Tight Multi-User Security of CCM and Enhancement by Tag-Based Key Derivation Applied to GCM and CCM

YUSUKE NAITO, Mitsubishi Electric Corporation, Japan
YU SASAKI, NTT Social Informatics Laboratories, Japan and Associate of National Institute of Standards and Technology, US
TAKESHI SUGAWARA, The University of Electro-Communications, Japan

Abstract. GCM and CCM are block cipher (BC) based authenticated encryption modes. In multi-user (mu) security, a total number of BC invocations by all users σ and the maximum number of BC invocations per user σ_{u} are crucial factors. For GCM, the tight mu-security bound has been identified as $\frac{\sigma_{\mathrm{u}}\sigma}{2^n} + \frac{up+u^2}{2^k}$, where k and n are respectively the key and block sizes, u is the number of users, p is the number of offline queries. In contrast, the CCM's mu-security bound is still unclear. Two bounds of $\frac{u\sigma_{\mathrm{u}}^2}{2^n} + \frac{up+u^2}{2^k}$ and $\frac{\sigma^2}{2^n} + \frac{up+u\sigma}{2^k}$ have been derived by Luykx et al. (Asiacrypt 2017) and Zhang et al. (CCS 2024), respectively, but both are not tight and worse than the GCM's bound. Moreover, methods to enhance mu security without disruptive changes in the scheme have been considered for GCM, namely nonce randomization (NR) to improve offline security and nonce-based key derivation (NKD) to improve online security, but their applicability to CCM has never been discussed.

In this paper, we prove an improved mu-security bound of CCM, which is tight, and reaches the GCM's bound. We then prove that NR and NKD applied to CCM result in the same bounds for the case to GCM. An important takeaway is that CCM is now proved to be as secure as GCM. Moreover, we argue that NR and NKD can be insufficient for some applications with massive data, and propose a new enhancement method called nonce-based and tag-based key derivation (NTKD) that is applied to GCM and CCM. We prove that the resulting schemes meet such real-world needs.

Additional Key Words and Phrases: CCM, GCM, Multi-User Security, Security Proof, Nonce Randomization, Nonce-Based and Tag-Based Key Derivation

1 Introduction

Privacy and message authenticity are two fundamental properties required for secure and reliable information systems. An authenticated encryption (AE) scheme is a symmetric-key cryptosystem that provides both properties and has been widely deployed, especially standard AE schemes, AES-GCM [11], AES-GCM-SIV [13], ChaCha20-Poly1305 [28], and AES-CCM [10, 40]. Hence, proving the security of these AE schemes is an important research topic.

Conventionally, AE security has been discussed only for a single user (su) with a fixed key. On the other hand, in recent years, multi-user (mu) security, which ensures security for all users with their own keys, has been discussed. In mu security, in addition to the behavior of each user, the total amount of data for all users (σ) affects the security. The value of σ that can be processed securely depends on the maximum message size and the maximum number of messages that each user can process. A superior AE scheme ensures large σ without imposing strong limitations on each user.

Researchers have studied mu-security of widely standardised algorithms, including AES-GCM [3, 16, 24], AES-GCM-SIV [6], and ChaCha20-Poly1305 [9]. Mu-security impacts how the schemes are used in real-world protocols. In particular, TLS, DTLS, and QUIC determine the rekeying intervals of AES-GCM according to the mu-security limit [32, 33, 37]. Moreover, the ongoing discussion on the usage limit of AEs, published as Internet-Draft [15], considers mu-security for other schemes, including AES-CCM.

Authors' Contact Information: Yusuke Naito, Naito.Yusuke@ce.MitsubishiElectric.co.jp, Mitsubishi Electric Corporation, Japan; Yu Sasaki, yusk.sasaki@ntt.com, NTT Social Informatics Laboratories, Japan and Associate of National Institute of Standards and Technology, US; Takeshi Sugawara, sugawara@uec.ac.jp, The University of Electro-Communications, Japan.

Both GCM and CCM are block cipher (BC) modes that build an AE scheme by combining encryption and message authentication code (MAC). GCM uses the counter (CTR) mode for encryption and a polynomial-hash-based authentication for MAC. CCM uses the CTR mode for encryption and CBC-MAC for MAC. Those can be computed efficiently with hardware accelerators available in devices. GCM has been widely standardized e.g. by ISO/IEC 19772 [19] and NIST SP800-38D [11]. GCM is used in many practical protocols, including Ethernet security [36], WPA3 Wifi security protocol [41], IPSec [38], and TLS [35]. CCM was designed as a patent-free solution for the IEEE 802.11 standard for wireless LANs [34, 39], and was subsequently standardized as RFC 3610 [40] and NIST SP800-38C [10]. Several practical systems now use CCM, such as ZigBee [44], IPSec [5, 17], and Bluetooth [42]. In particular, TLS 1.3 defined in 2018 recommends AES-CCM [25, 32].

GCM's mu-security bound has been already identified. With a BC with n-bit block and k-bit key, Hoang et al. [16] in their work show that the GCM's mu-security bound is represented by $\frac{\sigma_{\mathrm{u}}\sigma}{2^n} + \frac{q_{\mathrm{d}}}{2^t} + \mathbf{Adv}_E^{\mathrm{muprp}}$, wherein t is the tag length, q_{d} is the number of decryption queries, σ is the total number of BC invocations in online queries (queries to the encryption and decryption oracles), σ_{u} is the upper bound of the number of BC invocations in online queries for each user, and $\mathbf{Adv}_E^{\mathrm{muprp}}$ is the mu-pseudorandom-permutation (mu-PRP) advantage of E, the underlying BC. Assuming that $\frac{q_{\mathrm{d}}}{2^t} \leq \frac{\sigma_{\mathrm{u}}\sigma}{2^n}$, the bound matches the collision finding attack on CTR and is tight regarding online security.[1]

CCM's mu-security bound is, however, still unclear. Jonsson [21] proved that CCM's su-security bound in the ideal cipher (IC) model is $\frac{\sigma^2}{2^n} + \frac{q_{\mathrm{d}}}{2^t} + \frac{p}{2^k}$, wherein p is the number of offline queries to IC. The aforementioned Internet-Draft document [15] evaluates mu-security of AES-CCM with a generic bound, i.e., an mu-bound obtained from an su-bound with a hybrid argument. The generic mu-bound is given by $\frac{u\sigma^2}{2^n} + \frac{uq_{\mathrm{d}}}{2^t} + \frac{up+u\sigma}{2^k}$, which is degraded from the su-bound by the number of users u.

Such a generic mu-bound is not guaranteed to be tight, and improving it with dedicated proofs has been the central research challenge [24, 43]. Luykx et al. [24] showed a condition on deriving an mu-bound from an su-bound without security degradation, providing the improved mu-bound of CCM, given by $\frac{u\sigma_{\mathrm{u}}^2}{2^n} + \frac{q_{\mathrm{d}}}{2^t} + \mathbf{Adv}_E^{\mathrm{muprp}}$. In the IC model, $\mathbf{Adv}_E^{\mathrm{muprp}}$ is bounded by $\frac{up+u^2}{2^k}$.

At CCS 2024, Zhang et al. [43] showed another mu-bound of CCM; $\frac{\sigma^2}{2^n} + \frac{q_{\mathrm{d}}}{2^t} + \frac{up+u\sigma}{2^k}$. They showed the tightness of the bound under some conditions. The second term $\frac{q_{\mathrm{d}}}{2^t}$ is tight with generic forgery attacks that exhaustively guess the tags. The third term $\frac{up+u\sigma}{2^k}$ corresponds to attacks with offline queries, and this is tight when the offline query overwhelms the online query, i.e., $\sigma \leq p$. In contrast, the first term $\frac{\sigma^2}{2^n}$ is proved to be tight only in the extreme case with $\sigma_{\mathrm{u}} \approx \sigma$, i.e., an adversary sends all online queries to a single user. This case is essentially equivalent that the adversary performs an su-attack even the access to multi-users is given, thus it does not demonstrate truly meaningful tightness w.r.t. mu-security.

Which of $\frac{u\sigma_{\mathrm{u}}^2}{2^n}$ and $\frac{\sigma^2}{2^n}$ is better depends on parameters u, σ_{u}, and σ; Zhang et al.'s $\frac{\sigma^2}{2^n}$ is better than Luykx et al's $\frac{u\sigma_{\mathrm{u}}^2}{2^n}$ in the extreme cases with $u \approx \sigma$ (e.g., $\sigma_{\mathrm{u}} = \sigma^{3/4}$ and $u \approx \sigma$), but Zhang et al's bound is worse in other cases, such as $\sigma \approx u\sigma_{\mathrm{u}}$, i.e. each of u users is queried with σ_{u} BC invocations. However, both bounds have critical problems. Zhang et al.'s $\frac{\sigma^2}{2^n}$ indicates that CCM's security is broken when σ reaches the birthday bound, and this cannot be avoided no matter how strong limitations are imposed on each user. Luykx et al's $\frac{u\sigma_{\mathrm{u}}^2}{2^n}$ only considers the maximum BC invocation

[1]The adversary has access to u users and makes encryption queries such that all plaintexts are zero strings and the number of plaintext blocks per user is σ_{u}, thus $\sigma = u\sigma_{\mathrm{u}}$. Since no collision occurs in the BC's outputs within the same user, the birthday analysis offers the distinguishing probability $\Omega(\frac{u\sigma_{\mathrm{u}}^2}{2^n}) = \Omega(\frac{\sigma_{\mathrm{u}}\sigma}{2^n})$.

Table 1. Mu-bounds of GCM, CCM, and their enhancements with NR and NKD. For NKD, the mu-PRF advantage of KDF is omitted in this table, since it can be negligible by choosing a KDF and its key length appropriately.

Reference	Target	Bound
Hoang et al. [16]	GCM	$\frac{\sigma_u\sigma}{2^n} + \frac{q_d}{2^t} + \frac{up+u^2}{2^k}$
Hoang et al. [16]	GCM w/ NR	$\frac{\sigma_u\sigma}{2^n} + \frac{q_d}{2^t} + \frac{dp}{2^k}$
Hoang et al. [16]	GCM w/ NR + NKD	$\frac{\sigma_n\sigma}{2^n} + \frac{q_d}{2^t} + \frac{dp}{2^k}$
Generic Bound	CCM	$\frac{u\sigma^2}{2^n} + \frac{uq_d}{2^t} + \frac{up+u\sigma}{2^k}$
Luykx et al. [24]	CCM	$\frac{u\sigma_u^2}{2^n} + \frac{q_d}{2^t} + \frac{up+u^2}{2^k}$
Zhang et al. [43]	CCM	$\frac{\sigma^2}{2^n} + \frac{q_d}{2^t} + \frac{up+u\sigma}{2^k}$
This Work	CCM	$\frac{\sigma_u\sigma}{2^n} + \frac{q_d}{2^t} + \frac{up+u^2}{2^k}$
This Work	CCM w/ NR	$\frac{\sigma_u\sigma}{2^n} + \frac{q_d}{2^t} + \frac{dp}{2^k}$
This Work	CCM w/ NR + NKD	$\frac{\sigma_n\sigma}{2^n} + \frac{q_d}{2^t} + \frac{dp}{2^k}$
This Work	TAE w/ NR + NTKD	$\frac{\sqrt{\sigma_n}\sigma}{2^n} + \frac{q_d}{2^t} + \frac{dp}{2^k}$

per user, and thus non-tight when data from some users do not reach the maximum. Moreover, both bounds are worse than GCM's mu-bound of $\frac{\sigma_u\sigma}{2^n}$ [16].

Another line of research work aims to enhance the security without disruptive changes in the scheme. For example, both GCM's and CCM's mu-offline security is already tight bounded by Biham's attack [4], which lowers the amount of offline queries to $\frac{2^k}{u}$, and there is no room for improvement as long as GCM's and CCM's specification are maintained. GCM in TLS 1.3 implements a countermeasure called nonce randomization (NR) that preprocesses the nonce without changing the GCM's implementation interface. NR uses a randomized nonce $N_{\mathrm{rand}} = N_{\mathrm{orig}} \oplus R$ with the original ν-bit nonce N_{orig} and a user-specific random mask $R \in \{0,1\}^\nu$. With this modification, Biham's attack additionally requires a collision in N_{rand} and the offline security is improved from $\frac{2^k}{u}$ to 2^k.

Bellare and Tackmann [3] proved confidentiality of NR, but the analysis is merely non-tight and did not consider integrity. Hoang et al. [16] formalized NR by introducing the d-bound model, where the number of the same randomized nonces across distinct users is bounded by d; in the d-bound and the IC models, the new mu-bound of GCM with NR becomes $\frac{\sigma_u\sigma}{2^n} + \frac{q_d}{2^t} + \frac{dp}{2^k}$, as summarized in Table 1.

Nonce-based key derivation (NKD) [14] is another method that enhances online (cf. offline) security. Note that NKD is a meaningful technique for real-world applications, and in fact, NIST recently announced their interest in revising NIST SP800-38D to standardize the combination of GCM and NKD [27]. In NKD, each pair of a randomized nonce and a key for key derivation function (KDF) generates a fresh key of an AE scheme, and combining it with the mu-bound in the d-bound model provides the following mu-bound of GCM with NKD, $\frac{\sigma_n\sigma}{2^n} + \frac{q_d}{2^t} + \frac{dp}{2^k} + \mathbf{Adv}_{\mathsf{F}}^{\mathrm{muprf}}$, where σ_n is the maximum number of BC invocations per nonce in a single user and $\mathbf{Adv}_{\mathsf{F}}^{\mathrm{muprf}}$

Table 2. Upper-bounds of BC invocations for online mu-security with concrete parameters: $n = 128$ and several per-user usage limits, (i) $\sigma_u = 2^{34.5}$, (ii) $\sigma_u = 2^{48}$, and (iii) $\sigma_u = 2^{53}$, from practical standards. σ_n is upper-bounded by about $v\ell$ and this table evaluates σ for $v = 2^{10}$ and $\ell = 2^{24}$. We approximate that $k - \log_2 d \approx k$.

Reference	Target	Online				Offline
		Generic	$\sigma_u = 2^{34.5}$	$\sigma_u = 2^{48}$	$\sigma_u = 2^{53}$	Generic
Zhang et al. [43]	CCM	$\sigma \le 2^{n/2}$	$\sigma \le 2^{64}$	$\sigma \le 2^{64}$	$\sigma \le 2^{64}$	$k - \log_2 u$
This work	CCM	$\sigma \le \frac{2^n}{\sigma_u}$	$\sigma \le 2^{93.5}$	$\sigma \le 2^{80}$	$\sigma \le 2^{75}$	$k - \log_2 u$
This work	CCM w/ NR	$\sigma \le \frac{2^n}{\sigma_u}$	$\sigma \le 2^{93.5}$	$\sigma \le 2^{80}$	$\sigma \le 2^{75}$	k
This work	CCM w/ NR + NKD	$\sigma \le \frac{2^n}{\sigma_n}$	$\sigma \le 2^{94}$			k
This work	GCM/CCM w/ NR + NTKD	$\sigma \le \frac{2^n}{\sqrt{\sigma_n}}$	$\sigma \le 2^{111}$			k

is an mu-pseudorandom-function (mu-PRF) advantage of the KDF F. Note that $\mathbf{Adv}_F^{\mathrm{muprf}}$ can be negligible by choosing a KDF and its key length appropriately. Since $\sigma_n \le \sigma_u$, NKD enhances the security of GCM. In particular, we can significantly improve security by limiting the number of decryption failures to some constant, i.e., $\sigma_n \ll \sigma_u$.

So far, enhancing methods such as NR and NKD have only been discussed for GCM, but their applicability to CCM has never been discussed. CCM is far behind GCM also in this respect.

It is also necessary to consider whether the enhanced security by NR and NKD is sufficient. The offline security term $\frac{dp}{2^k}$ is almost tight, because $k - \log_2 d \approx k$. Hence, possible concerns are on online security. Amazon AWS showed that AE schemes should allow to encrypt 2^{92} messages [22]. By combining it with the limitation of TLS 1.3 [32] that the maximum size of each message, ℓ, is 2^{10} blocks, AE schemes must be secure for $\sigma = 2^{102}$ BC invocations. Let us assume that the BC is AES having $n = 128$. With the original GCM and only with NR, the online term is $\frac{\sigma_u \sigma}{2^{128}}$. TLS 1.3 [32] limits $\sigma_u = 2^{34.5}$ BC invocations in AES-GCM,[2] and the aforementioned Internet Draft document [15] is establishing similar limits for other schemes. NIST standards have the same kind of limits: NIST SP800-38B for CMAC [12] recommends $\sigma_u = 2^{48}$ BC invocations when $n = 128$, and NIST SP800-38D for AES-GCM [11] limits $\sigma_u = 2^{53}$ BC invocations.[3] Even with the strongest limitation of $\sigma_u = 2^{34.5}$ by TLS1.3, the maximum σ is $2^{93.5}$ as shown in Table 2, which does not reach the goal of 2^{102}. When NKD is used, the online term is $\frac{\sigma_n \sigma}{2^{128}}$. Adversaries can make queries under the same nonce up to v, the number of acceptable verification failures in decryption, hence σ_n is upper-bounded by about $v\ell$. To ensure security for $\sigma = 2^{102}$ with $\ell = 2^{24}$ coming from the maximum counter size of CCM, v can be at most 4. Practical systems can limit v to a constant threshold by implementing lockdown with failed decryption attempts, however $v = 4$ is too strong limitation, which significantly lowers usability.

