mathsf{Y}}_i)$ (forward query), else it queries $(\widehat{\mathsf{Z}}_i, \widehat{\mathsf{X}}_i)$ and gets response $\widehat{\mathsf{Y}}_i := \mathsf{IC}^-(\widehat{\mathsf{Z}}_i, \widehat{\mathsf{X}}_i)$ (backward query). We store the q query-response tuples in a list $\mathcal{L}$. Sometimes, we also write $\mathcal{L}' := ((\widehat{\mathsf{Z}}_0, \widehat{\mathsf{Y}}_0, \widehat{\mathsf{X}}_0), \ldots, (\widehat{\mathsf{Z}}_{q-1}, \widehat{\mathsf{Y}}_{q-1}, \widehat{\mathsf{X}}_{q-1}))$ which drops information about query direction. Fix a linear map L over $\{0,1\}^n$ and consider the graph $\mathcal{G}^{\mathsf{L}}_{\mathcal{L}'}$. Let

[1] A forgery attack that does not involve exhaustive guessing of internal state or key.

$\mathsf{W}_{t,k}(\mathcal{L}')$ denote the maximum over the size of all t-sink super-chains of length k in $\mathcal{G}^{\mathsf{L}}_{\mathcal{L}'}$. Then, $\mathsf{W}_{t,k}(\mathcal{L})$ is a random variable where the randomness is induced by IC.

Lemma 5.1. *Let* $\nu := \max_{i \in (q]} \left| \{ j : \widehat{\mathsf{Z}}_j = \mathsf{U}(\widehat{\mathsf{Z}}_i) \} \right|$*. For any non-trivial adversary* $\mathscr{A}$ *and an ideal cipher* IC*, we have*

$$\mathsf{Ex}\left[\mathsf{W}_{t,k}(\mathcal{L})\right] \leq 2\widehat{\mu}(q,n) + (t-1) \cdot \widehat{\mu}'(q,n,\mathsf{rank}(\mathsf{L})) + k \cdot \mu'(q\nu, n, \mathsf{rank}(\mathsf{L})).$$

The proof of this lemma is postponed to supplementary material B.

6 Security of gCOMET

In this section, we give a detailed security analysis of gCOMET. Theorem 6.1 gives the combined AEAD security of gCOMET in the ideal cipher model.

Theorem 6.1. *For* $N, r > 0$*, let* $\mathsf{cycle}(\Psi) = N$ *and* $\mathsf{rank}(\Phi') = r$*. Then, for* $n, \nu_{ed} > 0$*,* $\sigma_c < \min\left\{N, 2^{n-2}\right\}$*,* $q_p < 2^{\kappa-2}$ *and* $(q_e, q_d, \sigma_e, \sigma_d, q_p)$*-adversary* $\mathscr{A}$*, we have*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{gCOMET}}(\mathscr{A}) \leq & \left(\frac{2q_p}{2^\kappa} + \frac{6\sigma_c}{2^{\kappa-\mathsf{c}'}} + \frac{4\sigma_d}{2^{\kappa-\mathsf{c}'+n}}\right)\mu(\sigma_c, n) + \frac{4q_d}{2^\kappa}\widehat{\mu}(q_p, n) + \frac{q_c}{2^{\kappa-\mathsf{c}'}} \\ & + \min\left\{\frac{2\sigma_d\sigma_e}{2^\kappa}\widehat{\mu}'(q_p,n,r), \frac{2\sigma_d\sigma_e}{2^{\kappa-\mathsf{c}'}} + \frac{2\sigma_d}{2^\kappa}\widehat{\mu}'(q_p,n,r)\right\} + \frac{q_p+\sigma_c}{2^\kappa} \\ & + \frac{2\sigma_d}{2^\kappa}\mu'(q_p\nu_{ed}, n, r) + \frac{q_p\sigma_c}{\nu_{ed}2^\kappa} + \frac{2q_d(\sigma_e+q_e)}{2^{\kappa-\mathsf{c}'+n}} + \frac{4q_p\sigma_d}{2^{\kappa+n}} + \frac{2q_d}{2^n}. \quad (4)\end{aligned}$$

The proof is given in the rest of this section. In relation to the expectation method (high level tool used in the proof), we largely reuse the definitions and notations from section 2.2.

6.1 Initial Setup and Description of Oracles

We denote the query-response tuple of $\mathscr{A}$'s interaction with its oracle by a transcript $\omega = \{\omega_e, \omega_d, \omega_p\}$, where $\omega_e := \{(\mathsf{N}^i, \mathsf{A}^i, \mathsf{M}^i, \mathsf{C}^i, \mathsf{T}^i) : i \in (q_e]\}$, $\omega_d := \{(\bar{\mathsf{N}}^j, \bar{\mathsf{A}}^j, \bar{\mathsf{C}}^j, \bar{\mathsf{T}}^j, \bar{\mathsf{D}}^j) : j \in (q_d]\}$, and $\omega_p := \{(\widehat{\mathsf{Z}}_k, \widehat{\mathsf{Y}}_k, \widehat{\mathsf{X}}_k, \widehat{\mathsf{d}}_k) : k \in (q_p]\}$. Here,

- $(\mathsf{N}^i, \mathsf{A}^i, \mathsf{M}^i, \mathsf{C}^i, \mathsf{T}^i)$ denotes the i-th encryption query-response tuple, where N^i, A^i, M^i, C^i, and T^i, denote the nonce, associated data, message, ciphertext, and tag, respectively. Let $\left\lceil \frac{|\mathsf{A}^i|}{n} \right\rceil = a^i$, $\left\lceil \frac{|\mathsf{C}^i|}{n} \right\rceil = \left\lceil \frac{|\mathsf{M}^i|}{n} \right\rceil = m^i$, and $\ell^i = a^i + m^i$.
- $(\bar{\mathsf{N}}^j, \bar{\mathsf{A}}^j, \bar{\mathsf{C}}^j, \bar{\mathsf{T}}^j, \bar{\mathsf{D}}^j)$ denotes the j-th decryption query-response tuple, where $\bar{\mathsf{N}}^j$, $\bar{\mathsf{A}}^j$, $\bar{\mathsf{C}}^j$, $\bar{\mathsf{T}}^j$, and $\bar{\mathsf{D}}^j$, denote the nonce, associated data, ciphertext, tag, and the authentication result, respectively. $\bar{\mathsf{D}}^j$ equals to a message $\bar{\mathsf{M}}^j$ when authentication succeeds, and $\perp$ otherwise. Let $\left\lceil \frac{|\bar{\mathsf{A}}^i|}{n} \right\rceil = \bar{a}^j$ and $\left\lceil \frac{|\bar{\mathsf{C}}^i|}{n} \right\rceil = \bar{m}^j$, and $\bar{\ell}^j = \bar{a}^j + \bar{m}^j$.

- $(\widehat{\mathsf{Z}}_k, \widehat{\mathsf{Y}}_k, \widehat{\mathsf{X}}_k, \widehat{\mathsf{d}}_k)$ denotes the k-th primitive query-response tuple, where $\widehat{\mathsf{Z}}_k$, $\widehat{\mathsf{Y}}_k$, $\widehat{\mathsf{X}}_k$, and $\widehat{\mathsf{d}}_k$, denote the key, input, output, and direction of query, respectively. $\widehat{\mathsf{d}}_k = 0$ if the k-th query is forward, and $\widehat{\mathsf{d}}_k = 1$ if the k-th query is backward.

In addition, for all $(i,j) \in (q_e] \times (\ell^i + 1]$ and $(i',j') \in (q_d] \times (\bar{\ell}^i + 1]$, $(\mathsf{Z}^i_j, \mathsf{Y}^i_j, \mathsf{X}^i_j)$ and $(\bar{\mathsf{Z}}^{i'}_{j'}, \bar{\mathsf{Y}}^{i'}_{j'}, \bar{\mathsf{X}}^{i'}_{j'})$ are defined analogous to Figure 3.1 and Algorithm 3.1.

Ideal Oracle Description: The ideal oracle works as follows:

- For the i-th primitive query:
  return $\widehat{\mathsf{X}}_i = \mathsf{IC}^+(\widehat{\mathsf{Z}}_i, \widehat{\mathsf{Y}}_i)$ if $\widehat{\mathsf{d}}_i = 0$, and return $\widehat{\mathsf{Y}}_i = \mathsf{IC}^-(\widehat{\mathsf{Z}}_i, \widehat{\mathsf{X}}_i)$ otherwise.
- For the i-the encryption query:
  - $(\mathsf{X}^i_0, \ldots, \mathsf{X}^i_{\ell^i}) \leftarrow\!\!\$\ \{0,1\}^n$.
  - for $j \in (m^i]$ and $k = a^i + j$, set $(\mathsf{Y}^i_{k+1}, \mathsf{C}^i_j) = \mathsf{L}_{\mathrm{pt}}(\mathsf{X}^i_k, \mathsf{M}^i_j)$ and $\mathsf{T}^i = \mathsf{X}^i_{\ell^i}$.
  - for $j \in (a^i]$, set $\mathsf{Y}^i_{j+1} = \mathsf{L}_{\mathrm{ad}}(\mathsf{X}^i_j, \mathsf{A}^i_j)$.
  - return $(\mathsf{C}^i, \mathsf{T}^i)$.
- For the i-th decryption query: simply return $\perp$.

Note that, the sampling mechanism in the ideal world is slightly indirect in nature. We compute ciphertext and tag outputs by first sampling X values and then using operations identical to gCOMET. However, owing to the invertibility of Φ, the marginal distribution of (C, T) is identical to the case where they are sampled uniform at random.

Real Oracle Description: The real oracle faithfully responds to $\mathscr{A}$'s encryption, decryption, and primitive queries using $\mathsf{IC}^{\pm}$.

