On Post-Quantum Multi-Key Security of GCM

Akinori Hosoyamada[1,2]

[1] NTT Social Informatics Laboratories, Tokyo, Japan
[2] NTT Research Center for Theoretical Quantum Information, Atsugi, Japan
akinori.hosoyamada@ntt.com

Abstract. This paper studies the post-quantum multi-key security of Galois/Counter Mode (GCM) in the Quantum Ideal Cipher Model (QICM). GCM is one of the most widely used AEAD schemes. In practice, widely deployed cryptosystems are often instantiated under many independent keys, making the multi-key setting practically relevant. A trivial reduction from the multi-key setting to the single-key setting incurs a security loss proportional to the number of keys. In particular, in the post-quantum setting, the term corresponding to exhaustive key search becomes $up^2/2^k$, where u is the number of keys, k is the key length, and p is the number of quantum queries to the underlying block cipher E and its inverse, with p serving as a coarse measure of the amount of offline (quantum) computation performed by the adversary. For example, when $u = 2^{32}$, the term $up^2/2^k$ reaches constant order at $p = 2^{48}$ for $k = 128$ and at $p = 2^{80}$ for $k = 192$.

We show that, at the cost of additional loss terms, the u-dependent quantum key-search term $up^2/2^k$ can be replaced by a term of order $\sqrt{dp^2/2^k}$, where d denotes the maximum number of keys under which the same nonce appears in encryption queries. Thus, when d is much smaller than u and the additional loss terms remain small, our bound improves upon the trivial multi-key bound in that the security bound reaches constant order only at a substantially larger value of p. This can be viewed as a post-quantum counterpart of the classical result of Hoang et al. (CCS 2018). As in their work, for randomized nonce-generation mechanisms that abstract patterns used in protocols such as TLS, the parameter d remains small even when u is large. Although our bounds are not tight and leave room for improvement, they give non-trivial post-quantum multi-key guarantees for several concrete parameter settings. To the best of our knowledge, this is the first non-trivial post-quantum multi-key security bound for an AEAD mode in the QICM. Our proof combines the reprogramming-and-resampling approach of Alagic et al. (EUROCRYPT 2022) with counting arguments used in the classical multi-user analysis of GCM by Hoang et al.

Keywords: GCM · post-quantum security · multi-key security · quantum ideal cipher model.

1 Introduction

Multi-Key Security in Symmetric-Key Cryptography. The security of symmetric-key cryptographic schemes is often defined and analyzed in the single-key setting. In practice, however, the same symmetric-key primitive is deployed under many different keys, often across many users or sessions. It is therefore important to consider multi-key security, which is also commonly referred to as multi-user security [9,15]. The relevance of the multi-key setting is also recognized in standards and standards-related documents. NIST explicitly considers multi-key security in its recent specifications and evaluations of symmetric-key cryptography [69,55], while RFC 9771 identifies multi-user security as a relevant security property of AEAD schemes, listing AES-GCM [54] as an example and protocols such as TLS, IPsec, and SRTP among its applications [17].

Multi-key security is typically formulated by considering u independently keyed instances and declaring the adversary successful if it breaks any one of them. A trivial way to derive a multi-key security bound is to reduce multi-key security to single-key security. For example, if the single-key security advantage of a scheme is upper bounded by X, then its u-key security advantage is trivially upper bounded by uX.

Such a reduction, however, can incur an excessively large security loss. For example, (single-key) security bounds for block-cipher-based schemes typically contain a term of the form $\mathsf{Adv}^{\mathsf{prp}}_E + \cdots$, where $\mathsf{Adv}^{\mathsf{prp}}_E$ denotes the PRP advantage of the underlying block cipher E. If the key length of E is k-bit, the exhaustive key search with p evaluations of E or E^{-1} already achieves $\mathsf{Adv}^{\mathsf{prp}}_E \approx p/2^k$. The trivial multi-key reduction then yields a term of order $up/2^k$ in the security bound. For instance, when $k = 128$ and $u = 2^{32}$ keys are considered, the resulting bound no longer guarantees security once p reaches approximately 2^{96}.

In many practically deployed schemes, however, security does not necessarily degrade this rapidly as the number of keys increases. For GCM, for example, Bellare and Tackmann [11] and Hoang et al. [36] showed that nonce randomization can substantially improve multi-user security in the ideal-cipher model. In particular, Hoang et al. showed that the problematic factor u in the term $up/2^k$ can essentially be avoided, yielding a term much closer to the single-key term $p/2^k$. Specifically, they first showed that if each nonce is used under at most d keys, then a term containing the factor $p/2^k$, corresponding to exhaustive key search, can depend on d rather than on the total number u of keys: namely, the relevant term becomes $dp/2^k$ instead of $up/2^k$. When the nonces contain sufficient randomness, d can remain small even when u is large, resulting in a term close to $p/2^k$. The nonce-randomization analysis of Hoang et al. was later sharpened by Degabriele et al. [24]. Separately, Luykx et al. [51] showed that GCM as a mode incurs no multi-key security loss beyond that of the underlying block cipher.

These results are of significant practical importance. GCM is standardized by NIST [27] and ISO/IEC [42], and is used in widely deployed protocols and standards including TLS [63], QUIC [68], IPsec [70], SSH [39]. Moreover, several

protocols and standards employ nonce-generation mechanisms that provide the type of randomization relevant to the analyses in [11,36,24].

Post-Quantum Security. The effect of the trivial multi-key reduction remains significant in the post-quantum setting. Due to Grover's algorithm [31], the generic key-search term for the single-key PRP security of a block cipher deteriorates from $p/2^k$ in the classical setting to roughly $p^2/2^k$ in the post-quantum setting. If multi-key security is evaluated using the trivial reduction from single-key security, this immediately gives a term $up^2/2^k$. For example, taking $u = 2^{32}$ as an illustrative large-scale choice, the resulting bound no longer guarantees security for $p \geq 2^{48}$ when $k = 128$, and even when $k = 192$, it does not guarantee security for $p \geq 2^{80}$.

This latter threshold of $p = 2^{80}$ is already below the complexity $2^{256/3} \approx 2^{85.3}$ of BHT collision search [18] for a hash function with a 256-bit output. This contrast is particularly notable in light of post-quantum guidance from some countries and organizations recommending secret-key lengths of $k \geq 192$ alongside hash output lengths of at least 384 bits [1,25]: even at the recommended key length of $k = 192$, the trivial multi-key bound for[3] $u = 2^{32}$ ceases to guarantee security before reaching the query complexity of generic collision search on a 256-bit-output hash function, whereas the same guidance recommends at least 384-bit output lengths for hash functions. Tighter multi-key security bounds can be practically useful, especially if they alleviate such a mismatch.

In the classical setting, the non-trivial multi-key security results for GCM by Hoang et al. [36] are proved in the classical ideal-cipher model. They do not automatically extend to the post-quantum setting, where an adversary may make quantum queries to the underlying block cipher. Establishing a quantum analogue therefore requires an analysis in the Quantum Ideal Cipher Model (QICM) [38]. This leads us to the following question:

Can we avoid the term $up^2/2^k$ in a post-quantum multi-key security analysis of GCM, and obtain a security bound analogous to the classical bound of [36]?

1.1 Our Contributions

In this paper, we answer this question affirmatively. More specifically, we prove a bound in which the quantum key-search term, which would trivially be $up^2/2^k$, is replaced by a term of order $\sqrt{dp^2/2^k}$, at the cost of additional loss terms. This can be viewed as a post-quantum counterpart of the classical result of Hoang et al. [36]. To the best of our knowledge, our result gives the first non-trivial post-quantum multi-key security bound for an AEAD mode in the QICM.

[3] The scale $u = 2^{32}$ is not implausibly large for cloud-scale multi-key settings. For example, Campagna and Gueron, both affiliated with Amazon Web Services, discussed the cryptographic design of AWS KMS at Real World Crypto 2018 [19] and subsequently gave a detailed analysis [20] that considers target limits of even 2^{40} users, each associated with a master key.

We focus on the form of GCM in which nonces are mapped to block-cipher inputs by a key-independent injective encoding, rather than being processed through the keyed hash. This follows [36] and includes, in particular, the standard 96-bit-IV instantiation of AES-GCM.

Our proof uses the reprogramming-and-resampling approach, which was introduced by Alagic et al. [3] and has been applied to several symmetric-key constructions [4,37,8,65,5,66]. Previous applications of this approach to AEAD modes use it in a largely modular manner, with the quantum argument confined to the security of an underlying primitive or intermediate construction. For the Ascon AEAD mode [69], Hosoyamada [37] reduces the quantum part to the TPRP security of a tweakable Even-Mansour cipher [28]; once this cipher is replaced with a secret random object, the remaining AEAD-specific security is captured by a classical security bound. Similarly, Alagic et al. [5] derive single-key post-quantum security of GCM and other block-cipher modes through a general lifting argument that combines existing classical mode-specific security bounds with the post-quantum PRP or SPRP security of the underlying block cipher. In contrast, in our proof the quantum-specific argument is no longer confined to such a primitive-replacement step: reprogramming and resampling are applied directly within the privacy and authenticity games and are intertwined with mode-specific counting and forgery arguments for GCM. Directly incorporating reprogramming and resampling into the mode-specific privacy and authenticity analysis is essential for obtaining a bound depending on the nonce-overlap parameter d rather than the total number of keys u. Our result provides further evidence of the broad applicability of the reprogramming-and-resampling framework.

For privacy, we combine the reprogramming-and-resampling framework with counting arguments used in the classical H-coefficient proof [62] of Hoang et al. [36]. Using this combination, we prove the indistinguishability of the real encryption oracle from the ideal encryption oracle returning fresh random strings. The main technical challenge is to control collisions among the dynamically reprogrammed points while keeping all reprogramming operations mutually consistent, so that the relevant loss term depends on d rather than on u. Moreover, the hash-key values $E_K(0)$ must be left untouched, since naively reprogramming them would reintroduce a u-dependent quantum loss. For authenticity, we extend the privacy hybrid argument by recording verification queries and evaluating them later with respect to the reprogrammed ideal cipher, and then bound the probability that any such query is accepted.

Motivated by the nonce-randomization analyses of Hoang et al. [36] and Degabriele et al. [24], we also analyze security when nonces are randomized. In particular, we consider nonce-generation mechanisms based on adding a random salt to per-key distinct offsets (e.g., counter values) and concatenating a random salt with a nonce. The former (resp., latter) can be viewed as an abstraction of the nonce-generation patterns used in TLS 1.3 [63], QUIC [68], SRTP [53], and the successor-based instantiation of the RBG-based construction of NIST SP 800-38D [27] (resp., TLS 1.2 [64] and IPsec ESP [70]).

Although our security bounds are somewhat involved, evaluating them under concrete parameter settings illustrates a notable improvement over the trivial multi-key bound. Under the illustrative workload assumptions detailed in Section 7, using again the choice $u = 2^{32}$ considered above and setting $k = 192$, the p-dependent part of our bound reaches constant order at[4] $p \approx 2^{92.8}$, in contrast to $p \approx 2^{80}$ for the trivial multi-key bound. As another illustrative setting, when $u = 2^{40}$ and $k = 256$, the corresponding thresholds are $p \approx 2^{124.8}$ for our bound while $p \approx 2^{108}$ for the trivial multi-key bound.

Our security bounds are not tight and leave room for further improvement. We hope that this work motivates further study of post-quantum multi-key security for AEADs, as well as other schemes.

1.2 Related Works

Security of GCM (in the Single-Key Setting). Galois/Counter Mode (GCM) was introduced by McGrew and Viega together with its original security analysis [54]. Iwata et al. later identified a flaw in the proof, exhibited a distinguishing attack, and provided repaired privacy and authenticity bounds [44]; Niwa et al. subsequently tightened these bounds substantially [61]. The tightness of GCM's authenticity bounds and corresponding forgery attacks was further investigated by Luykx and Preneel [52] and Nandi [58]. GCM has also been studied under stronger robustness notions in, e.g., nonce-misuse and nonce-misuse resilience [47,7], authenticity under RUP [40], and committing security [26,22], to name a few.

Variants of GCM. Various variants of GCM have been proposed to improve its misuse resistance, concrete security, robustness, or nonce and key handling. GCM-SIV was introduced by Gueron and Lindell as an efficient nonce-misuse-resistant AEAD based on GCM-like components [33,32]; its security was further studied and strengthened by Iwata and Minematsu [43] and Iwata and Seurin [45], while Bose et al. established improved multi-user bounds for AES-GCM-SIV [16]. Chung et al. proposed eGCM and eGCM-SIV, which target almost full security while supporting longer nonces or nonce-misuse resistance [23]. Bhattacharya and Nandi proposed mGCM, a simple variant of GCM based on a variable-output-length pseudorandom function that provides stronger concrete security [12]. Bellare and Hoang proposed modifications of GCM and AES-GCM-SIV that additionally provide committing security [10]. GCM-SST, specified by Campagna et al. [21], is designed to provide stronger authentication security, in particular for truncated tags; its generic and multi-user security have been analyzed by Inoue et al. [41] and Naito et al. [56], respectively.

Randomized-nonce variants of GCM, including RGCM and XGCM, and their multi-user security were studied by Bellare and Tackmann [11] and Hoang et

[4] Here, this comparison refers to the quantum-query threshold at which the security upper bound becomes of constant order, rather than to a usage limit for any particular target advantage.

al. [36]. The nonce-randomization analysis of Hoang et al. was later sharpened by Degabriele et al. [24] using some improved combinatorial arguments. Another approach to improving the security bounds of GCM is nonce-based key derivation, as studied by Gueron and Lindell [34]. More recently, Gueron and Ristenpart proposed the derived-nonce, derived-key (DNDK) framework and its concrete instantiation DNDK-GCM, aiming to improve the secure scalability of GCM [35]. Naito et al. proposed nonce-based and tag-based key derivation (NTKD) and applied it to GCM and CCM to obtain improved multi-user security for large amounts of data [57]. Bhaumik et al. proposed ♯Pencil, domain-extended pseudorandom functions that can be used to strengthen GCM through key derivation while accommodating larger nonce domains [14].

Related directions have also been considered in the ongoing revision of NIST SP 800-38D. NIST has considered nonce-key-derivation proposals [59], and has subsequently proposed extending GCM to 256-bit-block ciphers under the name wGCM [60].

Our work instead studies the post-quantum multi-key security of standard GCM itself, without modifying the underlying mode, while additionally analyzing randomized nonce-generation mechanisms.

Post-Quantum Security of Symmetric-Key Encryption. We briefly review post-quantum security results most closely related to our work, focusing on symmetric-key encryption and authenticated encryption. Post-quantum provable security of symmetric-key schemes has been studied under several oracle models. Recent works have developed post-quantum security analyses for a variety of constructions in the so-called Q1 model [49], where the adversary has only classical access to the keyed construction, while quantum access to the underlying (possibly idealized) primitive is allowed [3,46,4,37,8,65,5,66]. In particular, Shiba and Iwata study post-quantum multi-key security of tweakable Even-Mansour, FX, and tweakable key-alternating Feistel ciphers [65,66].

A separate line of work studies security in the so-called Q2 model [49], which allows quantum superposition access to the construction itself. Anand et al. analyzed standard block-cipher modes against quantum adversaries [6], while Soukharev et al. studied post-quantum authenticated encryption more generally [67]. Kaplan et al. showed that modes including GMAC and GCM become insecure in such a setting [48], while QCB provides a positive construction designed for quantum superposition queries [13]. Lang and Lucks further studied GCM and other classical AE schemes under such attacks [50], and Fischlin and Schmalz recently proposed quantum-resistant variants of GCM and GCM-SIV achieving security in a fully quantum model [30].

Our result is established in the Q1 model, which we regard as the most practically relevant setting for conventional deployments where honest users perform classical computation and communicate over classical channels.

2 Preliminaries

For a bit string x, by $\mathsf{msb}_i(x)$ (resp., $\mathsf{lsb}_i(x)$) we denote the most significant i bits (resp., least significant i bits) of x, and $|x|$ denotes the bit length of x. We often identify n-bit strings x with integers such that $0 \leq x < 2^n$. When x and y are n-bit strings or integers such that $0 \leq x, y < 2^n$, the value $x + y$ is understood as their sum modulo 2^n, unless otherwise noted. For a finite set $\mathcal{X}$, let $x \xleftarrow{\$} \mathcal{X}$ denote that an element x is sampled from $\mathcal{X}$ uniformly at random. For any distribution D over a finite set $\mathcal{X}$, we denote its min-entropy by $H_\infty(D)\,(= -\log_2 \max_{x \in \mathcal{X}} D(x))$.

An n-bit block cipher with k-bit keys is a function $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ such that $E(K, \cdot)$ is a permutation for all $K \in \{0,1\}^k$. $\mathsf{BC}(n,k)$ denotes the set of all such E. By E^{-1} we denote the inverse cipher satisfying $E^{-1}(K, E(K,x)) = x$ for all K and x. We often write $E_K(x)$ and $E_K^{-1}(y)$ instead of $E(K,x)$ and $E^{-1}(K,y)$.

Quantum Computation. We assume that the readers are familiar with the basics of quantum computation. When $\mathcal{Y}$ is a set of bit strings, the quantum oracle of a function $f : \mathcal{X} \to \mathcal{Y}$ is given as the unitary operator $U_f : |x, y\rangle \to |x, y \oplus f(x)\rangle$. When we talk about a security notion in the Quantum Ideal Cipher Model (QICM), we assume that an $E \in \mathsf{BC}(n,k)$ is chosen uniformly at random at the beginning of a security game, and an adversary is given the quantum oracle access to E and E^{-1}. We often write like $\mathcal{A}^{|E^{\pm}\rangle}$ to indicate that this oracle access is given to an adversary $\mathcal{A}$.

2.1 Almost XOR Universal Hash Functions

A keyed hash function with a key space $\mathcal{K}$, an input space $\mathcal{X}$, and an output $\mathcal{Y}$, is a function $H : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. We often write $H_K(X)$ instead of $H(K,X)$.

Following [36], we define almost XOR universal hash functions as follows: Let n be a positive integer, $\mathcal{X} := \{0,1\}^* \times \{0,1\}^*$, $\mathcal{Y} := \{0,1\}^n$. For $(X_1, X_2) \in \mathcal{X}$, define $\ell^H(X_1, X_2) := \max\{1, \lceil |X_1|/n \rceil\} + \max\{1, \lceil |X_2|/n \rceil\}$. For a value $c \geq 1$, H is called a c-almost XOR universal hash (c-AXU) if it satisfies

$$\Pr_K [H_K(X_1, X_2) \oplus H_K(X_1', X_2') = Y] \leq c \cdot \frac{\max\left\{\ell^H(X_1, X_2), \ell^H(X_1', X_2')\right\}}{2^n}$$

for any distinct $(X_1, X_2), (X_1', X_2') \in \{0,1\}^* \times \{0,1\}^*$, and any $Y \in \{0,1\}^n$. Here, the probability is taken over a uniform random choice of $K \in \mathcal{K}$.

2.2 Authenticated Encryption with Associated Data

Syntax and Notations. An Authenticated Encryption with Associated Data (AEAD) scheme Π is a pair of functions $\mathcal{E} : \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T}$ and $\mathcal{D} : \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \cup \{\perp\}$. Here, $\mathcal{K}$ is a key space, $\mathcal{N}$ is a nonce space, $\mathcal{AD}$ is a space of associated data, $\mathcal{M}$ is a message space, $\mathcal{C}$ is a

ciphertext space, and $\mathcal{T}$ is a tag space, all of which are sets of some bit strings. We require that $\mathcal{D}(K, N, AD, \mathcal{E}(K, N, AD, M)) = M$ for any $(K, N, AD, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{M}$, and assume that $|C| = |M|$ when $\mathcal{E}(K, N, AD, M) = (C, T)$ for some K, N, AD, T. We often write $\mathcal{E}_K(N, AD, M)$ (resp., $\mathcal{D}_K(N, AD, C, T)$) instead of $\mathcal{E}(K, N, AD, M)$ (resp., $\mathcal{D}(K, N, AD, C, T)$).

In addition, the verification function $\mathcal{V} : \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{C} \times \mathcal{T} \to \{\top, \bot\}$ is defined by $\mathcal{V}(K, N, AD, C, T) = \top$ if $\mathcal{D}(K, N, AD, C, T) \neq \bot$, and $\mathcal{V}(K, N, AD, C, T) = \bot$ if $\mathcal{D}(K, N, AD, C, T) = \bot$.