In summary, mu-security of CCM still falls short compared to GCM in online security and the existing enhancements, as summarized in Table 1. Moreover, the existing enhancements may not be sufficient for some practical use cases. This paper aims to fill the gaps between CCM and GCM,

[2] $2^{34.5}$ is derived from the maximum number of messages ($2^{24.5}$) and $\ell = 2^{10}$ blocks.
[3] NIST SP800-38D [11] tolerates 2^{21} messages for each key with 96-bit IV and 2^{32} blocks per message restricted by the counter length, totaling 2^{53} blocks for each key.

and to present a new enhancement method to reach an ideal mu-security level. In particular, we address the following research questions.

- What is the tight online mu-bound of CCM? Is it better or worse compared to GCM?
- Do conventional enhancing methods, i.e., NR and NKD, improve mu-security of CCM? If yes, how much?
- Is it possible to further enhance mu-security beyond NR and NKD, which works for both GCM and CCM?

1.1 Contributions

In the first part of this paper, we prove that CCM is as good as GCM with respect to mu-security for the standard model, NR, and NKD, with the following contributions.

Tight Mu-bound in the Standard Model (Section 5). We first improve the mu-bound of CCM in the standard model to

$$\frac{\sigma_{\mathrm{u}}\sigma}{2^n} + \frac{q_{\mathrm{d}}}{2^t} + \mathbf{Adv}_E^{\mathrm{muprp}} \quad .$$

The first two terms $\frac{\sigma_{\mathrm{u}}\sigma}{2^n} + \frac{q_{\mathrm{d}}}{2^t}$ represent the online security, which are better than the corresponding terms in the previous work, i.e., $\frac{\sigma^2}{2^n} + \frac{q_{\mathrm{d}}}{2^t}$ and $\frac{u\sigma_{\mathrm{u}}^2}{2^n} + \frac{q_{\mathrm{d}}}{2^t}$, for any adversary since $\sigma \le u\sigma_{\mathrm{u}}$ and $\sigma_{\mathrm{u}} \le \sigma$. The new online terms are tight, i.e. match the generic bounds of the distinguishing attack on CTR (see footnote 1 for details) and a generic forgery attack. Furthermore, with condition $\sigma_{\mathrm{u}} \ll 2^{n/2}$, which can be ensured by adequate rekeying, CCM achieves beyond-birthday-bound online security. The offline security of CCM, on the other hand, is derived from the last term $\mathbf{Adv}_E^{\mathrm{muprp}}$. This mu-PRP term offers the bound $\frac{up+u^2}{2^k}$ in the IC model, which is also tight, matching the bounds of the generic attacks [4]. In summary, the entire bound is tight, and CCM achieves the same level of mu security as GCM, as summarized in Table 1.

Enhancing Offline Security with NR *(Section 7).* Next, we prove that the mu-bound of CCM with NR in the d-bound and IC models is

$$\frac{\sigma_{\mathrm{u}}\sigma}{2^n} + \frac{q_{\mathrm{d}}}{2^t} + \frac{dp}{2^k} \quad .$$

The last term $\frac{dp}{2^k}$ represents offline security, where d is $\approx \frac{n}{\log_2 n}$ and negligible. Thus, NR enhances offline security from $\frac{k}{u}$ to k bits, making it independent of the number of users. The online security represented by the first two terms, on the other hand, is identical to that of bare CCM in the standard model, which is tight. The bound is again the same as that of GCM in the d-bound and IC models, as shown in Table 1.

Enhancing Online Security with NKD *(Section 9).* While NR enhances the offline security of CCM, the online security bound remains unchanged with the term $\frac{\sigma_{\mathrm{u}}\sigma}{2^n}$. We improve it using NKD, following the previous approach for GCM [16]. The mu-security of CCM with NKD in the d-bound and IC models is

$$\frac{\sigma_{\mathrm{n}}\sigma}{2^n} + \frac{q_{\mathrm{d}}}{2^t} + \frac{dp}{2^k} + \mathbf{Adv}_{\mathsf{F}}^{\mathrm{muprf}} \quad ,$$

obtained by replacing σ_{u} with σ_{n} and adding the mu-PRF advantage of F in the above mu-bound of CCM with NR. Although the mu-PRF advantage is added as a new offline term, it becomes negligible by choosing an appropriate KDF with sufficient key length, and the overall offline security is k bits.

The online security is enhanced under the condition that $\frac{\sigma_{\mathrm{n}}\sigma}{2^n} \ge \frac{q_{\mathrm{d}}}{2^t}$ and $\sigma_{\mathrm{n}} \ll \sigma_{\mathrm{u}}$. Because σ_{n} is upper-bounded by about $v\ell$, the online term is improved to about $\frac{v\ell\sigma}{2^n}$. The bound is the same as

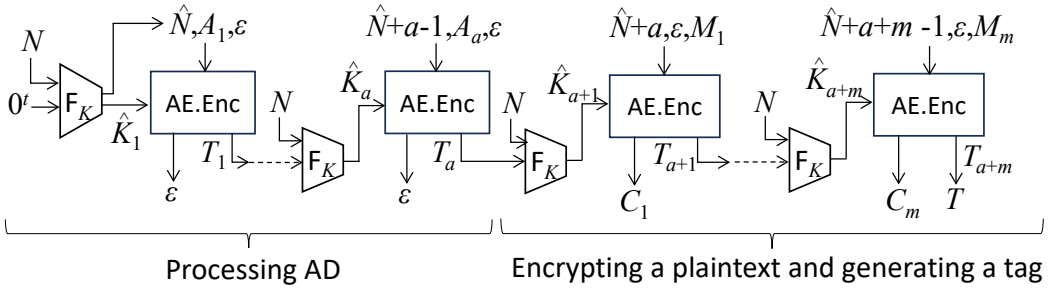

Fig. 1. Encryption of AE_NTKD. AE.Enc is the underlying tag-based AE encryption. F_K is the KDF that takes nonce and a tag and generates a key of AE.Enc (and a nonce-based IV $\hat{N}$ for the first AE.Enc call). $A_1, \ldots, A_a$ are AD sectors, $M_1, \ldots, M_m$ are plaintext sectors, $C_1, \ldots, C_m$ are ciphertext sectors, $T_1, \ldots, T_{a+m}$ are tags of AE.Enc, T is a tag of AE_NTKD. $\hat{N}$ with a counter addition is used as nonce of each AE.Enc call.

the one of GCM with NKD in the IC and d-bound models, showing that CCM achieves the same level of mu security as GCM.

*Nonce-Based and Tag-Based Key Derivation (*NTKD*) (Section 10).* In the second part of this paper, we present a new method called NTKD to further enhance online mu-security. NTKD can be applied to both GCM and CCM, and can be applied in generic for any tag-based AE: an AE such that (i) encryption generates a ciphertext C and a tag T and (ii) decryption generates T' without using T and authenticates the data by matching T and T'. The basic idea of NTKD is to separate the input data into multiple *sectors*, an appropriately parameterized number of data blocks, and apply AE to each sector by setting the key for the ith sector to an output of KDF that is computed from the nonce and the tag for $i-1$th sector (Fig. 1). This has the effect of rekeying in every sector and improves security. By setting the sector length to $\sqrt{\sigma_{\mathsf{n}}}$, the mu-bound becomes

$$\frac{\sqrt{\sigma_{\mathsf{n}}}\sigma}{2^n} + \frac{q_{\mathsf{d}}}{2^t} + \frac{dp}{2^k}.$$

Because σ_{n} is upper-bounded by about $v\ell$, with $n = 128$, $\ell = 2^{24}$, and $\sigma = 2^{102}$, security is ensured as long as $v \leq 2^{28}$, which is significantly higher than $v \leq 4$ for NKD with the same setting. Also we evaluate the value of σ that can be securely processed for some v. with $n = 128$, $v = 2^{10}$, and $\ell = 2^{24}$, NTKD ensures security up to $\sigma = 2^{111}$, while NKD ensures security up to $\sigma = 2^{94}$, which does not reach $\sigma = 2^{102}$ as shown in Table 2.

2 Notations

Let ε be the empty string, $\emptyset$ the empty set, and $\{0,1\}^*$ the set of all bit strings. For integers $i \leq j$, let $[i,j] := \{i, i+1, \ldots, j\}$ and $[j] := [1,j]$. If $i > j$ then $[i,j] := \emptyset$. For an integer $n \geq 0$, let $\{0,1\}^n$ be the set of all n-bit strings, $\{0,1\}^0 := \{\varepsilon\}$, $\{0,1\}^{\leq n} := \cup_{i\in[0,n]}\{0,1\}^i$, and $\{0,1\}^{n*} := \{X \in \{0,1\}^* \mid |X| > 0, |X| \bmod n = 0\}$. Let 0^i be the bit string of i-bit zeros. For a bit-string $D \in \{0,1\}^*$ and a positive integer n, let $|D|_n := \lceil |D|/n \rceil$ be the n-bit block length of D. For $X \in \{0,1\}^j$, let $|X| := j$. The concatenation of two bit strings X and Y is written as $X\|Y$ or XY when no confusion is possible. For integers $0 \leq j \leq i$ and $X \in \{0,1\}^i$, let $\mathrm{msb}_j(X)$ (resp. $\mathrm{lsb}_j(X)$) be the most (resp. least) significant j bits of X. For a non-empty set $\mathcal{S}$, $S \xleftarrow{\$} \mathcal{S}$ means that an element is chosen uniformly at random from $\mathcal{S}$ and assigned to S. For two sets $\mathcal{S}$ and $\mathcal{S}'$, $\mathcal{S} \xleftarrow{\cup} \mathcal{S}'$ means $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}'$. For an integer $l \geq 0$ and $X \in \{0,1\}^*$, $X_1, \ldots, X_\ell \xleftarrow{l} X$ means parsing

Algorithm 1 CTR

Encryption/Decryption $\mathsf{CTR}[E_K](N, D)$

1: $m \leftarrow |D|_n$; **for** $i = 1, \ldots, m$ **do** $X_{2,i} \leftarrow \mathsf{add}(N, i)$; $Y_{2,i} \leftarrow E_K(X_{2,i})$ **end for**
2: $KS \leftarrow \mathsf{msb}_{|D|}(Y_{2,1} \| \cdots \| Y_{2,m})$; $D' \leftarrow D \oplus KS$; **return** D'

Algorithm 2 CBC

MAC $\mathsf{CBC}[E_K](B)$

1: $b \leftarrow |B|_n$; $B_1, \ldots, B_b \xleftarrow{n} B$; $Y_{1,0} \leftarrow 0^n$
2: **for** $i = 1, \ldots, b$ **do** $X_{1,i} \leftarrow B_i \oplus Y_{1,i-1}$; $Y_{1,i} \leftarrow E_K(X_{1,i})$ **end for**
3: **return** $Y_{1,b}$

Algorithm 3 CCM

Encryption $\mathsf{CCM.Enc}[E_K](N, A, M)$

1: $B \leftarrow \mathsf{f}_{\mathsf{CCM}}(N, A, M)$; $S \leftarrow \mathsf{CBC}[E_K](B)$;
2: $X_{2,0} \leftarrow \mathsf{add}(N, 0)$ $Y_{2,0} \leftarrow E_K(X_{2,0})$
3: $T \leftarrow \mathsf{lsb}_t(S \oplus Y_{2,0})$; $C \leftarrow \mathsf{CTR}[E_K](N, M)$; **return** (C, T)

Decryption $\mathsf{CCM.Dec}[E_K](N, A, C, \widetilde{T})$

1: $M \leftarrow \mathsf{CTR}[E_K](N, C)$; $B \leftarrow \mathsf{f}_{\mathsf{CCM}}(N, A, M)$;
2: $S \leftarrow \mathsf{CBC}[E_K](B)$
3: $X_{2,0} \leftarrow \mathsf{add}(N, 0)$; $Y_{2,0} \leftarrow E_K(X_{2,0})$; $T \leftarrow \mathsf{lsb}_t(S \oplus Y_{2,0})$
4: **if** $T = \widetilde{T}$ **then return** M **else return reject end if**

of X into fixed-length l-bit strings, where if $X \neq \varepsilon$ then $X = X_1 \| \cdots \| X_\ell$, $|X_i| = l$ for $i \in [\ell - 1]$, and $0 < |X_\ell| \leq l$; if $X = \varepsilon$ then $\ell = 1$ and $X_1 = \varepsilon$. For integers $m, n \geq 0$, let $\mathsf{Func}(m, n)$ be the set of all functions from $\{0,1\}^m$ to $\{0,1\}^n$. For an integer $n \geq 0$, let $\mathsf{Perm}(n)$ be the set of all n-bit permutations. For a set $\mathcal{S}$ and $j \in [l]$, let $(y_1, \ldots, y_{j-1}, *, y_{j+1}, \ldots, y_l) \in \mathcal{S}$ be a condition that $\exists y$ s.t. $(y_1, \ldots, y_{j-1}, y, y_{j+1}, \ldots, y_l) \in \mathcal{S}$.

3 Specification of CCM

CCM is a block-cipher(BC)-based and nonce-based AE scheme with the Encrypt-and-MAC structure. The encryption part is the CTR mode and the MAC part is CBC-MAC, which is simply denoted by CBC throughout the paper.

3.1 Block Cipher (BC)

A BC is a set of permutations indexed by a key. For positive integers k and n, let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be an encryption of a BC with k-bit keys and n-bit blocks. Let $E^{-1} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be its decryption. Let $E^{\pm} := (E, E^{-1})$. E with a key K is denoted by E_K or $E(K, \cdot)$. Similarly, E^{-1} with a key K is denoted by E_K^{-1} or $E^{-1}(K, \cdot)$.

3.2 CTR Mode

CTR is a parallelizable encryption scheme with a BC E_K. The specification of CTR is given in Algorithm 1 and Fig. 2(right). Let c be a parameter for the counters. $\mathsf{CTR}[E_K] : \{0,1\}^v \times \{0,1\}^{\leq n(2^c-2)} \to \{0,1\}^{\leq n(2^c-2)}$ takes a tuple of a key K, a nonce N, and a plaintext/ciphertext D, and returns its ciphertext/plaintext D' such that $|D| = |D'|$. If D is a plaintext (resp. ciphertext), then D' is the

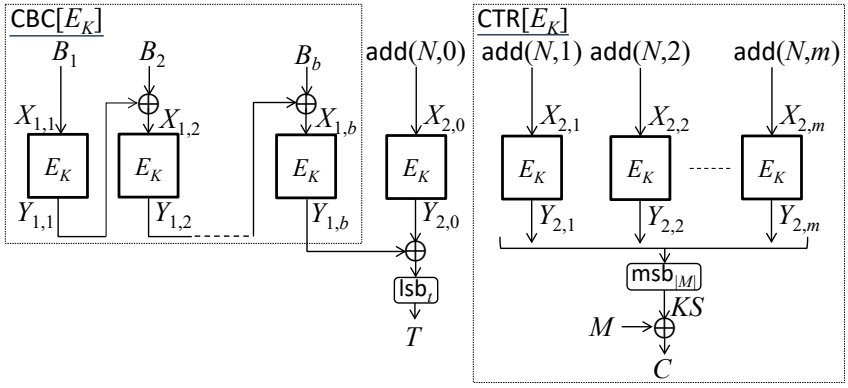

Fig. 2. The encryption of CCM, where $B \leftarrow \mathsf{f}_{\mathsf{CCM}}(N, A, M)$ and $B_1, \ldots, B_b \xleftarrow{n} B$.

ciphertext (resp. plaintext). KS is a key stream with which a ciphertext (resp. plaintext) is defined by XORing a plaintext (resp. ciphertext). add : $\{0,1\}^\nu \times [0, 2^c] \rightarrow \{0,1\}^n$ is a function that on an input pair of a nonce and a counter, returns an input block of E such that for any $N \in \{0,1\}^\nu$ and distinct values $i, j \in [0, 2^c]$, $\mathsf{add}(N, i) \neq \mathsf{add}(N, j)$. Note that "$2^c$" is reserved for the first block of CBC and "0" is reserved for masking CBC outputs.

3.3 CBC Mode

CBC is a BC-based MAC that is an iterated construction of E_K. $\mathsf{CBC}[E_K] : \{0,1\}^{n*} \rightarrow \{0,1\}^n$ takes a message B of length multiple of n, and returns an n-bit tag $Y_{1,b}$. The specification of CBC is given in Algorithm 2 and Fig. 2(left).

3.4 CCM Mode

CCM is a nonce-based AE scheme with E_K. Let ν be the nonce size such that $\nu \leq n$. Let $\mathcal{M} = \{0,1\}^{\leq n(2^c-2)}$ be plaintext/ciphertext spaces and $\mathcal{A} \subset \{0,1\}^*$ an associated data (AD) space. Let t be the tag size of CCM such that $t \leq n$. The specification of CCM is given in Algorithm 3 and Fig. 2. Let $\mathsf{f}_{\mathsf{CCM}} : \{0,1\}^\nu \times \mathcal{A} \times \mathcal{M} \rightarrow \{0,1\}^*$ be an injective formatting function that takes a nonce N, an AD A, and a plaintext M, and returns an encoded message $B = \mathsf{f}_{\mathsf{CCM}}(N, A, M)$ such that its first n-bit block is $B_1 = \mathsf{add}(N, 2^c)$, meaning that all first input blocks of CBC are distinct from all input blocks of CTR. The input blocks defined by add, namely $X_{1,1}, X_{2,0}, X_{2,1}, \ldots, X_{2,m}$, are called the nonce-dependent input blocks, and the other input blocks, $X_{1,2}, \ldots, X_{1,b}$, are called the nonce-independent input blocks.