Releasing additional information: After the query-response phase is over, the oracles additionally release $(\mathsf{X}^i_0, \ldots, \mathsf{X}^i_{\ell^i})$ to the adversary. We add $(\mathsf{X}^i_0, \ldots, \mathsf{X}^i_{\ell^i})$ to the encryption transcript, i.e. $\mathtt{I}_e$ in case of ideal oracle and $\mathtt{R}_e$ in case of real oracle. Note that, A, M, X tuples completely define $(\mathsf{Y}^i_1, \ldots, \mathsf{Y}^i_{\ell_i})$.

Decryption blocks information from encryption blocks: Consider a decryption query $i \in (q_d]$. If $\bar{\mathsf{N}}^i \neq \mathsf{N}^{i'}$, for all $i' \in (q_e]$, then we define the index of longest common prefix, denoted p_i as -1. If there exists a unique index $i' \in (q_e]$, such that $\bar{\mathsf{N}}^i = \mathsf{N}^{i'}$, then we have

$$p_i := \begin{cases} \max\{j : (\bar{\mathsf{A}}^i_0, \ldots, \bar{\mathsf{A}}^i_{j-1}) = (\mathsf{A}^{i'}_0, \ldots, \mathsf{A}^{i'}_{j-1})\} & \text{if } \bar{\mathsf{A}}^i \neq \mathsf{A}^{i'}, \\ \max\{\bar{a}^i + j : (\bar{\mathsf{C}}^i_0, \ldots, \bar{\mathsf{C}}^i_{j-1}) = (\bar{\mathsf{C}}^{i'}_0, \ldots, \bar{\mathsf{C}}^{i'}_{j-1})\} & \text{otherwise.} \end{cases}$$

It is clear that whenever $p_i \geq 0$, then $(\bar{\mathsf{Z}}^i_0, \bar{\mathsf{Y}}^i_0) = (\mathsf{Z}^{i'}_0, \mathsf{Y}^{i'}_0)$. Further, $\bar{\mathsf{Y}}^i_j$, and $\bar{\mathsf{X}}^i_j$ are determined for all $j \in (p_i + 1]$, due to $\mathsf{Y}^{i'}_j$, $\mathsf{X}^{i'}_j$, and $\bar{\mathsf{C}}^i_j$. Note that, this holds in both the real and ideal world due to the way we define the ideal oracle responses.

At this point, the transcript random variables, viz. $\mathtt{R}$ and $\mathtt{I}$, are completely defined. For the sake of notational simplicity, we use the same notation to represent the constituent random variables in the transcripts of both the world. However, they can be easily separated via their probability distribution which will be determined from their exact definitions in the two worlds. For any transcript ω, we define

– $\theta_e^b := \max_{c \in \{0,1\}^n} |\{(i,j) \in (q_e] \times (m^i + 1] : \mathsf{Y}_j^i = c\}|$.
– $\theta_e^f := \max_{c \in \{0,1\}^n} |\{(i,j) \in (q_e] \times (m^i + 1] : \mathsf{X}_j^i = c\}|$.

Definition 6.1 (Useful index and transcript set). *For $\nu > 0$, the ν-useful index set corresponding to some primitive transcript ω_p, is defined as the maximal set $\mathcal{I}$, such that for all $i \in \mathcal{I}$ we have $\left|\{j \in (q_p] : \widehat{\mathsf{Z}}^j = \widehat{\mathsf{Z}}^i\}\right| \leq \nu$, and the ν-useful transcript set is defined as $\mathcal{Q}_\nu := \{(\widehat{\mathsf{Z}}^i, \widehat{\mathsf{Y}}^i, \widehat{\mathsf{X}}^i) : i \in \mathcal{I}\}$.*

A useful set signifies the keys that do not occur often in primitive queries. Specifically, our aim is to bound the number of keys that appear in both primitive and construction queries. Since, the construction key is not released to the adversary one can get good bounds on ν. Looking ahead momentarily, a useful set will represent the subset of primitive queries that the adversary can use to herd some decryption query to the desired tag value.

6.2 Ratio of Interpolation Probabilities

Fix a transcript $\omega := (\omega_e, \omega_d, \omega_p)$. Since the transcript is attainable, we must have $\omega_d = \perp^{q_d}$. Analogous to the transcript $(\omega_e, \omega_d, \omega_p)$, we also view $\mathtt{I}$ and $\mathtt{R}$ as $(\mathtt{I}_e, \mathtt{I}_d, \mathtt{I}_p)$ and $(\mathtt{R}_e, \mathtt{R}_d, \mathtt{R}_p)$, respectively.

Ideal World: With respect to the encryption transcript, the ideal oracle samples exactly $\sigma_e + q_e$ mutually independent blocks uniformly at random. The decryption transcript holds with probability 1 as the ideal oracle always responds with $\perp$. Using the independence of construction and primitive transcripts in ideal world, we have

$$\Pr[\mathtt{I} = \omega] = \Pr[\mathtt{I}_e = \omega_e, \mathtt{I}_d = \omega_d, \mathtt{I}_p = \omega_p] = \Pr[\mathtt{I}_p = \omega_p] \times \frac{1}{2^{n(\sigma_e + q_e)}}. \tag{5}$$

Consider the multiset, $\mathcal{Z}_p := \{\widehat{\mathsf{Z}}^i : i \in (q_p]\}$. Let $(\mathsf{L}_0, \ldots, \mathsf{L}_{s-1})$ denote the tuple of distinct keys in $\mathcal{Z}_p$ and λ_i^p be the multiplicity of L_i in $\mathcal{Z}_p$ for all $i \in (s]$. Then, in Eq. (5) we have

$$\Pr[\mathtt{I} = \omega] = \frac{1}{\prod_{i \in (s]} (2^n)_{\lambda_i^p}} \times \frac{1}{2^{n(\sigma_e + q_e)}}. \tag{6}$$

Real World: The interpolation probability of ω with respect to the real oracle $\mathcal{R}$ is slightly involved. In particular, we bound the interpolation probability for a special class of values for the internal transcript (i.e. K, Y_0, Z and $\bar{\mathsf{Z}}$) that are compatible with ω. Loosely, the quadruple $(\mathsf{K}, \mathsf{Y}_0, \mathsf{Z}, \bar{\mathsf{Z}})$ is incompatible when it might result in some inconsistent input/output relations for the underlying ideal cipher. Formally, we say that $(\mathsf{K}, \mathsf{Y}_0, \mathsf{Z}, \bar{\mathsf{Z}})$ is incompatible with the external transcript ω, if one of the following events hold:

B0 : $\exists i \in (q_p]$, such that $\mathsf{K} = \widehat{\mathsf{Z}}^i$.

$\texttt{B1}:\ \exists(i,j)\in(q_e]\times(\ell^i+1]$, such that $\mathsf{K}=\mathsf{Z}^i_j$.
$\texttt{B2}:\ \exists(i,j)\in(q_d]\times(\bar{\ell}^i+1]$, such that $\mathsf{K}=\bar{\mathsf{Z}}^i_j$.
$\texttt{B3}:\ \exists i\in(q_e]$, such that $\mathsf{Z}^i_0=*\|0^{\kappa-\mathfrak{c}'}$.
$\texttt{B4}:\ \exists i\in(q_d]$, such that $\bar{\mathsf{Z}}^i_0=*\|0^{\kappa-\mathfrak{c}'}$.
$\texttt{B5}:\ \exists(i,j)\in(q_e]\times(\ell^i+1],(i',j')\in(q_e]\times(\ell^{i'}+1]$, such that $(\mathsf{Z}^i_j,\mathsf{Y}^i_j)=(\mathsf{Z}^{i'}_{j'},\mathsf{Y}^{i'}_{j'})$.
$\texttt{B6}:\ \exists(i,j)\in(q_e]\times(\ell^i+1],(i',j')\in(q_e]\times(\ell^{i'}+1]$, such that $(\mathsf{Z}^i_j,\mathsf{X}^i_j)=(\mathsf{Z}^{i'}_{j'},\mathsf{X}^{i'}_{j'})$.
$\texttt{B7}:\ \exists(i,j)\in(q_e]\times(\ell^i+1],\ i'\in(q_p]$, such that $(\mathsf{Z}^i_j,\mathsf{Y}^i_j)=(\widehat{\mathsf{Z}}^{i'},\widehat{\mathsf{Y}}^{i'})$.
$\texttt{B8}:\ \exists(i,j)\in(q_e]\times(\ell^i+1],\ i'\in(q_p]$, such that $(\mathsf{Z}^i_j,\mathsf{X}^i_j)=(\widehat{\mathsf{Z}}^{i'},\widehat{\mathsf{X}}^{i'})$.
$\texttt{B9}:\ \exists(i,j)\in(q_e]\times(\ell^i+1]$ such that $|\{j\in(q_p]:\widehat{\mathsf{Z}}^j=\mathsf{Z}^i\}|\geq\nu_{ed}$.
$\texttt{B10}:\ \exists(i,j)\in(q_d]\times(\bar{\ell}^i+1]$ such that $|\{j\in(q_p]:\widehat{\mathsf{Z}}^j=\bar{\mathsf{Z}}^i\}|\geq\nu_{ed}$.

For brevity we accumulate the incompatibility events in certain compound events as follows:

$\texttt{Kcoll}:\ \texttt{B0}\cup\texttt{B1}\cup\texttt{B2}\cup\texttt{B3}\cup\texttt{B4}$.
$\texttt{EEmatch}:\ \texttt{B5}\cup\texttt{B6}$.
$\texttt{EPmatch}:\ \texttt{B7}\cup\texttt{B8}$.
$\texttt{PKcount}:\ \texttt{B9}\cup\texttt{B10}$.