When multiple keys $K_1, \ldots, K_u \in \mathcal{K}$ are considered, we define Enc and Vrf by $\mathsf{Enc}(\mathsf{id}, N, AD, M) := \mathcal{E}_{K_{\mathsf{id}}}(N, AD, M)$ and $\mathsf{Vrf}(\mathsf{id}, N, AD, C, T) := \mathcal{V}_{K_{\mathsf{id}}}(N, AD, C, T)$, respectively $(\mathsf{id} = 1, \ldots, u)$.

This paper considers AEADs that are built on top of a block cipher E. We write Π^E, $\mathcal{E}^E$, Enc^E, and so on, to indicate that the schemes depend on E.

Security (in the QICM). We study privacy and authenticity separately. Let $\Pi^E = (\mathcal{E}^E, \mathcal{D}^E)$ be an AEAD scheme built on top of a block cipher E. We consider security in the multi-key setting (also commonly referred to as the multi-user setting)[5], where u keys $K_1, \ldots, K_u \in \mathcal{K}$ are chosen uniformly at random for some u (in the real world), and the oracles Enc^E and Vrf^E perform the encryption and verification with respect to these u keys. We discuss security in the QICM, where an adversary is given the quantum oracle of an ideal cipher E and its inverse $E^{\pm}$. We refer to the oracles of E and E^{-1} as the primitive oracles, and the queries to them as primitive queries.

Privacy. The multi-key privacy ($\mathsf{mk\text{-}priv}$) advantage of an adversary $\mathcal{A}$ in the QICM is defined by

$$\mathsf{Adv}^{\mathsf{mk\text{-}priv}}_{\Pi}(\mathcal{A}) := \left| \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{Enc}^E, |E^{\pm}\rangle}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\$, |E^{\pm}\rangle}\right] \right|.$$

Here, \$ is the oracle that returns a random string of length $|M| + \tau$ for any input (id, N, AD, M), and the keys $K_1, \ldots, K_u$ for the real encryption oracle are chosen independently and uniformly at random. The oracles of Enc^E and \$ are classical, while the access to the ideal cipher E and its inverse is quantum. We refer to Enc^E and \$ as the real and ideal encryption oracles, respectively, and queries to them as encryption queries. We focus on the nonce-respecting setting: For each $(\mathsf{id}, N) \in \{1, \ldots, u\} \times \mathcal{N}$, once $\mathcal{A}$ queries (id, N, AD, M) to the encryption oracle for some (AD, M), it is prohibited to query $(\mathsf{id}, N, AD', M')$ afterwards for any (AD', M').

Authenticity. Following prior analyses of GCM [44], we consider an online authenticity notion in which the adversary may make multiple forgery attempts.

[5] We use the term "multi-key" instead of "multi-user" because the indices in our model refer to independently generated keys rather than physical users; in particular, a single user may use multiple keys.

We use an accept/reject verification oracle, as in the multi-user AE formulations of [11,36], rather than a decryption oracle.

The multi-key authenticity (mk-auth) advantage of an adversary $\mathcal{A}$ in QICM is defined by

$$\mathsf{Adv}^{\mathsf{mk\text{-}auth}}_{\Pi}(\mathcal{A}) := \Pr\left[\mathcal{A}^{\mathsf{Enc}^E,\mathsf{Vrf}^E,|E^{\pm}\rangle} \text{ forges }\right],$$

where "$\mathcal{A}^{\mathsf{Enc}^E,\mathsf{Vrf}^E,|E^{\pm}\rangle}$ forges " means that the oracle Vrf^E returns $\top$ while $\mathcal{A}$ is running, and the keys $K_1,\ldots,K_u$ for the real encryption oracle are chosen independently and uniformly at random. The oracles of Enc^E and Vrf^E are classical, while the oracles of E and E^{-1} are quantum. To avoid trivial wins, we prohibit $\mathcal{A}$ from forwarding the response from the encryption oracle to the verification oracle. We refer to the oracle Vrf^E as the verification oracle, and queries to it as verification queries.

As in the definition of privacy, we require that $\mathcal{A}$ is nonce-respecting for the encryption oracle, i.e., it is prohibited to query the same pair (id, N) to the encryption oracle. (This restriction is not applied to verification queries.)

2.3 Reprogramming and Resampling Lemmas

Our proof is based on the resampling-and-reprogramming proof technique, which was introduced by Alagic et al. [3] and has been applied in several follow-up works [4,37,8,65,5,66]. Their core technical lemmas are the arbitrary reprogramming lemma and the resampling lemma, which we recall below (for the resampling lemma, we use a variant shown in [5]).

Lemma 1 (Arbitrary reprogramming lemma [3]). *Let $\mathcal{A}'$ be a quantum algorithm that runs in the following three-stage game.*

1. *After performing some computation, $\mathcal{A}'$ outputs the descriptions of a function $F_0 : \{0,1\}^m \to \{0,1\}^n$ and a randomized algorithm $\mathcal{B}$. Here, $\mathcal{B}$ is an algorithm that outputs another function $F_1 : \{0,1\}^m \to \{0,1\}^n$.*
2. *Run $\mathcal{B}$ to sample F_1, sample a bit b uniformly at random, and resume running $\mathcal{A}'$ relative to the quantum oracle of F_b.*
3. *Provide $\mathcal{A}'$ with the randomness used to invoke $\mathcal{B}$, and continue running $\mathcal{A}'$ without an oracle. Finally, $\mathcal{A}'$ outputs a bit b'.*

Suppose that the expected value of the number of the quantum queries by $\mathcal{A}'$ is at most Q when $b = 0$. Then,

$$|\Pr[1 \leftarrow \mathcal{A}'|b = 0] - \Pr[1 \leftarrow \mathcal{A}'|b = 1]| \leq 2Q\sqrt{\varepsilon}$$

holds, where $\varepsilon := \max_{x\in\{0,1\}^m} \Pr[F_1(x) \neq F_0(x)]$.

For $K_* \in \{0,1\}^k$ and $s_0, s_1 \in \{0,1\}^n$, define a function $\mathsf{swap}^{K_*}_{s_0,s_1} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^k \times \{0,1\}^n$ by

$$\mathsf{swap}^{K_*}_{s_0,s_1}(K,x) = \begin{cases} (K,x) & \text{if } K \neq K_* \text{ or } K = K_* \wedge x \notin \{s_0,s_1\}, \\ (K, s_{b\oplus 1}) & \text{if } K = K_* \text{ and } x = s_b. \end{cases} \tag{1}$$

Lemma 2 (Resampling lemma for ideal ciphers [5]). *Let E be an ideal cipher chosen from $\mathsf{BC}(n,k)$ uniformly at random, and $\mathcal{A}'$ be a quantum adversary that runs in the following two-stage game.*

1. *$\mathcal{A}'$ runs relative to the quantum oracles of E and E^{-1}. After performing some computation and making some queries, $\mathcal{A}'$ outputs a distribution D over $\{0,1\}^k \times \{0,1\}^n \times \{0,1\}^n$.*
2. *A tuple (K_*, s_0, s_1) is sampled according to the distribution D, and a bit b is sampled uniformly at random. Set $E^{(0)} := E$ and $E^{(1)} := E \circ \mathsf{swap}^{K_*}_{s_0,s_1}$. Give (K_*, s_0, s_1) to $\mathcal{A}'$, and continue running $\mathcal{A}'$ relative to the quantum oracles of $E^{(b)}$ and its inverse. Finally, $\mathcal{A}'$ outputs a bit b'.*

If $\mathcal{A}'$ makes at most p quantum queries at the first stage, then

$$|\Pr[1 \leftarrow \mathcal{A}'|b=0] - \Pr[1 \leftarrow \mathcal{A}'|b=1]| \leq 4\sqrt{p \cdot 2^{n-H_\infty(D)}}.$$

3 Description of CAU

Here we describe the AEAD scheme CAU [11,36], which abstracts the structure of GCM [54,27] in which the nonce is mapped directly to a block-cipher input, without being processed by the keyed hash. For 128-bit block ciphers and 96-bit nonces, this nonce processing coincides with the standard processing of 96-bit IVs in AES-GCM. We do not consider the alternative GCM processing for non-96-bit IVs, where the nonce is first processed by the keyed hash.

CAU is built from a block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ and a keyed hash function $H : \{0,1\}^n \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^n$. Its key, nonce, associated-data, message, and tag spaces are $\{0,1\}^k$, $\{0,1\}^r$, $\{0,1\}^*$, $\{0,1\}^*$, and $\{0,1\}^\tau$, respectively, where $r < n$ and $\tau \leq n$. The encryption and decryption algorithms of CAU are defined as follows.

Encryption algorithm of CAU.

1. *Setup.* Suppose that a key $K \in \{0,1\}^k$, nonce $N \in \{0,1\}^r$, associated data $AD \in \{0,1\}^*$, and message $M \in \{0,1\}^*$ are given as input. Compute $Y := \mathsf{pad}(N)$, where $\mathsf{pad} : \{0,1\}^r \to \{0,1\}^n$ is defined by $\mathsf{pad}(N) := N||0^{n-r-1}||1$. If M is non-empty, parse it as $M = M[1]||\cdots||M[\ell]$, where $|M[i]| = n$ for $i < \ell$ and $1 \leq |M[\ell]| \leq n$.
2. *Counter-mode encryption.* If M is empty, set C to the empty string. Otherwise, compute the ciphertext using counter mode with initial counter $Y+1$ as follows. For $i = 1, \ldots, \ell$, compute $V[i] := E_K(Y+i)$. Set $C[i] := M[i] \oplus V[i]$ for $i = 1, \ldots, \ell-1$ and $C[\ell] := M[\ell] \oplus \mathsf{msb}_{|M[\ell]|}(V[\ell])$. Then, set $C := C[1]||\cdots||C[\ell]$.
3. *Tag computation.* Set $L := E_K(0)$ and compute $T := \mathsf{msb}_\tau(H_L(AD, C) \oplus E_K(Y))$.
4. *Output.* Return (C, T).

Decryption algorithm of CAU.

1. *Setup.* Suppose that a key $K \in \{0,1\}^k$, nonce $N \in \{0,1\}^r$, associated data $AD \in \{0,1\}^*$, ciphertext $C \in \{0,1\}^*$, and tag $T \in \{0,1\}^\tau$ are given as input. Compute $Y := \mathsf{pad}(N)$. If C is non-empty, parse it as $C = C[1]||\cdots||C[\ell]$, where $|C[i]| = n$ for $i < \ell$ and $1 \le |C[\ell]| \le n$.
2. *Tag verification.* Set $L := E_K(0)$ and compute $T' := \mathsf{msb}_\tau(H_L(AD, C) \oplus E_K(Y))$. If $T \neq T'$, return $\perp$; otherwise, proceed to the next step.
3. *Counter-mode decryption.* If C is empty, set M to the empty string. Otherwise, compute the message using counter mode with initial counter $Y+1$ as follows. For $i = 1, \ldots, \ell$, compute $V[i] := E_K(Y+i)$. Set $M[i] := C[i] \oplus V[i]$ for $i = 1, \ldots, \ell-1$ and $M[\ell] := C[\ell] \oplus \mathsf{msb}_{|C[\ell]|}(V[\ell])$. Then, set $M := M[1]||\cdots||M[\ell]$.
4. *Output.* Return M.

We measure the length of variable-length strings in n-bit blocks. For a non-empty string $X \in \{0,1\}^*$, its block length is $\ell := \lceil |X|/n \rceil$, and we write $X = X[1]||\cdots||X[\ell]$, where $|X[i]| = n$ for $i < \ell$ and $1 \le |X[\ell]| \le n$. We refer to $X[i]$ as the ith block of X, and define the block length of the empty string to be 0. We use this terminology for messages, associated data, ciphertexts, and other variable-length strings appearing below.

In what follows, we assume that $k \ge n \ge 128$ and that the block length of every message is at most $2^{n-r} - 2$. Under the latter assumption, in the nonce-respecting setting, for each fixed key index id and each $x \neq 0$, the real encryption oracle evaluates $E_{K_\mathsf{id}}(x)$ at most once[6] among all encryption queries associated with id.

4 Post-Quantum Privacy of CAU in the QICM

Recall that the value $\ell^H(AD, C)$ is defined for each (AD, C) as in Section 2.1. Regarding the post-quantum privacy of CAU, the following theorem holds.

Theorem 1. *Suppose that $u \le 2^k$ and that H is c-AXU. Suppose that $\mathcal{A}$ makes at most q_e classical encryption queries, with total message length at most σ_e blocks, and that each nonce appears in encryption queries under at most d key indices. In addition, for each key index, suppose that $\mathcal{A}$ makes at most Q_e encryption queries and that the total message length is at most B_e blocks. Assume also that, for every encryption query, $\ell^H(AD, C) \le \ell^H_{\max}$ holds for the hash input (AD, C). Finally, suppose that $\mathcal{A}$ makes at most p quantum queries in total to the ideal cipher E and its inverse E^{-1}. Then, we have $\mathsf{Adv}^{\mathsf{mk\text{-}priv}}_{\mathsf{CAU}}(\mathcal{A}) \le \delta_{\mathsf{priv}}$,*

[6] This statement is with respect to each key index id; the key values $K_1, \ldots, K_u$ are not assumed to be distinct. If $K_\mathsf{id} = K_{\mathsf{id}'}$ for distinct indices $\mathsf{id} \neq \mathsf{id}'$, the same primitive input may be evaluated once for each of these indices.

where

$$\begin{aligned}\delta_{\mathsf{priv}} :=\ & 4(\sigma_e+q_e)\sqrt{\frac{p}{2^k}}+2p\sqrt{\frac{4d}{2^k}}\\ &+\frac{16(\sigma_e+q_e)\ell^H_{\max}}{2^{(\ell^H_{\max}-2)n/2}}+\frac{2(\sigma_e+q_e)}{2^{2n}}+\frac{4(\sigma_e+q_e+1)^{d+2}}{(d+1)!\,2^{(d-1)n}}\\ &+\frac{2\left(n\ell^H_{\max}+8\right)(\sigma_e+q_e)Q_e+2\left(2n+8\right)(\sigma_e+q_e)B_e+4(\sigma_e+q_e+1)^2}{2^{n+1}}\\ &+\frac{2(\sigma_e+q_e)(Q_e+B_e+2d)}{2^k}. \end{aligned} \tag{2}$$

The interpretation of the bound is discussed later in Section 7.

Proof Overview. As outlined in Section 1.1, our proof applies the reprogramming-and-resampling framework directly to the privacy game. Roughly speaking, we replace the response blocks of the real encryption oracle by fresh random blocks one by one. After each replacement, we reprogram the ideal cipher E so that its evaluations under the secret keys $K_1, \ldots, K_u$ remain consistent with the classical encryption transcript generated so far. Formally, this is implemented by a sequence of hybrids indexed by the number of response blocks that have already been replaced.

An important point is that we never directly reprogram the values $E_{K_{\mathsf{id}}}(0)$, which serve as the hash keys. Indeed, the input 0 is shared by all u keys, and hence the assumption that a single nonce is used under at most d keys gives no useful control at this point. Naively reprogramming $E_{K_{\mathsf{id}}}(0)$ for all u keys would therefore reintroduce a quantum security loss depending explicitly on u, with essentially the same multi-key query threshold as the trivial bound $up^2/2^k$. Instead, we avoid using 0 as a designated reprogramming input and analyze the resulting dependence between the hash function and the ideal cipher directly.

Like the previous works relying on the reprogramming-and-resampling approach, there are two main sources of quantum-specific security loss in the proof. First, when a fresh random response block is introduced, the resampling lemma (Lemma 2) is used to modify one corresponding value of the ideal cipher. Applying this argument block by block gives the term of order $(\sigma_e+q_e)\sqrt{p/2^k}$. Second, after a response block has been fixed, we use the arbitrary reprogramming lemma (Lemma 1) to shift the timing of the reprogramming from immediately after that block to just before the next response block is generated. The most important thing is that, except for unlikely collision events, the number of secret keys that can cause the same relevant reprogramming point is controlled by the parameter d, rather than by the total number u of keys. This yields the term of $O(p\sqrt{d/2^k})$.

The technically most involved part is establishing this latter property while ensuring that all reprogramming operations are mutually consistent. We define bad events corresponding to collisions among the input/output points involved in the reprogramming operations and bound their probabilities. For this purpose, we combine the quantum reprogramming argument with maximum-load

and balls-into-bins techniques appearing in the classical multi-key analysis of CAU [36]. The proof framework, however, is substantially different from [36]: the analysis of [36] is based on the H-coefficient technique, whereas ours proceeds by gradually resampling and reprogramming the ideal cipher. Thus, incorporating their classical counting arguments into our proof requires controlling how the corresponding collision events interact with the dynamically reprogrammed points. In particular, these arguments control the number of keys K satisfying transcript-induced relations of the forms $E_K(x) = y$ and $E_K(x) \oplus H_{E_K(0)}(X) = y$. Together with a bound on the multiplicity of values appearing in the random encryption transcript, this allows us to replace a potential dependence on the total number of keys u by the nonce-overlap parameter d.

Modification to the Encryption Oracles. For the hybrid argument below, it is convenient to regard every ciphertext response and tag as a sequence of full n-bit blocks. We therefore consider slightly strengthened versions of the real and ideal encryption oracles, which reveal some additional bits to the adversary.

First, it suffices to consider the case $\tau = n$. Indeed, in the privacy game the adversary is not given a verification oracle, and hence an oracle returning a full n-bit tag is at least as informative as one returning only its first τ bits. We additionally complete the last ciphertext block to n bits.

More precisely, the modified oracles are defined as follows. Suppose that (id, N, AD, M) is queried to the encryption oracle, and let $\ell := \lceil |M|/n \rceil$. In both modified oracles, we assume that the n-bit response blocks $C[1], \ldots, C[\ell], T$ are returned to $\mathcal{A}$ sequentially, in this order, when they are generated. (These blocks constitute a single encryption response, and $\mathcal{A}$ does not regain control until the last block T is returned.)

Real encryption oracle. Pad M with zeroes to a multiple of n bits, and denote the resulting string by $\bar{M}$; that is, $\bar{M} = M||0^m$ for some m, $|\bar{M}| = \ell n$, and $\bar{M} = M$ whenever $|M|$ is already a multiple of n. The ciphertext blocks $C[1], \ldots, C[\ell]$ are computed for the padded message $\bar{M}$ (using the counter mode), so that the length of the overall ciphertext $C := C[1]|| \cdots ||C[\ell]$ will be ℓn.
Next, let $\tilde{C} := \mathsf{msb}_{|M|}(C)$. By construction, $\tilde{C}$ is exactly the ciphertext that the original encryption oracle would return on input (id, N, AD, M). The tag is computed from $\tilde{C}$ exactly as in the original scheme, i.e., $T := H_L(AD, \tilde{C}) \oplus E_{K_{\mathsf{id}}}(Y)$ (with $L := E_{K_{\mathsf{id}}}(0)$).

Ideal encryption oracle. Sample and return independent uniform n-bit strings $C[1], \ldots, C[\ell], T$ sequentially, in this order.

The original encryption response can be simulated by buffering these blocks until T is returned and then outputting $\big(\mathsf{msb}_{|M|}(C), \mathsf{msb}_\tau(T)\big)$. This holds in both the real and ideal worlds. Consequently, any adversary distinguishing the original real and ideal oracles can be simulated using the modified oracles, and hence it suffices to bound the distinguishing advantage between the two modified games.

In what follows, C always denotes the ciphertext returned by the modified oracle, $\tilde{C} := \mathsf{msb}_{|M|}(C)$ denotes the ciphertext used as input to the hash function, and $\bar{M}$ denotes the zero-padded message defined above.

Notations and Terminology. For each encryption query, the modified oracle returns n-bit blocks $C[1], \ldots, C[\ell], T$ sequentially as its $\ell+1$ response blocks. We write $C := C[1] \| \cdots \| C[\ell]$ and refer to T as the last response block. We index all response blocks globally in the order in which they are generated and returned throughout the game. For example, if the first two responses are (C_1, T_1) and (C_2, T_2), then the first block of C_2 is the (ℓ_1+2)nd response block overall, where $\ell_1 := |C_1|/n$.

Transcript. Let $\mathsf{T}_z^{\mathsf{enc}}$ denote the transcript of the classical encryption queries and responses up to and including the zth response block in the global ordering (in the ideal world). Then,

$$\mathsf{T}_z^{\mathsf{enc}} = \Big(\big((\mathsf{id}_1, N_1, AD_1, M_1), (C_1, T_1) \big), \ldots, \big((\mathsf{id}_i, N_i, AD_i, M_i), (C_i, T_i) \big) \Big) \quad (3)$$

for some i. Here, $(\mathsf{id}_j, N_j, AD_j, M_j)$ is the jth encryption query and (C_j, T_j) its response. The responses for $j < i$ are complete, while the last response (C_i, T_i) may be only partially defined. More precisely, letting $b_{\mathsf{fin}} := z - \sum_{j=1}^{i-1}(\ell_j + 1)$, the transcript contains exactly the first b_{fin} blocks of (C_i, T_i).