$\mathsf{CCM.Enc}[E_K] : \{0,1\}^\nu \times \mathcal{A} \times \mathcal{M} \rightarrow \mathcal{M} \times \{0,1\}^t$ is the encryption of CCM with E_K. It accepts a nonce $N \in \{0,1\}^\nu$, an AD $A \in \mathcal{A}$, and a plaintext $M \in \mathcal{M}$, and returns a ciphertext $C \in \mathcal{M}$ such that $|C| = |M|$.

$\mathsf{CCM.Dec}[E_K] : \{0,1\}^\nu \times \mathcal{A} \times \mathcal{M} \times \{0,1\}^t \rightarrow \mathcal{M} \cup \{\textbf{reject}\}$ is the decryption of CCM with E_K. It accepts a nonce $N \in \{0,1\}^\nu$, an AD $A \in \mathcal{A}$, a cipher $C \in \mathcal{M}$, and a tag $T \in \{0,1\}^t$ and returns, deterministically, either the distinguished invalid symbol **reject** $\notin \mathcal{M}$ or a valid plaintext $M \in \mathcal{M}$.

We define a nonce extracting function $\mathsf{ext}_{\mathsf{nonce}} : \{0,1\}^n \rightarrow \{0,1\}^\nu$ that takes an n-bit input block $X \in \{0,1\}^n$ and returns a ν-bit value such that for an input block X, if $\exists N \in \{0,1\}^\nu, i \in [0, 2^c]$ s.t. $X = \mathsf{add}(N, i)$, then $\mathsf{ext}_{\mathsf{nonce}}(X) = N$.

4 Security Definitions and Proof Tools

4.1 Distinguishing Advantage

We consider distinguishing-type security notions for BCs and AEs. For the security notions, we define the following distinguishing advantage of an adversary $\mathbf{A}$ that has access to either O_1 or O_2 and returns a decision bit. For $i \in [2]$, let $\mathbf{A}^{O_i} = 1$ be an event that $\mathbf{A}$ with O_i returns 1. Then, the distinguishing advantage of $\mathbf{A}$ is defined as $\mathbf{Adv}^{\mathsf{dist}}_{O_1,O_2}(\mathbf{A}) \coloneqq \Pr\left[\mathbf{A}^{O_1} = 1\right] - \Pr\left[\mathbf{A}^{O_2} = 1\right]$.

4.2 Security Models for BCs

In the mu-security proofs of CCM, we consider two models for BCs: the standard multi-user-pseudorandom-permutation (mu-PRP) security and the ideal cipher (IC) models.

4.2.1 Standard Model. In the standard model, the underlying BCs are assumed to be mu-PRP secure, where BC instantiations with independent keys are securely replaced with independent random permutations (RPs). Let u be the number of users. In the mu-PRP game, an adversary interacts with either the real-world oracles $(E_{K_1}, \ldots, E_{K_u})$ or the ideal-world oracles $(P_1, \ldots, P_u)$, where $\forall \omega \in [u] : K_\omega \xleftarrow{\$} \{0,1\}^k$ and RPs are defined as $\forall \omega \in [u] : P_\omega \xleftarrow{\$} \mathsf{Perm}(n)$. At the end of this game, $\mathbf{A}$ returns a decision bit in $\{0,1\}$. The mu-PRP advantage function of $\mathbf{A}$ is defined as

$$\mathbf{Adv}^{\mathsf{muprp}}_{E}(\mathbf{A}) \coloneqq \mathbf{Adv}^{\mathsf{dist}}_{(E_{K_1},\ldots,E_{K_u}),(P_1,\ldots,P_u)}(\mathbf{A}) \ .$$

For all possible adversaries $\mathbf{A}$ that have access to u users, make at most q queries, and run in time τ, the maximum advantage is defined as $\mathbf{Adv}^{\mathsf{muprp}}_{E}(u,q,\tau) \coloneqq \max_{\mathbf{A}} \mathbf{Adv}^{\mathsf{muprp}}_{E}(\mathbf{A})$.

4.2.2 Ideal Cipher (IC) Model. Let $\mathcal{BC}$ be the set of all encryptions of k-bit key and n-bit block BCs. An IC is an ideal BC and defined as $E \xleftarrow{\$} \mathcal{BC}$. In the IC model, all parties including CCM oracles and adversaries obtain IC's outputs by accessing an IC $E^{\pm} = (E, E^{-1})$.

4.3 Security Models for CCM

Multi-user-AE (mu-AE) security is the indistinguishability between the real and ideal worlds. Let u be the number of users. Let $\$_\omega$ be a random-bit oracle of the ω-th user that takes an input tuple (N, A, M) of a nonce, an AD, and a plaintext, and returns a pair of a random ciphertext and a tag defined as $(C, T) \xleftarrow{\$} \{0,1\}^{|\mathsf{CCM.Enc}[E_K](N,A,M)|}$. Let $\perp_\omega$ be a reject oracle that returns **reject** for any query. Let $K_1, \ldots, K_u$ be users' keys defined as $K_\omega \xleftarrow{\$} \{0,1\}^k$ for each $\omega \in [u]$. In the mu-AE game in the standard or IC model, an adversary $\mathbf{A}$ has access to either real-world oracles O_{real} or ideal-world oracles O_{ideal} defined as follows.

Standard Model: $O_{\mathsf{real}} \coloneqq (\mathsf{CCM}[E_{K_1}], \ldots, \mathsf{CCM}[E_{K_u}])$
$O_{\mathsf{ideal}} \coloneqq ((\$_1, \perp_1), \ldots, (\$_u, \perp_u))$.

IC Model: $O_{\mathsf{real}} \coloneqq (\mathsf{CCM}[E_{K_1}], \ldots, \mathsf{CCM}[E_{K_u}], E^{\pm})$
$O_{\mathsf{ideal}} \coloneqq ((\$_1, \perp_1), \ldots, (\$_u, \perp_u), E^{\pm})$.

At the end of this game, $\mathbf{A}$ return a decision bit in $\{0,1\}$. The mu-AE-security advantage function of $\mathbf{A}$ is defined as

$$\mathbf{Adv}^{\mathsf{muae}}_{\mathsf{CCM}}(\mathbf{A}) \coloneqq \mathbf{Adv}^{\mathsf{dist}}_{O_{\mathsf{real}},O_{\mathsf{ideal}}}(\mathbf{A}) \ .$$

Queries to each user are called online queries. Queries to encryption oracles $\mathsf{CCM.Enc}[E_{K_\omega}]$ or $\$_\omega$ (resp. decryption oracles $\mathsf{CCM.Dec}[E_{K_\omega}]$ or $\perp_\omega$) are called encryption (resp. decryption) queries. In the IC model, Queries to an IC are called offline queries, and offline queries to E (resp. E^{-1}) are called forward (resp. inverse) queries.

We consider nonce-respecting adversaries where for each user, all nonces in encryption queries are distinct. In this game, making a repeated query and a trivial decryption query is forbidden, where the trivial query $(N, A, C, \overline{T})$ is such that the query tuple was obtained by some previous encryption query to the same user.

4.3.1 Adversaries and Its Resources. In our proofs, we consider computationally-bounded and/or computationally-unbounded adversaries. Queries to encryption oracles $\mathsf{CCM.Enc}[E_{K_\omega}]$ or $\$_\omega$ (resp. decryption oracles $\mathsf{CCM.Dec}[E_{K_\omega}]$ or $\perp_\omega$) are called encryption (resp. decryption) queries. Let q_e be the number of encryption queries, q_d be the number of decryption queries, and σ be the number of BC invocations in online queries. and σ_ω the number of BC invocations in online queries to the ω-th user such that $\sum_{\omega\in[u]}\sigma_\omega = \sigma$. For computationally-bounded (resp. computationally-unbounded) adversaries, the time resources are expressed by its running time τ (resp. the number of offline queries to an IC denoted by p). Let σ_u be the maximum number of BC invocations per user, i.e., $\forall\omega \in [u] : \sigma_\omega \le \sigma_\mathsf{u}$.

Let $\mathcal{A}_\mathsf{sm}$ (resp. $\mathcal{A}_\mathsf{icm}$) be the set of all possible adversaries in the standard (resp. IC) model with the above resources.

4.4 Coefficient-H Technique

The distinguishing advantage of $\mathbf{A}$ with access to either O_1 or O_2 can be upper-bounded by using Patarin's coefficient-H technique [31]. A set of values that an adversary obtains in the security game is called a "transcript." For $i \in [2]$, let T_i be a transcript obtained by random samples of O_i. We call a transcript τ *valid* if $\Pr[\mathsf{T}_2 = \tau] > 0$. Let $\mathcal{T}$ be the set of all valid transcripts such that $\forall\tau \in \mathcal{T} : \Pr[\mathsf{T}_1 = \tau] > \Pr[\mathsf{T}_2 = \tau]$. Then, we have $\mathbf{Adv}^{\mathsf{dist}}_{O_1,O_2}(\mathbf{A}) \le \mathrm{SD}(\mathsf{T}_1, \mathsf{T}_2) := \sum_{\tau\in\mathcal{T}}(\Pr[\mathsf{T}_1 = \tau] - \Pr[\mathsf{T}_2 = \tau])$.

The statistical distance $\mathrm{SD}(\mathsf{T}_1, \mathsf{T}_2)$ can be bounded by using the coefficient-H technique [31].

Lemma 4.1. *Let $\mathcal{T}_\mathsf{good}$ and $\mathcal{T}_\mathsf{bad}$ be good and bad transcripts into which $\mathcal{T}$ is partitioned. If $\forall\tau \in \mathcal{T}_\mathsf{good} : \frac{\Pr[\mathsf{T}_1=\tau]}{\Pr[\mathsf{T}_2=\tau]} \ge 1-\varepsilon$ s.t. $0 \le \varepsilon \le 1$, then $\mathrm{SD}(\mathsf{T}_1, \mathsf{T}_2) \le \Pr[\mathsf{T}_2 \in \mathcal{T}_\mathsf{bad}] + \varepsilon$.*

Hence, we can obtain an upper-bound of $\mathbf{Adv}^{\mathsf{dist}}_{O_1,O_2}(\mathbf{A})$ by (1) defining good and bad transcripts; (2) upper-bounding $\Pr[\mathsf{T}_2 \in \mathcal{T}_\mathsf{bad}]$; and (3) lower-bounding $\frac{\Pr[\mathsf{T}_1=\tau]}{\Pr[\mathsf{T}_2=\tau]}$ for $\forall\tau \in \mathcal{T}_\mathsf{good}$.

4.5 Definitions for Proofs

In our proofs, we use the following notations and definitions.

- For $\alpha \in [p]$, let $(\hat{K}^{(\alpha)}, \hat{X}^{(\alpha)}, \hat{Y}^{(\alpha)})$ be the α-th offline query-response tuples such that $\hat{Y}^{(\alpha)} = E(\hat{K}^{(\alpha)}, \hat{X}^{(\alpha)})$.
- For $\omega \in [u]$ and $\alpha \in [q]$, ω is called "user index" and α is called "query index."
- For $\alpha \in [q]$, values corresponding with the α-th query are denoted by using the superscript symbol of (α) such as $M^{(\alpha)}, C^{(\alpha)}, N^{(\alpha)}, A^{(\alpha)}$, etc.
- The lengths b and m for the α-th online query are denoted by b_α and m_α, respectively.
- For $\alpha \in [q]$, let $\mathsf{u}_\alpha \in [u]$ be the user index for the α-th online query. If an α-th online query is to an ω-th user, then $\mathsf{u}_\alpha = \omega$.
- Let $\mathcal{Q}_\mathsf{Enc} \subseteq [q]$ (resp. $\mathcal{Q}_\mathsf{Dec} \subseteq [q]$) be the set of encryption (resp. decryption) query indexes.
- Let $\mathcal{Q}^{[\omega]}_\mathsf{Enc} \subseteq \mathcal{Q}_\mathsf{Enc}$ (resp. $\mathcal{Q}^{[\omega]}_\mathsf{Dec} \subseteq \mathcal{Q}_\mathsf{Dec}$) be the set of encryption (resp. decryption) query indexes of the ω-th user.
- Let $\mathcal{Q}^{[\omega]} := \mathcal{Q}^{[\omega]}_\mathsf{Enc} \cup \mathcal{Q}^{[\omega]}_\mathsf{Dec}$ be the set of online query indexes of the ω-th user.
- For $\alpha \in [q]$, let $\mathsf{Index}^{(\alpha)} := (\{1\} \times [b_\alpha]) \cup (\{2\} \times [0, m_\alpha])$ be the set of indexes of input-output pairs in the α-th online query.

- Let $\mathcal{X}^{[\omega]} := \{X_{i,j}^{(\alpha)} \mid \alpha \in \mathcal{Q}^{[\omega]}, (i,j) \in \mathsf{Index}^{(\alpha)}\}$ be all input blocks for the ω-th user.
- Let $\mathcal{X}_{\mathsf{Enc}}^{[\omega]} := \{X_{i,j}^{(\alpha)} \mid \alpha \in \mathcal{Q}_{\mathsf{Enc}}^{[\omega]}, (i,j) \in \mathsf{Index}^{(\alpha)}\}$ be all input blocks for encryption queries to the ω-th user.
- Let $\mathcal{X}_{2}^{[\omega]} := \{(X_{2,j}^{(\gamma)}, Y_{2,j}^{(\gamma)}) \mid \gamma \in \mathcal{Q}^{[\omega]}, j \in [0, m_\gamma]\}$ be all input-output pairs defined in CTR and the tag generation of the ω-th user.
- Let $\mathcal{X}_{\mathsf{N}}^{[\omega]} := \{X_{1,1}^{(\alpha)}, X_{2,1}^{(\alpha)}, \ldots, X_{2,m_\alpha}^{(\alpha)} \mid \alpha \in \mathcal{Q}^{[\omega]}\}$ be the set of nonce-dependent input blocks for the ω-th user.
- Let $\mathcal{X}_{\neq\mathsf{N}}^{[\omega]} := \{X_{1,2}^{(\alpha)}, \ldots, X_{1,b_\alpha}^{(\alpha)} \mid \alpha \in \mathcal{Q}^{[\omega]}\}$ be the set of nonce-independent input blocks for the ω-th user.
- Let $\mathcal{Y}^{[\omega]} := \{Y_{i,j}^{(\alpha)} \mid \alpha \in \mathcal{Q}^{[\omega]}, (i,j) \in \mathsf{Index}^{(\alpha)}\}$ be the set of output blocks for the ω-th user.
- We call "a query phase" a phase that an adversary makes queries to its oracles and "a decision phase" a phase after finishing all queries and before outputting a decision bit.

5 Mu-Security of CCM in the Standard Model

In this section, we show an mu-bound of CCM in the standard model, followed by the security proof.

5.1 Security Bound

Theorem 5.1. $\forall \mathbf{A} \in \mathcal{A}_{\mathsf{sm}}$:

$$\mathbf{Adv}^{\mathsf{muae}}_{\mathsf{CCM}}(\mathbf{A}) \le \mathbf{Adv}^{\mathsf{muprp}}_{E}(u, \sigma, \tau + O(\sigma)) + \sum_{\omega \in [u]} \frac{\sigma_\omega^2}{2^n} + \frac{q_{\mathsf{d}}}{2^t} \quad .$$

With the parameters σ and σ_{u}, $\forall \mathbf{A} \in \mathcal{A}_{\mathsf{sm}}$:

$$\mathbf{Adv}^{\mathsf{muae}}_{\mathsf{CCM}}(\mathbf{A}) \le \mathbf{Adv}^{\mathsf{muprp}}_{E}(u, \sigma, \tau + O(\sigma)) + \frac{\sigma_{\mathsf{u}}\sigma}{2^n} + \frac{q_{\mathsf{d}}}{2^t} \quad .$$

5.2 Proof of Theorem 5.1

Without loss of generality, assume that $\mathbf{A}$ is deterministic. Let σ_ω be the number of BC calls in online queries to the ω-th user, where $\sigma_\omega \le \sigma_{\mathsf{u}}$. In the following evaluation, we consider four games.

Real World → Game 2. We start the proof from the real world, followed by Game 2. In the real world, $\mathbf{A}$ has access to $\mathcal{O}_{\mathsf{real}}$. From the real world to Game 2, the u BCs $(E_{K_\omega})_{\omega\in[u]}$ are replaced with u RPs $(P_\omega)_{\omega\in[u]}$, where $\forall\omega \in [u] : P_\omega \xleftarrow{\$} \mathsf{Perm}(n)$. Hence, in Game 2, $\mathbf{A}$ has access to the modified oracles $\mathcal{O}_2 := (\mathsf{CCM}[P_\omega])_{\omega\in[u]}$. The BC-RP switch yields the following bound. $\mathbf{Adv}^{\mathsf{dist}}_{\mathcal{O}_{\mathsf{real}},\mathcal{O}_2}(\mathbf{A}) \le \mathbf{Adv}^{\mathsf{muprp}}_{E}(\sigma, \tau + O(\sigma))$.