The `Kcoll` event handles all the scenarios which might lead to key recovery or internal key collisions. `EEmatch` handles the event that two encryption query block states collide, and `EPmatch` handles a similar scenario for an encryption query block and a primitive query. The event `PKcount` is more of a technical requirement that accounts for the adversarial strategy of exhausting a particular encryption/decryption block key via primitive queries. If this happens, then the adversary can guess the block cipher outputs (or inputs) with higher probability. Let

$$\texttt{Comp}:=\neg\left(\texttt{Kcoll}\cup\texttt{EEmatch}\cup\texttt{EPmatch}\cup\texttt{PKcount}\right).$$

Then, in the real world we have

$$\begin{aligned}\Pr\left[\texttt{R}=\omega\right]&\geq\Pr\left[\texttt{R}=\omega,\texttt{Comp}\right]\\&\geq\Big(1-\Pr\left[\neg\texttt{Comp}\right]\Big)\times\Pr\left[\texttt{R}=\omega\mid\texttt{Comp}\right]\\&\geq\Big(1-\Pr\left[\neg\texttt{Comp}\right]\Big)\times\Pr\left[\texttt{R}_p=\omega_p\mid\texttt{Comp}\right]\\&\qquad\qquad\times\Pr\left[\texttt{R}_e=\omega_e\mid\texttt{Comp}\wedge\texttt{R}_p=\omega_p\right]\\&\qquad\qquad\times\Pr\left[\texttt{R}_d=\omega_d\mid\texttt{Comp}\wedge(\texttt{R}_p,\texttt{R}_e)=(\omega_p,\omega_e)\right].\quad(7)\end{aligned}$$

For any compatible quadruple $(\mathsf{K},\mathsf{Y}_0,\mathsf{Z},\bar{\mathsf{Z}})$, in addition to the multiset $\mathcal{Z}_p$, consider the following two multisets,

$$\mathcal{Z}_e:=\{\mathsf{Z}^i_j:i\in(q_e]\times(m^i]\}\qquad\mathcal{Z}_d:=\{\bar{\mathsf{Z}}^i_j:i\in(q_d]\times(\bar{m}^i]\}$$

We extend $(\mathsf{L}_0, \ldots, \mathsf{L}_{s-1})$ to $(\mathsf{L}_0, \ldots, \mathsf{L}_{s-1}, \ldots, \mathsf{L}_{s'-1})$ for some $s' \geq s$ to denote the tuple of distinct keys in $\mathcal{Z}_p \cup \mathcal{Z}_e$ and let λ_i^t be the multiplicity of L_i in $\mathcal{Z}_t$ for all $t \in \{p, e\}$ and $i \in (s']$. Then, by continuing Eq. (7) we have

$$\begin{aligned}
\Pr[\mathtt{R} = \omega] &\geq \Big(1 - \Pr[\neg\mathtt{Comp}]\Big) \times \frac{1}{\prod_{i \in (s']} (2^n)_{\lambda_i^p}} \times \frac{1}{\prod_{i \in (s']} (2^n - \lambda_i^p)_{\lambda_i^e}} \\
&\qquad \times \Pr[\mathtt{R}_d = \omega_d \mid \mathtt{Comp} \wedge (\mathtt{R}_p, \mathtt{R}_e) = (\omega_p, \omega_e)] \\
&\overset{(*)}{\geq} \Big(1 - \Pr[\neg\mathtt{Comp}]\Big) \times \frac{1}{\prod_{i \in (s]} (2^n)_{\lambda_i^p}} \times \frac{1}{2^{n(\sigma_e + q_e)}} \\
&\qquad \times \Big(1 - \Pr[\mathtt{R}_d \neq \omega_d \mid \mathtt{Comp} \wedge (\mathtt{R}_p, \mathtt{R}_e) = (\omega_p, \omega_e)]\Big) \\
\frac{\Pr[\mathtt{R} = \omega]}{\Pr[\mathtt{I} = \omega]} &\overset{(**)}{\geq} \Big(1 - \Pr[\neg\mathtt{Comp}] - \Pr[\mathtt{R}_d \neq \omega_d \mid \mathtt{Comp} \wedge (\mathtt{R}_p, \mathtt{R}_e) = (\omega_p, \omega_e)]\Big).
\end{aligned} \tag{8}$$

At inequality $(*)$ we use two facts. First, ω_p contains only s distinct keys, and second, $\sum_{i \in (s']} \lambda_i^e = \sigma_e + q_e$. Inequality $(**)$ follows from Eq. (6). In Lemma 6.1 and 6.2 we bound $\Pr[\neg\mathtt{Comp}]$ and $\Pr[\mathtt{R}_d \neq \omega_d \mid \mathtt{Comp} \wedge (\mathtt{R}_p, \mathtt{R}_e) = (\omega_p, \omega_e)]$, respectively.

Lemma 6.1. *For $\sigma_c < \min\{N, 2^{n-2}\}$ and $q_p \leq 2^{\kappa-2}$, we have*

$$\Pr[\neg\mathtt{Comp}] \leq \frac{q_p + \sigma_c + q_p(\theta_e^b + \theta_e^f)}{2^\kappa} + \frac{q_c + 2\sigma_e(\theta_e^b + \theta_e^f)}{2^{\kappa - \mathfrak{c}'}} + \frac{q_p \sigma_c}{\nu_{ed} 2^\kappa}.$$

The proof of this lemma is postponed to supplementary material C.

Lemma 6.2. *Let $\mathtt{E}$ denote the event $\mathtt{Comp} \wedge (\mathtt{R}_p, \mathtt{R}_e) = (\omega_p, \omega_e)$. For $\sigma_c < \min\{N, 2^{n-2}\}$ and $q_p \leq 2^{\kappa-2}$, we have*

$$\begin{aligned}
\Pr[\mathtt{R}_d \neq \omega_d \mid \mathtt{E}] \leq{}& \frac{2q_d(\sigma_e + q_e) + 4\theta_e^b \sigma_d}{2^{\kappa - \mathfrak{c}' + n}} + \frac{2\theta_e^b q_d}{2^{\kappa - \mathfrak{c}'}} + \frac{4 q_p \sigma_d}{2^{\kappa + n}} + \frac{2 q_d}{2^n} \\
&+ \sum_{i \in (q_d]} \min\left\{ \frac{2\mathsf{W}_{\bar{\ell}^i \sigma_e, \bar{\ell}^i}(\mathcal{Q}_{\nu_{ed}})}{2^\kappa}, \frac{2\bar{\ell}^i \sigma_e}{2^{\kappa - \mathfrak{c}'}} + \frac{2\mathsf{W}_{\bar{\ell}^i, \bar{\ell}^i}(\mathcal{Q}_{\nu_{ed}})}{2^\kappa} \right\}.
\end{aligned}$$

The proof of this lemma is postponed to supplementary material D.

On substituting these bounds in Eq. (8), and applying Theorem 2.1, we get

$$\begin{aligned}
\mathbf{Adv}_{\mathsf{gCOMET}}^{\mathsf{aead}}(\mathscr{A}) \leq{}& \left(\frac{q_p}{2^\kappa} + \frac{4\sigma_c}{2^{\kappa - \mathfrak{c}'}} + \frac{4\sigma_d}{2^{\kappa - \mathfrak{c}' + n}} \right) \mathsf{Ex}\left[\theta_e^b\right] + \left(\frac{q_p}{2^\kappa} + \frac{2\sigma_e}{2^{\kappa - \mathfrak{c}'}} \right) \mathsf{Ex}\left[\theta_e^f\right] \\
&+ \sum_{i \in (q_d]} \min\left\{ \frac{2\mathsf{Ex}\left[\mathsf{W}_{\bar{\ell}^i \sigma_e, \bar{\ell}^i}(\mathcal{Q}_{\nu_{ed}})\right]}{2^\kappa}, \frac{2\bar{\ell}^i \sigma_e}{2^{\kappa - \mathfrak{c}'}} + \frac{2\mathsf{Ex}\left[\mathsf{W}_{\bar{\ell}^i, \bar{\ell}^i}(\mathcal{Q}_{\nu_{ed}})\right]}{2^\kappa} \right\} \\
&+ \frac{q_p + \sigma_c}{2^\kappa} + \frac{q_c}{2^{\kappa - \mathfrak{c}'}} + \frac{q_p \sigma_c}{\nu_{ed} 2^\kappa} + \frac{2q_d(\sigma_e + q_e)}{2^{\kappa - \mathfrak{c}' + n}} + \frac{4 q_p \sigma_d}{2^{\kappa + n}} + \frac{2 q_d}{2^n}.
\end{aligned} \tag{9}$$

Note that, θ_e^b and θ_e^f correspond to $\Theta_{\sigma_e,n}$ and $\Theta_{\sigma_d,n}$ respectively (see section 4.1). Thus, $\mathsf{Ex}\left[\theta_e^b\right], \mathsf{Ex}\left[\theta_e^f\right] \leq \mu(\sigma_c, n)$. Further, $|\mathcal{Q}_{\nu_{ed}} \times \mathcal{Q}_{\nu_{ed}}| \leq q_p \nu_{ed}$, as $\mathcal{Q}_{\nu_{ed}}$ is a ν_{ed}-useful transcript set. The result follows from these facts and the application of Lemma 5.1.

6.3 Desired Properties from Ψ and Φ' Matrices

Theorem 6.1 sheds some light on the properties required from Ψ and Φ' in order to get a secure gCOMET instance. Specifically, in a secure gCOMET instance we must have:

- *Large period for Ψ matrix*: Let ℓ denote the maximum permissible message length. For any $i > j \in (\ell]$, and some non zero $Z \in \{0,1\}^\kappa$, we want to avoid $\mathsf{U}^i(Z) = \mathsf{U}^j(Z)$. In words, this roughly translates to key repetition within an encryption/decryption query. We can rewrite it as $\mathsf{U}^{i-j} = \mathsf{I}$. Clearly, if $\mathsf{cycle}(\mathsf{U}) \geq \ell$, then we are done. Now, due to the nature of U, we have $\mathsf{cycle}(\mathsf{U}) = \mathsf{cycle}(\Psi)$. Hence, the property $\mathsf{cycle}(\Psi) \geq \ell$ helps in avoiding key repetitions within a query.
- *Small value for $\mathfrak{c}'$:* As evident from Theorem 6.1, the value of $\mathfrak{c}'$ directly affects the security bound, as $\mathsf{rank}(\Psi) = \kappa - \mathfrak{c}'$. In other words, smaller the value of $\mathfrak{c}'$, higher the rank of Ψ, which directly translates to better security guarantee for gCOMET.
- *High rank for Φ' matrix*: In decryption phase, the rank of Φ' function quantifies the effect of the previous block cipher output on the next block cipher input. For example, if $\Phi' = 0$ (possible when $\Phi = \mathsf{I}$), the next input is independent of previous output. In other words, the adversary can fully control the next input. In particular, the adversary can collide the input of a large number of blocks. This can be verified from Theorem 6.1 as well, where some multicollision bounds are inversely proportional to $\mathsf{rank}(\Phi')$.