For each j and $w \le \ell_j$, whenever $C_j[w]$ is defined, let $V_j[w] := \bar{M}_j[w] \oplus C_j[w]$. This agrees with the corresponding variable in the description of CAU on page 10. We also write $Y_j := \mathsf{pad}(N_j)$ and $L_j := E_{K_{\mathsf{id}_j}}(0)$.

Except for the hash-key evaluation defining L_j, the jth encryption query makes $\ell_j + 1$ evaluations of E, i.e., $E_{K_{\mathsf{id}_j}}(Y_j + 1) \ldots, E_{K_{\mathsf{id}_j}}(Y_j + \ell_j)$ for the ciphertext, and $E_{K_{\mathsf{id}_j}}(Y_j)$ for the tag. Hence, up to the ith encryption query, the total number of such evaluations is $\sigma_i + i$, where $\sigma_i := \ell_1 + \cdots + \ell_i$.

Reprogramming of E. We reprogram E so that its evaluations under the secret keys are consistent with the transcript generated so far. Let $\mathsf{T}_z^{\mathsf{enc}}$ and i be as in Eq. (3), and let $\mathbf{K} := (K_1, \ldots, K_u)$ be the keys for E.

For each $j < i$ and $w \le \ell_j + 1$, define $\mathsf{ReProg}_j^{\le w} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^k \times \{0,1\}^n$ by

$$\mathsf{ReProg}_j^{\le w} := \begin{cases} \text{identity map} & \text{if } w = 0, \\ \mathsf{swap}^{K_{\mathsf{id}_j}}_{Y_j + w, E^{-1}_{K_{\mathsf{id}_j}}(V_j[w])} \circ \cdots \circ \mathsf{swap}^{K_{\mathsf{id}_j}}_{Y_j + 1, E^{-1}_{K_{\mathsf{id}_j}}(V_j[1])} & \text{if } 1 \le w \le \ell_j, \\ \mathsf{swap}^{K_{\mathsf{id}_j}}_{Y_j, E^{-1}_{K_{\mathsf{id}_j}}(T_j \oplus H_{L_j}(AD_j, \tilde{C}_j))} \circ \mathsf{ReProg}_j^{\le \ell_j} & \text{if } w = \ell_j + 1. \end{cases} \quad (4)$$

For $j = i$, define $\mathsf{ReProg}_i^{\leq w}$ analogously whenever the corresponding response blocks are defined. Then, define[7] $b_{\mathsf{fin}} := z - \sum_{j=1}^{i-1}(\ell_j + 1)$ and the reprogrammed block cipher as

$$E[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!] := E \circ \mathsf{ReProg}_i^{\leq b_{\mathsf{fin}}} \circ \mathsf{ReProg}_{i-1}^{\leq (\ell_{i-1}+1)} \circ \cdots \circ \mathsf{ReProg}_1^{\leq (\ell_1+1)} \tag{5}$$

for $z \geq 1$, and set $E[\![\mathsf{T}_0^{\mathsf{enc}}, \mathbf{K}]\!] := E$.

Intuitively, each swap in $\mathsf{ReProg}_j^{\leq w}$ is intended to make the corresponding block-cipher evaluation agree with the associated response block from the ideal encryption oracle, while leaving the hash-key evaluation $L_j = E_{K_{\mathsf{id}_j}}(0)$ untouched. Accordingly, $E[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]$ is obtained by composing the swaps corresponding to the first z response blocks. When the points involved in these swaps do not collide, the resulting cipher is consistent with all these response blocks (except for the hash-key evaluations); possible collisions are handled explicitly in the hybrid analysis below.

Hybrids. For each $z = 0, \ldots, \sigma_e + q_e$, we define hybrids $\mathbf{H}_1^z, \ldots, \mathbf{H}_4^z$ as follows. In each hybrid, the execution initially proceeds in the ideal world and is paused at the specified point, at which some computations and oracles are modified. All response blocks generated before that point remain unchanged from the ideal world. If $\mathcal{A}$ terminates before the specified point is reached, the hybrid simply coincides with the ideal world.

$\mathbf{H}_1^z$. The adversary $\mathcal{A}$ runs relative to the ideal encryption oracle \$ and the quantum ideal-cipher oracle $|E^{\pm}\rangle$ until immediately before the $(z+1)$st response block (in the global ordering, counted from the beginning of the game) is sampled. At this point, sample $\mathbf{K} = (K_1, \ldots, K_u)$ uniformly from $(\{0,1\}^k)^u$, switch the generation of the current and subsequent encryption response blocks from \$ to $\mathsf{Enc}^{E[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]}$ (the encryption oracle defined with respect to the reprogrammed cipher $E[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]$), replace the quantum oracles of E and E^{-1} with those of $E[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]$ and its inverse, and resume the execution.

$\mathbf{H}_2^z$. This hybrid is identical to $\mathbf{H}_1^z$ until the execution is paused. Suppose that the $(z+1)$st response block is the wth block of the response to the current encryption query (id, N, AD, M). Let $Y := \mathsf{pad}(N)$ and let ℓ be the block length of M. Sample $\mathbf{K} = (K_1, \ldots, K_u)$ uniformly from $(\{0,1\}^k)^u$ and $s_1 \xleftarrow{\$} \{0,1\}^n$, and define

$$E' := \begin{cases} E \circ \mathsf{swap}_{Y+w, s_1}^{K_{\mathsf{id}}} & \text{if } w \leq \ell, \\ E \circ \mathsf{swap}_{Y, s_1}^{K_{\mathsf{id}}} & \text{if } w = \ell + 1. \end{cases}$$

Then, switch the generation of the current and subsequent encryption response blocks from \$ to $\mathsf{Enc}^{E'[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]}$, replace the quantum oracles of E and E^{-1} with

[7] ChatGPT was used to suggest the choice of the value to be reprogrammed at this point. The authors subsequently adapted this suggestion to the hybrid argument and verified the resulting construction.

those of $E'[\![\mathsf{T}^{\mathsf{enc}}_z, \mathbf{K}]\!]$ and its inverse, and resume the execution.

$\mathbf{H}^z_3$. This hybrid is identical to $\mathbf{H}^z_2$, except for the order in which s_1 and the $(z+1)$st response block are determined. First, sample the $(z+1)$st response block uniformly at random and return it as the next response block. This is the ciphertext block $C[w]$ if $w \leq \ell$ and the tag block T if $w = \ell + 1$. Then, before resuming the execution, define s_1 by $s_1 := E^{-1}_{K_{\mathsf{id}}}(C[w] \oplus \bar{M}[w])$ if $w \leq \ell$, and $s_1 := E^{-1}_{K_{\mathsf{id}}}(T \oplus H_L(AD, \tilde{C}))$ if $w = \ell + 1$, where $L := E_{K_{\mathsf{id}}}(0)$. Afterwards, make the same oracle modifications as in $\mathbf{H}^z_2$ and resume the execution, generating all subsequent response blocks accordingly.

$\mathbf{H}^z_4$. The adversary $\mathcal{A}$ runs relative to the ideal encryption oracle \$ and the quantum ideal-cipher oracle $|E^{\pm}\rangle$ until it *receives* the $(z+1)$st response block in the global ordering. Immediately afterwards, sample $\mathbf{K} = (K_1, \ldots, K_u)$ uniformly from $(\{0,1\}^k)^u$, switch the generation of all subsequent encryption response blocks from \$ to $\mathsf{Enc}^{E[\![\mathsf{T}^{\mathsf{enc}}_{z+1}, \mathbf{K}]\!]}$, replace the quantum oracles of E and E^{-1} with those of $E[\![\mathsf{T}^{\mathsf{enc}}_{z+1}, \mathbf{K}]\!]$ and its inverse, and resume the execution.

The difference between $\mathbf{H}^z_1$ and $\mathbf{H}^z_2$ is bounded as follows.

Lemma 3. $|\Pr[1 \leftarrow \mathcal{A} \textit{ in } \mathbf{H}^z_1] - \Pr[1 \leftarrow \mathcal{A} \textit{ in } \mathbf{H}^z_2]| \leq 4\sqrt{p \cdot 2^{-k}}$.

The proof is a straightforward application of the resampling lemma (Lemma 2). See Section B in the appendix for details.

The difference between $\mathbf{H}^z_2$ and $\mathbf{H}^z_3$ is bounded as follows.

Lemma 4. $|\Pr[1 \leftarrow \mathcal{A} \textit{ in } \mathbf{H}^z_2] - \Pr[1 \leftarrow \mathcal{A} \textit{ in } \mathbf{H}^z_3]| \leq \frac{8\ell^H_{\max}}{2^{(\ell^H_{\max}-2)n/2}} + \frac{1}{2^{2n}} + \frac{z^{d+1}}{(d+1)!\,2^{(d-1)n}} + \frac{(n\ell^H_{\max}+8)Q_e+(2n+8)B_e+2z+2}{2^{n+1}} + \frac{Q_e+B_e+2d}{2^k}$.

To prove this, we define a bad event capturing collisions among the points involved in the reprogramming. Its probability is bounded using maximum-load and balls-into-bins arguments similar to those in [36]. While the final probability bounds rely only on classical counting arguments, adapting them to our setting is non-trivial: the relevant collision events involve dynamically reprogrammed cipher points, whose interactions must be carefully controlled so that the resulting bounds depend on d rather than on u. See Section C in the appendix for a complete proof.

The difference between $\mathbf{H}^z_3$ and $\mathbf{H}^z_4$ is bounded as follows.

Lemma 5. $|\Pr[1 \leftarrow \mathcal{A} \textit{ in } \mathbf{H}^z_3] - \Pr[1 \leftarrow \mathcal{A} \textit{ in } \mathbf{H}^z_4]| \leq \frac{8\ell^H_{\max}}{2^{(\ell^H_{\max}-2)n/2}} + \frac{1}{2^{2n}} + \frac{z^{d+1}}{(d+1)!\,2^{(d-1)n}} + \frac{(n\ell^H_{\max}+8)Q_e+(2n+8)B_e+2z+2}{2^{n+1}} + \frac{Q_e+B_e+2d}{2^k}$.

For this lemma, the same bad event as in the proof of Lemma 4 guarantees that the two hybrids are identical whenever it does not occur, so this lemma has the same bound as Lemma 4. See Section D in the appendix for details.

Finally, the difference between $\mathbf{H}^z_4$ and $\mathbf{H}^{z+1}_1$ is bounded as follows.

Lemma 6. *Let p_z be the expected number of quantum primitive queries to E/E^{-1} made by $\mathcal{A}$ in the ideal world after receiving the $(z+1)$th response block in the global ordering and before receiving the $(z+2)$th response block. If the $(z+1)$st response block is the last one received during the execution, p_z instead counts the quantum primitive queries made after receiving that block and before $\mathcal{A}$ terminates. Then,*

$$\left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_4^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| \leq 2p_z\sqrt{\frac{4d}{2^k}} + \frac{2(z+1)^{d+1}}{(d+1)!2^{(d-1)n}}. \quad (6)$$

Proof. For each $j \in \{1,\ldots,z+1\}$ and $L \in \{0,1\}^n$, let $(\mathsf{id}(j), N, AD, M)$ and (C,T) be the encryption query and response containing the jth response block (in the global ordering, counted from the beginning of the game). Define[8]

$$W_j(L) := \begin{cases} \bar{M}[w] \oplus C[w], & \text{if the } j\text{th response block is } C[w], \\ T \oplus H_L(AD, \tilde{C}), & \text{if it is } T. \end{cases}$$

Next, define[9] mcoll to be the event that $W_{j_1}(L) = W_{j_2}(L) = \cdots = W_{j_{d+1}}(L)$ holds for some $1 \leq j_1 < \cdots < j_{d+1} \leq z+1$ and some $L \in \{0,1\}^n$. Then, since the responses (C,T) from the ideal encryption oracle \$ are always freshly sampled random strings, we have $\Pr[W_{j_1}(L) = \cdots = W_{j_{d+1}}(L)] = 2^{-nd}$, which implies

$$\Pr[\mathsf{mcoll}] \leq 2^n \binom{z+1}{d+1} 2^{-nd} \leq \frac{(z+1)^{d+1}}{(d+1)!2^{(d-1)n}}. \quad (7)$$

1-1. $\mathcal{A}'$ samples an ideal cipher E and runs $\mathcal{A}$. It simulates the modified ideal encryption oracle \$ for classical encryption queries and uses E and E^{-1} to answer quantum primitive queries. As in $\mathbf{H}_4^z$, $\mathcal{A}'$ pauses the execution immediately after the $(z+1)$st response block in the global ordering is returned, and checks whether mcoll has occurred.

1-2. Define $F_0 : \{0,1\} \times \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ by $F_0(a,K,x) := E_K^{1-2a}(x)$. Let $\mathcal{B}$ be the randomized algorithm that samples $\mathbf{K} = (K_1,\ldots,K_u)$ uniformly and outputs the function $F_1 : \{0,1\} \times \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ defined by $F_1(a,K,x) := \left(E[\![\mathsf{T}_{z+1}^{\mathsf{enc}}, \mathbf{K}]\!]\right)_K^{1-2a}(x)$ if mcoll has not occurred, and by $F_1 := F_0$ otherwise. Here, $\mathsf{T}_{z+1}^{\mathsf{enc}}$ is the encryption transcript defined so far. $\mathcal{A}'$ outputs F_0 and $\mathcal{B}$ at the end of the first stage.

2-1. Run $\mathcal{B}$ to sample $\mathbf{K} = (K_1,\ldots,K_u)$ and define F_1. Sample a uniform bit b.

[8] Recall that $\bar{M}$ is the zero-padded version of M, and $\tilde{C} = \mathsf{msb}_{|M|}(C)$. Note that L is not necessarily equal to $E_{K_{\mathsf{id}}}(0)$. $W_j(L)$ is in fact independent from L if the jth block is not the last (tag) block, but still we define it in this way for ease of later discussions.

[9] Within the proof approach developed by the authors, ChatGPT suggested introducing the event mcoll and the key counting ideas used to establish Eqs. (10)–(12), and produced intermediate drafts of the corresponding argument. The authors then revised these drafts, integrated the resulting argument and bounds into the proof, and independently verified the final proof.

2-2. $\mathcal{A}'$ is given quantum oracle access to F_b. If the $(z+1)$st response block is not the final (tag) block of the current encryption response, $\mathcal{A}'$ proceeds immediately to the next stage. Otherwise, $\mathcal{A}'$ continues running $\mathcal{A}$, answering its quantum primitive queries using $F_b(0,\cdot,\cdot)$ and $F_b(1,\cdot,\cdot)$, until $\mathcal{A}$ makes its next classical encryption query or terminates. In the former case, $\mathcal{A}'$ pauses the execution before answering that encryption query; in the latter case, it records the output of $\mathcal{A}$. Then, $\mathcal{A}'$ proceeds to the next stage.

3-1. $\mathcal{A}'$ is given the randomness used to define F_1, namely $\mathbf{K} = (K_1, \ldots, K_u)$. If $\mathcal{A}$ has not yet terminated, $\mathcal{A}'$ resumes the execution, generating the current and subsequent encryption response blocks using $\mathsf{Enc}^{E[\![\mathsf{T}^{\mathsf{enc}}_{z+1},\mathbf{K}]\!]}$ and answering quantum primitive queries using $E[\![\mathsf{T}^{\mathsf{enc}}_{z+1},\mathbf{K}]\!]$ and its inverse.

3-2. $\mathcal{A}'$ returns the output of $\mathcal{A}$ as its own output.

When $b=0$, the above $\mathcal{A}'$ perfectly simulates $\mathbf{H}_1^{z+1}$, and the expected number of quantum queries by $\mathcal{A}'$ made in the second stage is p_z. Moreover, when $b=1$ and mcoll does not occur, $\mathcal{A}'$ also perfectly simulates $\mathbf{H}_4^z$. In particular, we have

$$\begin{aligned}
&\left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_4^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| \\
&\quad \le \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_4^z | \neg\mathsf{mcoll}\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| + \Pr[\mathsf{mcoll}] \\
&\quad \le \left|\Pr\left[1 \leftarrow \mathcal{A}' | b=1 \wedge \neg\mathsf{mcoll}\right] - \Pr\left[1 \leftarrow \mathcal{A}' | b=0\right]\right| + \Pr[\mathsf{mcoll}] \\
&\quad \le \left|\Pr\left[1 \leftarrow \mathcal{A}' | b=1\right] - \Pr\left[1 \leftarrow \mathcal{A}' | b=0\right]\right| + 2\Pr[\mathsf{mcoll}]. \qquad (8)
\end{aligned}$$

In what follows, we bound the parameter ε of Lemma 1. Fix $(a,K,x) \in \{0,1\} \times \{0,1\}^k \times \{0,1\}^n$ arbitrarily. If $F_0(a,K,x) \neq F_1(a,K,x)$, then the point (K,x), viewed as an input to E if $a=0$ and to E^{-1} if $a=1$, must be affected by one of the reprogramming operations used to define $E[\![\mathsf{T}^{\mathsf{enc}}_{z+1},\mathbf{K}]\!]$ from E (see Eq. (5)). Thus, at least one of the following four conditions holds.

(i) $a=0$, and there is an encryption query (id, N, AD, M) appearing in $\mathsf{T}^{\mathsf{enc}}_{z+1}$ such that $K = K_{\mathsf{id}}$ and $x = \mathsf{pad}(N) + w$ for some $0 \le w \le \ell$, where ℓ is the block length of M.

(ii) $a=1$, and there is an encryption query (id, N, AD, M) appearing in $\mathsf{T}^{\mathsf{enc}}_{z+1}$ such that $K = K_{\mathsf{id}}$ and $E_K^{-1}(x) = \mathsf{pad}(N) + w$ for some $0 \le w \le \ell$.

(iii) $a=0$, and there is some $j \in \{1, \ldots, z+1\}$ such that $K = K_{\mathsf{id}(j)}$ and $E_K(x) = W_j(E_K(0))$.

(iv) $a=1$, and there is some $j \in \{1, \ldots, z+1\}$ such that $K = K_{\mathsf{id}(j)}$ and $x = W_j(E_K(0))$.

Recall that (a,K,x) is fixed. For the case of (i), by the definition of pad and the bound on ℓ, for any fixed x there is at most one nonce N such that $x = \mathsf{pad}(N) + w$ for some $0 \le w \le \ell$. Since $\mathcal{A}$ queries each nonce under at most d keys,

$$\Pr_{K_1,\ldots,K_u}\left[F_1(a,K,x) \neq F_0(a,K,x) \wedge \text{Case (i)}\right] \le \frac{d}{2^k}. \qquad (9)$$

For Case (ii), $E_K^{-1}(x)$ is fixed. By the same argument as in the Case (i), there is at most one nonce N such that $E_K^{-1}(x) = \mathsf{pad}(N) + w$ for some $0 \le w \le \ell$. Since $\mathcal{A}$ queries each nonce under at most d key indices,

$$\Pr_{K_1,\ldots,K_u}[F_1(a,K,x) \neq F_0(a,K,x) \wedge \text{Case (ii)}] \le \frac{d}{2^k}. \tag{10}$$

For Case (iii), if mcoll occurs, then $F_1 = F_0$ by definition. Otherwise, since $E_K(0)$ and $E_K(x)$ are fixed, the definition of mcoll gives

$$|\{j \in \{1,\ldots,z+1\} : W_j(E_K(0)) = E_K(x)\}| \le d.$$

Hence,

$$\begin{aligned}&\Pr_{K_1,\ldots,K_u}[F_1(a,K,x) \neq F_0(a,K,x) \wedge \text{Case (iii)}]\\ &\le \sum_{j:W_j(E_K(0))=E_K(x)} \Pr_{K_1,\ldots,K_u}\left[K_{\mathsf{id}(j)} = K\right] \le \frac{d}{2^k}.\end{aligned} \tag{11}$$

We can similarly show

$$\Pr_{K_1,\ldots,K_u}[F_1(a,K,x) \neq F_0(a,K,x) \wedge \text{Case (iv)}] \le \frac{d}{2^k}. \tag{12}$$

By Eqs. (9)-(12), the parameter ε in Lemma 1 satisfies

$$\varepsilon = \max_{a,K,x} \Pr_{K_1,\ldots,K_u}[F_1(a,K,x) \neq F_0(a,K,x)] \le \frac{4d}{2^k}.$$

Therefore, by Lemma 1, Eq. (7), and Eq. (8),

$$\left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_4^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| \le 2p_z\sqrt{\frac{4d}{2^k}} + \frac{2(z+1)^{d+1}}{(d+1)!2^{(d-1)n}}.$$

□

Finishing the Proof of Theorem 1. The real and ideal worlds in the definition of $\mathsf{mk\text{-}priv}$ (with the modified oracles as defined below Theorem 1) correspond to $\mathbf{H}_1^0$ and $\mathbf{H}_1^{\sigma_e+q_e}$, respectively. Thus, Theorem 1 follows from a straightforward combination of Lemmas 3-6. See Section E in the appendix for details.