Game 2 → Game 3. We next consider Game 3. Hereafter, we consider a computationally-unbounded adversary $\mathbf{A}$. From Game 2 to Game 3, the RPs $(P_\omega)_{\omega\in[u]}$ are replaced with random functions (RFs) $(\mathcal{R}_\omega)_{\omega\in[u]}$, where $\forall\omega \in [u] : \mathcal{R}_\omega \xleftarrow{\$} \mathsf{Func}(n,n)$. Hence, in Game 3, $\mathbf{A}$ has access to the modified oracles $\mathcal{O}_3 := (\mathsf{CCM}[\mathcal{R}_\omega])_{\omega\in[u]}$. For each $\omega \in [u]$, a RF $\mathcal{R}_\omega$ is the same as a RP as long as no output collision occurs, and the collision probability is $\binom{\sigma_\omega}{2} \cdot \frac{1}{2^n} \le \frac{0.5\sigma_\omega^2}{2^n}$. Hence, by the RP-RF switch, we have $\mathbf{Adv}^{\mathsf{dist}}_{\mathcal{O}_2,\mathcal{O}_3}(\mathbf{A}) \le \sum_{\omega\in[u]} \frac{0.5\sigma_\omega}{2^n}$.

Game 3 → Ideal World. Finally, we evaluate the difference between Game 3 and the ideal world. We derive the following bound.

Lemma 5.2. *For any computationally-unbounded adversary* $\mathbf{A}$*,*

$$\mathbf{Adv}^{\mathsf{dist}}_{O_3,O_{\mathsf{ideal}}}(\mathbf{A}) \le \frac{q_{\mathsf{d}}}{2^t} + \sum_{\omega\in[u]} \frac{0.5\sigma_\omega^2}{2^n} \ .$$

where $O_3 = (\mathrm{CCM}[\mathcal{R}_\omega])_{\omega\in[u]}$ *and* $O_{\mathsf{ideal}} = (\$_\omega, \bot_\omega)_{\omega\in[u]}$.

Hereafter, we provide a high-level overview of the proof of Lemma 5.2, and the formal proof is given in Section 6.

5.2.1 Proof of Lemma 5.2 (Overview). We first consider encryption queries in Game 3. For each user, all input blocks $X^{(\alpha)}_{2,i}$ in CTR are distinct, and the outputs $Y^{(\alpha)}_{2,i}$ are chosen independently and uniformly at random from $\{0,1\}^n$. Hence, the responses $(C^{(\alpha)}, T^{(\alpha)})$ to the encryption queries are indistinguishable from those defined by $\$_\omega$ in the ideal world.

The remaining work is to evaluate the difference of responses to decryption queries between Game 3 and the ideal world. In Game 3, for some response of the decryption query, a valid plaintext ($\neq$ **reject**) is probabilistically returned, and we have

$$\mathbf{Adv}^{\mathsf{dist}}_{O_3,O_{\mathsf{ideal}}}(\mathbf{A}) \le \Pr[\exists\beta \in \mathcal{Q}_{\mathsf{Dec}} \text{ s.t. } T^{(\beta)} = \widetilde{T}^{(\beta)}].$$

We evaluate the probability by using the following event.

$$\mathsf{coll}_{X_{1,b}} \Leftrightarrow \exists\beta \in \mathcal{Q}_{\mathsf{Dec}} \text{ s.t. } X^{(\beta)}_{1,b_\beta} \in \mathcal{X}^{[\mathsf{u}_\beta]}_{\mathsf{Enc}}.$$

The event means that for some decryption query, the last input block in CBC collides with some input block defined by the encryption query. In other worlds, if the event does not occur, then all tags $T^{(\beta)}$ are defined independently of the responses of the encryption queries. Thus, we have $\Pr[\exists\beta \in \mathcal{Q}_{\mathsf{Dec}} \text{ s.t. } T^{(\beta)} = \widetilde{T}^{(\beta)} \mid \neg\mathsf{coll}_{X_{1,b}}] \le \frac{q_{\mathsf{d}}}{2^t}$, and

$$\begin{aligned}\mathbf{Adv}^{\mathsf{dist}}_{O_3,O_{\mathsf{ideal}}}(\mathbf{A}) &\le \Pr[\exists\beta \in \mathcal{Q}_{\mathsf{Dec}} \text{ s.t. } T^{(\beta)} = \widetilde{T}^{(\beta)} \mid \neg\mathsf{coll}_{X_{1,b}}] + \Pr[\mathsf{coll}_{X_{1,b}}] \\ &\le \frac{q_{\mathsf{d}}}{2^t} + \Pr[\mathsf{coll}_{X_{1,b}}] \ .\end{aligned}$$

We evaluate the probability $\Pr[\mathsf{coll}_{X_{1,b}}]$. By the iterated structure of CBC and the property of add,[4] the event $\mathsf{coll}_{X_{1,b}}$ implies that there exists $\beta \in \mathcal{Q}_{\mathsf{Dec}}$, and $j \in [b_\beta]$ such that the j-th CBC input block is $X^{(\beta)}_{1,j} \in \mathcal{X}^{[\mathsf{u}_\beta]}_{\mathsf{Enc}}$ but the previous input block is $X^{(\beta)}_{1,j-1} \notin \mathcal{X}^{[\mathsf{u}_\beta]}_{\mathsf{Enc}}$. The j-th input block has the form of $X^{(\beta)}_{1,j} = B^{(\beta)}_j \oplus Y^{(\beta)}_{1,j-1}$ and $Y^{(\beta)}_{1,j-1}$ is chosen independently of all output blocks for the encryption queries. Using the randomness of $Y^{(\beta)}_{1,j-1}$, we have the following birthday bound: $\Pr[\mathsf{coll}_{X_{1,b}}] \le \sum_{\omega\in[u]} \binom{\sigma_\omega}{2} \cdot \frac{1}{2^n} \le \sum_{\omega\in[u]} \frac{0.5\sigma_\omega^2}{2^n}$.

By using the above bounds, we obtain the bound in Lemma 5.2. Note that the formal proof given in Appendix 6 uses the coefficient-H technique and these events are evaluated in the ideal world by introducing dummy internal input-output blocks.

[End of Proof of Lemma 5.2 (Overview)] ■

[4] The property of add ensures that for each $\omega \in [u]$, $\alpha \in \mathcal{Q}^{[\omega]}_{\mathsf{Enc}}$, and $\beta \in \mathcal{Q}^{[\omega]}_{\mathsf{Dec}}$, the messages $B^{(\alpha)}$ and $B^{(\beta)}$ of CBC are distinct and the first input block $B^{(\beta)}_1$ is distinct from all input blocks in CTR, offering the condition $\exists\beta, j$ s.t. $(X^{(\beta)}_{1,j-1} \notin \mathcal{X}^{[\mathsf{u}_\beta]}_{\mathsf{Enc}}) \wedge (X^{(\beta)}_{1,j} \in \mathcal{X}^{[\mathsf{u}_\beta]}_{\mathsf{Enc}})$.

Conclusion of the Proof. By using these bounds, we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{muae}}_{\mathsf{CCM}}(\mathbf{A}) &= \mathbf{Adv}^{\mathsf{dist}}_{O_{\mathsf{real}},O_2}(\mathbf{A}) + \mathbf{Adv}^{\mathsf{dist}}_{O_2,O_3}(\mathbf{A}) + \mathbf{Adv}^{\mathsf{dist}}_{O_3,O_{\mathsf{ideal}}}(\mathbf{A}) \\ &\le \mathbf{Adv}^{\mathsf{muprp}}_{E}(u,\sigma,\tau+O(\sigma)) + \sum_{\omega\in[u]} \frac{0.5\sigma_\omega^2}{2^n} + \frac{q_{\mathsf{d}}}{2^t} \ .\end{aligned}$$

[End of Proof of Theorem 5.1] ■

6 Proof of Lemma 5.2

We derive the following bound by using the coefficient-H technique (See Section 4.4). For any computationally-unbounded adversary **A**,

$$\mathbf{Adv}^{\mathsf{dist}}_{O_3,O_{\mathsf{ideal}}}(\mathbf{A}) \le \frac{q_{\mathsf{d}}}{2^t} + \sum_{\omega\in[u]} \frac{0.5\sigma_\omega^2}{2^n} \ ,$$

where $O_3 = (\mathsf{CCM}[\mathcal{R}_\omega])_{\omega\in[u]}$ and $O_{\mathsf{ideal}} = (\$_\omega, \perp_\omega)_{\omega\in[u]}$. Let T_3 (resp. T_I) be a transcript obtained by random samplings of O_3 (resp. O_{ideal}).

6.1 Extended Transcript

In this poof, we permit **A** to obtain all input-output pairs $\{(X^{(\alpha)}_{i,j}, Y^{(\alpha)}_{i,j}) \mid \alpha \in [q], (i,j) \in \mathsf{Index}^{(\alpha)}\}$ in the decision phase. In the ideal world, the (dummy) internal pairs are defined by using Algorithm 4 in the decision phase. Note that giving the additional pairs does not reduce the **A**'s advantage, since **A** can ignore the additional pairs. Thus, the (extended) transcript τ consists of

- encryption query-responses $(N^{(\alpha)}, A^{(\alpha)}, M^{(\alpha)}, C^{(\alpha)}, T^{(\alpha)})$ for $\alpha \in \mathcal{Q}_{\mathsf{Enc}}$,
- decryption query-responses $(N^{(\alpha)}, A^{(\alpha)}, C^{(\alpha)}, \widetilde{T}^{(\alpha)}, RV^{(\alpha)})$ for $\alpha \in \mathcal{Q}_{\mathsf{Dec}}$, where $RV^{(\alpha)} \in \mathcal{M} \cup \{\mathbf{reject}\}$ is the response to the α-th decryption query, and
- (dummy) internal pairs $\{(X^{(\alpha)}_{i,j}, Y^{(\alpha)}_{i,j}) \mid \alpha \in [q], (i,j) \in \mathsf{Index}^{(\alpha)}\}$.

We explain Algorithm 4. The algorithm define dummy input-output pairs $\{(X^{(\alpha)}_{i,j}, Y^{(\alpha)}_{i,j}) \mid \alpha \in [q], (i,j) \in \mathsf{Index}^{(\alpha)}\}$ by using online query-response tuples. First, the algorithm initializes tables R_ω for $\omega \in [u]$ that will keep dummy input-output pairs. Second, in Steps 2-18, dummy input-output pairs for encryption queries are defined according to the structure of CCM.Enc. Finally, in Steps 19-38, dummy input-output pairs for decryption queries are defined according to the structure of CCM.Dec.

For the α-th encryption query, in Steps 4-7, dummy input-output pairs in CTR are defined with the relation $KS^{(\alpha)} = M^{(\alpha)} \oplus C^{(\alpha)}$. If $|M^{(\alpha)}| \bmod n \neq 0$, then the last block is extended to n bits by appending a random-bit string KS^*. In Steps 8-14, dummy input-output pairs in CBC are defined, where each output block is randomly chosen if the input is new. In Steps 15-17, a dummy input-output pair $(X^{(\alpha)}_{2,0}, Y^{(\alpha)}_{2,0})$ for a tag are defined by using the relation $T^{(\alpha)} = \mathsf{lsb}_t(Y^{(\alpha)}_{1,b_\alpha} \oplus Y^{(\alpha)}_{2,0})$. If $t < n$, then the truncated $(n-t)$ bits are randomly chosen.

Similarly, the dummy input-output pairs of the α-th decryption query are defined according to the structure of CCM.Dec. Note that in the (original) ideal world, the tag $T^{(\alpha)}$ is not introduced, thus in Steps 35-37, $Y^{(\alpha)}_{2,0}$ is randomly chosen and the dummy tag is defined as $T^{(\alpha)} = \mathsf{lsb}_t(Y^{(\alpha)}_{1,b_\alpha} \oplus Y^{(\alpha)}_{2,0})$. Also note that for a repeated input block, the output is equal to the previous output by using the table R_ω

Algorithm 4 Procedure to define internal values in the ideal world

for $\omega \in [u], X \in \{0,1\}^n$ **do** $\mathsf{R}_\omega[X] \leftarrow \varepsilon$ **end for** ▷ Initialization of the RF's tables
// Defining dummy input-output pairs for encryption queries
for $\alpha \in Q_{\mathsf{Enc}}$ **do**
  // Defining dummy input-output pairs $(X^{(\alpha)}_{2,i}, Y^{(\alpha)}_{2,i})$ in CTR s.t. $i \neq 0$
  $m_\alpha \leftarrow |M^{(\alpha)}|_n$; $KS^{(\alpha)} \leftarrow M^{(\alpha)} \oplus C^{(\alpha)}$; $\omega \leftarrow \mathsf{u}_\alpha$
  $KS^* \xleftarrow{\$} \{0,1\}^{m_\alpha n - |M^{(\alpha)}|}$; $Y^{(\alpha)}_{2,1}, \ldots, Y^{(\alpha)}_{2,m_\alpha} \xleftarrow{n} KS^{(\alpha)} \| KS^*$
  for $i \in [m_\alpha]$ **do** $X^{(\alpha)}_{2,i} \leftarrow \mathsf{add}(N^{(\alpha)}, i)$; $\mathsf{R}_\omega[X^{(\alpha)}_{2,i}] \leftarrow Y^{(\alpha)}_{2,i}$ **end for**
  // Defining dummy input-output pairs $(X^{(\alpha)}_{1,i}, Y^{(\alpha)}_{1,i})$ in CBC
  $B^{(\alpha)} \leftarrow \mathsf{f}_{\mathsf{CCM}}(N^{(\alpha)}, A^{(\alpha)}, M^{(\alpha)})$; $b_\alpha \leftarrow |B^{(\alpha)}|_n$; $B^{(\alpha)}_1, \ldots, B^{(\alpha)}_{b_\alpha} \xleftarrow{n} B^{(\alpha)}$
  $Y^{(\alpha)}_{1,0} \leftarrow 0^n$
  for $i \in [b_\alpha]$ **do**
    $X^{(\alpha)}_{1,i} \leftarrow B^{(\alpha)}_1 \oplus Y^{(\alpha)}_{1,i-1}$; **if** $\mathsf{R}_\omega[X^{(\alpha)}_{1,i}] = \varepsilon$ **then** $\mathsf{R}_\omega[X^{(\alpha)}_{1,i}] \xleftarrow{\$} \{0,1\}^n$ **end if**
    $Y^{(\alpha)}_{1,i} \leftarrow \mathsf{R}_\omega[X^{(\alpha)}_{1,i}]$
  end for
  // Defining a dummy input-output pair $(X^{(\alpha)}_{2,0}, Y^{(\alpha)}_{2,0})$ for a tag
  $T^* \xleftarrow{\$} \{0,1\}^{n-t}$; $X^{(\alpha)}_{2,0} \leftarrow \mathsf{add}(N^{(\alpha)}, 0)$; $Y^{(\alpha)}_{2,0} \leftarrow Y^{(\alpha)}_{1,b_\alpha} \oplus (T^* \| T^{(\alpha)})$
  $\mathsf{R}_\omega[X^{(\alpha)}_{2,0}] \leftarrow Y^{(\alpha)}_{2,0}$
end for
// Defining dummy input-output pairs for decryption queries
for $\alpha \in Q_{\mathsf{Dec}}$ **do**
  // Defining dummy input-output pairs $(X^{(\alpha)}_{2,i}, Y^{(\alpha)}_{2,i})$ in CTR s.t. $i \neq 0$
  $m_\alpha \leftarrow |C^{(\alpha)}|_n$; $\omega \leftarrow \mathsf{u}_\alpha$
  for $i \in [m_\alpha]$ **do**
    $X^{(\alpha)}_{2,i} \leftarrow \mathsf{add}(N^{(\alpha)}, i)$; **if** $\mathsf{R}_\omega[X^{(\alpha)}_{2,i}] = \varepsilon$ **then** $\mathsf{R}_\omega[X^{(\alpha)}_{2,i}] \xleftarrow{\$} \{0,1\}^n$ **end if**
    $Y^{(\alpha)}_{2,i} \leftarrow \mathsf{R}_\omega[X^{(\alpha)}_{2,i}]$
  end for
  $M^{(\alpha)} \leftarrow C^{(\alpha)} \oplus \mathsf{msb}_{|C^{(\alpha)}|}\left(Y^{(\alpha)}_{2,1} \| \cdots \| Y^{(\alpha)}_{2,m_\alpha}\right)$
  // Defining dummy input-output pairs $(X^{(\alpha)}_{1,i}, Y^{(\alpha)}_{1,i})$ in CBC
  $B^{(\alpha)} \leftarrow \mathsf{f}_{\mathsf{CCM}}(N^{(\alpha)}, A^{(\alpha)}, M^{(\alpha)})$; $b_\alpha \leftarrow |B^{(\alpha)}|_n$; $B^{(\alpha)}_1, \ldots, B^{(\alpha)}_{b_\alpha} \xleftarrow{n} B^{(\alpha)}$
  $Y^{(\alpha)}_{1,0} \leftarrow 0^n$
  for $i \in [b_\alpha]$ **do**
    $X^{(\alpha)}_{1,i} \leftarrow B^{(\alpha)}_1 \oplus Y^{(\alpha)}_{1,i-1}$; **if** $\mathsf{R}_\omega[X^{(\alpha)}_{1,i}] = \varepsilon$ **then** $\mathsf{R}_\omega[X^{(\alpha)}_{1,i}] \xleftarrow{\$} \{0,1\}^n$ **end if**
    $Y^{(\alpha)}_{1,i} \leftarrow \mathsf{R}_\omega[X^{(\alpha)}_{1,i}]$
  end for
  // Defining a dummy input-output pair $(X^{(\alpha)}_{2,0}, Y^{(\alpha)}_{2,0})$ for a tag
  $X^{(\alpha)}_{2,0} \leftarrow \mathsf{add}(N^{(\alpha)}, 0)$; **if** $\mathsf{R}_\omega[X^{(\alpha)}_{2,0}] = \varepsilon$ **then** $\mathsf{R}_\omega[X^{(\alpha)}_{2,0}] \xleftarrow{\$} \{0,1\}^n$ **end if**
  $Y^{(\alpha)}_{2,0} \leftarrow \mathsf{R}_\omega[X^{(\alpha)}_{2,0}]$; $T^{(\alpha)} \leftarrow \mathsf{lsb}_t(Y^{(\alpha)}_{1,b_\alpha} \oplus Y^{(\alpha)}_{2,0})$
end for

6.2 Bad and Good Transcripts

We define conditions $\mathsf{coll}_{X_{1,b}}$ and forge on bad transcripts. The set of bad transcripts $\mathcal{T}_{\mathsf{bad}}$ is a subset of $\mathcal{T}$ such that one of the conditions holds. The set of good transcripts is defined as $\mathcal{T}_{\mathsf{good}} := \mathcal{T} \setminus \mathcal{T}_{\mathsf{bad}}$, which is the subset of $\mathcal{T}$ such that the conditions do not hold.