7 Instantiating gCOMET

For any $S \in \{0,1\}^+$ and $s \in (|S|]$, $S \ggg s$ denotes the "circular right shift by s" operation on S. The set $\{0,1\}^{\kappa-\mathfrak{c}'}$ can be viewed as the Galois field $\mathrm{GF}(2^{\kappa-\mathfrak{c}'})$ consisting of $2^{\kappa-\mathfrak{c}'}$ elements. Let $f(x)$ denote the primitive polynomial used to represent the field $\mathrm{GF}(2^{\kappa-\mathfrak{c}'})$, and α_f denote a fixed primitive element in this representation. The set $\{0,1\}^{\kappa-\mathfrak{c}'}$ can also be viewed as a $(\kappa - \mathfrak{c}')$-dimensional vector space over GF(2). In this context, α_f can be viewed as an invertible linear map over $\{0,1\}^{\kappa-\mathfrak{c}'}$. By a slight abuse of notation, we denote the binary matrix associated with α_f by α_f itself. It is well-known that $\mathsf{cycle}(\alpha_f) = 2^{\kappa-\mathfrak{c}'} - 1$.

7.1 COMETv1 and Its Security

The NIST LwC candidate COMET, hereafter referred as COMETv1, can be easily obtained from gCOMET in the following manner:

- Key size, κ is set to 128.

Algorithm 7.1 Control sequence generator for COMETv1 (left) and COMETv2 (right).

```
1: function $\Delta(A, I)$
2:     $a \leftarrow \left\lceil \frac{|A|}{n} \right\rceil$, $m \leftarrow \left\lceil \frac{|I|}{n} \right\rceil$, $\ell := a + m$
3:     $\delta^{\ell+2} \leftarrow (0^5)^{\ell+2}$
4:     if $a \neq 0$ then
5:         $\delta_0 \leftarrow \delta_0 \oplus 00001$
6:         if $n \nmid |A|$ then $\delta_{a-1} \leftarrow \delta_{a-1} \oplus 00010$
7:     if $m \neq 0$ then
8:         $\delta_a \leftarrow \delta_a \oplus 00100$
9:         if $n \nmid |I|$ then $\delta_{\ell-1} \leftarrow \delta_{\ell-1} \oplus 01000$
10:    $\delta_{\ell+1} \leftarrow \delta_{\ell+1} \oplus 10000$
11:    return $(a, m, \ell, \delta^{\ell+2})$
```

```
1: function $\Delta(A, I)$
2:     $a \leftarrow \left\lceil \frac{|A|}{n} \right\rceil$, $m \leftarrow \left\lceil \frac{|I|}{n} \right\rceil$, $\ell := a + m$
3:     $\delta^{\ell+2} \leftarrow (0^5)^{\ell+2}$
4:     if $a \neq 0$ then
5:         $\delta_1 \leftarrow \delta_1 \oplus 00001$
6:         if $n \nmid |A|$ then $\delta_a \leftarrow \delta_a \oplus 00010$
7:     if $m \neq 0$ then
8:         $\delta_{a+1} \leftarrow \delta_{a+1} \oplus 00100$
9:         if $n \nmid |I|$ then $\delta_\ell \leftarrow \delta_\ell \oplus 01000$
10:    $\delta_{\ell+1} \leftarrow \delta_{\ell+1} \oplus 10000$
11:    return $(a, m, \ell, \delta^{\ell+2})$
```

- Block size, n is set to 128 and 64 in fatCOMETv1 and tinyCOMETv1, respectively.
- The control size $\mathfrak{c}$ is set to 5 and the invariant-prefix size $\mathfrak{c}'$ is set to $\kappa/2 = 64$.
- Δ is defined in Algorithm 7.1 (left).
- Φ is defined by the mapping $(X_3, X_2, X_1, X_0) \longmapsto X_1 \| X_0 \| (X_2 \ggg 1) \| X_3$, where $(X_3, X_2, X_1, X_0) \overset{n/4}{\longleftarrow} X$. One can verify that $\mathsf{rank}(\Phi') = n - 1$.
- The Ψ function is defined as the binary matrix α_f, where α_f denotes the primitive element of $\mathrm{GF}(2^{64})$ with respect to $f(x) = x^{64} + x^4 + x^3 + x + 1$.

In Corollary 7.1, we apply Theorem 6.1 and relevant multicollision bounds from Propositions 4.1-4.4, to obtain security bounds for fatCOMETv1 and tinyCOMETv1.

Corollary 7.1. *For $n \geq 4$, $q_p < 2^{126}$, and any $(q_e, q_d, \sigma_e, \sigma_d, q_p)$-adversary $\mathscr{A}$, we have*

1. *For $\sigma_c < 2^{64}$, and $\nu_{ed} = \frac{2^{55}}{\sqrt{11}}$:*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{fatCOMETv1}}(\mathscr{A}) \leq \frac{q_p}{2^{125.19}} + \frac{\sigma_c}{2^{59.75}} + \frac{\sigma_d \sigma_e}{2^{120.8}} + \frac{q_p \sigma_c}{2^{180.24}}.$$

2. *For $\sigma_c < 2^{39}$, and $\nu_{ed} = \frac{2^{24}}{\sqrt{11}}$:*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{tinyCOMETv1}}(\mathscr{A}) \leq \frac{q_p}{2^{121.58}} + \frac{\sigma_c}{2^{55.98}} + \frac{\sigma_d \sigma_e}{2^{126}} + \frac{q_p \sigma_d}{2^{149.24}} + \frac{q_p \sigma_e \sigma_d}{2^{188.68}}.$$

Corollary 7.1 clearly shows that fatCOMETv1 (or the NIST submission COMET-128) is secure while $\sigma_c < 2^{63.75}$ bytes[2] (data complexity), $q_p < 2^{125.19}$ (time complexity), and $q_p \sigma_c < 2^{184.24}$ (data-time trade-off). Similarly, under the assumption that $\sigma_c < 2^{42}$ bytes[3] (data complexity), tinyCOMETv1 (or the NIST submis-

[2] Each block of fatCOMETv1 is built of 16 bytes.
[3] Each block of tinyCOMETv1 is built of 8 bytes.

sion COMET-64) is secure while $q_p < 2^{112}$ (time complexity) and $q_p\sigma_c < 2^{152.24}$ (data-time trade-off).

In supplementary material E, we summarize the two known cryptanalytic works [12,14] on COMETv1. Although these works are largely inconsequential in relation to the validity of COMETv1's security claims, they show that large value of $\mathfrak{c}'$ can lead to a large class of weak keys. We observe that the value of $\mathfrak{c}'$ can be reduced significantly without much degradation in performance. Particularly, we observe that the Ψ function can be defined over a larger field which avoids the above given strategies. In fact, a similar remedy has been also offered in [12].

7.2 COMETv2 and its Security

We describe a variant of COMETv1, called COMETv2, that differs in the following components:

- The control size $\mathfrak{c}$ is set to 5 and the invariant-prefix size $\mathfrak{c}'$ is set to 8.
- The Δ function is defined in Algorithm 7.1 (right).
- The Ψ function is defined as the binary matrix α_f, where α_f denotes the primitive element of $\mathrm{GF}(2^{120})$ with respect to $f(x) = x^{120} + x^9 + x^6 + x^2 + 1$.

From the above discussion, it is clear that COMETv2 differs from COMETv1 in just two components, namely Δ and Ψ functions. The modified Δ function helps in reducing the hardware footprint as the earlier version required an additional n-bit of memory. Further, the strategies from [12,14] have significantly higher data/time complexity against COMETv2 due to the small value of $\mathfrak{c}'$ and the updated Ψ function.

In Corollary 7.2, we apply Theorem 6.1 and relevant multicollision bounds from Propositions 4.1-4.4, to obtain security bounds for fatCOMETv2 and tinyCOMETv2.

Corollary 7.2. *For $n \geq 4$, $q_p < 2^{126}$, and any $(q_e, q_d, \sigma_e, \sigma_d, q_p)$-adversary $\mathscr{A}$, we have*

1. *For $\sigma_c < 2^{64}$, and $\nu_{ed} = \frac{2^{55}}{\sqrt{11}}$:*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{fatCOMETv2}}(\mathscr{A}) \leq \frac{q_p}{2^{125.19}} + \frac{\sigma_c}{2^{115.62}} + \frac{\sigma_d\sigma_e}{2^{120}} + \frac{q_p\sigma_c}{2^{180.24}}.$$

2. *For $\sigma_c < 2^{62}$, and $\nu_{ed} = \frac{2^{24}}{\sqrt{11}}$:*

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathsf{tinyCOMETv2}}(\mathscr{A}) \leq \frac{q_p}{2^{121.58}} + \frac{\sigma_c}{2^{63}} + \frac{\sigma_d\sigma_e}{2^{120}} + \frac{q_p\sigma_d}{2^{149.24}}.$$

On the Benefits of fatCOMETv2 over fatCOMETv1: Note that the advantage expressions for the two versions look similar. However, fatCOMETv2 has subtle advantages over fatCOMETv1. For instance, when we restrict $q_p < 2^{119}$ (NIST prescribed), the dominating terms are

- for v1: $\sigma_d\sigma_e/2^{120} + \sigma_c/2^{59}$
- for v2: $\sigma_d\sigma_e/2^{120}$

In fact, the additional term $\sigma_c/2^{59}$ for v1, is not just an artifact of the proof. Indeed, the previous works by Khairallah [12] and Bernstein et al. [14] (although violate the designers' prescribed limits) achieve a lower bound which almost matches this term using encryption queries only. On the other hand, our security proofs guarantee that even such strategies do not work against v2. Clearly, when $\sigma_e \approx 2^{60}$, $\sigma_d \ll 2^{60}$ and $q_p \ll 2^{119}$, v2 has much better security than v1. This is an improved security feature of v2, in addition to the fact that it has obvious implementation advantages. Note that, for $q_p > 2^{119}$ or $\sigma_e, \sigma_d \approx 2^{60}$, the two versions enjoy similar security guarantees.