5 Post-Quantum Authenticity of CAU in the QICM

Recall that the value $\ell^H(AD,C)$ is defined for each (AD,C) as in Section 2.1. As for the post-quantum authenticity of CAU, the following theorem holds.

Theorem 2. *Suppose that $u \leq 2^k$ and that H is c-AXU. Suppose that $\mathcal{A}$ makes at most q_e classical encryption queries with total message length at most σ_e blocks and at most q_d classical verification queries, where each nonce appears in encryption queries under at most d key indices. For each key index, suppose further that $\mathcal{A}$ makes at most Q_e encryption queries with total message length at most B_e blocks. Assume also that, for every encryption or verification query, $\ell^H(AD, C) \leq \ell^H_{\max}$ holds for the hash input (AD, C), where $\ell^H_{\max} \geq 2$. Finally, suppose that $\mathcal{A}$ makes at most p quantum queries in total to the ideal cipher E and its inverse E^{-1}. Then, we have $\mathsf{Adv}^{\mathsf{mk\text{-}auth}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + \delta'$, where δ_{priv} is as defined in Eq. (2), and*

$$\begin{aligned}\delta' := q_d &\left(\frac{nc\ell^H_{\max}}{2^\tau} + \frac{\left(\ell^H_{\max} n + 1\right) q_e + (n+1)(\sigma_e + 1)}{2^{n+k}} + \frac{\ell^H_{\max} n + 2}{2^\tau}\right) \\ &+ (\sigma_e + q_e + q_d)\left(\frac{(n\ell^H_{\max} + 4)Q_e + (n+5)(B_e+1)}{2^n} + \frac{2d}{2^k}\right) \\ &+ \ell^H_{\max} 2^{-(\ell^H_{\max}-2)n/2+3} + \frac{2}{2^{2n}} + \frac{(\sigma_e + q_e)^{d+1}}{(d+1)!2^{(d-1)n}}. \end{aligned} \tag{13}$$

The interpretation of the bound is discussed later in Section 7.

Proof Overview. The proof[10] first postpones the evaluation of all verification queries until the end of the game. More precisely, we replace the verification oracle by an oracle that always returns $\perp$, while recording all verification queries made by the adversary. This modification does not change the adversary's view before a successful forgery occurs, and we may therefore equivalently determine at the end of the game whether any of the recorded verification queries would have been accepted.

Once the verification oracle always returns $\perp$, we next replace the encryption oracle with the ideal encryption oracle \$, while the recorded verification queries are eventually evaluated with respect to a block cipher reprogrammed to be compatible with the encryption transcript. This replacement is validated by extending the hybrid argument in the privacy proof of Theorem 1. More precisely, we augment each hybrid in that proof by recording all verification queries and evaluating them only after the adversary terminates, using the block cipher associated with that hybrid. The bounds for all hybrid transitions remain unchanged, and hence the total distinguishing loss is at most δ_{priv}. After this trans-

[10] The proof is shown with the assistance of ChatGPT. The author proposed separating privacy and authenticity and, in the authenticity proof, reusing the hybrid argument developed for the privacy proof. ChatGPT identified a gap in a direct reuse of that argument and suggested the augmented-hybrid construction in which verification queries are recorded and evaluated after the adversary terminates. The technical details of the proof were then worked out through multiple rounds of interaction with ChatGPT, with the author providing feedback, corrections, and refinements throughout the process; ChatGPT also produced an intermediate proof draft. The author revised and verified the final proof.

formation, the encryption transcript consists of independently sampled random strings, and all verification queries are evaluated only at the end with respect to the reprogrammed ideal cipher.

Then, it remains to bound the probability that one of the recorded verification queries is accepted in this final game. For this purpose, we condition on suitable good events extending those used in the privacy proof, which guarantee, in particular, that the points of E modified by different reprogramming operations do not collide and that the relevant key-to-output mappings have bounded maximum load. Under these good events, the forgery probability is bounded using the c-AXU property of H together with counting and maximum-load arguments based on [36].

Modification to the Encryption/Verification Oracles. We use the same modified real and ideal encryption oracles as in the privacy proof, including the convention that response blocks are returned sequentially. Note that all the ciphertext and tag blocks are full n-bit blocks. We also reuse the notations (see p. 13 for details).

The verification oracle is unchanged: on a query $(\mathsf{id}, N, AD, \tilde{C}, \tilde{T})$, it returns $\mathsf{Vrf}^E(\mathsf{id}, N, AD, \tilde{C}, \tilde{T})$. Here, $|\tilde{T}| = \tau$, and $\tilde{C}$ can be any length. To exclude trivial forgeries, such a query is prohibited if there is a previous encryption query of the form (id, N, AD, M) such that the response (C, T) satisfies $\tilde{C} = \mathsf{msb}_{|M|}(C)$ and $\tilde{T} = \mathsf{msb}_\tau(T)$.

As in the privacy proof, the original encryption response can be simulated by buffering the response blocks until T is returned and then outputting $\big(\mathsf{msb}_{|M|}(C), \mathsf{msb}_\tau(T)\big)$, while verification responses are unchanged. Thus, any adversary in the original authenticity game can be perfectly simulated in this modified game, and it suffices to upper bound the forging probability in the latter. Whenever the ideal encryption oracle is used below, we use its modified version defined in the privacy proof.

In what follows, the length of the ciphertext C (resp., tag T) returned from the encryption oracle is always a multiple of n (resp., equal to n), and the truncated ciphertext given to the hash function H or verification oracle (resp., τ-bit tags given to the verification oracle) is denoted by $\tilde{C}$ (resp., denoted by $\tilde{T}$). $\bar{M}$ denotes the zero-padded message.

Notations and Terminology. We reuse the notations and terminology from the proof of Theorem 1. In particular, $\mathsf{T}^{\mathsf{enc}}_z$ denotes the transcript of the classical *encryption* queries and responses up to and including the zth response block in the global ordering, and $E[\![\mathsf{T}^{\mathsf{enc}}_z, \mathbf{K}]\!]$ denotes the corresponding reprogrammed cipher, as defined there. We additionally let $\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}} := \mathsf{T}^{\mathsf{enc}}_{\sigma_e+q_e}$ denote the complete encryption transcript at the end of the game, and let $\mathsf{T}^{\mathsf{vf}}_{\mathsf{all}}$ denote the transcript of all classical verification queries.

Hybrids. To prove the theorem, we consider the following hybrids $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{H}_4$. In all hybrids, the flag win is initialized to 0.

$\mathbf{H}_1$. This is the real authenticity game, with the encryption oracle modified as above. The adversary $\mathcal{A}$ is given the (modified) classical encryption oracle Enc^E, the verification oracle Vrf^E, and the quantum oracles of E and E^{-1}. Whenever Vrf^E returns $\top$, set $\mathsf{win} \leftarrow 1$.

$\mathbf{H}_2$. This hybrid is identical to $\mathbf{H}_1$, except that the verification oracle always returns $\perp$ to $\mathcal{A}$. Internally, however, each verification query $(\mathsf{id}, N, AD, \tilde{C}, \tilde{T})$ is evaluated using Vrf^E, and win is set as $\mathsf{win} \leftarrow 1$ if $\mathsf{Vrf}^E(\mathsf{id}, N, AD, \tilde{C}, \tilde{T}) = \top$. The encryption oracle is unchanged.

$\mathbf{H}_3$. The oracles are the same as in $\mathbf{H}_2$, but the internal evaluation of verification queries is deferred until the end of the game. Specifically, every verification query is recorded in $\mathsf{T}^{\mathsf{vf}}_{\mathsf{all}}$ and the oracle returns $\perp$ without evaluating it. When $\mathcal{A}$ terminates, evaluate $\mathsf{Vrf}^E(\mathsf{id}, N, AD, \tilde{C}, \tilde{T})$ for every $(\mathsf{id}, N, AD, \tilde{C}, \tilde{T}) \in \mathsf{T}^{\mathsf{vf}}_{\mathsf{all}}$, and set $\mathsf{win} \leftarrow 1$ if any of them returns $\top$.

$\mathbf{H}_4$. This hybrid has the same structure as $\mathbf{H}_3$, except that $\mathcal{A}$ runs relative to the modified ideal encryption oracle \$ from the privacy proof, while the verification oracle continues to return $\perp$. The adversary retains quantum oracle access to E and E^{-1}. When $\mathcal{A}$ terminates, sample $\mathbf{K} = (K_1, \ldots, K_u)$ uniformly from $(\{0,1\}^k)^u$ and evaluate every recorded verification query $(\mathsf{id}, N, AD, \tilde{C}, \tilde{T}) \in \mathsf{T}^{\mathsf{vf}}_{\mathsf{all}}$ using the verification oracle $\mathsf{Vrf}^{E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]}$ defined with respect to the reprogrammed cipher $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]$. Set $\mathsf{win} \leftarrow 1$ if any of these evaluations returns $\top$.

First, the following lemma holds.

Lemma 7. $\Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_1] = \Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_2] = \Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_3]$.

Proof. The first (resp., second) equation holds as $\mathbf{H}_1$ and $\mathbf{H}_2$ are identical until win is set as 1 (resp., as $\mathcal{A}$'s views are identical between $\mathbf{H}_2$ and $\mathbf{H}_3$). □

The difference between $\mathbf{H}_3$ and $\mathbf{H}_4$ is bounded as follows. (This lemma is proven by extending the hybrid argument in the privacy proof of Theorem 1. See Section F in the appendix for details.)

Lemma 8. $|\Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_3] - \Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_4]| \le \delta_{\mathsf{priv}}$ *holds, where* δ_{priv} *is the value defined as in the statement of Theorem 1.*

Finally, the winning probability in $\mathbf{H}_4$ is bounded as follows.

Lemma 9. $\Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_4]$ *is upper bounded by* $q_d\left(\frac{nc\ell^H_{\max}}{2^\tau} + \frac{\left(\ell^H_{\max}n+1\right)q_e+(n+1)(\sigma_e+1)}{2^{n+k}} + \frac{\ell^H_{\max}n+2}{2^\tau}\right) + (\sigma_e+q_e+q_d)\left(\frac{(n\ell^H_{\max}+4)Q_e+(n+5)(B_e+1)}{2^n} + \frac{2d}{2^k}\right) + \ell^H_{\max}2^{-(\ell^H_{\max}-2)n/2+3} + \frac{2}{2^{2n}} + \frac{(\sigma_e+q_e)^{d+1}}{(d+1)!2^{(d-1)n}}$.

The quantum-specific part in the proofs is complete after Lemma 8. In the proof of Lemma 9, conditioning on good events that ensure consistency of the reprogramming and bounded maximum loads, we bound the remaining forgery probability using the c-AXU property of H and classical counting arguments similar to those in [36]. See Section G in the appendix for a complete proof.

Finishing the Proof of Theorem 2. Since $\Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_1]$ upper bounds the probability of an adversary forging, the claim of the theorem follows from Lemmas 7, 8, and 9.

6 Security when Nonces are Randomized

As shown in the previous works in the classical setting [11,36,24], the multi-key security bounds of GCM (CAU) can be improved when the nonce-generation mechanism is randomized. Hoang et al. [36] study nonce-randomization techniques based on addition (e.g., XOR) and concatenation, mainly treating the randomizer as part of the secret key, and the analysis of the former was later sharpened by [24]. In addition, [36] also discuss public salting, including a construction in which a public random salt is added to a counter. Motivated by their work, we consider the following two public-salt variants.

1. (Adding a public random salt to per-key distinct offsets.) A random public r-bit salt $R_{\mathsf{id}} \in \{0,1\}^r$ is chosen (for each key index id), and the honest parties use nonces $(R_{\mathsf{id}} \boxplus c_{\mathsf{id},0}), (R_{\mathsf{id}} \boxplus c_{\mathsf{id},1}), (R_{\mathsf{id}} \boxplus c_{\mathsf{id},2}), \ldots$ in a sequential order (here, "$\boxplus$" is an arbitrary abelian group operation defined over $\{0,1\}^r$, e.g., XOR operation), where $c_{\mathsf{id},j}$ are pre-defined constants such that $c_{\mathsf{id},j} \neq c_{\mathsf{id},j'}$ for $j \neq j'$. The typical case is that these constants represent counter or sequence-number values.
2. (Concatenating a public random salt.) A parameter $t \leq r$ is chosen, a random t-bit public salt $R'_{\mathsf{id}} \in \{0,1\}^t$ is sampled (for each key index id), and nonces are always of the form $R'_{\mathsf{id}}||N$ for some $N \in \{0,1\}^{r-t}$.

The first (resp., the second) variant can be regarded as the nonce-generation patterns of AES-GCM used in TLS 1.3 [63], QUIC [68], SRTP [53], and the successor-based instantiation of the RBG-based construction of NIST SP 800-38D [27] (resp., TLS 1.2 [64] and IPsec ESP [70]). In the first type, a fixed per-key nonce component is added to a counter- or sequence-dependent value (e.g., using XOR), whereas in the second type it is concatenated with an explicit per-record or per-packet value.

In several of these protocols, the value corresponding to our random salt is derived from secret keying material using a KDF, rather than sampled as an independent public random string. We abstract away this key-derivation layer and model the value as an independently uniform public salt. If an independently sampled salt is kept secret rather than revealed, our public-salt security bound

continues to hold, since revealing the salt only gives additional information to the adversary[11].

We show that these two nonce-generation mechanisms yield improved post-quantum multi-key security bounds for GCM (CAU). The analysis follows the classical nonce-randomization arguments of [36]; for the first variant, we additionally use some combinatorial arguments from [24].

Regarding the first modification of adding a public random salt to per-key distinct offsets, the following corollary holds.

Corollary 1 (Security of CAU, adding a public random salt to per-key distinct offsets). *Suppose that all the assumptions of Theorem 1 (resp., Theorem 2) hold, except for the assumption that every nonce is queried under at most d keys. For each key index id, let $R_{\mathsf{id}} \in \{0,1\}^r$ be an independent uniform public salt sampled at the beginning of the game. Assume that the jth encryption query under key index id uses nonce $R_{\mathsf{id}} \boxplus c_{\mathsf{id},j-1}$, where the values $c_{\mathsf{id},j}$ are fixed and pairwise distinct for each id, and assume that $uQ_e < 2^r$. Then, for any $\gamma > 0$ and any positive integer d satisfying $d \geq \lceil \max\{2euQ_e 2^{-r}, (1+\gamma)r\} / \max\{1, r - \log_2(uQ_e)\} \rceil - 1$, we have $\mathsf{Adv}^{\mathsf{mk\text{-}priv}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + 2^{-\gamma r}$, (resp., $\mathsf{Adv}^{\mathsf{mk\text{-}auth}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + \delta' + 2^{-\gamma r}$). Here, δ_{priv} (and δ' in the authenticity case) are as defined in the corresponding theorem, with d set to the chosen value above.*

Although the randomizer in the XN transform of [24] is kept secret while our salt is public, the relevant collision-counting argument of [24] still applies, since the salts are independent and uniform across keys and the per-key offsets are distinct. See Section H in the appendix for details[12].

Regarding the second modification of concatenating a public random salt, the following corollary holds. (The proof is straightforward. See Section I in the appendix for details.)

Corollary 2 (Security of CAU with a concatenated public random salt). *Let t and d be positive integers with $t \leq r$. Suppose that all the assumptions of Theorem 1 (resp., Theorem 2) hold, except for the assumption that every nonce is queried under at most d keys. For each key index id, let $R'_{\mathsf{id}} \in \{0,1\}^t$ be an independent uniform public salt sampled at the beginning of the game. Assume that every encryption query under key index id uses a nonce of the form $R'_{\mathsf{id}} \| N'$ for some $N' \in \{0,1\}^{r-t}$. Then, $\mathsf{Adv}^{\mathsf{mk\text{-}priv}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + \frac{u^{d+1}}{(d+1)!2^{dt}}$, (resp., $\mathsf{Adv}^{\mathsf{mk\text{-}auth}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + \delta' + \frac{u^{d+1}}{(d+1)!2^{dt}}$) holds. Here, δ_{priv} (and δ' in the authenticity case) are as defined in the corresponding theorem with this choice of d.*

[11] Strictly speaking, relating KDF-derived value to this idealized model additionally requires an appropriate post-quantum pseudorandomness assumption on the KDF, but we do not get into more details about this.

[12] ChatGPT produced an initial draft of the proof in response to an author-specified proof goal. The author then checked the counting argument and parameter substitution, corrected and revised the draft, and verified the final proof. The authors take full responsibility for the correctness.

7 Interpretation of the Security Bounds

This section discusses interpretation of the security bounds shown in Theorem 1 and Theorem 2 (and the corollaries in the previous section).

Overall, our bounds in Theorem 1 and Theorem 2 say that the conclusion analogous to the classical work by Hoang et al. [36] also holds in the post-quantum setting. The u-dependent quantum term $up^2/2^k$ arising from the trivial multi-key reduction is replaced by a d-dependent term of order $\sqrt{dp^2/2^k}$, at the cost of an additional term of order $\sqrt{\sigma_e^2 p/2^k}$. In the parameter regimes considered below, this yields better multi-key security bounds than the trivial ones, in that the security bound reaches constant order at a substantially larger value of p.

In what follows, we provide more detailed discussions on privacy and authenticity separately, assuming that nonces are generated by adding a random salt to per-key distinct offsets (e.g., counter values) as in Corollary 1. We use the query complexity at which the relevant security bound reaches constant advantage as a coarse measure of security[13].

On Privacy. First, we study the interpretation of the privacy bound (Theorem 1, combined with Corollary 1). According to Corollary 1, we can use[14] arbitrary d satisfying $d \geq \lceil \max\{2euQ_e 2^{-r},\, (1+\gamma)r\} / \max\{1,\, r - \log_2(uQ_e)\} \rceil - 1$. For the coarse comparison below, we set $\gamma = 1/2$.

General Conditions for Small Advantage. If $\ell_{\max}^H \geq 4$ and $d \geq 5$, then the terms $\frac{16(\sigma_e+q_e)\ell_{\max}^H}{2^{(\ell_{\max}^H-2)n/2}}$ and $\frac{4(\sigma_e+q_e+1)^{d+2}}{(d+1)!\,2^{(d-1)n}}$ in Eq. (2) become negligible compared to other terms. Assuming that $\sigma_e \gg q_e$ and that d is a small constant, the dominant terms in the bound of Theorem 1 are

$$\sqrt{\frac{16\sigma_e^2 p}{2^k}} + \sqrt{\frac{16p^2 d}{2^k}} + \frac{2n(\ell_{\max}^H Q_e + B_e)\sigma_e + 4\sigma_e^2}{2^n} + \frac{2\sigma_e B_e}{2^k}. \tag{14}$$

The third and fourth terms of Eq. (14) do not involve p and depend only on the classical encryption queries. They are significantly smaller than 1 as long as

$$\sigma_e \ll 2^{n/2-1}, \quad B_e\sigma_e \ll 2^{n-1-\log_2 n}, \quad \text{and} \quad \ell_{\max}^H Q_e \sigma_e \ll 2^{n-1-\log_2 n}. \tag{15}$$

Meanwhile, the first two terms of Eq. (14) depend on the parameter p, which is the number of quantum queries to the cipher E and the inverse E^{-1}. The terms

[13] This provides only a coarse comparison of the bounds in terms of query complexity, and should not be interpreted as a usage limit for a prescribed small target advantage. For a fixed target advantage such as 2^{-32} or 2^{-60}, the quantitative comparison may be different.

[14] An additional term $2^{-r\gamma}$ appears in the bound, but this does not play an important role in considering the query complexity for the advantages to reach a constant order, and we omit it from this coarse comparison.