For encryption queries, all output blocks $Y_{1,j}^{(\alpha)}$ $(j \geq 1)$ are independently chosen, and pairs of ciphertext and tag in Game 3 are indistinguishable from those in the ideal world. On the other hand, responses to decryption queries between Game 3 and the ideal world are probabilistically distinct, since in Game 3, some response is not **reject**. The condition forge is defined so that if the condition does not hold, responses to decryption queries are all **reject** and the condition $\mathsf{coll}_{X_{1,b}}$ is defined to support the evaluation of forge.

The first condition $\mathsf{coll}_{X_{1,b}}$ is defined as follows.

$$\mathsf{coll}_{X_{1,b}} \Leftrightarrow \exists \beta \in \mathcal{Q}_{\mathsf{Dec}} \text{ s.t. } X_{1,b_\beta}^{(\beta)} \in \mathcal{X}_{\mathsf{Enc}}^{[\mathsf{u}_\beta]}.$$

$\mathsf{coll}_{X_{1,b}}$ ensures that if the condition does not hold, all last input blocks in CBC are new, and the output blocks can be seen as fresh random values. Next, the second condition forge is defined as follows.

$$\mathsf{forge} \Leftrightarrow \exists \beta \in \mathcal{Q}_{\mathsf{Dec}} \text{ s.t. } \widetilde{T}^{(\beta)} = T^{(\beta)}.$$

In Game 3, if the event does not occur, then all responses to decryption queries are **reject**, thus there is no difference between Game 3 and the ideal world.

6.3 Upper-bounding $\Pr[\mathsf{T}_I \in \mathcal{T}_{\mathsf{bad}}]$

Let $\mathsf{coll}^*_{X_{1,b}}$ (resp. forge^*) be an event that $\mathsf{coll}_{X_{1,b}}$ (resp. forge) occurs before forge (resp. $\mathsf{coll}_{X_{1,b}}$) occurs. Since one of the conditions on $\mathcal{T}_{\mathsf{bad}}$ occurs before the other occurs, we have

$$\Pr[\mathsf{T}_I \in \mathcal{T}_{\mathsf{bad}}] \leq \Pr[\mathsf{coll}^*_{X_{1,b}}] + \Pr[\mathsf{forge}^*] \leq \left(\sum_{\omega \in [u]} \frac{0.5\sigma_\omega^2}{2^n} \right) + \frac{q_{\mathsf{d}}}{2^t} \ .$$

These bounds are derived in the followings.

6.3.1 Upper-bounding $\Pr[\mathsf{coll}^*_{X_{1,b}}]$. Fix $\omega \in [u]$. Let $\sigma_{\omega,2} := |\mathcal{X}_2^{[\omega]}|$ be the total number of input blocks in CTR of the ω-th user, and $\sigma_{\omega,1} := \sum_{\alpha \in \mathcal{Q}^{[\omega]}} b_\alpha$ the total number of input-output pairs $(X_{1,j}^{(\alpha)}, Y_{1,j}^{(\alpha)})$ in CBC of the ω-th user.

In this evaluation, we use the following event.

$$\mathsf{coll}_{X_1,X_2} \Leftrightarrow \exists \beta \in \mathcal{Q}^{[\omega]}, j \in [b_\beta] \text{ s.t. } X_{1,j}^{(\beta)} \in \mathcal{X}_2^{[\omega]}.$$

The event considers a collision in input blocks between CBC and CTR (with the tag generation) for the ω-th user. Since $X_{1,1}^{(\beta)} \notin \mathcal{X}_2^{[\omega]}$, the output block $Y_{1,1}^{(\beta)}$ is chosen independently of all output blocks in CTR. By the iterated structure of CBC, if coll_{X_1,X_2} occurs, then there exists $j \in [b_\beta]$ such that $X_{1,j-1}^{(\beta)} \notin \mathcal{X}_2^{[\omega]} \wedge X_{1,j}^{(\beta)} \in \mathcal{X}_2^{[\omega]}$. For each $X_{1,j-1}^{(\beta)} \notin \mathcal{X}_2^{[\omega]}$ and $X_{2,i}^{(\alpha)} \in \mathcal{X}_2^{[\omega]}$, $Y_{1,j-1}^{(\beta)}$ is chosen independently of $X_{2,i}^{(\alpha)}$, we have

$$\Pr[X_{1,j}^{(\beta)} = X_{2,i}^{(\alpha)}] = \Pr[Y_{1,j-1}^{(\beta)} \oplus B_j^{(\beta)} = X_{2,i}^{(\alpha)}] \leq \frac{1}{2^n},$$

and

$$\Pr[\mathsf{coll}_{X_1,X_2}] \leq \frac{\sigma_{\omega,2}\sigma_{\omega,1}}{2^n} \ .$$

We next evaluate $\Pr[\mathsf{coll}^*_{X_{1,b}}]$ under the assumption that coll_{X_1,X_2} does not occur. Since $\mathsf{f}_{\mathsf{CCM}}$ is an injective function, for any $\alpha, \beta \in \mathcal{Q}^{[\omega]}$ such that $\alpha \neq \beta$, $B^{(\beta)} \neq B^{(\alpha)}$ holds. By this property and

the iterated structure of CBC, if $\mathsf{coll}^*_{X_{1,b}}$ occurs, then $X^{(\alpha)}_{2,i-1} \neq X^{(\beta)}_{2,j-1}$ and $X^{(\alpha)}_{2,i} = X^{(\beta)}_{2,j}$ must hold for some $\alpha, \beta \in \mathcal{Q}^{[\omega]}, i \in [b_\alpha], j \in [2, b_\beta]$ such that $(\alpha, i) \neq (\beta, j)$. Note that if $i = 1$, then $X^{(\alpha)}_{2,0} := \varepsilon$ for the sake of convenience. For two input blocks $X^{(\alpha)}_{2,i-1} \neq X^{(\beta)}_{2,j-1}$, the output blocks are independently chosen, and we have

$$\Pr[X^{(\alpha)}_{2,i} = X^{(\beta)}_{2,j}] = \Pr[X^{(\alpha)}_{2,i} = Y^{(\beta)}_{2,j-1} \oplus B^{(\beta)}_j] \leq \frac{1}{2^n} \ ,$$

and

$$\Pr[\mathsf{coll}^*_{X_{1,b}}] \leq \binom{\sigma_{\omega,1}}{2} \cdot \frac{1}{2^n} \leq \frac{0.5\sigma^2_{\omega,1}}{2^n} \ .$$

Summing the above bounds for each ω, we have

$$\Pr[\mathsf{coll}^*_{X_{1,b}}] \leq \sum_{\omega \in [u]} \frac{0.5\sigma^2_{\omega,1}}{2^n} + \frac{\sigma_{\omega,2}\sigma_{\omega,1}}{2^n} \leq \sum_{\omega \in [u]} \frac{0.5\sigma^2_{\omega}}{2^n} \ .$$

6.3.2 Upper-bounding $\Pr[\mathsf{forge}^*]$. For each $\beta \in \mathcal{Q}_{\mathsf{Dec}}$, $\mathsf{coll}_{X^*_{1,b}}$ does not occur, and $Y^{(\beta)}_{1,b_\beta}$ is chosen independently of all output blocks defined in encryption queries. Hence, we have

$$\begin{aligned}\Pr[\mathsf{forge}^*] &\leq \sum_{\beta \in \mathcal{Q}_{\mathsf{Dec}}} \Pr[\widetilde{T}^{(\beta)} = T^{(\beta)}] \\ &= \sum_{\beta \in \mathcal{Q}_{\mathsf{Dec}}} \Pr[\widetilde{T}^{(\beta)} = \mathsf{lsb}_t(Y^{(\beta)}_{1,b_\beta} \oplus Y^{(\beta)}_{2,0})] \leq \frac{q_{\mathsf{d}}}{2^t} \ .\end{aligned}$$

6.4 Lower-bounding $\frac{\Pr[\mathsf{T}_3 = \tau]}{\Pr[\mathsf{T}_I = \tau]}$

Fix a good transcript $\tau \in \mathcal{T}_{\mathsf{good}}$. By $\neg\mathsf{forge}$, the input-output pairs in τ are defined so that $\forall \beta \in \mathcal{Q}_{\mathsf{Dec}} : \widetilde{T}^{[\beta]} \neq T^{[\beta]}$ holds. Since all responses $(C^{(\alpha)}, T^{(\beta)})$ for $\alpha \in \mathcal{Q}_{\mathsf{Enc}}$ are uniquely fixed from input-output pairs $(X^{(\alpha)}_{i,j}, Y^{(\alpha)}_{i,j})$, we evaluate the probability that all input-output pairs result in the good transcript τ.

In the ideal world, all output blocks of CTR for encryption queries are independent defined by random-bit oracles via Algorithm 4. In Game 3, the nonce-respect setting ensures that all input blocks in CTR are distinct, ensuring that all output blocks in CTR are independently defined. For the other outputs, for a new input, the output is chosen uniformly at random from $\{0,1\}^n$ in both Game 3 and the ideal world, and for a repeated input, the output is defined as the same one. Hence, the above evaluation shows that $\Pr[\mathsf{T}_3 = \tau] = \Pr[\mathsf{T}_I = \tau]$, thus we have

$$\frac{\Pr[\mathsf{T}_3 = \tau]}{\Pr[\mathsf{T}_I = \tau]} = 1.$$

6.5 Deriving the Upper-bound

Combining Lemma 4.1 with the upper-bound of $\Pr[\mathsf{T}_I \in \mathcal{T}_{\mathsf{bad}}]$ and the lower-bound of $\frac{\Pr[\mathsf{T}_3 = \tau]}{\Pr[\mathsf{T}_I = \tau]}$, we obtain the upper-bound in Lemma 5.2.

[End of Proof of Lemma 5.2] ■

7 Mu-Security of CCM with NR

We evaluate the security of CCM with randomized nonce in the IC model. We use the d-bound model by Hoang and Tessaro [16], which is a generalization of the randomized nonce.

7.1 d-bound Adversaries

In the d-bound model, the number of collisions of nonces in encryption queries across users is bounded by d. Note that there is no collision in nonces in encryption queries within the same user.

Definition 7.1 (d-bound model). For $\omega \in [u]$, let $\mathcal{N}^{[\omega]}$ be the set of nonces in encryption queries to the ω-th user. A d-bound adversary is such that for any $N \in \{0,1\}^\nu$, $|\{\omega \in [u] \mid N \in \mathcal{N}^{[\omega]}\}| \le d$.

We study the bound d. We consider the following randomized nonce: each original nonce N_{orig} is defined by incrementing 1, i.e., $N_{\mathrm{orig}} \leftarrow N_{\mathrm{orig}} + 1$ (initially $N_{\mathrm{orig}} = 0^n$), and a randomized nonce N is defined as $N = N_{\mathrm{orig}} \oplus R$ with the ν-bit original nonce N_{orig} and a user-specific random mask $R \in \{0,1\}^\nu$. Then, for each of d randomized nonces $N^{(\alpha_1)}, \ldots, N^{(\alpha_d)}$ such that the user indexes $\mathsf{u}_{\alpha_1}, \ldots, \mathsf{u}_{\alpha_d}$ are all distinct, we have $\Pr[N^{(\alpha_1)} = \ldots = N^{(\alpha_d)}] \le \left(\frac{1}{2^\nu}\right)^{d-1}$. Using the bound with $d := \frac{\nu}{\log_2 \nu}$, we have

$$\begin{aligned}
&\Pr[\exists \alpha_1, \ldots, \alpha_d \text{ s.t. } N^{(\alpha_1)} = \ldots = N^{(\alpha_d)}] \\
&\le \binom{q_{\mathrm{e}}}{d} \cdot \left(\frac{1}{2^\nu}\right)^{d-1} \le 2^\nu \left(\frac{eq_{\mathrm{e}}}{d2^\nu}\right)^d = 2^\nu \left(\frac{eq_{\mathrm{e}}}{\frac{\nu}{\log_2 \nu} \cdot 2^\nu}\right)^{\frac{\nu}{\log_2 \nu}} \\
&\le \left((2^\nu)^{\frac{\log_2 \nu}{\nu}} \cdot \frac{eq_{\mathrm{e}}}{\frac{\nu}{\log_2 \nu} \cdot 2^\nu}\right)^{\frac{\nu}{\log_2 \nu}} \le \left(\frac{3(\log_2 \nu) q_{\mathrm{e}}}{2^\nu}\right)^{\frac{\nu}{\log_2 \nu}},
\end{aligned}$$

using Stirling's approximation ($d! \ge (d/e)^d$ for any d). We then consider for the common parameter for CCM: the nonce size is $\nu = 3n/4$ ($\nu = 96$ when using the AES parameter $n = 128$). In this case, $d = \frac{3n/4}{\log_2(3n/4)}$ and the bound of d can be ensured up to $q_{\mathrm{e}} \approx 2^{3n/4}$ encryption queries.

7.2 Security Bound

Theorem 7.2. *$\forall \mathbf{A} \in \mathcal{A}_{\mathrm{icm}}$ such that $\mathbf{A}$ is a d-bound adversary:*

$$\mathbf{Adv}^{\mathrm{muae}}_{\mathrm{CCM}}(\mathbf{A}) \le \frac{q_{\mathrm{d}}}{2^t} + \sum_{\omega \in [u]} \frac{\sigma_\omega^2}{2^n} + \frac{\left(d + \frac{n}{\log_2 n}\right)(p+\sigma)}{2^k} + \left(\frac{3(\log_2 n)\sigma}{2^n}\right)^{\frac{n}{\log_2 n}} + \frac{\sigma(p+\sigma)}{2^{k+n}} .$$

With the parameters σ and σ_{u}, $\forall \mathbf{A} \in \mathcal{A}_{\mathrm{icm}}$ such that $\mathbf{A}$ is a d-bound adversary:

$$\mathbf{Adv}^{\mathrm{muae}}_{\mathrm{CCM}}(\mathbf{A}) \le \frac{q_{\mathrm{d}}}{2^t} + \frac{\sigma_{\mathrm{u}}\sigma}{2^n} + \frac{\left(d + \frac{n}{\log_2 n}\right)(p+\sigma)}{2^k} + \left(\frac{3(\log_2 n)\sigma}{2^n}\right)^{\frac{n}{\log_2 n}} + \frac{\sigma(p+\sigma)}{2^{k+n}} .$$

Assume that $n \le k$. The last three terms excluding p are of online security and become a constant if σ is about 2^n. On the other hand, the second term becomes a constant if σ is about $\frac{2^n}{\sigma_{\mathrm{u}}}$. Hence, the first two terms are dominant online terms. The last term excluding σ is of offline security and becomes a constant if $p = \frac{2^{k+n}}{\sigma}$. Since $\sigma \le 2^n$, the third term excluding σ is a dominant offline term. Since d is about $\frac{n}{\log_2 n}$, dominant terms in the bound is $\frac{q_{\mathrm{d}}}{2^t} + \frac{\sigma_{\mathrm{u}}\sigma}{2^n} + \frac{dp}{2^k}$.

7.3 Proof of Theorem 7.2

Without loss of generality, we assume that $\mathbf{A}$ is deterministic. In this evaluation, we consider three games.

Real World → Game 2. We start the proof from the real world, followed by Game 2. In the real world, $\mathbf{A}$ has access to O_{real}. From the real world to Game 2, the u BCs $(E_{K_\omega})_{\omega\in[u]}$ are replaced with u RFs $(\mathcal{R}_\omega)_{\omega\in[u]}$. Hence, in Game 2, $\mathbf{A}$ has access to the modified oracles

$$O_2 \coloneqq ((\mathsf{CCM}[\mathcal{R}_\omega])_{\omega\in[u]}, E^{\pm}),$$

where $E \xleftarrow{\$} \mathcal{BC}$ and $\forall\omega\in[u] : \mathcal{R}_\omega \xleftarrow{\$} \mathsf{Func}(n,n)$. The following lemma shows an upper-bound of the difference between the real world and Game 2.