8 Conclusion

In this paper, we proposed a generalization of the COMET mode of operation, called gCOMET, and gave a detailed security proof of gCOMET. We view COMET as an instance of gCOMET and derive its security bounds. Finally, we propose a refinement of COMET, called COMETv2, that seems to have better performance and security as compared to COMET. We note that our security proofs are not complemented with matching attacks, and it is possible that the security bounds can be improved, particularly for the COMET-64 versions.

Acknowledgments. Shay Gueron is supported by The Israel Science Foundation (grants No. 1018/16 and 3380/19); NSF-BSF Grant 2018640; The BIU Center for Research in Applied Cryptography and Cyber Security and the Center for Cyber Law and Policy at the University of Haifa, both in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office. Ashwin Jha's work was carried out in the framework of the French-German-Center for Cybersecurity, a collaboration of CISPA and LORIA. Mridul Nandi is supported by the project "Study and Analysis of IoT Security" under Government of India at R. C. Bose Centre for Cryptology and Security, Indian Statistical Institute, Kolkata.

References

1. NIST: Lightweight cryptography (2018) Online: https://csrc.nist.gov/Projects/Lightweight-Cryptography. Accessed: August 31, 2020.
2. Gueron, S., Jha, A., Nandi, M.: COMET: Counter mode encryption with tag. Submission to NIST LwC Standardization Process (Round 1) (2019) Online: https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/comet-spec.pdf. Access: June 26, 2020.
3. Gueron, S., Jha, A., Nandi, M.: COMET: Counter mode encryption with tag. Submission to NIST LwC Standardization Process (Round 2) (2020) Online: https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/comet-spec-round2.pdf. Access: June 26, 2020.
4. Chakraborti, A., Iwata, T., Minematsu, K., Nandi, M.: Blockcipher-based authenticated encryption: How small can we go? In: Cryptographic Hardware and Embedded Systems - CHES 2017. Proceedings. (2017) 277–298

5. Chakraborti, A., Datta, N., Nandi, M., Yasuda, K.: Beetle Family of Lightweight and Secure Authenticated Encryption Ciphers. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2018**(2) (2018) 218–241
6. Dworkin, M.: Recommendation for Block Cipher Modes of Operation – Methods and Techniques. NIST Special Publication 800-38A, National Institute of Standards and Technology, U. S. Department of Commerce (2001)
7. NIST: Announcing the ADVANCED ENCRYPTION STANDARD (AES). Fedral Information Processing Standards Publication FIPS 197, National Institute of Standards and Technology, U. S. Department of Commerce (2001)
8. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK Lightweight Block Ciphers. In: 52nd Annual Design Automation Conference. Proceedings. (2015) 175:1–175:6
9. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: SIMON and SPECK: Block Ciphers for the Internet of Things. IACR Cryptology ePrint Archive **2015** (2015) 585
10. Koo, B., Roh, D., Kim, H., Jung, Y., Lee, D., Kwon, D.: CHAM: A family of lightweight block ciphers for resource-constrained devices. In: Information Security and Cryptology - ICISC 2017, Revised Selected Papers. (2017) 3–25
11. Weatherley, R.: Performance of aead algorithms on AVR (2020) Online: https://rweather.github.io/lightweight-crypto/performance_avr.html#perf_avr_overall. Accessed: September 14, 2020.
12. Khairallah, M.: Weak keys in the rekeying paradigm: Application to COMET and mixfeed. IACR Trans. Symmetric Cryptol. **2019**(4) (2019) 272–289
13. Gueron, S., Jha, A., Nandi, M.: On the security of COMET authenticated encryption scheme. Presented at NIST Lightweight Cryptography Workshop 2019 (2019) Online: https://csrc.nist.gov/CSRC/media/Presentations/on-the-security-of-comet-authenticated-encryption/images-media/session2-gueron-security-of-comet.pdf. Accessed: September 14, 2020.
14. Bernstein, D.J., Gilbert, H., Turan, M.S.: Observations on COMET. Personal communication (2020)
15. Chakraborty, B., Jha, A., Nandi, M.: On the security of sponge-type authenticated encryption modes. IACR Trans. Symmetric Cryptol. **2020**(2) (2020) 93–119
16. Hoang, V.T., Tessaro, S.: Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In: Advances in Cryptology - CRYPTO '16. Proceedings, Part I. (2016) 3–32
17. Jha, A., Nandi, M.: Applications of h-technique: Revisiting symmetric key security analysis. IACR Cryptol. ePrint Arch. **2018** (2018) 1130
18. Patarin, J.: Etude de Générateurs de Permutations Basés sur les Schémas du DES. PhD thesis, Université de Paris (1991)
19. Patarin, J.: The "coefficients H" technique. In: Selected Areas in Cryptography - SAC '08. Revised Selected Papers. (2008) 328–345
20. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Duplexing the sponge: Single-pass authenticated encryption and other applications. In: Selected Areas in Cryptography - SAC 2011, Revised Selected Papers. (2011) 320–337
21. AlTawy, R., Gong, G., He, M., Jha, A., Mandal, K., Nandi, M., Rohit, R.: SpoC: An authenticated cipher submission to the nist lwc competition. Submission to NIST LwC Standardization Process (Round 2) (2019) Online: https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-2/spec-doc-rnd2/spoc-spec-round2.pdf. Access: July 09, 2020.

22. Chakraborty, B., Jha, A., Nandi, M.: On the security of sponge-type authenticated encryption modes. IACR Cryptol. ePrint Arch. **2019** (2019) 1475
23. Gueron, S., Jha, A., Nandi, M.: Revisiting the Security of COMET Authenticated Encryption Scheme. IACR Cryptology ePrint Archive **2021** (2021)

Supplementary Material

A Proofs Related to the Multicollision Results

A.1 Proof of Proposition 4.1

For any integer $\rho \geq 2$, we have

$$\Pr\left[\Theta_{q,n} \geq \rho\right] \leq 2^n \times \left(\frac{qe}{\rho 2^n}\right)^{\rho}. \tag{10}$$

See [15] for a justification for Eq. (10). Using Eq. (3) and (10), we get Proposition 4.1 by plugging in some suitable value for ρ.

It is sufficient to upper bound $q2^n \times \left(\frac{qe}{\rho 2^n}\right)^{\rho}$. Consider $q \leq 2^{\frac{n}{2}}$. Taking $\rho = 3$, we have

$$q2^n \times \left(\frac{qe}{\rho 2^n}\right)^{\rho} \leq 2^{\frac{3n}{2}} \times \left(\frac{1}{2^{\frac{n}{2}}}\right)^3 \leq 1.$$

Consider $2^{\frac{n}{2}} < q \leq 2^n$. Taking $\rho = \frac{4n}{\log_2 n}$, we have

$$q2^n \times \left(\frac{qe}{\rho 2^n}\right)^{\rho} \leq 2^{2n} \times \left(\frac{1}{\sqrt{n}}\right)^{\frac{4n}{\log_2 n}} \leq 1,$$

where the first inequality follows from $n \geq 2$. Consider $2^n < q \leq n2^n$. Taking $\rho = 4n$, we have

$$q2^n \times \left(\frac{qe}{\rho 2^n}\right)^{\rho} \leq n2^{2n} \times \left(\frac{e}{4}\right)^{4n} \leq n.$$

Now, using Eq. (3), we get the desired bound for $q \leq n2^n$. For $q \geq n2^n$, we divide the queries into groups of size $n2^n$ (add additional queries, if required), and apply the above argument to each group. The result follows by accumulating the bounds for all the groups. □

A.2 Proof of Proposition 4.3

For $\rho \geq 2$, we have

$$\begin{aligned}\Pr\left[\widehat{\Theta}'_{q,n,r} \geq \rho\right] &\leq \sum_{a \in \{0,1\}^n} \Pr\left[|\{i : \mathsf{L}(\mathsf{X}_i) = a\}| \geq \rho\right] \\ &\overset{(*)}{\leq} 2^n \times \frac{\binom{q}{\rho}}{2^{r\rho}} \leq 2^n \times \frac{q^{\rho}}{2^{r\rho}\rho!} \overset{(**)}{\leq} 2^n \times \left(\frac{qe}{\rho 2^r}\right)^{\rho},\end{aligned} \tag{11}$$

where at inequality $(*)$ we use the fact that the number of solutions for $\mathsf{L}(\mathsf{X}_i) = a$ is bounded by at most 2^r since rank of L is r, and at inequality $(**)$ we use the simple observation that $e^\rho = \sum_{i \geq 0} \rho^i/i! \geq \rho^\rho/\rho!$.