$\sqrt{\frac{16\sigma_e^2 p}{2^k}}$ and $\sqrt{\frac{16p^2 d}{2^k}}$ are significantly smaller than 1 as long as

$$p \ll 2^{k-4-2\log_2 \sigma_e} \quad \text{and} \quad p \ll 2^{k/2-2-\frac{1}{2}\log_2 d}. \tag{16}$$

Concrete Parameter Settings. Next, we analyze the practical effect of Eq. (16) in more detail, assuming that CAU is instantiated as AES-GCM with 96-bit nonces ($n = 128$ and $r = 96$). We also conservatively set $Q_e = 2^{32}$, which matches the 2^{32}-invocation-per-key limit prescribed by NIST SP 800-38D for the RBG-based IV construction [27] and is substantially larger than the current TLS 1.3 AES-GCM limit of $2^{24.5}$ full-size records under a given set of keys [63].

Let us estimate σ_e as $\sigma_e \approx uB_{\mathsf{avg}}$, where B_{avg} is the average number of the blocks processed by the encryption function per key. The value B_{avg} heavily depends on applications and can sometimes be large, but remains small in some practical protocols. For HTTPS connections using AES-GCM, the TCP payload volume provides a rough proxy for the amount of data processed by AES-GCM under the corresponding TLS traffic keys. For example, Cloudflare [2] reports mean server-to-client TCP payload volumes of 224 KB and 390 KB per HTTP/1.x and HTTP/2 connection, respectively. Dividing these values by the 16-byte AES block size gives workload proxies of approximately $2^{13.8}$ and $2^{14.6}$ blocks per connection, suggesting that $B_{\mathsf{avg}} = 2^{15}$ is a reasonable illustrative workload value.

With this in mind, suppose $\sigma_e = uB_{\mathsf{avg}}$ and $B_{\mathsf{avg}} = 2^{15}$. In addition, assume $u \le 2^{k/4-16+\frac{1}{4}\log_2 d}$. Then the first condition of Eq. (16) is automatically satisfied as long as the second condition $p \ll 2^{k/2-2-\frac{1}{2}\log_2 d}$ is satisfied.

Now, assume $k = 192$, and set $u = 2^{k/4-16} = 2^{32}$. In this case, $d = 5$ suffices to apply Corollary 1 (since $Q_e = 2^{32}$), and thus Eq. (16) is satisfied if $p \ll 2^{k/2-2-\frac{1}{2}\log_2 d} \approx 2^{92.8}$. This is in contrast to the trivial multi-key bound obtained from the single-key bound having the term $up^2/2^k$, which does not guarantee security for $p \ge 2^{80}$ when $k = 192$ and $u = 2^{32}$.

When $k = 256$, Eq. (15) can still be satisfied even when $u = 2^{40}$ for reasonable parameter settings (assuming $B_{\mathsf{avg}} = 2^{15}$), and Eq. (16) is also satisfied for $p \ll 2^{k/2-2-\frac{1}{2}\log_2 d} \approx 2^{124.8}$ (with $d = 5$). Meanwhile, the trivial multi-key bound reaches 1 at $p = 2^{108}$ in the same setting.

Summary. For the privacy bound of Theorem 1, when combined with Corollary 1, the advantage is approximately simplified as in Eq. (14). It is significantly smaller than 1 as long as the conditions of Eq. (15) and Eq. (16) are satisfied. Eq. (15) contains only the classical query complexity, while Eq. (16) involves the parameter p, which is the number of quantum queries to E and E^{-1}. The parameter d is basically a small constant under reasonable assumptions.

For AES-GCM with 96-bit nonces, if we additionally assume $\sigma_e \approx uB_{\mathsf{avg}}$, $B_{\mathsf{avg}} \approx 2^{15}$, and $u \le 2^{k/4-16+\frac{1}{4}\log_2 d}$, then the condition for p can be simplified as $p \ll 2^{k/2-2-\frac{1}{2}\log_2 d}$. In particular, provided that Eq. (15) is also satisfied, the advantage remains significantly small for $p \ll 2^{92.8}$ (resp., $p \ll 2^{124.8}$) when

$k = 192$ and $u = 2^{32}$ (resp., $k = 256$ and $u = 2^{40}$), in contrast to the trivial multi-key bound.

On Authenticity. Next, we consider the interpretation of the authenticity bounds (Theorem 2, combined with Corollary 1). Again, we can use arbitrary d satisfying $d \geq \lceil \max\{2euQ_e 2^{-r}, (1+\gamma)r\} / \max\{1, r - \log_2(uQ_e)\} \rceil - 1$.

In Theorem 2, the bound is given as $\delta_{\mathsf{priv}} + \delta'$. Assuming $\ell^H_{\max} \geq 4$ and $d \geq 5$, the terms $\frac{16(\sigma_e+q_e)\ell^H_{\max}}{2^{(\ell^H_{\max}-2)n/2}} + \frac{4(\sigma_e+q_e+1)^{d+2}}{(d+1)!\,2^{(d-1)n}}$ (in δ_{priv}) and $\ell^H_{\max} 2^{-(\ell^H_{\max}-2)n/2+3} + \frac{(\sigma_e+q_e)^{d+1}}{(d+1)!2^{(d-1)n}}$ (in δ') become negligible compared to other terms. Additionally assuming that $\sigma_e \gg q_e$ and that d is a small constant, the dominant terms are approximately equal to

$$\begin{aligned}
&\sqrt{\frac{16\sigma_e^2 p}{2^k}} + \sqrt{\frac{16p^2 d}{2^k}} + \frac{2n(\ell^H_{\max} Q_e + B_e)\sigma_e + 4\sigma_e^2}{2^n} + \frac{2\sigma_e B_e}{2^k} \\
&\quad + q_d \left(\frac{n(c+1)\ell^H_{\max}}{2^\tau} + \frac{\ell^H_{\max} n q_e + n\sigma_e}{2^{n+k}} \right) + (\sigma_e + q_d) \left(\frac{n\ell^H_{\max} Q_e + nB_e}{2^n} \right) \\
&\lessapprox \sqrt{\frac{16\sigma_e^2 p}{2^k}} + \sqrt{\frac{16p^2 d}{2^k}} + \frac{3n(\ell^H_{\max} Q_e + B_e)\sigma_e + 4\sigma_e^2}{2^n} + \frac{2\sigma_e B_e}{2^k} \\
&\quad + q_d \left(\frac{n(c+1)\ell^H_{\max}}{2^\tau} + \frac{n\ell^H_{\max} Q_e + nB_e}{2^n} \right) \qquad (17)
\end{aligned}$$

(note that we dropped the terms with denominator 2^{2n} or 2^{n+k} as those terms are also negligible compared to other terms).

In Eq. (17), the first four terms are in the same order as those appearing in the privacy bound, which we have already analyzed. The remaining terms contain only the parameters regarding classical queries. The remaining terms are significantly smaller than 1 as long as

$$(c+1)nq_d\ell^H_{\max} \ll 2^\tau, \quad n\ell^H_{\max} Q_e q_d \ll 2^n, \text{ and } nB_e q_d \ll 2^n. \qquad (18)$$

The parameter c is at most 1.5 for AES-GCM [36].

7.1 On Tightness and Room for Improvements of the Bounds

In both the privacy and authenticity bounds (Theorems 1 and 2), the terms that involve p (the number of quantum queries to E and E^{-1}) are those of the order $\sqrt{p^2 d/2^k}$ and $\sqrt{\sigma_e^2 p/2^k}$. The former term $\sqrt{p^2 d/2^k}$ roughly reflects the complexity of the multi-target key search by Grover's algorithm (if a single nonce N is queried to the encryption oracle for d keys, then the d-target preimage search with Grover's algorithm recovers one of those d keys with $p \approx \sqrt{2^k/d}$))[15]. Meanwhile, the latter term $\sqrt{\sigma_e^2 p/2^k}$ appears to be an artifact arising from

[15] Strictly speaking, the advantage by Grover's algorithm is $p^2 d/2^k$, so the term $\sqrt{p^2 d/2^k}$ is tight only with respect to the number of queries.

the repeated application of the resampling lemma (Lemma 2). Similar terms also appear in the previous works taking the reprogramming-and-resampling approach [4,37,8,65], but there are no known attacks matching those terms. (Due to the latter term, our bound does not guarantee security when σ_e is large. For example, if 2^{32} encryption queries (with non-empty message) are made for every key and $u = 2^{32}$, then the term becomes at least $\sqrt{2^{128}p/2^k}$, and the security is not guaranteed for $p \geq 2^{k-128}$.)

The remaining terms involve only the complexity of classical queries. In the privacy bound, the order of essentially the largest term among them is $O(\sigma_e^2/2^n)$. This roughly reflects the fact that the classical tight bound of GCM (CAU) is the birthday bound, and tight in the single-key setting ($u = 1$). However, the leading term of the classical bound shown in [36] is $O(B_e\sigma_e/2^n)$, and there is room for improvement (if $\sigma_e = uB_e$, then our term is $O(u^2B_e^2/2^n)$ while the classical bound is $O(uB_e^2/2^n)$).

The authenticity bound is the sum of the privacy bound (δ_{priv} of Theorem 1) and the additional term δ' of Eq. (13). In δ', the largest terms among those that we have not discussed are roughly in the order of $q_d\ell_{\max}^H/2^\tau$ and $q_d(\ell_{\max}^H Q_e + B_e)/2^n$. Regarding the first term, Ferguson's forgery attack on GCM [29] roughly matches the dependence on q_d, $\ell_{\max}^H$, and $2^{-\tau}$, up to additional multiplicative factors in our bound. On the latter term $q_d(\ell_{\max}^H Q_e + B_e)/2^n$, we do not know a matching attack, and this term might be removable with another proof technique.

References

1. Agence nationale de la sécurité des systèmes d'information (ANSSI), France: Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques. Tech. Rep. ANSSI-PG-083, ANSSI (Mar 2026), https://messervices.cyber.gouv.fr/documents-guides/anssi-guide-mecanismes-crypto-3.00.pdf, version 3.00, 20 March 2026
2. Ahmad, S., Wu, P.: Measuring characteristics of TCP connections at internet scale. Cloudflare Blog (Oct 2025), published October 29, 2025. Accessed August 4, 2026. https://blog.cloudflare.com/measuring-network-connections-at-scale/
3. Alagic, G., Bai, C., Katz, J., Majenz, C.: Post-quantum security of the Even-Mansour cipher. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part III. LNCS, vol. 13277, pp. 458–487. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07082-2_17
4. Alagic, G., Bai, C., Katz, J., Majenz, C., Struck, P.: Post-quantum security of tweakable Even-Mansour, and applications. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part I. LNCS, vol. 14651, pp. 310–338. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58716-0_11
5. Alagic, G., Bai, C., Majenz, C., Shi, K.: Post-quantum security of block cipher constructions. Cryptology ePrint Archive, Paper 2026/209 (2026), https://eprint.iacr.org/2026/209, to appear in Selected Areas in Cryptography (SAC) 2026
6. Anand, M.V., Targhi, E.E., Tabia, G.N., Unruh, D.: Post-quantum security of the CBC, CFB, OFB, CTR, and XTS modes of operation. In: Takagi, T. (ed.) PQCrypto 2016. pp. 44–63. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8_4

7. Ashur, T., Dunkelman, O., Luykx, A.: Boosting authenticated encryption robustness with minimal modifications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part III. LNCS, vol. 10403, pp. 3–33. Springer, Cham (Aug 2017). https://doi.org/10.1007/978-3-319-63697-9_1
8. Basak, J., Bhaumik, R., Chauhan, A.K., Jejurikar, R., Jha, A., Roy, A., Schrottenloher, A., Talnikar, S.: Post-quantum security of key-alternating feistel ciphers. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025, Part I. LNCS, vol. 16245, pp. 446–478. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5018-0_15
9. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Berlin, Heidelberg (May 2000). https://doi.org/10.1007/3-540-45539-6_18
10. Bellare, M., Hoang, V.T.: Efficient schemes for committing authenticated encryption. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 845–875. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3_29
11. Bellare, M., Tackmann, B.: The multi-user security of authenticated encryption: AES-GCM in TLS 1.3. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 247–276. Springer, Berlin, Heidelberg (Aug 2016). https://doi.org/10.1007/978-3-662-53018-4_10
12. Bhattacharya, S., Nandi, M.: Revisiting variable output length xor pseudorandom function. IACR Trans. Symm. Cryptol. **2018**(1), 314–335 (2018). https://doi.org/10.13154/tosc.v2018.i1.314-335
13. Bhaumik, R., Bonnetain, X., Chailloux, A., Leurent, G., Naya-Plasencia, M., Schrottenloher, A., Seurin, Y.: QCB: Efficient quantum-secure authenticated encryption. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part I. LNCS, vol. 13090, pp. 668–698. Springer, Cham (Dec 2021). https://doi.org/10.1007/978-3-030-92062-3_23
14. Bhaumik, R., Degabriele, J.P., Dhar, C.: ♯Pencil: A Domain-Extended Committing BBB PRF for Strengthening GCM. In: Heninger, N., Rosulek, M. (eds.) Advances in Cryptology – CRYPTO 2026. Lecture Notes in Computer Science, vol. 16805. Springer, Cham (2026)
15. Biham, E.: How to decrypt or even substitute DES-encrypted messages in 2^{28} steps. Information Processing Letters **84**(3), 117–124 (2002). https://doi.org/10.1016/S0020-0190(02)00269-7
16. Bose, P., Hoang, V.T., Tessaro, S.: Revisiting AES-GCM-SIV: Multi-user security, faster key derivation, and better bounds. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 468–499. Springer, Cham (Apr / May 2018). https://doi.org/10.1007/978-3-319-78381-9_18
17. Bozhko, A.: Properties of authenticated encryption with associated data (AEAD) algorithms. RFC 9771 (May 2025). https://doi.org/10.17487/RFC9771
18. Brassard, G., Høyer, P., Tapp, A.: Quantum cryptanalysis of hash and claw-free functions. In: LATIN'98: Theoretical Informatics. Lecture Notes in Computer Science, vol. 1380, pp. 163–169. Springer (1998). https://doi.org/10.1007/BFb0054319
19. Campagna, M., Gueron, S.: AWS Key Management Service (KMS): Handling Cryptographic Bounds for Use of AES-GCM. Presented at Real World Crypto 2018 (2018), https://rwc.iacr.org/2018/Slides/Gueron.pdf
20. Campagna, M., Gueron, S.: Key management systems at the cloud scale. Cryptography **3**(3), 23 (2019), https://doi.org/10.3390/cryptography3030023

21. Campagna, M., Maximov, A., Preuß Mattsson, J.: Galois counter mode with secure short tags (GCM-SST). Internet-Draft draft-mattsson-cfrg-aes-gcm-sst-03 (Mar 2024), https://datatracker.ietf.org/doc/html/draft-mattsson-cfrg-aes-gcm-sst-03, work in Progress
22. Chan, J., Rogaway, P.: On committing authenticated-encryption. In: Atluri, V., Di Pietro, R., Jensen, C.D., Meng, W. (eds.) ESORICS 2022, Part II. LNCS, vol. 13555, pp. 275–294. Springer, Cham (Sep 2022). https://doi.org/10.1007/978-3-031-17146-8_14
23. Chung, W., Hwang, S., Kim, S., Lee, B., Lee, J.: Making GCM great again: Toward full security and longer nonces. In: Fehr, S., Fouque, P.A. (eds.) EUROCRYPT 2025, Part I. LNCS, vol. 15601, pp. 33–61. Springer, Cham (May 2025). https://doi.org/10.1007/978-3-031-91107-1_2
24. Degabriele, J.P., Govinden, J., Günther, F., Paterson, K.G.: The security of ChaCha20-Poly1305 in the multi-user setting. In: Vigna, G., Shi, E. (eds.) ACM CCS 2021. pp. 1981–2003. ACM Press (Nov 2021). https://doi.org/10.1145/3460120.3484814
25. Digital Agency and Ministry of Internal Affairs and Communications and Ministry of Economy, Trade and Industry, Japan: The list of ciphers that should be referred to in the procurement for the e-government system (CRYPTREC ciphers list). Tech. Rep. CRYPTREC LS-0001-2022R2, CRYPTREC (Mar 2026), https://www.cryptrec.go.jp/list/cryptrec-ls-0001-2022r2.pdf, originally published March 30, 2023; last updated March 30, 2026
26. Dodis, Y., Grubbs, P., Ristenpart, T., Woodage, J.: Fast message franking: From invisible salamanders to encryptment. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 155–186. Springer, Cham (Aug 2018). https://doi.org/10.1007/978-3-319-96884-1_6
27. Dworkin, M.: Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac. NIST Special Publication 800-38D, National Institute of Standards and Technology (2007)
28. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. Journal of Cryptology **10**(3), 151–162 (Jun 1997). https://doi.org/10.1007/s001459900025
29. Ferguson, N.: Authentication weaknesses in gcm (May 2005), https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/comments/cwc-gcm/ferguson2.pdf, public comment submitted to NIST, May 20, 2005
30. Fischlin, M., Schmalz, T.: Quantum-resistant authenticated encryption variants of GCM. IACR Transactions on Symmetric Cryptology **2026**(2) (2026)
31. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: 28th ACM STOC. pp. 212–219. ACM Press (May 1996). https://doi.org/10.1145/237814.237866
32. Gueron, S., Langley, A., Lindell, Y.: AES-GCM-SIV: Specification and analysis. Cryptology ePrint Archive, Report 2017/168 (2017), https://eprint.iacr.org/2017/168
33. Gueron, S., Lindell, Y.: GCM-SIV: Full nonce misuse-resistant authenticated encryption at under one cycle per byte. In: Ray, I., Li, N., Kruegel, C. (eds.) ACM CCS 2015. pp. 109–119. ACM Press (Oct 2015). https://doi.org/10.1145/2810103.2813613
34. Gueron, S., Lindell, Y.: Better bounds for block cipher modes of operation via nonce-based key derivation. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 1019–1036. ACM Press (Oct / Nov 2017). https://doi.org/10.1145/3133956.3133992

35. Gueron, S., Ristenpart, T.: DNDK: Combining nonce and key derivation for fast and scalable AEAD. Cryptology ePrint Archive, Report 2025/785 (2025), https://eprint.iacr.org/2025/785
36. Hoang, V.T., Tessaro, S., Thiruvengadam, A.: The multi-user security of GCM, revisited: Tight bounds for nonce randomization. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 1429–1440. ACM (2018). https://doi.org/10.1145/3243734.3243816, https://eprint.iacr.org/2018/993, full version: Cryptology ePrint Archive, Paper 2018/993, revision dated November 17, 2025
37. Hosoyamada, A.: Post-quantum security of keyed sponge-based constructions through a modular approach. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025, Part I. LNCS, vol. 16245, pp. 411–445. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5018-0_14
38. Hosoyamada, A., Yasuda, K.: Building quantum-one-way functions from block ciphers: Davies-Meyer and Merkle-Damgård constructions. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part I. LNCS, vol. 11272, pp. 275–304. Springer, Cham (Dec 2018). https://doi.org/10.1007/978-3-030-03326-2_10
39. Igoe, K., Solinas, J.: AES Galois Counter Mode for the Secure Shell Transport Layer Protocol. RFC 5647 (Aug 2009). https://doi.org/10.17487/RFC5647, https://www.rfc-editor.org/info/rfc5647/
40. Inoue, A., Iwata, T., Minematsu, K.: Comprehensive robustness analysis of GCM, CCM, and OCB3. In: Patra, A. (ed.) CT-RSA 2025. LNCS, vol. 15598, pp. 75–98. Springer, Cham (Apr / May 2025). https://doi.org/10.1007/978-3-031-88661-4_4
41. Inoue, A., Jha, A., Mennink, B., Minematsu, K.: Generic security of GCM-SST. In: Fischlin, M., Moonsamy, V. (eds.) ACNS 2025, Part II. LNCS, vol. 15826, pp. 342–369. Springer, Cham (Jun 2025). https://doi.org/10.1007/978-3-031-95764-2_14
42. ISO/IEC: Information security – authenticated encryption. ISO/IEC 19772:2020 (Nov 2020), https://www.iso.org/standard/81550.html
43. Iwata, T., Minematsu, K.: Stronger security variants of GCM-SIV. IACR Trans. Symm. Cryptol. **2016**(1), 134–157 (2016). https://doi.org/10.13154/tosc.v2016.i1.134-157, https://tosc.iacr.org/index.php/ToSC/article/view/539
44. Iwata, T., Ohashi, K., Minematsu, K.: Breaking and repairing GCM security proofs. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 31–49. Springer, Berlin, Heidelberg (Aug 2012). https://doi.org/10.1007/978-3-642-32009-5_3
45. Iwata, T., Seurin, Y.: Reconsidering the security bound of AES-GCM-SIV. IACR Trans. Symm. Cryptol. **2017**(4), 240–267 (2017). https://doi.org/10.13154/tosc.v2017.i4.240-267
46. Jaeger, J., Song, F., Tessaro, S.: Quantum key-length extension. In: Nissim, K., Waters, B. (eds.) TCC 2021, Part I. LNCS, vol. 13042, pp. 209–239. Springer, Cham (Nov 2021). https://doi.org/10.1007/978-3-030-90459-3_8
47. Joux, A.: Authentication failures in NIST version of GCM. Public comment on the first draft of NIST Special Publication 800-38D (Apr 2006), https://csrc.nist.gov/CSRC/media/Projects/Block-Cipher-Techniques/documents/BCM/Comments/800-38-series-drafts/GCM/Joux_comments.pdf
48. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 207–237. Springer, Berlin, Heidelberg (Aug 2016). https://doi.org/10.1007/978-3-662-53008-5_8

49. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Quantum differential and linear cryptanalysis. IACR Trans. Symm. Cryptol. **2016**(1), 71–94 (2016). https://doi.org/10.13154/tosc.v2016.i1.71-94, https://tosc.iacr.org/index.php/ToSC/article/view/536
50. Lang, N., Lucks, S.: On the post-quantum security of classical authenticated encryption schemes. In: El Mrabet, N., De Feo, L., Duquesne, S. (eds.) AFRICACRYPT 23. LNCS, vol. 14064, pp. 79–104. Springer, Cham (Jul 2023). https://doi.org/10.1007/978-3-031-37679-5_4
51. Luykx, A., Mennink, B., Paterson, K.G.: Analyzing multi-key security degradation. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 575–605. Springer, Cham (Dec 2017). https://doi.org/10.1007/978-3-319-70697-9_20
52. Luykx, A., Preneel, B.: Optimal forgeries against polynomial-based MACs and GCM. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 445–467. Springer, Cham (Apr / May 2018). https://doi.org/10.1007/978-3-319-78381-9_17
53. McGrew, D., Igoe, K.: AES-GCM Authenticated Encryption in the Secure Real-time Transport Protocol (SRTP). RFC 7714 (Dec 2015). https://doi.org/10.17487/RFC7714, https://www.rfc-editor.org/info/rfc7714/
54. McGrew, D.A., Viega, J.: The security and performance of the Galois/counter mode (GCM) of operation. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 343–355. Springer, Berlin, Heidelberg (Dec 2004). https://doi.org/10.1007/978-3-540-30556-9_27
55. Mouha, N.: Review of the advanced encryption standard. NIST Interagency/Internal Report 8319, National Institute of Standards and Technology (Jul 2021). https://doi.org/10.6028/NIST.IR.8319
56. Naito, Y., Sasaki, Y., Sugawara, T.: The multi-user security of GCM-SST and further enhancements. In: Cha, S.K., Park, J. (eds.) ISC 2025. LNCS, vol. 16186, pp. 45–65. Springer, Cham (Oct 2025). https://doi.org/10.1007/978-3-032-08124-7_3
57. Naito, Y., Sasaki, Y., Sugawara, T.: Tight multi-user security of CCM and enhancement by tag-based key derivation. In: Jee, K., Kate, A. (eds.) Applied Cryptography and Network Security - 24th International Conference, ACNS 2026, Stony Brook, NY, USA, June 22-25, 2026, Proceedings, Part I. Lecture Notes in Computer Science, vol. 16571, pp. 211–240. Springer (2026), https://doi.org/10.1007/978-3-032-32560-0_8
58. Nandi, M.: Bernstein bound on WCS is tight - repairing luykx-preneel optimal forgeries. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 213–238. Springer, Cham (Aug 2018). https://doi.org/10.1007/978-3-319-96881-0_8
59. National Institute of Standards and Technology: Pre-draft call for comments: GCM and GMAC block cipher modes of operation. NIST SP 800-38D Rev. 1 (Initial Preliminary Draft) (Jan 2025), https://csrc.nist.gov/pubs/sp/800/38/d/r1/iprd
60. National Institute of Standards and Technology: Second pre-draft call for comments: GCM and GMAC block cipher modes of operation. NIST SP 800-38D Rev. 1 (2nd Preliminary Draft) (Jun 2026), https://csrc.nist.gov/pubs/sp/800/38/d/r1/2prd
61. Niwa, Y., Ohashi, K., Minematsu, K., Iwata, T.: GCM security bounds reconsidered. In: Leander, G. (ed.) FSE 2015. LNCS, vol. 9054, pp. 385–407. Springer,

Berlin, Heidelberg (Mar 2015). https://doi.org/10.1007/978-3-662-48116-5_19

62. Patarin, J.: The "coefficients H" technique (invited talk). In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 328–345. Springer, Berlin, Heidelberg (Aug 2009). https://doi.org/10.1007/978-3-642-04159-4_21
63. Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 9846 (Jul 2026). https://doi.org/10.17487/RFC9846, https://www.rfc-editor.org/info/rfc9846/
64. Salowey, J., Choudhury, A., McGrew, D.: AES Galois Counter Mode (GCM) Cipher Suites for TLS. RFC 5288 (Aug 2008). https://doi.org/10.17487/RFC5288, https://www.rfc-editor.org/info/rfc5288/
65. Shiba, R., Iwata, T.: Multi-key security in the quantum world: Revisiting tweakable even-mansour and FX. IACR Transactions on Symmetric Cryptology **2026**(2) (2026), https://doi.org/10.46586/tosc.abksudsuurh6
66. Shiba, R., Iwata, T.: Post-quantum security of tweakable key-alternating feistel ciphers in the multi-key setting. In: Security and Cryptography for Networks (SCN 2026) (2026), to appear
67. Soukharev, V., Jao, D., Seshadri, S.: Post-quantum security models for authenticated encryption. In: Takagi, T. (ed.) PQCrypto 2016. pp. 64–78. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8_5
68. Thomson, M., Turner, S.: Using TLS to Secure QUIC. RFC 9001 (May 2021). https://doi.org/10.17487/RFC9001, https://www.rfc-editor.org/info/rfc9001/
69. Turan, M.S., McKay, K.A., Chang, D., Kang, J., Kelsey, J.: Ascon-based lightweight cryptography standards for constrained devices: Authenticated encryption, hash, and extendable output functions. NIST Special Publication 800-232, National Institute of Standards and Technology (Aug 2025). https://doi.org/10.6028/NIST.SP.800-232
70. Viega, J., McGrew, D.: The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP). RFC 4106 (Jun 2005). https://doi.org/10.17487/RFC4106, https://www.rfc-editor.org/info/rfc4106/

AI Tool Use Disclosure

OpenAI's ChatGPT (GPT-5.6 Sol and GPT-6) was used as an interactive research and writing aid during the preparation of this manuscript. Its uses included assisting with literature searches, drafting or revising selected expository passages, checking calculations and parameter analyses, suggesting or helping refine selected proof ideas and technical arguments, producing intermediate proof drafts, and reviewing arguments for potential gaps or errors. The authors developed the main research direction and overall proof strategy, while ChatGPT made several substantive local contributions to the technical development, refinement, and debugging of the proofs. The principal substantive uses that materially affected the final technical arguments are identified in footnotes at the relevant locations. The authors reviewed, revised, and verified the final manuscript and take full responsibility for the correctness of all claims and proofs in this work.

A Balls-into-Bins Lemma

Like the classical work of Hoang et al. [36], we also rely on the following balls-into-bins lemma, which is used in the subsequent sections in the appendix.

Lemma 10 ([16]). *Let n, ℓ, a, and q be integers such that $n \geq 128$, $\ell \geq 2$, $a \geq 1$, and $q \leq a \cdot 2^n$. Let $X_1, \ldots, X_q$ be random variables taking values in $\{0,1\}^n$, which may not be independent from each other but satisfy*

$$\Pr[X_i = Y | E] \leq 2/2^n$$

for any $Y \in \{0,1\}^n$ and all $i = 1, \ldots, q$, and for any event E on $X_1, \ldots, X_{i-1}$. Then,

$$\Pr\left[\exists Y \in \{0,1\}^n : \#\{i : X_i = Y\} \geq \lceil a\ell n/2 \rceil\right] \leq 2^{-1.5\ell n}.$$

B Proof of Lemma 3

Proof (of Lemma 3). We prove the lemma by using the resampling lemma (Lemma 2). Let $\mathcal{A}'$ be a distinguisher for the game of Lemma 2 defined as follows.

1-0. The distinguisher is given quantum oracle access to E and E^{-1}.
1-1. The distinguisher $\mathcal{A}'$ runs $\mathcal{A}$, simulating the classical encryption oracle by \$ and forwarding quantum queries to E/E^{-1}.
1-2. As in $\mathbf{H}_1^z$, pause the execution just before the $(z+1)$th response block (in the global ordering, counted from when $\mathcal{A}$ started) is sampled. Suppose that this block is the wth block in the response to the classical encryption query (id, N, AD, M), and let $Y := \mathsf{pad}(N)$ and ℓ be the block length of M.
1-3. Define a distribution D over $\{0,1\}^k \times \{0,1\}^n \times \{0,1\}^n$ as $D(K_*, s_0, s_1) := \delta_{Y+w, s_0}/2^{k+n}$ if $w \leq \ell$, and $D(K_*, s_0, s_1) := \delta_{Y, s_0}/2^{k+n}$ if $w = \ell + 1$. Return D as the output of $\mathcal{A}'$ in the first stage.
2-0. (K_*, s_0, s_1) is sampled according to D, and a bit b is chosen uniformly at random. The distinguisher $\mathcal{A}'$ is given the tuple (K_*, s_0, s_1), and the quantum oracle access to $E^{(0)} := E$ if $b = 0$ (resp., $E^{(1)} := E \circ \mathsf{swap}_{s_0, s_1}^{K_*}$ if $b = 1$).
2-1. $\mathcal{A}'$ sets $K_{\mathsf{id}} := K_*$ (note that $\mathsf{id} \in \{1, \ldots, u\}$), randomly samples $K_1, \ldots, K_u$ from $\{0,1\}^k$ except for K_{id}, and sets $\mathbf{K} = (K_1, \ldots, K_u)$. $\mathcal{A}'$ resumes the execution, simulating the classical encryption oracle for $\mathcal{A}$ by $\mathsf{Enc}^{E^{(b)}[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]}$ and quantum primitive oracles for $\mathcal{A}$ by $E^{(b)}[\![\mathsf{T}_z^{\mathsf{enc}}, \mathbf{K}]\!]$ and its inverse. Finally, $\mathcal{A}'$ outputs the output of $\mathcal{A}$ as its own output.

Then, $\mathcal{A}'$ perfectly simulates $\mathbf{H}_1^z$ and $\mathbf{H}_2^z$ when $b = 0$ and $b = 1$, respectively, and at most p primitive queries are made to E in the first stage of $\mathcal{A}'$. Hence, from Lemma 2, it follows that

$$\begin{aligned}
&|\Pr[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^z] - \Pr[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_2^z]| \\
&\quad \leq |\Pr[1 \leftarrow \mathcal{A}' | b = 0] - \Pr[1 \leftarrow \mathcal{A}' | b = 1]| \\
&\quad \leq 4\sqrt{p \cdot 2^{n - H_\infty(D)}} = 4\sqrt{p \cdot 2^{-k}}.
\end{aligned}$$

□

C Proof of Lemma 4

Proof (of Lemma 4). Even if the sampling of the $(z+1)$th block and s_1 in $\mathbf{H}_3^z$ are modified as below, their distribution does not change.

(i) Sample $s_1 \in \{0,1\}^n$ uniformly at random.
(ii) Set the $(z+1)$th block as $C[w] := \bar{M}[w] \oplus E_{K^\star}(s_1)$ for $w \le \ell$, and $T := H_L(AD, \tilde{C}) \oplus E_{K^\star}(s_1)$ for $w = \ell + 1$.

Here, $\tilde{C}$ denotes the truncated version of C such that $|\tilde{C}| = |M|$, and $K^\star$ is the key involved in the $(z+1)$th block. Thus, in what follows, we regard $\mathbf{H}_3^z$ as sampling the $(z+1)$th block and s_1 in this way.

Let[16] $\mathsf{id}^\star \in \{1, \ldots, u\}$ be the key index involved in the $(z+1)$th block, and

$$x^\star := \begin{cases} Y + w & \text{if } w \le \ell, \\ Y & \text{if } w = \ell + 1 \end{cases}$$

(note that $K^\star := K_{\mathsf{id}^\star}$ holds).

For each $j \in \{1, \ldots, z+1\}$, let $(\mathsf{id}(j), N(j), AD(j), M(j))$ and $(C(j), T(j))$ be the unique classical encryption query and its response containing the jth response block in the global ordering (indexed in global ordering, counted from the beginning of the game). Let $\ell(j)$ be the block length of $M(j)$, and let $w(j) \in \{1, \ldots, \ell(j)+1\}$ denote the position of the jth response block within the response to this query, where $w(j) \le \ell(j)$ means that the block is the ciphertext block $C(j)[w(j)]$, and $w(j) = \ell(j) + 1$ means that it is the last (tag) block $T(j)$. Let $\bar{M}(j)$ denote the zero-padded version of $M(j)$, and let $\widetilde{C}(j)$ denote the truncated ciphertext corresponding to this query.

For $L \in \{0,1\}^n$, define[17] $W_j(L) \in \{0,1\}^n$ by

$$W_j(L) := \begin{cases} \bar{M}(j)[w(j)] \oplus C(j)[w(j)], & \text{if } w(j) \le \ell(j), \\ T(j) \oplus H_L\big(AD(j), \widetilde{C}(j)\big), & \text{if } w(j) = \ell(j) + 1. \end{cases}$$

We also define

$$\widetilde{x}_j := E^{-1}_{K_{\mathsf{id}(j)}}\big(W_j(E_{K_{\mathsf{id}(j)}}(0))\big).$$

Finally, let $Y(j) := \mathsf{pad}(N(j))$ and define

$$x_j := \begin{cases} Y(j) + w(j), & \text{if } w(j) \le \ell(j), \\ Y(j), & \text{if } w(j) = \ell(j) + 1. \end{cases}$$

[16] From this point on, we define the bad event bad and bound its probability. The authors developed the high-level idea underlying the definition of bad, while the technical details of the argument were worked out with the assistance of ChatGPT through multiple rounds of interaction, including feedback, corrections, and refinements by the authors. The authors independently checked the resulting argument for correctness and take full responsibility for its contents.

[17] This $W_j(L)$ is essentially the same as that used in the proof of Lemma 6. Note that L is not necessarily equal to $E_{K_{\mathsf{id}(j)}}(0)$. $W_j(L)$ is in fact independent of L when $w(j) \le \ell(j)$, but still we define it in this way for ease of later discussions.

Define bad to be the event that at least one of the following conditions holds when s_1 and the $(z+1)$th block are defined:

1. $(K^\star, x^\star) = (K_{\mathsf{id}(j)}, x_j)$ for some $j \le z$;
2. $(K^\star, x^\star) = (K_{\mathsf{id}(j)}, \widetilde{x}_j)$ for some $j \le z$;
3. $(K^\star, s_1) = (K_{\mathsf{id}(j)}, x_j)$ for some $j \le z$;
4. $(K^\star, s_1) = (K_{\mathsf{id}(j)}, \widetilde{x}_j)$ for some $j \le z$;
5. $s_1 = 0$.
6. $\tilde{x}_j = 0$ for some $j \le z$.

For $i = 1, \ldots, 6$, let bad_i be the event that bad happens and the ith condition among the six conditions holds. (When defining $E'[\![\mathsf{T}^{\mathsf{enc}}_z, \boldsymbol{K}]\!]$ or $E[\![\mathsf{T}^{\mathsf{enc}}_{z+1}, \boldsymbol{K}]\!]$ from E, the values x_j and $\widetilde{x}_j$ are swapped for each j with respect to the key $K_{\mathsf{id}(j)}$. If bad does not occur, then the newly introduced swap points $(K^\star, x^\star)$ and $(K^\star, s_1)$ are disjoint from every previously used swap point $(K_{\mathsf{id}(j)}, x_j)$ and $(K_{\mathsf{id}(j)}, \widetilde{x}_j)$ for $j \le z$, and none of the swap points involved here is at input 0. In particular, the new swap leaves all previously defined $\widetilde{x}_j$ unchanged and commutes with the previously applied swaps.)

Then, the probability of bad is the same between $\mathbf{H}^z_2$ and $\mathbf{H}^z_3$, and the behavior of the oracles is identical between the two hybrids as long as bad does not happen. Thus,

$$\left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}^z_2\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}^z_3\right]\right| \le \Pr\left[\mathsf{bad}\right] \tag{19}$$

holds.

As for bad_1, we have

$$\Pr\left[\mathsf{bad}_1 \wedge \mathsf{id}^\star \neq \mathsf{id}(j)\right] \le \frac{d}{2^k}, \quad \Pr\left[\mathsf{bad}_1 \wedge \mathsf{id}^\star = \mathsf{id}(j)\right] = 0,$$

because $\mathcal{A}$ is nonce-respecting, and $\mathcal{A}$ queries every nonce for at most d keys. Hence,

$$\Pr\left[\mathsf{bad}_1\right] \le \frac{d}{2^k}. \tag{20}$$

Next, we bound $\Pr[\mathsf{bad}_2]$. Let

$$\Lambda_1 := \left\lceil \frac{\ell^H_{\max} n}{2}\, 2^{k-n} \right\rceil, \quad \Lambda_2 := n 2^{k-n}.$$

Define[18] load_1 to be the event that

$$\left|\left\{K \in \{0,1\}^k : E_K(x) \oplus H_{E_K(0)}(X) = t\right\}\right| \ge \Lambda_1$$

for some $x \neq 0$, $t \in \{0,1\}^n$, and some hash input $X = (AD, \tilde{C})$ of the block length at most $\ell^H_{\max}$, and load_2 be the event that

$$\left|\left\{K \in \{0,1\}^k : E_K(x) = t\right\}\right| \ge \Lambda_2,$$

[18] The events load_1 and load_2 are defined based on balls-into-bins arguments in [36].

for some $x \neq 0$ and $t \in \{0,1\}^n$.

For fixed $x \neq 0$ and X, view each key K as a ball placed in the bin

$$E_K(x) \oplus H_{E_K(0)}(X).$$

Conditioned on the preceding balls, the probability that the next ball falls into any fixed bin is at most

$$\frac{1}{2^n - 1} \leq 2^{1-n}.$$

Hence, by applying Lemma 10 (with $q = 2^k$, $a = 2^{k-n}$, and $\ell = \ell^H_{\max}$), the probability that the maximum load is at least Λ_1 is upper bounded by $2^{-1.5\ell^H_{\max} n}$. Since the number of all possible $X = (AD, \tilde{C})$ is at most

$$\begin{aligned}\#\left\{(AD,\tilde{C}) : \ell^H(AD,\tilde{C}) \leq \ell^H_{\max}\right\} &\leq \sum_{\substack{a,b\geq 1,\\ a+b\leq \ell^H_{\max}}} 2^{an+1}2^{bn+1} = 4\sum_{2\leq w\leq \ell^H_{\max}} (w-1)2^{wn} \\ &\leq 8\ell^H_{\max}2^{\ell^H_{\max} n}, \end{aligned} \tag{21}$$

taking a union bound over all possible x and X gives

$$\Pr\left[\mathsf{load}_1\right] \leq 2^n \cdot \left(8\ell^H_{\max}2^{\ell^H_{\max} n}\right) \cdot 2^{-1.5\ell^H_{\max} n} = \ell^H_{\max}2^{n-\ell^H_{\max} n/2+3}.$$

We can also show

$$\Pr\left[\mathsf{load}_2\right] \leq 2^n \cdot 2^{-3n} = 2^{-2n}$$

in the same way (by applying Lemma 10 with $q = 2^k$, $a = 2^{k-n}$, and $\ell = 2$, and then taking the union bound for x).