Lemma 7.3. *For any computationally-unbounded adversary* $\mathbf{A}$*,*

$$\mathbf{Adv}^{\mathsf{dist}}_{O_{\mathsf{real}},O_2}(\mathbf{A}) \le \frac{\left(d+\frac{n}{\log_2 n}\right)(p+\sigma)}{2^k} + \frac{\sigma(p+\sigma)}{2^{k+n}} + \left(\frac{3(\log_2 n)\sigma}{2^n}\right)^{\frac{n}{\log_2 n}} + \sum_{\omega\in[u]}\frac{0.5\sigma_\omega^2}{2^n},$$

where $O_{\mathsf{real}} = ((\mathsf{CCM}[E_{K_\omega}])_{\omega\in[u]}, E^{\pm})$ *and* $O_2 = ((\mathsf{CCM}[\mathcal{R}_\omega])_{\omega\in[u]}, E^{\pm})$.

Hereafter, we provide a high-level overview of the proof, and the formal proof is given in Section 8.

7.3.1 Proof of Lemma 7.3 (Overview). From the real world to Game 2, the underlying primitives are replaced from an IC E (with independent keys) to independent RFs $(\mathcal{R}_\omega)_{\omega\in[u]}$. We thus define the following three events that are taken into account the difference.

Event $\mathsf{coll}_{\mathsf{on},\neq\mathsf{u}}$. We first define an event $\mathsf{coll}_{\mathsf{on},\neq\mathsf{u}}$ that considers a collision of pairs of key and input/output block between distinct users. If it does not occur in the real world, for each user, the underlying primitive can be independent of those of the other users as Game 2.

$$\mathsf{coll}_{\mathsf{on},\neq\mathsf{u}} \Leftrightarrow \omega_1,\omega_2\in[u] \text{ s.t. } \omega_1\neq\omega_2 \wedge K^{[\omega_1]} = K^{[\omega_2]} \wedge (\mathcal{X}^{[\omega_1]}\cap\mathcal{X}^{[\omega_2]}\neq\emptyset \vee \mathcal{Y}^{[\omega_1]}\cap\mathcal{Y}^{[\omega_2]}\neq\emptyset).$$

$\mathcal{X}^{[\omega_1]}\cap\mathcal{X}^{[\omega_2]}\neq\emptyset$ is the condition on the input-block collision and $\mathcal{Y}^{[\omega_1]}\cap\mathcal{Y}^{[\omega_2]}\neq\emptyset$ is the one on the output-block collision.

We consider the collisions $K^{[\omega_1]} = K^{[\omega_2]} \wedge \mathcal{X}^{[\omega_1]}\cap\mathcal{X}^{[\omega_2]}\neq\emptyset$.

- If $\mathcal{X}_{\mathsf{N}}^{[\omega_1]}\cap\mathcal{X}_{\mathsf{N}}^{[\omega_2]}\neq\emptyset$, i.e., a collision occurs in nonce-dependent input blocks, then a collision of nonces between distinct users occurs. In the d-bound model, for each nonce $N^{(\alpha)}$ of the ω_1-th user, the number of the other different users with the same nonce is at most d. Hence, for each pair of key and nonce, the probability that the pair collides with one of the pairs of the other different users is at most $\frac{d}{2^k}$. Thus, we have $\Pr[K^{[\omega_1]} = K^{[\omega_2]} \wedge \mathcal{X}_{\mathsf{N}}^{[\omega_1]}\cap\mathcal{X}_{\mathsf{N}}^{[\omega_2]}\neq\emptyset] \le \sum_{\alpha\in[q]}\frac{d}{2^k} = \frac{dq}{2^k}$.
- If $\mathcal{X}_{\neq\mathsf{N}}^{[\omega_1]}\cap\mathcal{X}^{[\omega_2]}\neq\emptyset$, i.e., a collision with nonce-independent input blocks occurs, then each input block $X_{1,j}^{(\alpha)}\in\mathcal{X}_{\neq\mathsf{N}}^{[\omega_1]}$ is defined as $X_{1,j}^{(\alpha)} = B_{1,j}^{(\alpha)}\oplus Y_{1,j-1}^{(\alpha)}$ where $Y_{1,j-1}^{(\alpha)}$ is an n-bit random value. Hence, we can use the n-bit randomness, providing the bound $\Pr[K^{[\omega_1]} = K^{[\omega_2]} \wedge \mathcal{X}_{\neq\mathsf{N}}^{[\omega_1]}\cap\mathcal{X}^{[\omega_2]}\neq\emptyset] \le \binom{\sigma}{2}\cdot\frac{1}{2^{k+n}} \le \frac{\sigma^2}{2^{k+n}}$.

We next consider the collisions $K^{[\omega_1]} = K^{[\omega_2]} \wedge \mathcal{Y}^{[\omega_1]}\cap\mathcal{Y}^{[\omega_2]}\neq\emptyset$. The evaluation is the same as the one for the collisions $K^{[\omega_1]} = K^{[\omega_2]} \wedge \mathcal{X}_{\mathsf{N}}^{[\omega_1]}\cap\mathcal{X}_{\mathsf{N}}^{[\omega_2]}\neq\emptyset$. In this case, instead of the multi-collision bound d for input blocks, we use a multi-collision event for output blocks $\cup_{\omega\in[u]}\mathcal{Y}^{[\omega]}$. By using the randomness of the output blocks, the probability that $(\frac{n}{\log_2 n})$-multi-collision occurs in the output blocks can be bounded by $(\frac{3(\log_2 n)\sigma}{2^n})^{\frac{n}{\log_2 n}}$. Assuming that the multi-collision does not

occur, by the same evaluation (but d is replaced with the bound $\frac{n}{\log_2 n}$), we have $\Pr[K^{[\omega_1]} = K^{[\omega_2]} \wedge \mathcal{Y}^{[\omega_1]} \cap \mathcal{Y}^{[\omega_2]} \neq \emptyset] \leq \frac{\frac{n}{\log_2 n} \cdot \sigma}{2^k}$.

Summing these bounds, we have

$$\Pr[\mathsf{coll}_{\mathsf{on},\neq\mathsf{u}}] \leq \frac{\left(d+\frac{n}{\log_2 n}\right)\sigma}{2^k} + \left(\frac{3(\log_2 n)\sigma}{2^n}\right)^{\frac{n}{\log_2 n}} + \frac{\sigma^2}{2^{k+n}}.$$

Event $\mathsf{coll}_{\mathsf{on},\mathsf{off}}$. In Game 2, the underlying primitives $(\mathcal{R}_\omega)_{\omega\in[u]}$ are independent of $E^{\pm}$. On the other hand, in the real world, all underlying primitives are E (with independent keys). We thus define an event $\mathsf{coll}_{\mathsf{on},\mathsf{off}}$ for the difference. The event considers a collision of pairs of key and input/output block between online and offline queries. If it does not occur, outputs of user's primitives can be independent of offline query-response tuples.

$$\begin{aligned}\mathsf{coll}_{\mathsf{on},\mathsf{off}} \Leftrightarrow \alpha \in [q], (i,j) \in \mathsf{Index}^{(\alpha)}, \beta \in [p]\\ \text{s.t. } K^{[\mathsf{u}_\alpha]} = \hat{K}^{(\beta)} \wedge (X_{i,j}^{(\alpha)} = \hat{X}^{(\beta)} \vee Y_{i,j}^{(\alpha)} = \hat{Y}^{(\beta)}).\end{aligned}$$

The evaluation is similar to the evaluation for the event $\mathsf{coll}_{\mathsf{on},\neq\mathsf{u}}$. By using the d-bound model for the input-block collision and the $(\frac{n}{\log_2 n})$-multi-collision event for the output-block collision, we can obtain $\Pr[\mathsf{coll}_{\mathsf{on},\mathsf{off}}] \leq \frac{\left(d+\frac{n}{\log_2 n}\right)p}{2^k} + \frac{\sigma p}{2^{k+n}}$.

Event $\mathsf{coll}_{\mathsf{on},=\mathsf{u}}$. In Game 2, for each $\omega \in [u]$, each output of the underlying primitive $\mathcal{R}_\omega$ is chosen with replacement. On the other hand, in the real world, all underlying primitives are E (with independent keys) and for each $\omega \in [u]$, each output of the underlying primitive is chosen without replacement. We thus define an event $\mathsf{coll}_{\mathsf{on},=\mathsf{u}}$ for the RP-RF difference.

$$\begin{aligned}\mathsf{coll}_{\mathsf{on},=\mathsf{u}} \Leftrightarrow \exists \omega \in [u], X_{i_1,j_1}^{(\alpha_1)}, X_{i_2,j_2}^{(\alpha_2)} \in \mathcal{X}^{[\omega]} \text{ s.t.}\\ X_{i_1,j_1}^{(\alpha_1)} \neq X_{i_2,j_2}^{(\alpha_2)} \wedge Y_{i_1,j_1}^{(\alpha_1)} = Y_{i_2,j_2}^{(\alpha_2)},\end{aligned}$$

where $Y_{i_1,j_1}^{(\alpha_1)}$ and $Y_{i_2,j_2}^{(\alpha_2)}$ are independently chosen. By the birthday analysis, we have $\Pr[\mathsf{coll}_{\mathsf{on},=\mathsf{u}}] \leq \sum_{\omega\in[u]} \binom{\sigma_\omega}{2} \cdot \frac{1}{2^n} \leq \sum_{\omega\in[u]} \frac{0.5\sigma_\omega^2}{2^n}$.

Deriving the Bound in Lemma 7.3. These events cover the differences from the replacements of the underlying primitives from $(E_{K_\omega})_{\omega\in[u]}$ to $(\mathcal{R}_\omega)_{\omega\in[u]}$, thus by the above bounds, we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{dist}}_{O_{\mathsf{real}},O_2}(\mathbf{A}) &\leq \Pr[\mathsf{coll}_{\mathsf{on},\neq\mathsf{u}}] + \Pr[\mathsf{coll}_{\mathsf{on},\mathsf{off}}] + \Pr[\mathsf{coll}_{\mathsf{on},=\mathsf{u}}]\\ &\leq \frac{\left(d+\frac{n}{\log_2 n}\right)(p+\sigma)}{2^k} + \frac{\sigma(p+\sigma)}{2^{k+n}} + \left(\frac{3(\log_2 n)\sigma}{2^n}\right)^{\frac{n}{\log_2 n}} + \sum_{\omega\in[u]} \frac{0.5\sigma_\omega^2}{2^n}.\end{aligned}$$

Note that the formal proof given in Appendix 8 uses the coefficient-H technique and these events are evaluated in Game 2.

[End of Proof of Lemma 7.3 (Overview)] ■

Game 2 → Ideal World. For the difference between Game 2 and the ideal world, we use Lemma 5.2 in the proof of Theorem 5.1. In Lemma 5.2, an IC is absent, whereas in this evaluation, an IC is available. However, the responses of online queries are independent of the IC, thus an adversary can simulate an IC. Hence, the difference between Game 2 and the ideal world can be bounded by the bound in Lemma 5.2, and we have $\mathbf{Adv}^{\mathsf{dist}}_{O_2,O_{\mathsf{ideal}}}(\mathbf{A}) \leq \left(\sum_{\omega\in[u]} \frac{0.5\sigma_\omega^2}{2^n}\right) + \frac{q_{\mathsf{d}}}{2^t}$.

Conclusion of the Proof. Using the above bounds, we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{muae}}_{\mathsf{CCM}}(\mathbf{A}) &= \mathbf{Adv}^{\mathsf{dist}}_{O_{\mathsf{real}},O_2}(\mathbf{A}) + \mathbf{Adv}^{\mathsf{dist}}_{O_2,O_{\mathsf{ideal}}}(\mathbf{A}) \\ &\leq \frac{q_{\mathsf{d}}}{2^t} + \sum_{\omega\in[u]} \frac{\sigma_\omega^2}{2^n} + \frac{\left(d + \frac{n}{\log_2 n}\right)(p+\sigma)}{2^k} + \left(\frac{3(\log_2 n)\sigma}{2^n}\right)^{\frac{n}{\log_2 n}} + \frac{\sigma(p+\sigma)}{2^{k+n}}.\end{aligned}$$

[End of Proof of Theorem 7.2] ■

8 Proof of Lemma 7.3

We derive the following bound by using the coefficient-H technique (See Section 4.4). For any computationally-unbounded, d-bound adversary $\mathbf{A}$,

$$\mathbf{Adv}^{\mathsf{dist}}_{O_{\mathsf{real}},O_2}(\mathbf{A}) \leq \frac{\left(d + \frac{n}{\log_2 n}\right)\cdot(p+\sigma)}{2^k} + \frac{\sigma(p+\sigma)}{2^{k+n}} + \left(\frac{3(\log_2 n)\sigma}{2^n}\right)^{\frac{n}{\log_2 n}} + \sum_{\omega\in[u]} \frac{0.5\sigma_\omega^2}{2^n},$$

where $O_{\mathsf{real}} = (\mathsf{CCM}[E_{K_\omega}])_{\omega\in[u]}, E^{\pm})$ and $O_2 = ((\mathsf{CCM}[\mathcal{R}_\omega])_{\omega\in[u]}, E^{\pm})$. Let T_R (resp. T_2) be a transcript obtained by random samplings of O_{real} (resp. O_2).

8.1 Extended Transcript

In this poof, we permit $\mathbf{A}$ to obtain all internal values $\{(X_{i,j}^{(\alpha)}, Y_{i,j}^{(\alpha)}) \mid \alpha \in [q], (i,j) \in \mathsf{Index}^{(\alpha)}\}$ in the decision phase. Note that the additional values do not reduce the $\mathbf{A}$'s advantage, since $\mathbf{A}$ can ignore the values. Thus, the (extended) transcript τ consists of

- primitive query-responses $(\hat{K}^{(\alpha)}, \hat{X}^{(\alpha)}, \hat{Y}^{(\alpha)})$ for $\alpha \in [p]$,
- encryption query-responses $(N^{(\alpha)}, A^{(\alpha)}, M^{(\alpha)}, C^{(\alpha)}, T^{(\alpha)})$ for $\alpha \in Q_{\mathsf{Enc}}$,
- decryption query-responses $(N^{(\alpha)}, A^{(\alpha)}, C^{(\alpha)}, \widetilde{T}^{(\alpha)}, RV^{(\alpha)})$ for $\alpha \in Q_{\mathsf{Dec}}$, where $RV^{(\alpha)} \in \mathcal{M} \cup \{\mathbf{reject}\}$ is the response to the α-th decryption query, and
- (dummy) internal values $\{(X_{i,j}^{(\alpha)}, Y_{i,j}^{(\alpha)}) \mid \alpha \in [q], (i,j) \in \mathsf{Index}^{(\alpha)}\}$.

8.2 Bad and Good Transcripts

We define four conditions $\mathsf{coll}_{\mathsf{on}}$, $\mathsf{coll}_{\mathsf{on,off}}$, mcoll_Y and $\mathsf{coll}_{\mathsf{on},=\mathsf{u}}$ on bad transcripts. The set of bad transcripts $\mathcal{T}_{\mathsf{bad}}$ is a subset of $\mathcal{T}$ such that one of the conditions holds. The set of good transcripts is defined as $\mathcal{T}_{\mathsf{good}} := \mathcal{T} \setminus \mathcal{T}_{\mathsf{bad}}$, which is the subset of $\mathcal{T}$ such that the conditions do not hold.

8.2.1 Conditions on (In)dependence for Online and Offline Queries. In Game 2, for each $\omega \in [u]$, the internal input-output tuples are defined independently of $E^{\pm}$ and the input-output tuples of the other users. On the other hand, in the real world, all underlying primitives use the same IC E (with independent keys). Hence, we ensure the independence by introducing the following events.

$$\begin{aligned}\mathsf{coll}_{\mathsf{on},\neq\mathsf{u}} \Leftrightarrow{}& \omega_1,\omega_2 \in [u] \text{ s.t. } \omega_1 \neq \omega_2 \wedge K^{[\omega_1]} = K^{[\omega_2]} \wedge \\ &(\mathcal{X}^{[\omega_1]} \cap \mathcal{X}^{[\omega_2]} \neq \emptyset \vee \mathcal{Y}^{[\omega_1]} \cap \mathcal{Y}^{[\omega_2]} \neq \emptyset).\end{aligned}$$

$$\begin{aligned}\mathsf{coll}_{\mathsf{on,off}} \Leftrightarrow{}& \alpha \in [q], (i,j) \in \mathsf{Index}^{(\alpha)}, \beta \in [p] \\ &\text{s.t. } K^{[\mathsf{u}_\alpha]} = \hat{K}^{(\beta)} \wedge (X_{i,j}^{(\alpha)} = \hat{X}^{(\beta)} \vee Y_{i,j}^{(\alpha)} = \hat{Y}^{(\beta)}).\end{aligned}$$

The first event $\mathsf{coll}_{\mathsf{on},\neq\mathsf{u}}$ considers a collision in pairs of key and input (or output) block between different users. Thus, if the event does not occur, the independence of input-output tuples between distinct users is ensured. The second event $\mathsf{coll}_{\mathsf{on,off}}$ considers a collision in pairs of key and input

block (or output block) between online and offline queries. Thus, if the event does not occur, the independence of input-output tuples between online and offline queries is ensured.