Now, the proof is identical to the proof of Proposition 4.1, and follows from Eq. (3) and (11). $\square$

A.3 Proof of Proposition 4.4

For $\rho \geq 2$ and distinct $i_0, \ldots, i_{\rho-1} \in (q]$, first consider $\Pr\left[\mathsf{C}_{i_0} = a, \ldots, \mathsf{C}_{i_{\rho-1}} = a\right]$. For brevity, we write k for i_k for all $k \in (\rho]$. Without loss of generality, we can assume $a = 0^n$. Since otherwise, we can consider $\mathsf{IC}'_z(y) = \mathsf{IC}^+_z(y \oplus a)$, which is an equivalent problem if we consider $\bar{y}_i = y_i \oplus a$ instead of y_i for all $i \in (\rho]$. So, it is sufficient to consider

$$\Pr\left[\mathsf{C}_i = 0, \ldots, \mathsf{C}_i = 0 : i \in (q]\right] = \sum_{x^\rho} \Pr\left[\mathsf{IC}_{z_i}(y_i) = x_i, \mathsf{IC}_{z'_i}(\mathsf{L}(x_i)) = x'_i : i \in (q]\right].$$

For $i \in (q]$, let $y'_i = \mathsf{L}(x_i)$. We say that x^ρ is valid if $(z'_i, y'_i) = (z_j, y_j)$ if and only if $(z_j, x_j) = (z'_i, x'_i)$ for all $i, j(q]$. The set of all such valid tuples is denoted as V. For any valid x^ρ, we define

$$S := \{(z_i, y_i) : i \in (\rho]\} \cup \{(z'_i, y'_i) : i \in (\rho]\}.$$

Then, we have $\rho \leq |S| \leq 2\rho$. Suppose S contains $t \leq 2\rho$ many distinct keys $(\hat{z}_0, \ldots, \hat{z}_{t-1})$ and β_j denotes the number of occurrences of key $\hat{z}_j$ in some tuple in S. Then,

$$\Pr\left[\mathsf{IC}_{z_i}(y_i) = x_i, \mathsf{IC}_{z'_i}(y'_i) = x'_i : i \in (q]\right] = \frac{1}{\prod_{j \in (t]} (2^n)_{\beta_j}}.$$

On the other hand, the above probability is zero for an invalid x^ρ. Let V_s denote the number of valid tuples for which $|S| = s$.

We say that (z'_i, y'_i) is old if $(z'_i, y'_i) = (z_j, x_j)$ for some $i, j \in (\rho]$. If $|S| = 2\rho - k$, then we must have exactly k old y'_i values. The number of ways these k old y'_i values can be chosen is bounded by at most ρ^{2k}. The number of x_i values corresponding to old y'_i values is bounded by at most $\rho^{2k} 2^{k(n-r)}$, since for each y'_i there are at most 2^{n-r} choices for x_i such that $\mathsf{L}(x_i) = y'_i$.

Now, we have to choose the remaining x_i values corresponding to new y'_i values. We choose these values one at a time in lexicographic order, say $m_0, \ldots, m_{\rho-k}$. y'_{m_l} can be chosen in at most $(2^n - \gamma_{m_l})$, where γ_{m_l} denotes the number of previous indices (including the old ones) sharing the same key as m_l. By applying this to all the remaining indices, the number of ways to choose the remaining y'_i is $(2^n - \gamma_{m_0}) \cdots (2^n - \gamma_{m_{\rho-k}})$, whence the number of ways to choose the remaining x_i values is at most $2^{(\rho-k)(n-r)} (2^n - \gamma_{m_0}) \cdots (2^n - \gamma_{m_{\rho-k}})$. Hence,

$V_{2\rho-k} \le \rho^{2k} 2^{\rho(n-r)} \prod_{i=0}^{l} (2^n - \gamma_{m_i})$.

$$\begin{aligned}
\Pr\left[\mathsf{C}_0 = 0, \ldots, \mathsf{C}_{\rho-1} = 0\right] &= \sum_{s=\rho}^{2\rho} \sum_{x^\rho \in V_s} \Pr\left[\mathsf{IC}_{z_i}(y_i) = x_i, \mathsf{IC}_{z_i'}(y_i') = x_i' : i \in (\rho)\right] \\
&\le \sum_{k=0}^{\rho} \frac{|V_{2\rho-k}|}{\prod_{i\in(t]} (2^n)_{\beta_i}} \le \sum_{k=0}^{\rho} \frac{\rho^{2k} 2^{\rho(n-r)} \prod_{j\in(l]} (2^n - \gamma_{m_j})}{\prod_{i\in(t]} (2^n)_{\beta_i}} \\
&\le \sum_{k=0}^{\rho} \frac{\rho^{2(\rho-k)} 2^{\rho(n-r)}}{(2^n - 2\rho)^\rho} \le 2 \left(\frac{\rho^2 2^{n-r}}{2^n - 2\rho}\right)^\rho .
\end{aligned}$$

The number of ways we can choose the ρ indices is $\binom{q}{\rho}$, and the number of choices for a is 2^n. So, we have

$$\Pr\left[\Theta'_{q,n,r} \ge \rho\right] \le 2^{n+1} \left(\frac{qe\rho 2^{n-r}}{2^n - 2\rho}\right)^\rho . \tag{12}$$

Consider $q \le \frac{2^r}{22n}$. Taking $\rho = n$, we have

$$\begin{aligned}
q2^{n+1} \times \left(\frac{qe\rho 2^{n-r}}{2^n - 2\rho}\right)^\rho = q2^{n+1} \times \left(\frac{qen2^{n-r}}{2^n - 2n}\right)^n &\overset{(*)}{\le} q2^{n+1} \times \left(\frac{2qen}{2^r}\right)^n \\
&\overset{(**)}{\le} 2^{2n+1} \times \left(\frac{1}{4}\right)^n \le 2,
\end{aligned}$$

where we use $n \ge 4$ at $(*)$ and $(**)$ follows from the bound on q. Then, using Eq. (3), we have $\mu'(q,n,r) \le \rho + 1 < 2\rho$ for $q \le \frac{2^r}{22n}$. For $q > \frac{2^r}{22n}$, we apply the grouping argument to obtain the desired bound. □

B Proof of Lemma 5.1

We will bound $\mathsf{Ex}\left[\mathsf{W}_{t,k}(\mathcal{L})\right]$ in terms of four multicollision random variables defined below:
For $a \in \{0,1\}^n$,

1. let $\mathsf{W}^{\mathsf{bck},a} := \left|\{i : \widehat{\mathsf{d}}_i = 1 \wedge \widehat{\mathsf{Y}}_i = a\}\right|$.
2. let $\mathsf{W}^{\mathsf{fwd},a} := \left|\{i : \widehat{\mathsf{d}}_i = 0 \wedge \widehat{\mathsf{X}}_i = a\}\right|$, and $\mathsf{W}^{\mathsf{fwd}',a} := \left|\{i : \widehat{\mathsf{d}}_i = 0 \wedge \mathsf{L}(\widehat{\mathsf{X}}_i) = a\}\right|$.
3. let $\mathsf{W}^{\mathsf{mitm},a} := \left|\{\{i,j\} : \widehat{\mathsf{d}}_i = 1 \oplus \widehat{\mathsf{d}}_j \wedge \mathsf{L}(\widehat{\mathsf{X}}_i) \oplus \widehat{\mathsf{Y}}_j = a \wedge \mathsf{U}(\widehat{\mathsf{Z}}_i) = \widehat{\mathsf{Z}}_j\}\right|$.

We define: $\mathsf{W}^{\mathsf{bck}} := \max_a \mathsf{W}^{\mathsf{bck},a}$, $\mathsf{W}^{\mathsf{fwd}} := \max_a \mathsf{W}^{\mathsf{fwd},a}$, $\mathsf{W}^{\mathsf{fwd}'} := \max_a \mathsf{W}^{\mathsf{fwd}',a}$, and $\mathsf{W}^{\mathsf{mitm}} := \max_a \mathsf{W}^{\mathsf{mitm},a}$.
Now, we can divide the set of multi-chains into three sets:

– *Backward-only chains*: Each chain is constructed by IC^- queries only. By definition, the number of such chains is at most $\mathsf{W}^{\mathsf{bck}}$, as all the chains share a common source.

– *Forward-only chains*: Each chain is constructed by IC^+ queries only. Now, by definition of t-sink super-chain, we know that there are exactly t distinct sinks, out of which $t-1$ sinks can occur only in partial chains. By definition, the number of such chains is at most $(t-1)\cdot \mathsf{W}^{\mathsf{fwd}'}$. Further, the remaining sink occurs in complete chains. By definition, the number of such chains is at most $\mathsf{W}^{\mathsf{fwd}}$. In total, the number of forward-only chains is given by $(t-1)\mathsf{W}^{\mathsf{fwd}'}+\mathsf{W}^{\mathsf{fwd}}$.
– *Forward-backward chains*: Each chain is constructed by using both IC^+ and IC^- queries. Let us denote the number of such chains by $\mathsf{W}^{\mathsf{fwd\text{-}bck}}$.

Combining the three cases, we have

$$\mathsf{W}_{t,k}(\mathcal{L}) \le \mathsf{W}^{\mathsf{bck}} + (t-1)\mathsf{W}^{\mathsf{fwd}'} + \mathsf{W}^{\mathsf{fwd}} + \mathsf{W}^{\mathsf{fwd\text{-}bck}}.$$

We claim that $\mathsf{W}^{\mathsf{fwd\text{-}bck}} \le k\cdot\mathsf{W}^{\mathsf{mitm}}$. This can be shown using pigeonhole principle. Suppose $\mathsf{W}^{\mathsf{fwd\text{-}bck}} = N$. For each of the individual chain $\mathcal{C}$ constructed using both IC^+ and IC^- queries, we have at least one index $j \in (k]$ such that $\widehat{\mathsf{d}}_j = 1 \oplus \widehat{\mathsf{d}}_{j+1}$. Although the chains could be of different lengths, the preceding condition holds for any chain that contains both IC^+ and IC^- queries. We put the i-th chain in a bucket labeled j, if $\widehat{\mathsf{d}}_j = 1 \oplus \widehat{\mathsf{d}}_{j+1}$. As there are k buckets and N chains, by pigeonhole principle, we must have at least one[4] bucket $j \in (k]$, such that it holds at least $\lceil \frac{N}{k} \rceil$ chains. Thus, we have $\mathsf{W}^{\mathsf{fwd\text{-}bck}} \le k \cdot \mathsf{W}^{\mathsf{mitm}}$.