We also define mcoll to be the event that

$$W_{j_1}(L) = \cdots = W_{j_{d+1}}(L)$$

for some $L \in \{0,1\}^n$ and $1 \leq j_1 < \cdots < j_{d+1} \leq z$. Since each response block of the ideal encryption oracle is freshly and uniformly sampled,

$$\Pr[\mathsf{mcoll}] \leq 2^n \binom{z}{d+1} 2^{-dn} \leq \frac{z^{d+1}}{(d+1)!\,2^{(d-1)n}}.$$

Next, we bound the probability of bad_2 conditioned on $\neg\mathsf{load}_1$, $\neg\mathsf{load}_2$, and $\neg\mathsf{mcoll}$. For the case when $\mathsf{id}(j) = \mathsf{id}^\star$ and the jth block is the tag block, conditioned on $\neg\mathsf{load}_1$, we have

$$\begin{aligned}&\Pr\left[\mathsf{bad}_2 \wedge \mathsf{id}(j) = \mathsf{id}^\star \wedge j\text{th block is the tag block} | \neg\mathsf{load}_1\right] \\ &= \Pr\left[\exists j \leq z :\ \mathsf{id}(j) = \mathsf{id}^\star \wedge x^\star = \widetilde{x}_j \wedge j\text{th block is the tag block} \,\middle|\, \neg\mathsf{load}_1\right] \\ &\leq \sum_{\substack{j:\mathsf{id}(j)=\mathsf{id}^\star \\ j\text{th block is the tag block}}} \Pr\left[E_{K_{\mathsf{id}^\star}}(x^\star) = W_j(E_{K_{\mathsf{id}^\star}}(0)) | \neg\mathsf{load}_1\right] \\ &\leq \sum_{\substack{j:\mathsf{id}(j)=\mathsf{id}^\star \\ j\text{th block is the tag block}}} \frac{\Lambda_1}{2^k} \leq \frac{Q_e\Lambda_1}{2^k}. \end{aligned} \tag{22}$$

For the case when $\mathsf{id}(j) = \mathsf{id}^\star$ and the jth block is *not* the tag block, conditioned on $\neg\mathsf{load}_2$, we have

$$\begin{aligned}
&\Pr\left[\mathsf{bad}_2 \wedge \mathsf{id}(j) = \mathsf{id}^\star \wedge j\text{th block is not the tag block} | \neg\mathsf{load}_2\right] \\
&\quad = \Pr\left[\exists j \le z:\ \mathsf{id}(j) = \mathsf{id}^\star \wedge x^\star = \widetilde{x}_j \wedge j\text{th block is not the tag block} \,\middle|\, \neg\mathsf{load}_2\right] \\
&\quad \le \sum_{\substack{j:\mathsf{id}(j)=\mathsf{id}^\star \\ j\text{th block is not the tag block}}} \Pr\left[E_{K_{\mathsf{id}^\star}}(x^\star) = W_j(E_{K_{\mathsf{id}^\star}}(0)) | \neg\mathsf{load}_2\right] \\
&\quad \le \sum_{\substack{j:\mathsf{id}(j)=\mathsf{id}^\star \\ j\text{th block is not the tag block}}} \frac{\Lambda_2}{2^k} \le \frac{B_e\Lambda_2}{2^k}. \qquad (23)
\end{aligned}$$

For the case of $\mathsf{id}(j) \neq \mathsf{id}^\star$ conditioned on $\neg\mathsf{mcoll}$, we have

$$\begin{aligned}
&\Pr\left[\mathsf{bad}_2 \wedge \mathsf{id}(j) \neq \mathsf{id}^\star | \neg\mathsf{mcoll}\right] \\
&\quad = \Pr\left[\mathsf{id}(j) \neq \mathsf{id}^\star \wedge K_{\mathsf{id}^\star} = K_{\mathsf{id}(j)} \wedge x^\star = \widetilde{x}_j \middle| \neg\mathsf{mcoll}\right] \\
&\quad \le \frac{d}{2^k}. \qquad (24)
\end{aligned}$$

Combining the above bounds, we obtain

$$\begin{aligned}
\Pr[\mathsf{bad}_2] &\le \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) = \mathsf{id}^\star] + \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) \neq \mathsf{id}^\star] \\
&\le \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) = \mathsf{id}^\star \wedge j\text{th block is the tag block}] \\
&\quad + \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) = \mathsf{id}^\star \wedge j\text{th block is not the tag block}] \\
&\quad + \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) \neq \mathsf{id}^\star] \\
&\le \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) = \mathsf{id}^\star \wedge j\text{th block is the tag block} | \neg\mathsf{load}_1] \\
&\quad + \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) = \mathsf{id}^\star \wedge j\text{th block is not the tag block} | \neg\mathsf{load}_2] \\
&\quad + \Pr[\mathsf{bad}_2 \wedge \mathsf{id}(j) \neq \mathsf{id}^\star] \\
&\quad + \Pr[\mathsf{load}_1] + \Pr[\mathsf{load}_2] + \Pr[\mathsf{mcoll}] \\
&\le \frac{Q_e\Lambda_1}{2^k} + \frac{B_e\Lambda_2}{2^k} + \frac{d}{2^k} + \ell_{\max}^H 2^{n-\ell_{\max}^H n/2+3} + 2^{-2n} + \frac{z^{d+1}}{(d+1)!\,2^{(d-1)n}}. \qquad (25)
\end{aligned}$$

Since

$$\frac{\Lambda_1}{2^k} \le \frac{\ell_{\max}^H n}{2^{n+1}} + \frac{1}{2^k} \text{ and } \frac{\Lambda_2}{2^k} = \frac{n}{2^n},$$

it follows that

$$\Pr[\mathsf{bad}_2] \le \frac{8\ell_{\max}^H}{2^{(\ell_{\max}^H-2)n/2}} + \frac{1}{2^{2n}} + \frac{z^{d+1}}{(d+1)!\,2^{(d-1)n}} + \frac{n\ell_{\max}^H Q_e + 2nB_e}{2^{n+1}} + \frac{Q_e + B_e + d}{2^k}. \qquad (26)$$

As for bad_3, we have

$$\begin{aligned}\Pr[\mathsf{bad}_3] &\le \Pr[\mathsf{bad}_3 \wedge \mathsf{id}(j) = \mathsf{id}^\star] + \Pr[\mathsf{bad}_3 \wedge \mathsf{id}(j) \neq \mathsf{id}^\star] \\ &\le \frac{Q_e + B_e}{2^n} + \frac{u-1}{2^k} \cdot \frac{Q_e + B_e}{2^n} \\ &\le \frac{2(Q_e + B_e)}{2^n} \end{aligned} \tag{27}$$

because the keys and s_1 are chosen uniformly at random, and $u \le 2^k$. Similarly,

$$\Pr[\mathsf{bad}_4] \le \frac{2(Q_e + B_e)}{2^n} \tag{28}$$

and

$$\Pr[\mathsf{bad}_5] \le \frac{1}{2^n} \tag{29}$$

hold.

Furthermore, $\Pr[\tilde{x}_j = 0] = 1/2^n$ since $W_j(L)$ is uniformly distributed over $\{0,1\}^n$ for any L and $j \le z+1$ (when E and the keys are fixed). Thus,

$$\Pr[\mathsf{bad}_6] \le \frac{z}{2^n}. \tag{30}$$

By Eqs. (20), (26), and (27)-(30), we have

$$\begin{aligned}\Pr[\mathsf{bad}] \le{}& \frac{8\ell_{\max}^H}{2^{(\ell_{\max}^H - 2)n/2}} + \frac{1}{2^{2n}} + \frac{z^{d+1}}{(d+1)!\,2^{(d-1)n}} \\ &+ \frac{\left(n\ell_{\max}^H + 8\right) Q_e + (2n+8)\, B_e + 2z + 2}{2^{n+1}} + \frac{Q_e + B_e + 2d}{2^k}. \end{aligned} \tag{31}$$

The claim of the lemma follows from this inequality and Eq. (19). □

D Proof of Lemma 5

Proof. Let bad be the event as defined in the proof of Lemma 4. Then, the oracles of $\mathbf{H}_3^z$ and $\mathbf{H}_4^z$ are identical if bad does not happen. Hence, the difference between the two hybrids is bounded by $\Pr[\mathsf{bad}]$, which is further bounded as in Eq. (31). Thus, the claim of the lemma holds. □

E Proof of Theorem 1

Proof (of Theorem 1). For each $z \in \{0, 1, \ldots, \sigma_e + q_e - 1\}$ we have

$$\begin{aligned}
&\left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^0\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{\sigma_e+q_e}\right]\right| \\
&\quad \leq \sum_{0 \leq z < \sigma_e + q_e} \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| \\
&\quad \leq \sum_{\substack{0 \leq z < \sigma_e + q_e \\ i=1,2,3}} \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_i^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_{i+1}^z\right]\right| \\
&\qquad + \sum_{0 \leq z < \sigma_e + q_e} \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_4^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| \qquad (32)
\end{aligned}$$

In addition,

$$\sum_{0 \leq z < \sigma_e + q_e} \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_2^z\right]\right| \leq 4(\sigma_e + q_e)\sqrt{\frac{p}{2^k}} \qquad (33)$$

follows from Lemma 3,

$$\begin{aligned}
&\sum_{\substack{0 \leq z < \sigma_e + q_e \\ i=2,3}} \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_i^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_{i+1}^z\right]\right| \\
&\leq \frac{16(\sigma_e + q_e)\ell_{\max}^H}{2^{(\ell_{\max}^H - 2)n/2}} + \frac{2(\sigma_e + q_e)}{2^{2n}} + \frac{2(\sigma_e + q_e)^{d+2}}{(d+1)!\, 2^{(d-1)n}} \\
&\quad + \frac{2\left(n\ell_{\max}^H + 8\right)(\sigma_e + q_e)Q_e + 2\left(2n + 8\right)(\sigma_e + q_e)B_e + 4(\sigma_e + q_e + 1)^2}{2^{n+1}} \\
&\quad + \frac{2(\sigma_e + q_e)(Q_e + B_e + 2d)}{2^k} \qquad (34)
\end{aligned}$$

follows from Lemmas 4 and 5, and

$$\begin{aligned}
&\sum_{0 \leq z < \sigma_e + q_e} \left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_4^z\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{z+1}\right]\right| \\
&\quad \leq 2p\sqrt{\frac{4d}{2^k}} + \frac{2(\sigma_e + q_e + 1)^{d+2}}{(d+1)!2^{(d-1)n}} \qquad (35)
\end{aligned}$$

follows from Lemma 6. From Eq. (32)-(35), we obtain

$$\begin{aligned}
&\left|\Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^0\right] - \Pr\left[1 \leftarrow \mathcal{A} \text{ in } \mathbf{H}_1^{\sigma_e+q_e}\right]\right| \\
&\leq 4(\sigma_e + q_e)\sqrt{\frac{p}{2^k}} + 2p\sqrt{\frac{4d}{2^k}} \\
&\quad + \frac{16(\sigma_e + q_e)\ell_{\max}^H}{2^{(\ell_{\max}^H - 2)n/2}} + \frac{2(\sigma_e + q_e)}{2^{2n}} + \frac{4(\sigma_e + q_e + 1)^{d+2}}{(d+1)!\, 2^{(d-1)n}} \\
&\quad + \frac{2\left(n\ell_{\max}^H + 8\right)(\sigma_e + q_e)Q_e + 2\left(2n + 8\right)(\sigma_e + q_e)B_e + 4(\sigma_e + q_e + 1)^2}{2^{n+1}} \\
&\quad + \frac{2(\sigma_e + q_e)(Q_e + B_e + 2d)}{2^k} \qquad (36)
\end{aligned}$$

Since the real and ideal worlds (with the modified oracles as defined below Theorem 1) correspond to $\mathbf{H}_1^0$ and $\mathbf{H}_1^{\sigma_e+q_e}$, respectively, the claim of Theorem 1 follows. □

F Proof of Lemma 8

Proof (of Lemma 8). Let $m := \sigma_e + q_e$. We augment each hybrid $\mathbf{H}_j^z$ used in the proof of Theorem 1 ($j \in \{1,2,3,4\}$ and $0 \le z \le m$) as follows. In addition to the oracles provided in $\mathbf{H}_j^z$, the adversary $\mathcal{A}$ is given a verification oracle that always returns $\perp$, while all verification queries are recorded in a transcript $\mathsf{T}_{\mathsf{all}}^{\mathsf{vf}}$. When the adversary terminates, all the recorded verification queries are evaluated with respect to the block cipher used in the hybrid at that point, and win is set to 1 if at least one of them is accepted. We denote the resulting augmented hybrids by $\widehat{\mathbf{H}}_j^z$. (If the adversary terminates before the switching point of a hybrid is reached, we sample $\mathbf{K} = (K_1, \ldots, K_u)$ uniformly at random and use the reprogrammed block cipher $E[\![\mathsf{T}_{\mathsf{all}}^{\mathsf{enc}}, \mathbf{K}]\!]$ only for the final evaluation of the recorded verification queries.)

By construction, $\widehat{\mathbf{H}}_1^0$ is identical to $\mathbf{H}_3$ in the current proof. Indeed, at $z = 0$ the reprogrammed cipher is just E, so the encryption oracle is the real encryption oracle and the final verification is performed with respect to E. On the other hand, since at most m response blocks are produced by the encryption oracle, the switching point of $\widehat{\mathbf{H}}_1^m$ is never reached. Hence, the encryption oracle remains the ideal encryption oracle \$ throughout the execution, and at termination, the verification queries are evaluated with respect to $E[\![\mathsf{T}_{\mathsf{all}}^{\mathsf{enc}}, \mathbf{K}]\!]$. Thus, $\widehat{\mathbf{H}}_1^m$ is identical to $\mathbf{H}_4$ in the current proof.

We next observe that all the hybrid transitions in the proof of Theorem 1 continue to hold for the augmented hybrids with exactly the same bounds.

First, consider the transition from $\widehat{\mathbf{H}}_1^z$ to $\widehat{\mathbf{H}}_2^z$, corresponding to Lemma 3. We use exactly the same distinguisher $\mathcal{A}'$ for the resampling lemma (Lemma 2) as in the proof of Lemma 3, except that, after the adversary $\mathcal{A}$ terminates, the distinguisher $\mathcal{A}'$ evaluates the recorded verification queries with respect to the block cipher of the corresponding hybrid and outputs the resulting value of win. These additional evaluations occur in the second stage of the experiment of the resampling lemma. Since the bound of Lemma 2 depends only on the number of quantum queries made in its first stage, the additional evaluations do not change the bound. Consequently, the difference between the probabilities of $\widehat{\mathbf{H}}_1^z$ to $\widehat{\mathbf{H}}_2^z$ is upper bounded in the same way as in Lemma 3.

Next, consider the transitions from $\widehat{\mathbf{H}}_2^z$ to $\widehat{\mathbf{H}}_3^z$ and from $\widehat{\mathbf{H}}_3^z$ to $\widehat{\mathbf{H}}_4^z$, corresponding to Lemmas 4 and 5. The bad event used in those lemmas is unchanged, because the verification queries do not affect the adversary's view and do not affect the encryption transcript. Conditioned on $\neg\mathsf{bad}$, the corresponding block-cipher oracles in the two hybrids are identical. Moreover, the recorded verification transcripts are also identical, since the verification oracle always returns $\perp$. Hence, the final verification outcomes are identical as well. It follows that the

proofs of Lemmas 4 and 5 apply without any change, with exactly the same bounds.

Finally, consider the transition from $\widehat{\mathbf{H}}_4^z$ to $\widehat{\mathbf{H}}_1^{z+1}$, corresponding to Lemma 6. We modify the distinguisher $\mathcal{A}'$ used in the proof of Lemma 6 only in its final step. After the randomness used to define F_1, in particular the keys $\mathbf{K}$, is revealed in the third stage of the experiment in the arbitrary reprogramming lemma (Lemma 1), the distinguisher $\mathcal{A}'$ continues the simulation exactly as in the proof of Lemma 6. When the adversary $\mathcal{A}$ terminates, the distinguisher $\mathcal{A}'$ evaluates all recorded verification queries with respect to the block cipher of the corresponding hybrid and outputs the resulting value of win. Recall that the distinguisher in the proof of Lemma 6 samples E itself. Therefore, once $\mathbf{K}$ is revealed, this final verification can be performed in the third stage of the experiment for $\mathcal{A}'$ without making any additional query to the oracle given in the experiment of Lemma 1. In particular, the quantity p_z appearing in Lemma 6 is unchanged. The event mcoll and its probability bound are also unchanged, since they depend only on the encryption transcript. Thus, the bound of Lemma 6 continues to hold for the augmented hybrids $\widehat{\mathbf{H}}_4^z$ and $\widehat{\mathbf{H}}_1^{z+1}$.

Therefore, every transition of the augmented hybrids $\widehat{\mathbf{H}}_j^z$ here has exactly the same upper bound as the corresponding transition in the proof of Theorem 1. Applying the same argument for the proof of Theorem 1 around Eq. (32), we obtain

$$\begin{aligned}&|\Pr[\mathsf{win}=1 \text{ in } \mathbf{H}_3] - \Pr[\mathsf{win}=1 \text{ in } \mathbf{H}_4]| \\ &\quad = \left|\Pr[\mathsf{win}=1 \text{ in } \widehat{\mathbf{H}}_1^0] - \Pr[\mathsf{win}=1 \text{ in } \widehat{\mathbf{H}}_1^m]\right| \le \delta_{\mathsf{priv}}.\end{aligned}$$

□

G Proof of Lemma 9

Proof (of Lemma 9). Since $\mathbf{K}$ is independent of the execution of $\mathcal{A}$ in $\mathbf{H}_4$ and is not used until $\mathcal{A}$ stops, we may equivalently sample $\mathbf{K}$ at the beginning for the purpose of bounding the probability of some events.

Let $\Lambda_1 := \left\lceil \frac{\ell_{\max}^H n}{2} 2^{k-n} \right\rceil$ and $\Lambda_2 := \lceil n 2^{k-n} \rceil$ as in the proof of Lemma 4, load_1 and load_2 be the events from the proof of Lemma 4, and mcoll be the event from the proof of Lemma 6. The proof of Lemma 4 shows

$$\Pr[\mathsf{load}_1] \le \ell_{\max}^H \cdot 2^{-(\ell_{\max}^H - 2)n/2+3}, \qquad \Pr[\mathsf{load}_2] \le 2^{-2n}, \tag{37}$$

and, exactly as in the proof of Lemma 6,

$$\Pr[\mathsf{mcoll}] \le 2^n \binom{\sigma_e + q_e}{d+1} 2^{-dn} \le \frac{(\sigma_e + q_e)^{d+1}}{(d+1)! 2^{(d-1)n}}. \tag{38}$$

We also define load_0 to be the event that

$$\left|\{K \in \{0,1\}^k : E_K(0) = L\}\right| > n 2^{k-n}$$

for some $L \in \{0,1\}^n$. By applying Lemma 10 with $q = 2^k$, $a = 2^{k-n}$, and $\ell = 2$, we obtain

$$\Pr[\mathsf{load}_0] \le 2^{-3n}. \tag{39}$$

Next, we isolate the event on which all the reprogramming operations used to define $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]$ from E are compatible with each other. For each $j \in \{1, \ldots, \sigma_e + q_e\}$, let $\mathsf{id}(j)$ be the key index associated with the jth response block in the encryption queries (in the global ordering, counted from the beginning), and define x_j, W_j, and $\widetilde{x}_j$ as in the proof of Lemma 4. The reprogramming associated with the jth response block (see Eq. (4) and Eq. (5) for details) swaps the two points

$$\left(K_{\mathsf{id}(j)}, x_j\right) \quad \text{and} \quad \left(K_{\mathsf{id}(j)}, \widetilde{x}_j\right),$$

where

$$\widetilde{x}_j = E^{-1}_{K_{\mathsf{id}(j)}}\left(W_j\left(E_{K_{\mathsf{id}(j)}}(0)\right)\right).$$

Define $\mathsf{bad}_{\mathsf{repro}}$ to be the event that at least one of the following conditions holds:

(i) there exist $1 \le h < j \le \sigma_e + q_e$ such that $K_{\mathsf{id}(h)} = K_{\mathsf{id}(j)}$ and at least one of the following equalities holds:

$$x_j = x_h, \qquad x_j = \widetilde{x}_h, \qquad \widetilde{x}_j = x_h, \quad \text{or} \quad \widetilde{x}_j = \widetilde{x}_h.$$

(ii) $\widetilde{x}_j = 0$ for some $j \in \{1, \ldots, \sigma_e + q_e\}$.

Notice that $x_j \neq 0$ for every j, by the definition of the inputs used in CAU.