8.2.2 Condition on Output Blocks. We define a multi-collision event for output blocks across distinct users. The multi-collision event is used to evaluate the probabilities of $\mathsf{coll}_{\mathsf{on},\neq\mathsf{u}}$ and $\mathsf{coll}_{\mathsf{on},\mathsf{off}}$ with the output collisions. Note that for the probabilities with the input collisions, the bound d is used. Let $\mu := \frac{n}{\log_2 n}$.

$$\begin{aligned}\mathsf{mcoll}_Y \Leftrightarrow \exists \alpha_1, \ldots, \alpha_\mu \in [q], (i_1, j_1) \in \mathsf{Index}^{(\alpha_1)}, \ldots, (i_\mu, j_\mu) \in \mathsf{Index}^{(\alpha_\mu)} \\ \text{s.t. } \mathsf{u}_{\alpha_1}, \ldots, \mathsf{u}_{\alpha_\mu} \text{ are all distinct and } Y^{(\alpha_1)}_{i_1,j_1} = \cdots = Y^{(\alpha_\mu)}_{i_\mu,j_\mu}.\end{aligned}$$

8.2.3 Condition on RP-RF Switch. In the real world (resp. Game 2), output blocks are defined without (resp. with) replacement for each key element. We thus define the following event to cover the collision of output blocks in Game 2.

$$\mathsf{coll}_{\mathsf{on},=\mathsf{u}} \Leftrightarrow \exists \omega \in [u], X^{(\alpha_1)}_{i_1,j_1}, X^{(\alpha_2)}_{i_2,j_2} \in \mathcal{X}^{[\omega]} \text{ s.t. } X^{(\alpha_1)}_{i_1,j_1} \neq X^{(\alpha_2)}_{i_2,j_2} \wedge Y^{(\alpha_1)}_{i_1,j_1} = Y^{(\alpha_2)}_{i_2,j_2}.$$

8.3 Upper-bounding $\Pr[\mathsf{T}_I \in \mathcal{T}_{\mathsf{bad}}]$

Let $\mathbf{Event} := \{\mathsf{coll}_{\mathsf{on},\neq\mathsf{u}}, \mathsf{coll}_{\mathsf{on},\mathsf{off}}, \mathsf{mcoll}_Y, \mathsf{coll}_{\mathsf{on},=\mathsf{u}}\}$. For each event $\in \mathbf{Event}$, let event* be an event that event before the other events $\mathbf{Event}\backslash\{\mathsf{event}\}$ occur. Since one of the conditions on $\mathcal{T}_{\mathsf{bad}}$ occurs before the other occurs, we have

$$\begin{aligned}\Pr[\mathsf{T}_I \in \mathcal{T}_{\mathsf{bad}}] &\leq \sum_{\mathsf{event} \in \mathbf{Event}} \Pr[\mathsf{event}^*] \\ &\leq \frac{\left(d + \frac{n}{\log_2 n}\right)(p+\sigma)}{2^k} + \frac{\sigma(p+\sigma)}{2^{k+n}} + \left(\frac{3(\log_2 n)\sigma}{2^n}\right)^{\frac{n}{\log_2 n}} + \sum_{\omega \in [u]} \frac{0.5\sigma_\omega^2}{2^n} \quad .\end{aligned}$$

8.3.1 Upper-bounding $\Pr[\mathsf{coll}^*_{\mathsf{on},\neq\mathsf{u}}]$. We first consider the collisions $K^{[\omega_1]} = K^{[\omega_2]} \wedge \mathcal{X}^{[\omega_1]} \cap \mathcal{X}^{[\omega_2]} \neq \emptyset$.

- If $\mathcal{X}^{[\omega_1]}_{\mathsf{N}} \cap \mathcal{X}^{[\omega_2]}_{\mathsf{N}} \neq \emptyset$, i.e., a collision occurs in nonce-dependent input blocks between distinct users, then a nonce collision occurs, since $\exists X^{(\alpha_1)}_{i_1,j_1} \in \mathcal{X}^{[\omega_1]}_{\mathsf{N}}, X^{(\alpha_2)}_{i_2,j_2} \in \mathcal{X}^{[\omega_2]}_{\mathsf{N}}$ s.t. $X^{(\alpha_1)}_{i_1,j_1} = X^{(\alpha_2)}_{i_2,j_2} \Rightarrow N^{(\alpha_1)} = \mathsf{ext}_{\mathsf{nonce}}(X^{(\alpha_1)}_{i_1,j_1}) = \mathsf{ext}_{\mathsf{nonce}}(X^{(\alpha_2)}_{i_2,j_2}) = N^{(\alpha_2)}$. In the d-bound model, for each nonce $N^{(\alpha)}$ of the ω_1-th user, the number of the other users with the same nonce is at most d. Hence, for each pair of key and nonce, the probability that the pair collides with one of the pairs of the other users is at most $\frac{d}{2^k}$. Thus, we have $\Pr[K^{[\omega_1]} = K^{[\omega_2]} \wedge \mathcal{X}^{[\omega_1]}_{\mathsf{N}} \cap \mathcal{X}^{[\omega_2]}_{\mathsf{N}} \neq \emptyset] \leq \sum_{\alpha \in [q]} \frac{d}{2^k} = \frac{dq}{2^k}$.
- If $\mathcal{X}^{[\omega_1]}_{\neq\mathsf{N}} \cap \mathcal{X}^{[\omega_2]} \neq \emptyset$, i.e., a collision occurs for nonce-independent input blocks, then each input block $X^{(\alpha)}_{1,j} \in \mathcal{X}^{[\omega_1]}_{\mathsf{N}}$ is defined as $X^{(\alpha)}_{1,j} = B^{(\alpha)}_{1,j} \oplus Y^{(\alpha)}_{1,j-1}$ where $Y^{(\alpha)}_{1,j-1}$ is an n-bit random value. By using the n-bit randomness, we have $\Pr[K^{[\omega_1]} = K^{[\omega_2]} \wedge \mathcal{X}^{[\omega_1]}_{\neq\mathsf{N}} \cap \mathcal{X}^{[\omega_2]} \neq \emptyset] \leq \binom{\sigma}{2} \cdot \frac{1}{2^{k+n}} \leq \frac{0.5\sigma^2}{2^{k+n}}$.

Using the above bounds, we have

$$\Pr[\mathsf{coll}^*_{\mathsf{on},\neq\mathsf{u}} \wedge \mathcal{X}^{[\omega_1]} \cap \mathcal{X}^{[\omega_2]} \neq \emptyset] \leq \frac{dq}{2^k} + \frac{0.5\sigma^2}{2^{k+n}} \quad .$$

We next consider the collisions $K^{[\omega_1]} = K^{[\omega_2]} \wedge \mathcal{Y}^{[\omega_1]} \cap \mathcal{Y}^{[\omega_2]} \neq \emptyset$. By $\neg\mathsf{mcoll}_Y$, for each $\alpha \in [q]$ and $(i,j) \in \mathsf{Index}^{(\alpha)}$, the number of tuples $(\beta, (i', j')) \in [q] \times \mathsf{Index}^{(\beta)}$ s.t. $\mathsf{u}^{(\alpha)} \neq \mathsf{u}^{(\beta)} \wedge Y^{(\alpha)}_{i,j} = Y^{(\beta)}_{i',j'}$

is at most μ, thus the number of key candidates that yield the collisions is at most μ. Hence, we have

$$\Pr[\mathsf{coll}^{*}_{\mathsf{on},\neq\mathsf{u}} \wedge \mathcal{Y}^{[\omega_1]} \cap \mathcal{Y}^{[\omega_2]} \neq \emptyset] \leq \sum_{\alpha\in[q],(i,j)\in\mathsf{Index}^{(\alpha)}} \frac{\mu}{2^k} = \frac{\mu\sigma}{2^k} = \frac{\frac{n}{\log_2 n}\cdot\sigma}{2^k} \quad .$$

Using the above bounds, we have

$$\Pr[\mathsf{coll}_{\mathsf{on},\neq\mathsf{u}}] \leq \frac{dq + \frac{n}{\log_2 n}\cdot\sigma}{2^k} + \frac{\sigma^2}{2^{k+n}} \quad .$$

8.3.2 Upper-bounding $\Pr[\mathsf{coll}_{\mathsf{on},\mathsf{off}}]$. We first consider the collisions $K^{[\mathsf{u}_\alpha]} = \hat{K}^{(\beta)}$ and $X^{(\alpha)}_{i,j} = \hat{X}^{(\beta)}$.

- We consider the collisions with $X^{(\alpha)}_{i,j} \in \mathcal{X}^{[\omega]}_{\mathsf{N}}$ for some $\omega \in [u]$. In the d-bound model, for each $\beta \in [p]$, the number of input-output tuples $(K^{[\mathsf{u}_\alpha]}, X^{(\alpha)}_{i,j}, Y^{(\alpha)}_{i,j})$ such that $\hat{X}^{(\beta)} = X^{(\alpha)}_{i,j}$ is at most d, since $\hat{X}^{(\beta)} = X^{(\alpha)}_{i,j} \Rightarrow N^{(\alpha)} = \mathsf{ext}_{\mathsf{nonce}}(X^{(\alpha)}_{i,j}) = \mathsf{ext}_{\mathsf{nonce}}(\hat{X}^{(\beta)})$. Thus, the probability that $\hat{K}^{[\beta]}$ is equal to one of (at most) the d keys is $\frac{d}{2^k}$. Thus, we have $\Pr[\exists\alpha,\beta,i,j \text{ s.t. } K^{[\mathsf{u}_\alpha]} = \hat{K}^{(\beta)} \wedge X^{(\alpha)}_{i,j} = \hat{X}^{(\beta)}] \leq \sum_{\alpha\in[p]} \frac{d}{2^k} = \frac{dp}{2^k}$.
- We consider the collisions with $X^{(\alpha)}_{i,j} \in \mathcal{X}^{[\omega]}_{\neq\mathsf{N}}$ for some $\omega \in [u]$. In this case, $i = 1$ and $1 \leq j$. For $X^{(\alpha)}_{1,j} \in \mathcal{X}^{[\omega]}_{\neq\mathsf{N}}$, it is defined as $X^{(\alpha)}_{1,j} = B^{(\alpha)}_{1,j} \oplus Y^{(\alpha)}_{1,j-1}$ where $Y^{(\alpha)}_{1,j-1}$ is an n-bit random value. By using the n-bit randomness, we have $\Pr[\exists\alpha,\beta,i,j \text{ s.t. } K^{[\mathsf{u}_\alpha]} = \hat{K}^{(\beta)} \wedge X^{(\alpha)}_{i,j} = \hat{X}^{(\beta)}] \leq \sigma\cdot p\cdot\frac{1}{2^{k+n}} = \frac{\sigma p}{2^{k+n}}$.

Using the above bounds, we have

$$\Pr[\mathsf{coll}_{\mathsf{on},\mathsf{off}} \wedge X^{(\alpha)}_{i,j} = \hat{X}^{(\beta)}] \leq \frac{dp}{2^k} + \frac{\sigma p}{2^{k+n}} \quad .$$

We next consider the collisions $K^{[\mathsf{u}_\alpha]} = \hat{K}^{(\beta)}$ and $Y^{(\alpha)}_{i,j} = \hat{Y}^{(\beta)}$. For each $\beta \in [p]$, the number of input-output tuples $(K^{[\mathsf{u}_\alpha]}, X^{(\alpha)}_{i,j}, Y^{(\alpha)}_{i,j})$ such that $\hat{Y}^{(\beta)} = Y^{(\alpha)}_{i,j}$ is at most d due to $\neg\mathsf{mcoll}_Y$. Thus, the number of key candidates that yield the collisions is at most μ, and we have

$$\Pr[\mathsf{coll}_{\mathsf{on},\mathsf{off}} \wedge Y^{(\alpha)}_{i,j} = \hat{Y}^{(\beta)}] \leq p\cdot\frac{\mu}{2^k} = \frac{\frac{n}{\log_2 n}\cdot p}{2^k} \quad .$$

Using these bounds, we have

$$\Pr[\mathsf{coll}_{\mathsf{on},\mathsf{off}}] \leq \frac{\left(d + \frac{n}{\log_2 n}\right)p}{2^k} + \frac{\sigma p}{2^{k+n}} \quad .$$

8.3.3 Upper-bounding $\Pr[\mathsf{mcoll}^{*}_{Y}]$. For each $\alpha_1,\ldots,\alpha_\mu \in [q]$ and $(i_1,i_j) \in \mathsf{Index}^{(\alpha_1)},\ldots,(i_\mu,i_\mu) \in \mathsf{Index}^{(\alpha_\mu)}$ s.t. $\mathsf{u}_{\alpha_1},\ldots,\mathsf{u}_{\alpha_\mu}$ are all distinct, the μ tags $Y^{(\alpha_1)}_{i_1,j_1},\ldots,Y^{(\alpha_\mu)}_{i_\mu,j_\mu}$ are independently chosen, and we have $\Pr[Y^{(\alpha_1)}_{i_1,j_1} = \cdots = Y^{(\alpha_\mu)}_{i_\mu,j_\mu}] \leq \left(\frac{1}{2^n}\right)^{\mu-1}$. Thus,

$$\Pr[\mathsf{mcoll}^{*}_{Y}] \leq \binom{\sigma}{\mu}\left(\frac{1}{2^n}\right)^{\mu-1} \leq 2^n\left(\frac{e\sigma}{\mu 2^n}\right)^{\mu} = 2^n\left(\frac{e\sigma}{\frac{n}{\log_2 n}\cdot 2^n}\right)^{\frac{n}{\log_2 n}}$$
$$\leq \left((2^n)^{\frac{\log_2 n}{n}}\cdot\frac{e\sigma}{\frac{n}{\log_2 n}\cdot 2^n}\right)^{\frac{n}{\log_2 n}} \leq \left(\frac{3(\log_2 n)\sigma}{2^n}\right)^{\frac{n}{\log_2 n}},$$

using Stirling's approximation ($\mu! \geq (\mu/e)^\mu$ for any μ).

8.3.4 Upper-bounding $\Pr[\mathsf{coll}_{\mathsf{on},=\mathsf{u}}]$. For each $\omega \in [u], X_{i_1,j_1}^{(\alpha_1)}, X_{i_2,j_2}^{(\alpha_2)} \in \mathcal{X}^{[\omega]}$ s.t. $X_{i_1,j_1}^{(\alpha_1)} \neq X_{i_2,j_2}^{(\alpha_2)}$, we have $\Pr[Y_{i_1,j_1}^{(\alpha_1)} = Y_{i_2,j_2}^{(\alpha_2)}] \leq \frac{1}{2^n}$. Thus,

$$\Pr[\mathsf{coll}_{\mathsf{on},=\mathsf{u}}] \leq \sum_{\omega \in [u]} \binom{\sigma_\omega}{2} \cdot \frac{1}{2^n} \leq \sum_{\omega \in [u]} \frac{0.5\sigma_\omega^2}{2^n} \ .$$

8.4 Lower-bounding $\frac{\Pr[\mathsf{T}_R = \tau]}{\Pr[\mathsf{T}_2 = \tau]}$

Fix a good transcript $\tau \in \mathcal{T}_{\mathsf{good}}$. Since online-query response tuples are fixed from input-output pairs $(X_{i,j}^{(\alpha)}, Y_{i,j}^{(\alpha)})$ and user's keys, we evaluate the probability that all input-output pairs and user's keys result in the good transcript τ. By $\neg\mathsf{coll}_{\mathsf{on},\mathsf{off}}$, $\neg\mathsf{coll}_{\mathsf{on},\neq\mathsf{u}}$, and $\neg\mathsf{coll}_{\mathsf{on},=\mathsf{u}}$, for each of tuples (K, X, Y) and (K', X', Y') of $E^{\pm}$ in τ, if $K = K', X \neq X' \Leftrightarrow Y \neq Y'$ holds. Hence, $\Pr[\mathsf{T}_2 = \tau] > 0$.

Regarding offline queries, in both worlds, the responses are defined by an IC. Hence, there is no difference between the real world and Game 2.

Regarding user's keys, in both the real world and Game 2, each user's key is chosen uniformly at random from $\{0,1\}^k$, and there is no difference between the real world and Game 2.

Regarding online queries, in Game 2, for each new pair of key and input block, the output block is chosen from $\{0,1\}^n$, whereas it is chosen from $\{0,1\}^n$ excluding the previous output blocks with the same key. Hence, we have $\Pr[\mathsf{T}_R = \tau] \geq \Pr[\mathsf{T}_2 = \tau]$ and

$$\frac{\Pr[\mathsf{T}_R = \tau]}{\Pr[\mathsf{T}_2 = \tau]} \geq 1 \ .$$

8.5 Deriving the Upper-bound

Combining Lemma 4.1 with the upper-bound of $\Pr[\mathsf{T}_2 \in \mathcal{T}_{\mathsf{bad}}]$ and the lower-bound of $\frac{\Pr[\mathsf{T}_R = \tau]}{\Pr[\mathsf{T}_2 = \tau]}$, we obtain the upper-bound in Lemma 7.3.