Using the preceding discussion and linearity of expectation, we have

$$\begin{aligned}\mathsf{Ex}\left[\mathsf{W}_{t,k}(\mathcal{L})\right] &\le \mathsf{Ex}\left[\mathsf{W}^{\mathsf{bck}}\right] + (t-1)\cdot \mathsf{Ex}\left[\mathsf{W}^{\mathsf{fwd}'}\right] + \mathsf{Ex}\left[\mathsf{W}^{\mathsf{fwd}}\right] + k\cdot \mathsf{Ex}\left[\mathsf{W}^{\mathsf{mitm}}\right] \\ &\le 2\widehat{\mu}(q,n) + (t-1)\cdot \widehat{\mu}'(q,n,\mathsf{rank}(\mathsf{L})) + k\cdot \mu'(q\nu,n,\mathsf{rank}(\mathsf{L})).\end{aligned}$$

Observe that $\mathsf{W}^{\mathsf{fwd}}$ and $\mathsf{W}^{\mathsf{bck}}$ correspond to $\widehat{\Theta}_{q,n}$ (see section 4.1), and $\mathsf{W}^{\mathsf{fwd}'}$ corresponds to $\widehat{\Theta}'_{q,n,\mathsf{rank}(\mathsf{L})}$ (see section 4.1). Further, using the fact that all the chains have distinct keys, we can conclude that $\mathsf{W}^{\mathsf{mitm}} \le \Theta'_{q\nu,n,\mathsf{rank}(\mathsf{L})}$. This justifies the last inequality. □

C Proof of Lemma 6.1

We have

$$\begin{aligned}\Pr\left[\neg\mathtt{Comp}\right] &= \Pr\left[\mathtt{Kcoll} \cup \mathtt{EEmatch} \cup \mathtt{EPmatch} \cup \mathtt{PKcount}\right] \\ &\le \Pr\left[\mathtt{Kcoll}\right] + \Pr\left[\mathtt{EEmatch} | \neg\mathtt{Kcoll}\right] \\ &\quad + \Pr\left[\mathtt{EPmatch} | \neg\mathtt{Kcoll}\right] + \Pr\left[\mathtt{PKcount}\right] \qquad (13)\end{aligned}$$

We bound the right hand side as follows:

[4] In fact, 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.

1. *Bounding* $\Pr[\mathtt{Kcoll}]$: First, $\Pr[\mathtt{B0}]$ is bounded to at most $q_p 2^{-\kappa}$ since K is sampled uniformly at random. Second, $\Pr[\mathtt{B1}]$ is bounded to at most $\sigma_e 2^{-\kappa}$. This can be argued as follows: fix an encryption query index i. Consider two cases, pertaining to the two variants of gCOMET, namely $\mathsf{fatCOMET}$ and $\mathsf{tinyCOMET}$:
   - For $\mathsf{fatCOMET}$: We have the equation $\mathsf{K} = \mathsf{Z}^i_j = \mathsf{U}^{j+1}\left(\mathsf{IC}^+_{\mathsf{K}}(\mathsf{N}^i)\right)$. Conditioned on some arbitrary choice $\mathsf{K} = K$, the preceding equation holds with $2^{-\kappa}$ probability as IC^+_K is a random permutation. Since the conditional probability is independent of the choice of K, the joint event holds with identical probability.
   - For $\mathsf{tinyCOMET}$: We have the equation $\mathsf{K} = \mathsf{U}^{j+1}\left(\mathsf{K} \oplus \mathsf{N}^i\right)$. Since U is linear and K is sampled uniformly at random, the preceding equation holds with at most $2^{-\kappa}$ probability.

   So, for a fixed encryption query block, the probability is at most $2^{-\kappa}$. Summing over all choices we get the desired bound. Similarly, $\Pr[\mathtt{B2}]$ is bounded by at most $\sigma_d 2^{-\kappa}$. Following a similar line of argument as used in case of $\mathtt{B1}$, we also bound $\Pr[\mathtt{B3}]$ and $\Pr[\mathtt{B4}]$ to $q_e 2^{\mathfrak{c}'-\kappa}$ and $q_d 2^{\mathfrak{c}'-\kappa}$, respectively. Using union bound, we have

$$\Pr[\mathtt{Kcoll}] \leq \frac{q_p + \sigma_c}{2^\kappa} + \frac{q_c}{2^{\kappa-\mathfrak{c}'}}. \tag{14}$$

2. *Bounding* $\Pr[\mathtt{EEmatch}|\neg\mathtt{Kcoll}]$: We will bound $\Pr[\mathtt{B5}]$, while $\Pr[\mathtt{B6}]$ can be bounded in a similar fashion. Fix $(i,j) \neq (i',j')$. We must have $\mathsf{Z}^i_j = \mathsf{Z}^{i'}_{j'}$. Now, $i = i'$ and $\mathsf{Z}^i_j = \mathsf{Z}^{i'}_{j'}$ implies $\mathsf{U}^{j'-j} = \mathsf{I}_{\kappa-\mathfrak{c}'}$ (since $\neg(\mathtt{B3} \cup \mathtt{B4})$ holds). But this is not possible, due to the assumption that $j' \leq \sigma_c < \min\{N, 2^{n-2}\}$. Hence, $i \neq i'$. Now, we consider two cases:
   - For $\mathsf{fatCOMET}$: We have the equation $\mathsf{U}^{j+1}\left(\mathsf{IC}_{\mathsf{K}}(\mathsf{N}^i)\right) = \mathsf{U}^{j'+1}\left(\mathsf{IC}_{\mathsf{K}}(\mathsf{N}^{i'})\right)$, which holds with at most $1/(2^\kappa - 1)$ probability.
   - For $\mathsf{tinyCOMET}$: We have the equation $\mathsf{U}^{j+1}\left(\mathsf{K} \oplus \mathsf{N}^i\right) = \mathsf{U}^{j'+1}\left(\mathsf{K} \oplus \mathsf{N}^{i'}\right)$, which holds with at most $1/2^{\kappa-\mathfrak{c}'-1}$ probability (since $\neg(\mathtt{B3} \cup \mathtt{B4})$ holds).

   Thus, for a fixed pair of encryption query blocks, the probability is at most $1/2^{\kappa-\mathfrak{c}'-1}$. Now, for a fixed (i,j) we have at most θ^b_e choices for (i',j'). Summing over all choices we get an upper bound of $2\sigma_e\theta^b_e/2^{\kappa-\mathfrak{c}'}$. Finally, we have

$$\Pr[\mathtt{EEmatch}|\neg\mathtt{Kcoll}] \leq \frac{2\sigma_e(\theta^b_e + \theta^f_e)}{2^{\kappa-\mathfrak{c}'}}. \tag{15}$$

3. *Bounding* $\Pr[\mathtt{EPmatch}|\neg\mathtt{Kcoll}]$: We will bound $\Pr[\mathtt{B7}]$ here, while $\Pr[\mathtt{B8}]$ can be bounded similarly. For fixed (i,j) and i' the event holds with at most $2^{-\kappa}$ probability. This can be argued as in the previous cases. So, we have

$$\Pr[\mathtt{EPmatch}|\neg\mathtt{Kcoll}] \leq \frac{q_p(\theta^b_e + \theta^f_e)}{2^\kappa}. \tag{16}$$

4. *Bounding* $\Pr[\mathtt{PKcount}]$: We consider $\Pr[\mathtt{B9}]$, whereas $\Pr[\mathtt{B10}]$ is bounded similarly. $\mathtt{B9}$ accounts for the possibility of exhausting a particular encryption

block key via primitive queries. Let $\mathcal{L}$ denote the set of all primitive queries which are repeated at least ν_{ed} times. Then, we must have $|\mathcal{L}| \leq q_p/\nu_{ed}$. Thus, some encryption block key falls in $\mathcal{L}$ with at most $q_p/\nu_{ed}2^{\kappa}$ probability, whence we have

$$\Pr\left[\mathtt{PKcount}\right] \leq \frac{q_p\sigma_c}{\nu_{ed}2^{\kappa}}. \tag{17}$$

The result follows by accumulating bounds from Eq. (14)-(17) in Eq. (13).

D Proof of Lemma 6.2

Let $\mathtt{R}_d^i$ denote the output of the i-th decryption attempt in the real world. Then, we have

$$\Pr\left[\mathtt{R}_d \neq \omega_d \mid \mathtt{E}\right] \leq \sum_{i\in(q_d]} \Pr\left[\mathtt{R}_d^i \neq \bot \mid \mathtt{E}\right]. \tag{18}$$

Fix a decryption query index i. Now, there are two situations that lead to $\mathtt{R}_d^i \neq \bot$.

Forgery due to chains: First, suppose the i-th decryption query is completely determined via the primitive and encryption queries. In other words, one can construct a chain using primitive and encryption query blocks, such that a suffix of the decryption query matches with this chain. Let p_i' denote the largest block index such that $(\bar{\mathsf{Z}}^i_{p_i+1}, \bar{\mathsf{Y}}^i_{p_i+1}), \ldots, (\bar{\mathsf{Z}}^i_{p_i'}, \bar{\mathsf{Y}}^i_{p_i'})$ is in ω_p. If $(\bar{\mathsf{Z}}^i_{p_i+1}, \bar{\mathsf{Y}}^i_{p_i+1}) \notin \omega_p$, then $p_i' = p_i$. A forgery in this case implies that one of the following events occur:

$\mathtt{B11}:$ $\exists(i',j') \in (q_e] \times (m^{i'}+1]$, such that

$$(\bar{\mathsf{N}}^i, p_i+1) \neq (\mathsf{N}^{i'}, j') \text{ and } (\bar{\mathsf{Z}}^i_{p_i+1}, \bar{\mathsf{Y}}^i_{p_i+1}) = (\mathsf{Z}^{i'}_{j'}, \mathsf{Y}^{i'}_{j'}).$$

$\mathtt{B12}:$ $0 \leq p_i < p_i' = \bar{\ell}_i$ and $\bar{\mathsf{X}}^i_{\bar{\ell}^i} = \bar{\mathsf{T}}^i$.