Below, we bound the probability $\Pr\left[\mathsf{bad}_{\mathsf{repro}} \wedge \neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll}\right]$. Fix $j \in \{1, \ldots, \sigma_e + q_e\}$, and suppose that neither condition (i) nor condition (ii) has occurred for any of the first $(j-1)$ response blocks. Conditioned additionally on $\neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll}$, the event

$$K_{\mathsf{id}(j)} = K_{\mathsf{id}(h)} \quad \text{and} \quad x_j \in \{x_h, \widetilde{x}_h\}$$

for some $h < j$ corresponds to Conditions 1 and 2 in the definition of bad in the proof of Lemma 4 (p. 36). The analysis in Eqs. (20) and (26) of that proof therefore give

$$\begin{aligned}\Pr\left[\begin{array}{c} K_{\mathsf{id}(j)} = K_{\mathsf{id}(h)} \text{ and } x_j \in \{x_h, \widetilde{x}_h\} \text{for some } h < j \wedge \\ \neg\mathsf{bad}_{\mathsf{repro}} \text{ until the } j\text{th block} \wedge \neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll} \end{array}\right] \\ \le \frac{Q_e\Lambda_1 + B_e\Lambda_2 + 2d}{2^k} \le \frac{n\ell^H_{\max}Q_e + (n+1)B_e}{2^n} + \frac{2d}{2^k}. \end{aligned} \tag{40}$$

Similarly, the event

$$K_{\mathsf{id}(j)} = K_{\mathsf{id}(h)} \quad \text{and} \quad \widetilde{x}_j \in \{x_h, \widetilde{x}_h\}$$

for some $h < j$, together with the event $\widetilde{x}_j = 0$, corresponds to Conditions 3, 4, and 6 in the definition of bad in the proof of Lemma 4. The analysis of Eqs. (27),

(28), and (30) of that proof give

$$\Pr\left[\begin{array}{c} K_{\mathsf{id}(j)} = K_{\mathsf{id}(h)} \text{ and } \tilde{x}_j \in \{x_h, \widetilde{x}_h\} \text{ for some } h < j \ \vee (\widetilde{x}_j = 0), \text{ and} \\ \neg\mathsf{bad}_{\mathsf{repro}} \text{ until the } j\text{th block} \wedge \neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll} \end{array}\right] \leq \frac{4(Q_e + B_e) + 1}{2^n}. \tag{41}$$

From Eq. (40) and (41),

$$\Pr\left[\begin{array}{c} \mathsf{bad}_{\mathsf{repro}} \text{ occurs for the first time at the } j\text{th block} \wedge \\ \neg\mathsf{bad}_{\mathsf{repro}} \text{ until the } j\text{th block} \wedge \neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll} \end{array}\right] \leq \frac{(n\ell^H_{\max} + 4)Q_e + (n+5)B_e + 1}{2^n} + \frac{2d}{2^k} \tag{42}$$

follows. Taking a union bound over $j = 1, \ldots, \sigma_e + q_e$, we obtain

$$\Pr\left[\mathsf{bad}_{\mathsf{repro}} \wedge \neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll}\right] \leq (\sigma_e + q_e)\left(\frac{(n\ell^H_{\max} + 4)Q_e + (n+5)B_e + 1}{2^n} + \frac{2d}{2^k}\right) \tag{43}$$

Next, let good be the event defined as

$$\mathsf{good} := \neg\mathsf{bad}_{\mathsf{repro}} \wedge \neg\mathsf{load}_0 \wedge \neg\mathsf{load}_1 \wedge \neg\mathsf{load}_2 \wedge \neg\mathsf{mcoll}.$$

Then

$$\Pr\left[\neg\mathsf{good}\right] \leq (\sigma_e + q_e)\left(\frac{(n\ell^H_{\max} + 4)Q_e + (n+5)B_e + 1}{2^n} + \frac{2d}{2^k}\right) + \ell^H_{\max} 2^{-(\ell^H_{\max} - 2)n/2 + 3} + \frac{2}{2^{2n}} + \frac{(\sigma_e + q_e)^{d+1}}{(d+1)!\,2^{(d-1)n}}. \tag{44}$$

follows from Eqs. (37)-(39) and (43).

In what follows, we will bound the joint probability that $\mathsf{win} = 1$ and good occurs. If good occurs, then all the values swapped in defining $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]$ from E are pairwise disjoint, and every programmed encryption input has its prescribed output. In addition, every point that does not appear in the definition of $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]$ (Eq. (4) and (5)) retains its value under E. In particular,

$$E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]_{K_{\mathsf{id}}}(0) = E_{K_{\mathsf{id}}}(0) \tag{45}$$

for every $\mathsf{id} \in \{1, \ldots, u\}$.

Fix a verification query $V := (\mathsf{id}, N, AD, \tilde{C}, \tilde{T})$, and write $Y := \mathsf{pad}(N)$, $X := (AD, \tilde{C})$, and $L := E_{K_{\mathsf{id}}}(0)$. Below, we bound the probability that $\mathsf{Vrf}^{E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]}(V) = \top$. Let $\mathsf{win}_1(V)$ and $\mathsf{win}_2(V)$ be the events defined as follows.

$\underline{\mathsf{win}_1(V)}$: $\mathsf{Vrf}^{E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]}(V) = \top$ holds, and $\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}$ contains an encryption query of the form $(\mathsf{id}, N, AD', M')$ for some AD' and M' (the response is denoted by (C', T')).

$\underline{\mathsf{win}_2(V)}$: $\mathsf{Vrf}^{E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}},\mathbf{K}]\!]}(V) = \top$ holds, and $\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}$ does *not* contain an encryption query of the form $(\mathsf{id}, N, AD', M')$ for any AD' and M'.

Recall that E and the complete encryption and verification transcripts are determined before $\mathbf{K}$ is sampled in $\mathbf{H}_4$.

First, we bound the joint probability of $\mathsf{win}_1(V)$ and good. The encryption query of the form $(\mathsf{id}, N, AD', M')$ is unique by nonce-respectingness. Put $X' := (AD', \tilde{C}')$ (here, $\tilde{C}' := \mathsf{msb}_{|M'|}(C')$). Conditioned on good, acceptance of the verification query implies

$$H_L(X) \oplus H_L(X') = \left(\tilde{T}||Z\right) \oplus T' \tag{46}$$

for some $Z \in \{0,1\}^{n-\tau}$. If $X \neq X'$, then since H is c-AXU, the set of L satisfying Eq. (46) has size at most[19] $c\ell^H_{\max}$ (per each Z). Conditioned on $\neg\mathsf{load}_0$, each such L has at most $n2^{k-n}$ preimage keys $K \in \{0,1\}^k$ under the mapping $K \mapsto E_K(0)$, and the number of possible Z is $2^{n-\tau}$. Hence

$$\Pr[\mathsf{win}_1(V) \wedge X \neq X' \wedge \mathsf{good}] \leq \frac{c\ell^H_{\max} \cdot n2^{k-n} \cdot 2^{n-\tau}}{2^k} = \frac{nc\ell^H_{\max}}{2^\tau}.$$

If $X = X'$, acceptance requires $\tilde{T} = \mathsf{msb}_\tau(T')$. When the encryption response precedes the verification query, this is a prohibited replay. When the encryption query occurs later, T' is freshly and uniformly sampled by the ideal encryption oracle after the verification query has been fixed. Therefore,

$$\Pr[\mathsf{win}_1(V) \wedge X = X' \wedge \mathsf{good}] \leq 2^{-\tau}.$$

In summary, we have

$$\Pr[\mathsf{win}_1(V) \wedge \mathsf{good}] \leq \frac{nc\ell^H_{\max}}{2^\tau} + 2^{-\tau}. \tag{47}$$

Second, we bound the joint probability of $\mathsf{win}_2(V)$ and good. We introduce additional events kcoll, $\mathsf{coll}^1_{\mathsf{repro}}$, and $\mathsf{coll}^2_{\mathsf{repro}}$ as follows.

1. kcoll: $\mathsf{win}_2(V)$ happens, and there is an encryption query $(\mathsf{id}', N', AD', M')$ recorded in $\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}$ such that $\mathsf{id}' \neq \mathsf{id}$, $N' = N$, and $K_{\mathsf{id}} = K_{\mathsf{id}'}$.
2. $\mathsf{coll}^1_{\mathsf{repro}}$: $\mathsf{win}_2(V)$ happens, there is $j \leq q_e + \sigma_e$ such that $Y(= \mathsf{pad}(N)) = \tilde{x}_j$, and the jth block is in the response to an encryption query with key index id.
3. $\mathsf{coll}^2_{\mathsf{repro}}$: $\mathsf{win}_2(V)$ happens, there is $j \leq q_e + \sigma_e$ such that $Y(= \mathsf{pad}(N)) = \tilde{x}_j$, the jth block is in the response to an encryption query with key index id' for some $\mathsf{id}' \neq \mathsf{id}$, and $K_{\mathsf{id}'} = K_{\mathsf{id}}$.

[19] The argument of bounding the number of L satisfying the equation is based on the corresponding arguments using balls-into-bins in [36].

We will bound $\Pr[\mathsf{kcoll} \wedge \mathsf{good}]$. If $\mathsf{win}_2(V)$ happens, then there is no encryption query with the pair (id, N). There are at most d encryption queries with nonce N (with the key indices other than id). Hence,

$$\Pr\left[\mathsf{kcoll} \wedge \mathsf{good}\right] \leq \frac{d}{2^k}. \tag{48}$$

Next, we will bound $\Pr\left[\mathsf{coll}^1_{\mathsf{repro}} \wedge \mathsf{good}\right]$. For a non-tag response block with key index id, the equality $Y = \widetilde{x}_j$ is equivalent to

$$E_{K_{\mathsf{id}}}(Y) = W_j(E_{K_{\mathsf{id}}}(0)),$$

and for a tag block it is equivalent to

$$E_{K_{\mathsf{id}}}(Y) \oplus H_{E_{K_{\mathsf{id}}}(0)}(AD(j), \tilde{C}(j)) = T(j).$$

On $\neg\mathsf{load}_2$ and $\neg\mathsf{load}_1$, respectively, these equations have at most Λ_2 and Λ_1 solutions[20] for K_{id}. There are at most B_e non-tag blocks and Q_e tag blocks for the key index id. Hence,

$$\Pr\left[\mathsf{coll}^1_{\mathsf{repro}} \wedge \mathsf{good}\right] \leq \frac{B_e\Lambda_2 + Q_e\Lambda_1}{2^k} \leq \frac{\left(\ell^H_{\max} n + 1\right) Q_e + (n+1)B_e}{2^n}. \tag{49}$$

Similarly,

$$\Pr\left[\mathsf{coll}^2_{\mathsf{repro}} \wedge \mathsf{good}\right] \leq \frac{\sigma_e\Lambda_2 + q_e\Lambda_1}{2^{2k}} \leq \frac{\left(\ell^H_{\max} n + 1\right) q_e + (n+1)\sigma_e}{2^{n+k}}. \tag{50}$$

holds (note that the condition $K_{\mathsf{id}'} = K_{\mathsf{id}}$ is imposed here, and that we have σ_e and q_e instead of B_e and Q_e because $\mathsf{coll}^2_{\mathsf{repro}}$ concerns the key indices id' that are not id).

Conditioned on $\neg\mathsf{kcoll}$, $\neg\mathsf{coll}^1_{\mathsf{repro}}$, and $\neg\mathsf{coll}^2_{\mathsf{repro}}$ the value (K_{id}, Y) does not appear in defining $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]$ from E as in Eq. (4) and (5), and thus $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!](Y) = E_{K_{\mathsf{id}}}(Y)$. In addition, conditioned on good, $E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!](0) = E_{K_{\mathsf{id}}}(0)$ holds by Eq. (45). Thus, $\mathsf{Vrf}^{E[\![\mathsf{T}^{\mathsf{enc}}_{\mathsf{all}}, \mathbf{K}]\!]}(V) = \mathsf{Vrf}^{E}(V)$ holds for the verification query $V = (\mathsf{id}, N, AD, \tilde{C}, \tilde{T})$, and the response is $\top$ only if

$$E_{K_{\mathsf{id}}}(Y) \oplus H_{E_{K_{\mathsf{id}}}(0)}(X) = \tilde{T}||Z \tag{51}$$

for some $Z \in \{0,1\}^{n-\tau}$. Since Eq. (51) has at most Λ_1 solutions for K_{id} under good (especially $\neg\mathsf{load}_1$), by taking the union bound on Z, we have

$$\begin{aligned}\Pr\left[\mathsf{win}_2(V) \wedge \mathsf{good} \wedge \neg\mathsf{kcoll} \wedge \neg\mathsf{coll}^1_{\mathsf{repro}} \wedge \neg\mathsf{coll}^2_{\mathsf{repro}}\right] &\leq \frac{\Lambda_1}{2^k} \cdot 2^{n-\tau} \\ &\leq \frac{(\ell^H_{\max} n + 1)}{2^\tau}. \end{aligned} \tag{52}$$

[20] The argument of bounding the number of solutions here is also based on the corresponding balls-into-bins arguments in [36].

Combining Eqs. (48)-(50) and (52), we obtain

$$\begin{aligned}\Pr\left[\mathsf{win}_2(V) \wedge \mathsf{good}\right] \leq\ & \frac{\left(\ell^H_{\max} n + 1\right) Q_e + (n+1)(B_e+1)}{2^n} \\ & + \frac{\left(\ell^H_{\max} n + 1\right) q_e + (n+1)(\sigma_e+1)}{2^{n+k}} + \frac{d}{2^k} \\ & + \frac{\ell^H_{\max} n + 1}{2^\tau}. \end{aligned} \tag{53}$$

By taking the union bound for all the verification queries,

$$\begin{aligned}&\Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_4 \wedge \mathsf{good}] \\ &\leq \Pr\left[\begin{array}{l}\mathsf{win}_1(V) \text{ for some} \\ \text{verification query } V\end{array} \wedge \mathsf{good}\right] + \Pr\left[\begin{array}{l}\mathsf{win}_2(V) \text{ for some} \\ \text{verification query } V\end{array} \wedge \mathsf{good}\right] \\ &\leq q_d \left(\frac{nc\ell^H_{\max}}{2^\tau} + \frac{\left(\ell^H_{\max} n + 1\right) Q_e + (n+1)(B_e+1)}{2^n} \right. \\ &\qquad \left. + \frac{\ell^H_{\max} n + 2}{2^\tau} + \frac{\left(\ell^H_{\max} n + 1\right) q_e + (n+1)(\sigma_e+1)}{2^{n+k}} + \frac{d}{2^k} \right) \end{aligned} \tag{54}$$

follows from Eq. (47) and (53).

Consequently,

$$\begin{aligned}&\Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_4] \\ &\quad \leq \Pr[\mathsf{win} = 1 \text{ in } \mathbf{H}_4 \wedge \mathsf{good}] + \Pr[\neg\mathsf{good}] \\ &\quad \leq q_d \left(\frac{nc\ell^H_{\max}}{2^\tau} + \frac{\left(\ell^H_{\max} n + 1\right) Q_e + (n+1)(B_e+1)}{2^n} \right. \\ &\qquad \left. + \frac{\ell^H_{\max} n + 2}{2^\tau} + \frac{\left(\ell^H_{\max} n + 1\right) q_e + (n+1)(\sigma_e+1)}{2^{n+k}} + \frac{d}{2^k} \right) \\ &\qquad + (\sigma_e + q_e) \left(\frac{(n\ell^H_{\max} + 4) Q_e + (n+5) B_e + 1}{2^n} + \frac{2d}{2^k} \right) \\ &\qquad + \ell^H_{\max} 2^{-(\ell^H_{\max} - 2)n/2 + 3} + \frac{2}{2^{2n}} + \frac{(\sigma_e + q_e)^{d+1}}{(d+1)! 2^{(d-1)n}} \\ &\quad \leq q_d \left(\frac{nc\ell^H_{\max}}{2^\tau} + \frac{\left(\ell^H_{\max} n + 1\right) q_e + (n+1)(\sigma_e+1)}{2^{n+k}} + \frac{\ell^H_{\max} n + 2}{2^\tau} \right) \\ &\qquad + (\sigma_e + q_e + q_d) \left(\frac{(n\ell^H_{\max} + 4) Q_e + (n+5)(B_e+1)}{2^n} + \frac{2d}{2^k} \right) \\ &\qquad + \ell^H_{\max} 2^{-(\ell^H_{\max} - 2)n/2 + 3} + \frac{2}{2^{2n}} + \frac{(\sigma_e + q_e)^{d+1}}{(d+1)! 2^{(d-1)n}} \end{aligned} \tag{55}$$

follows from Eq. (44). □

H Proof of Corollary 1

To prove the corollary, we use the following result from [24].

Lemma 11 ([24, Lemma A.1]). *Let Q be a positive integer, $B, \tilde{m}, \lambda$ be positive real values such that $0 < B \leq 1$ and $\lambda > 1$, and e denote the Euler number. Let*

$$m := \left\lceil \frac{\max\left(QB\lambda e, \log_\lambda(B^{-1}) + \tilde{m}\right)}{\max\left(1, \log_\lambda((QB)^{-1})\right)} \right\rceil.$$

Then, $\binom{Q}{m} \cdot B^{m-1} \leq \lambda^{-\tilde{m}}$ holds.

Proof (of Corollary 1). We give the proof for mk-priv; the proof for mk-auth is analogous. For every key index $\mathsf{id} \in \{1, \ldots, u\}$ and $j \in \{1, \ldots, Q_e\}$, define

$$X_j^{\mathsf{id}} := R_{\mathsf{id}} \boxplus c_{\mathsf{id},j-1}.$$

We regard all these variables as being defined before $\mathcal{A}$ starts. Let bad be the event that $N = X_j^{\mathsf{id}}$ holds for a single $N \in \{0,1\}^r$ for more than d pairs of (id, j).

Since $c_{\mathsf{id},j} \neq c_{\mathsf{id},j'}$ for $j \neq j'$, we have $X_j^{\mathsf{id}} \neq X_{j'}^{\mathsf{id}}$ for every fixed id and distinct j, j'. Consequently, any $d+1$ variables of X_j^{id} taking the same value must be associated with $d+1$ distinct key indices.

Fix distinct key indices $\mathsf{id}_1, \ldots, \mathsf{id}_{d+1}$ and indices $j_1, \ldots, j_{d+1}$. Since the salts $R_{\mathsf{id}_1}, \ldots, R_{\mathsf{id}_{d+1}}$ are independent and uniform, the variables $X_{j_1}^{\mathsf{id}_1}, \ldots, X_{j_{d+1}}^{\mathsf{id}_{d+1}}$ are independent and uniform over $\{0,1\}^r$. Hence

$$\Pr\left[X_{j_1}^{\mathsf{id}_1} = \cdots = X_{j_{d+1}}^{\mathsf{id}_{d+1}}\right] = 2^{-rd}.$$

The number of variables X_j^{id} is at most uQ_e. A union bound therefore gives

$$\Pr[\mathsf{bad}] \leq \binom{uQ_e}{d+1} 2^{-rd}. \tag{56}$$

We now apply Lemma 11 by additionally setting $Q := uQ_e, B := 2^{-r}, \widetilde{m} := \gamma r, \lambda := 2$. Then we have $\log_\lambda(B^{-1}) = r, QB\lambda e = 2eQ2^{-r}, \log_\lambda((QB)^{-1}) = r - \log_2 Q$. Since

$$d \geq m - 1 = \left\lceil \frac{\max\{2eQ2^{-r},\ (1+\gamma)r\}}{\max\{1,\ r - \log_2 Q\}} \right\rceil - 1$$

holds by assumption, Lemma 11 gives

$$\binom{uQ_e}{d+1} 2^{-rd} = \binom{Q}{d+1} B^d \leq \binom{Q}{m} B^{m-1} \leq 2^{-\gamma r} \tag{57}$$

(here, for the first inequality, we used the fact that the sequence $\binom{Q}{s} B^{s-1}$ is non-increasing in s, which holds because $\frac{\binom{Q}{s+1}B^s}{\binom{Q}{s}B^{s-1}} = B\frac{Q-s}{s+1} < 1$.)

From Eqs. 56 and (57), it follows that

$$\Pr[\mathsf{bad}] \leq 2^{-\gamma r}.$$

As long as bad does not occur, every nonce is queried under at most d keys, and hence the setting satisfies the assumptions of Theorem 1. Therefore,

$$\mathsf{Adv}^{\mathsf{mk\text{-}priv}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + \Pr[\mathsf{bad}] \leq \delta_{\mathsf{priv}} + 2^{-\gamma r}.$$

□

I Proof of Corollary 2

Proof (of Corollary 2). We give a proof for mk-priv (the proof for mk-auth is analogous). Let bad be an event that $\mathcal{A}$ queries a single nonce to the encryption oracle for more than d keys. Then, bad happens only when the values of $(d+1)$ random t-bit salts collide, and thus $\Pr[\mathsf{bad}] \leq \binom{u}{d+1}/2^{dt} \leq (u^{d+1}/(d+1)!2^{dt})$. As long as bad does not happen, the setting satisfies the assumptions of Theorem 1, so we have $\mathsf{Adv}^{\mathsf{mk\text{-}priv}}_{\mathsf{CAU}}(\mathcal{A}) \leq \delta_{\mathsf{priv}} + \Pr[\mathsf{bad}] \leq \delta_{\mathsf{priv}} + (u^{d+1}/(d+1)!2^{dt})$. □