$\mathtt{B13}:$ $\exists(i',j') \in (q_e] \times (\ell^{i'}+1]$, such that

$$0 \leq p_i < p_i' < \bar{\ell}^i \text{ and } (\bar{\mathsf{Z}}^i_{p_i'+1}, \bar{\mathsf{Y}}^i_{p_i'+1}) = (\mathsf{Z}^{i'}_{j'}, \mathsf{Y}^{i'}_{j'}).$$

Let $\mathtt{Chain} := \mathtt{B11} \cup \mathtt{B12} \cup \mathtt{B13}$. First, we upper bound $\Pr\left[\mathtt{B11}|\mathtt{E}\right]$. We consider two cases:

- Case 1: $p_i = -1$ or $j = 0$. Note that, this condition is exclusive in nature, i.e., $p_i = -1$ and $j = 0$ is not possible (since initialization is injective for distinct tweaks). Using previously used arguments, we can upper bound the probability in this case to $2(\sigma_e + q_e)/2^{\kappa-\mathfrak{c}'+n}$.
- Case 2: $p_i \geq 0$ and $j > 0$. In this case, we have at most θ_e^b choices for (i',j') and the probability for each choice is bounded by at most $2/2^{\kappa-\mathfrak{c}'}$, resulting in an upper bound of $2\theta_e^b/2^{\kappa-\mathfrak{c}'}$.

On combining the two cases, we have

$$\Pr\left[\mathtt{B11}|\mathtt{E}\right] \leq \frac{2(\sigma_e+q_e)}{2^{\kappa-\mathfrak{c}'+n}} + \frac{2\theta_e^b}{2^{\kappa-\mathfrak{c}'}}.$$

We are left with $\Pr[\mathtt{B12} \cup \mathtt{B13} | \neg\mathtt{B11} \wedge \mathtt{E}]$. We use the super chain structure (see section 5) to bound this probability. Note that, the chains constructed via $\mathcal{Q}_{\nu_{ed}}$ (see Definition 6.1), are the only chains that can match with some decryption query (since $\neg\mathtt{PKcount}$ holds).

Now, $\mathtt{B12} \cup \mathtt{B13}$ implies that the decryption query is completed via a chain with starting node $(\bar{\mathsf{Z}}^i_{p_i+1}, \bar{\mathsf{Y}}^i_{p_i+1})$ and any prefix of $(\bar{\mathsf{C}}^i_{p_i+1}, \ldots, \bar{\mathsf{C}}^i_{\bar{m}^i})$ as label. Recall, from section 5.1, that the number of such chains is upper bounded by $\mathsf{W}_{t,k}(\mathcal{Q}_{\nu_{ed}})$, where t denotes the number of distinct sinks (see section 5) and k denotes the length of the longest possible chain, i.e., $k \leq \bar{\ell}^i - p_i$. In order to bound t, we can use one of the two approaches:

1. A trivial bound on t is at most $(\bar{\ell}^i - p_i)\sigma_e$ since any useful partial chain must end in a sink that collides with some encryption query block, and there are at most σ_e such blocks. Now, using the randomness of $\bar{\mathsf{Z}}^i_{p_i+1}$, we have

$$\Pr[\mathtt{B12} \cup \mathtt{B13} | \neg\mathtt{B11} \wedge \mathtt{E}] \leq \frac{2\mathsf{W}_{(\bar{\ell}^i - p_i)\sigma_e, (\bar{\ell}^i - p_i)}(\mathcal{Q}_{\nu_{ed}})}{2^\kappa},$$

2. In case $\kappa - \mathfrak{c}'$ is sufficient enough, we can bound t to at most $(\bar{\ell}^i - p_i)$ conditioned on another auxiliary event. Let

$$\mathtt{B5}': \quad \exists j \in (\bar{\ell}^i], (i', j') \in (q_e] \times (\ell^i + 1], \text{ such that } (\bar{\mathsf{N}}^i, j) \neq (\mathsf{N}^{i'}, j') \text{ and } \bar{\mathsf{Z}}^i_j = \mathsf{Z}^{i'}_{j'}.$$

It can be easily seen that conditioned on the event $\neg\mathtt{B5}'$, t is bounded to at most $(\bar{\ell}^i - p_i)$, since now any useful partial chain must end in a sink that collides with a unique encryption query block. Further, $\Pr[\mathtt{B5}' | \neg\mathtt{B11} \wedge \mathtt{E}] \leq \bar{\ell}^i \sigma_e 2^{\mathfrak{c}' - \kappa + 1}$. So, we have

$$\Pr[\mathtt{B12} \cup \mathtt{B13} | \neg\mathtt{B11} \wedge \mathtt{E}] \leq \frac{2\bar{\ell}^i \sigma_e}{2^{\kappa - \mathfrak{c}'}} + \frac{2\mathsf{W}_{(\bar{\ell}^i - p_i), (\bar{\ell}^i - p_i)}(\mathcal{Q}_{\nu_{ed}})}{2^\kappa},$$

Taking the minimum of the two bounds, we have

$$\Pr[\mathtt{Chain} | \mathtt{E}] \leq \frac{2(\sigma_e + q_e)}{2^{\kappa - \mathfrak{c}' + n}} + \frac{2\theta_e^b}{2^{\kappa - \mathfrak{c}'}} + \min\left\{ \frac{2\mathsf{W}_{\bar{\ell}^i \sigma_e, \bar{\ell}^i}(\mathcal{Q}_{\nu_{ed}})}{2^\kappa}, \frac{2\bar{\ell}^i \sigma_e}{2^{\kappa - \mathfrak{c}'}} + \frac{2\mathsf{W}_{\bar{\ell}^i, \bar{\ell}^i}(\mathcal{Q}_{\nu_{ed}})}{2^\kappa} \right\}. \tag{19}$$

Forgery due to guessing: Suppose $\neg\mathtt{Chain}$ happens, i.e., $(\bar{\mathsf{Z}}^i_{p'_i+1}, \bar{\mathsf{Y}}^i_{p'_i+1})$ is not in $\omega_p \cup \omega_e$. Then, $(\bar{\mathsf{Z}}^i_{p'_i+2}, \bar{\mathsf{Y}}^i_{p'_i+2})$ may collide with some primitive or encryption query block with probability at most $\frac{4q_p}{2^\kappa} + \frac{4\theta_e^b}{2^{\kappa - \mathfrak{c}'}}$. Applying this argument for all the successive blocks indices till the last one, we bound the probability that any one of them collide with some encryption or primitive query by $\frac{4q_p(\bar{\ell}^i - p'_i)}{2^{\kappa+n}} + \frac{4\theta_e^b(\bar{\ell}^i - p'_i)}{2^{\kappa - \mathfrak{c}' + n}}$. The conditional probability that the tag matches given that the tag generation input is fresh is bounded by $2/2^n$. Finally, we have

$$\Pr\left[\mathtt{R}^i_d \neq \bot | \neg\mathtt{Chain} \wedge \mathtt{E}\right] \leq \frac{4q_p \bar{\ell}^i}{2^{\kappa+n}} + \frac{4\theta_e^b \bar{\ell}^i}{2^{\kappa - \mathfrak{c}' + n}} + \frac{2}{2^n}. \tag{20}$$

The result follows by combining Eq. (18)-(20).

E Related Cryptanalytic Work on COMETv1

We briefly discuss two related cryptanalytic results on COMETv1. Although these results do not threaten the security claims of COMETv1, they show why the large value of $\mathfrak{c}'$ is not desirable. Further, their data and time complexity greatly endorse our conjecture on the security of tinyCOMETv1.

Khairallah's Work [12]: Khairallah [12] studied fatCOMETv1 under the weak key model. While the author also presents a multi-key analysis, here we only concentrate on the single-key analysis. The main observation behind Khairallah's work is based on the bad events B3 and B4 given in the proof of gCOMET (see section 6.2). Recall that,

$$\mathtt{B3}: \ \exists i \in (q_e], \text{ such that } \mathsf{Z}_0^i = *\|0^{\kappa-\mathfrak{c}'}.$$

If there exists an encryption query that satisfies B3, then all the block keys within this query will collide since Ψ applies on input value 0. This can be used to construct forgery and key recovery adversaries against fatCOMETv1. We remark that similar strategy also works against tinyCOMETv1. However, as evident from the proof of Lemma 6.1 (see supplementary material C), $\Pr[\mathtt{B3}] \leq q_c 2^{\kappa-\mathfrak{c}'}$. In other words, the adversary requires about 2^{68} bytes of data in order to get an appreciable advantage. However, the data complexity limit for fatCOMETv1 is capped at about 2^{63} bytes.

Observations of Bernstein, Gilbert and Turan [14]: In [13], Gueron et al. referred to a private communication in which Bernstein, Gilbert and Turan, proposed two observations on the security of tinyCOMETv1. The first observation builds upon Khairallah's work (and hence covered under B3), by constructing an encryption query only adversary. However, the data complexity of their adversary is worse than the previous one. Specifically, it requires about 2^{99} bytes of data. The second observation leads to a slide attack that tries to match two separate encryption query states (key and input). This strategy is covered under $\mathtt{EEmatch} = \mathtt{B5} \cup \mathtt{B6}$ (see section 6.2), and requires data complexity about 2^{68} bytes, which again goes beyond the data complexity limit of COMETv1.