[End of Proof of Lemma 7.3] ■

9 Mu-Security of CCM with NKD

In this section, we consider the mu-security of CCM_NKD, CCM with the nonce-based key derivation NKD, following the previous application to GCM [14]. Compared to CCM with NR, CCM_NKD replaces the dominant term $\frac{\sigma_{\mathsf{u}}\sigma}{2^n}$ to $\frac{\sigma_{\mathsf{n}}\sigma}{2^n}$, wherein σ_{n} is the maximum number of BC invocations within the same nonce and user's key, thus its online security becomes independent of σ_{u}.

9.1 Specification of CCM_NKD

Let $\mathsf{F}_K : \{0,1\}^\nu \to \{0,1\}^k$ be a KDF with a κ-bit key K that accepts a nonce and returns a nonce-based key of CCM. The encryption and decryption algorithms of CCM_NKD are defined in the following.

- For an input tuple $(N, A, M) \in \{0,1\}^\nu \times \mathcal{A} \times \mathcal{M}$, the encryption CCM_NKD.Enc is defined as

$$\mathsf{CCM\_NKD.Enc}[E, \mathsf{F}_K](N, A, M) := \mathsf{CCM.Enc}[E_{\mathsf{F}_K(N)}](N, A, M).$$

- For an input tuple $(N, A, C, \widetilde{T}) \in \{0,1\}^{\nu} \times \mathcal{A} \times \mathcal{M} \times \{0,1\}^{t}$, the decryption CCM_NKD.Dec is defined as

$$\mathsf{CCM\_NKD.Dec}[E, \mathsf{F}_K](N, A, C, \widetilde{T}) := \mathsf{CCM.Dec}[E_{\mathsf{F}_K(N)}](N, A, C, \widetilde{T}).$$

9.2 Multi-user PRF Security

In our proof, we assume that the KDF is mu-pseudorandom function (mu-PRF) secure. Let u be the number of users. In the mu-PRF-security game, an adversary $\mathbf{A}$ has access to either real-world oracles $(\mathsf{F}_{K_1}, \ldots, \mathsf{F}_{K_u})$ or ideal-world ones $(\mathcal{R}_1, \ldots, \mathcal{R}_u)$, where K_i is the i-th user's key defined as $K_i \xleftarrow{\$} \{0,1\}^{\kappa}$ and $\mathcal{R}_i$ is a random function of the i-th user defined as $\mathcal{R}_i \xleftarrow{\$} \mathsf{Func}(\nu, k)$. At the end of this game, $\mathbf{A}$ return a decision bit. Then, the mu-PRF-security advantage function of $\mathbf{A}$ is defined as

$$\mathbf{Adv}^{\mathsf{muprf}}_{\mathsf{F}}(\mathbf{A}) := \mathbf{Adv}^{\mathsf{dist}}_{(\mathsf{F}_{K_\omega})_{\omega\in[u]},(\mathcal{R}_\omega)_{\omega\in[u]}}(\mathbf{A}).$$

For all possible adversaries $\mathbf{A}$ that have access to u users, make at most q queries, and run in time τ, the maximum advantage is defined as $\mathbf{Adv}^{\mathsf{muprf}}_{\mathsf{F}}(u, q, \tau) := \max_{\mathbf{A}} \mathbf{Adv}^{\mathsf{muprf}}_{\mathsf{F}}(\mathbf{A})$.

9.3 Mu-Security of CCM_NKD

For each nonce, the KDF in CCM_NKD provides a fresh key of CCM under the assumption that F_K is a secure PRF. Hence, in the mu-setting, there are at most q keys of CCM via the KDF in CCM_NKD. By using the bounds in Theorems 5.1 and 7.2, we obtain the following bounds of the mu-AE security of CCM_NKD. Let σ_{n} be the maximum number of BC invocations whose keys are defined by CCM_NKD with the same nonce and user's key.

Corollary 9.1 (Mu-Security of CCM_NKD in the Standard Model). *For any computationally-bounded adversary* $\mathbf{A}$*,*

$$\mathbf{Adv}^{\mathsf{muae}}_{\mathsf{CCM\_NKD}}(\mathbf{A}) \le \mathbf{Adv}^{\mathsf{muprf}}_{\mathsf{F}}(u, q, \tau + O(\sigma)) + \mathbf{Adv}^{\mathsf{muprp}}_{E}(q, \sigma, \tau + O(\sigma)) + \frac{\sigma_{\mathsf{n}}\sigma}{2^n} + \frac{q_{\mathsf{d}}}{2^t}.$$

Corollary 9.2 (Mu-Security of CCM_NKD in the d-bound and IC Models). *For any computationally-bounded adversary* $\mathbf{A}$*,*

$$\mathbf{Adv}^{\mathsf{muae}}_{\mathsf{CCM\_NKD}}(\mathbf{A}) \le \mathbf{Adv}^{\mathsf{muprf}}_{\mathsf{F}}(u, q, \tau + O(\sigma)) + \frac{q_{\mathsf{d}}}{2^t} + \frac{\sigma_{\mathsf{n}}\sigma}{2^n} + \frac{\left(d + \frac{n}{\log_2 n}\right)(p+\sigma)}{2^k} + \left(\frac{3(\log_2 n)\sigma}{2^n}\right)^{\frac{n}{\log_2 n}} + \frac{\sigma(p+\sigma)}{2^{k+n}}.$$

With a discussion similar to Subsection 7.2, dominant terms in the bounds are $\frac{q_{\mathsf{d}}}{2^t} + \frac{\sigma_{\mathsf{n}}\sigma}{2^n} + \frac{dp}{2^k}$.

9.3.1 Choices for PRF. As mentioned in [6, 14, 26], the concatenation of truncated BCs and CENC [20] are nice choices for the KDF in CCM_NKD. Particularly, when implementing AES with AES-NI, the KDF can be efficiently performed.

10 Authenticated Encryption with NTKD

In this section, we present an AE mode AE_NTKD that enhances the mu-security of tag-based and BC-based AE schemes including CCM and GCM by respecting its interfaces. AE_NTKD equips a nonce- and tag-based key derivation NTKD.

10.1 AE_NTKD

10.1.1 Parameters. Let κ, ν, and t be lengths of keys, nonce, and tags of AE_NTKD such that $t \leq n$. In AE_NTKD, AD and a plaintext/ciphertext are divided into data blocks called *sectors*. Each sector is processed by using the underlying AE. Let s be the length of each sector.

10.1.2 The Underlying AE. We define the underlying AE scheme of AE_NTKD. Let AE be a tag-based AE scheme that is a pair of encryption and decryption algorithms $(\mathsf{AE.Enc}, \mathsf{AE.Dec})$. Let $\mathcal{K}$, $\mathcal{N}$, $\mathcal{A}$, $\mathcal{M}$, $\mathcal{C}$, and $\mathcal{T}$ be the sets of keys, nonce, AD, plaintexts, ciphertexts, and tags of AE, respectively. We define the set of tags of AE as $\mathcal{T} := \{0,1\}^t$. The encryption algorithm $\mathsf{AE.Enc} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T}$ takes a tuple (K, N, A, M), and returns, deterministically, a pair of a ciphertext and a tag (C, T). The decryption algorithm $\mathsf{AE.Dec} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \{\textbf{reject}\} \cup \mathcal{M}$ takes a tuple $(K, N, A, C, \widetilde{T})$ and returns, deterministically, either the distinguished invalid symbol $\textbf{reject} \notin \mathcal{M}$ or a plaintext $M \in \mathcal{M}$. We require that $\forall (K, N, A, M), (K', N', A', M') \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$ s.t. $|M| = |M'| : |\mathsf{AE.Enc}(K, A, M)| = |\mathsf{AE.Enc}(K', A', M')|$. We also require that $\forall K \in \mathcal{K}, N, A \in \mathcal{A}, M \in \mathcal{M} : \mathsf{AE.Dec}(K, A, \mathsf{AE.Enc}(K, N,$
$A, M)) = M$. AE.Enc and AE.Dec with a key $K \in \mathcal{K}$ are denoted by $\mathsf{AE.Enc}_K$ and $\mathsf{AE.Dec}_K$. Let $\mathsf{AE}_K := (\mathsf{AE.Enc}_K, \mathsf{AE.Dec}_K)$.

We extract a tag generation function and a core function of AE.Dec from AE. Let $\mathsf{TagGen} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \to \mathcal{T}$ be the tag generation function such that for any $(K, N, A, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$, $(C, \mathsf{TagGen}(K, N, A, C)) = \mathsf{AE.Enc}(K, N, A, M)$ holds. TagGen with a key K is denoted by TagGen_K. Let $\mathsf{AE.Dec}^* : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \times \mathcal{T}$ be the core function of AE.Dec that produces an unverified plaintext M and a tag T, i.e., for an input $(K, N, A, C, \widetilde{T}) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T}$ to AE.Dec, the output is defined as follows: $(M, T) \leftarrow \mathsf{AE.Dec}^*(K, N, A, C)$ and the output is M if $T = \widetilde{T}$; **reject** otherwise.

10.1.3 KDF. We define the underlying KDF F. Let $\mathsf{F} : \{0,1\}^\kappa \times \{0,1\}^\nu \times \{0,1\}^t \to \mathcal{K} \times \mathcal{N}$ be the KDF that takes a κ-bit key, nonce, and tag, and based on nonce and a tag, returns a pair of key and IV. F with a key K is denoted by F_K. For $(K, N, T) \in \{0,1\}^\kappa \times \{0,1\}^\nu \times \{0,1\}^t$, let $\mathsf{F}_K^1(N, T) := \hat{K}$ and $\mathsf{F}_K^2(N, T) := \hat{N}$ such that $\mathsf{F}_K(N, T) = (\hat{K}, \hat{N})$.

10.1.4 Specification of AE_NTKD. We define AE_NTKD. The specification is also given in Algorithm 5 and Figure 1. For $l_{\max}$ which is a maximum number of sectors in AD or a plaintext, let $\mathsf{add}_{\mathsf{ntk}} : \{0,1\}^\nu \times [l_{\max}] \to \mathcal{N}$ be a nonce-updating function that takes nonce $\hat{N}$ and a counter i, and returns nonce of the underlying AE such that $\forall \hat{N} \in \{0,1\}^\nu, i, j \in [l_{\max}]$ s.t. $i \neq j$: $\mathsf{add}_{\mathsf{ntk}}(\hat{N}, i) \neq \mathsf{add}_{\mathsf{ntk}}(\hat{N}, j)$. Note that in Figure 1, $\mathsf{add}_{\mathsf{ntk}}(\hat{N}, i) = \hat{N} + (i-1)$.

$\mathsf{AE\_NTKD.Enc}_K : \{0,1\}^\nu \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^t$ is the encryption algorithm with a key $K \in \{0,1\}^\kappa$ that takes a tuple of nonce, AD, and a plaintext, and returns a pair of a ciphertext and a tag. In AE_NTKD.Enc, AD and plaintext are respectively divided into sectors of s bits $A_1, \ldots, A_a$ and $M_1, \ldots, M_m$. Note that if $A = \varepsilon$, then $a = 0$. First, a nonce-based key $\hat{K}_1$ and a nonce-based IV $\hat{N}$ are defined by using the KDF F_K. Then, by iterating AE.Enc and F_K, AD sectors are processed, followed by the process of plaintext sectors. The KDF takes nonce N and a tag of the previous AE.Enc call, and returns a key of the next AE.Enc call.

$\mathsf{AE\_NTKD.Dec}_K : \{0,1\}^\nu \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^t \to \{0,1\}^* \times \{0,1\}^t$ is the decryption algorithm with a key $K \in \{0,1\}^\kappa$ that takes a tuple of nonce, AD, a plaintext, and a tag, and returns a valid plaintext if the inputs are authenticated; the reject symbol **reject** otherwise. In $\mathsf{AE\_NTKD.Dec}_K$, AD and a ciphertext are respectively divided into sectors $A_1, \ldots, A_a$ and $C_1, \ldots, C_m$. Then, similarly to $\mathsf{AE\_NTKD.Enc}_K$, the AD sector blocks are processed by iterating AE.Enc and F_K, and then the ciphertext sector blocks are processed by iterating $\mathsf{AE.Dec}^*$ (instead of AE.Enc) and F_K.

Algorithm 5 AE_NTKD

Encryption $\mathsf{AE\_NTKD.Enc}(K, N, A, M)$

1: $T_0 \leftarrow 0^t$; $\hat{N} \leftarrow \mathsf{F}^2_K(N, 0^t)$; $A_1, \ldots, A_a \xleftarrow{s} A$; $M_1, \ldots, M_m \xleftarrow{s} M$
2: **for** $i \in [a+m]$ **do**
3: $\quad \hat{K}_i \leftarrow \mathsf{F}^1_K(N, T_{i-1})$; $\hat{N}_i \leftarrow \mathsf{add}_{\mathsf{ntk}}(\hat{N}, i)$
4: $\quad$ **if** $i \le a$ **then** $(C_i, T_i) \leftarrow \mathsf{AE.Enc}(\hat{K}_i, \hat{N}_i, A_i, \varepsilon)$
$\qquad\qquad$ **else** $(C_{i-a}, T_i) \leftarrow \mathsf{AE.Enc}(\hat{K}_i, \hat{N}_i, \varepsilon, M_{i-a})$ **end if**
5: **end for**
6: $C \leftarrow C_1 \| \cdots \| C_m$; $T \leftarrow T_{a+m}$; **return** (C, T)

Decryption $\mathsf{AE\_NTKD.Dec}(K, N, A, C, \widetilde{T})$

1: $T_0 \leftarrow 0^t$; $\hat{N} \leftarrow \mathsf{F}^2_K(N, 0^t)$; $A_1, \ldots, A_a \xleftarrow{s} A$; $C_1, \ldots, C_m \xleftarrow{sn} C$
2: **for** $i \in [a+m]$ **do**
3: $\quad \hat{K}_i \leftarrow \mathsf{F}^1_K(N, T_{i-1})$; $\hat{N}_i \leftarrow \mathsf{add}_{\mathsf{ntk}}(\hat{N}, i)$
4: $\quad$ **if** $i \le a$ **then** $(M_i, T_i) \leftarrow \mathsf{AE.Enc}(\hat{K}_i, \hat{N}_i, A_i, \varepsilon)$
$\qquad\qquad$ **else** $(M_{i-a}, T_i) \leftarrow \mathsf{AE.Dec}^*(\hat{K}_i, \hat{N}_i, \varepsilon, C_{i-a})$ **end if**
5: **end for**
6: $M \leftarrow M_1 \| \cdots \| M_m$; $T \leftarrow T_{a+m}$
7: **if** $T = \widetilde{T}$ **then return** M **else return reject end if**

10.2 Security Model

10.2.1 Security Definition for AE_NTKD. We consider the mu-AE security of AE_NTKD in the IC model, since the security of the underlying AE AE is considered in this model. Let u be the number of users. We use the security definition given in Section 4.3 with
the real-world oracles $\mathcal{O}_{\mathsf{real}} := ((\mathsf{AE\_NTKD}_{K^{[\omega]}})_{\omega\in[u]}, E^{\pm})$ and
the ideal-world oracles $\mathcal{O}_{\mathsf{ideal}} := ((\$_\omega, \perp_\omega)_{\omega\in[u]}, E^{\pm})$,

where $\forall \omega \in [u] : K^{[\omega]} \xleftarrow{\$} \{0,1\}^\kappa$ and $E^{\pm}$ is an IC. Then, the advantage function of an adversary $\mathbf{A}$ is defined as $\mathbf{Adv}^{\mathsf{muae}}_{\mathsf{AE\_NTKD}}(\mathbf{A}) := \mathbf{Adv}^{\mathsf{dist}}_{\mathcal{O}_{\mathsf{real}},\mathcal{O}_{\mathsf{ideal}}}(\mathbf{A})$. Let p and σ be respectively the numbers of offline queries and of BC calls of AE_NTKD in online queries. Let q_{d} be the number of decryption queries. Let $\mathcal{A}$ be the set of all possible nonce-respecting computationally-unbounded adversaries with the resources.

10.2.2 Assumptions. Regarding the KDF, we assume that F is mu-PRF secure. The definition of mu-PRF security is given in Section 9.2.

Regarding the underlying AE, we assume that AE is mu-AE secure in the IC model. Let u_1 be the number of users. We use the security definition given in Section 4.3. In this case, we consider
the real-world oracles: $\mathcal{O}_{\mathsf{real}} := (\{\mathsf{AE}_{K^{[w]}}\}_{w\in[u_1]}, E^{\pm})$ and

the ideal-world oracles: $\mathcal{O}_{\mathsf{ideal}} := (\{(\$_\omega, \perp_\omega)\}_{\omega\in[u_1]}, E^{\pm})$, where $\forall w \in [u_1] : K^{[w]} \xleftarrow{\$} \mathcal{K}$ and $E^{\pm}$ is an IC. Then, the advantage function of an adversary $\mathbf{B}$ is defined as $\mathbf{Adv}^{\mathsf{muae}}_{\mathsf{AE}}(\mathbf{B}) := \mathbf{Adv}^{\mathsf{dist}}_{\mathcal{O}_{\mathsf{real}},\mathcal{O}_{\mathsf{ideal}}}(\mathbf{B})$. Let ℓ_1 be the maximum number of primitive calls of AE per online query. Let $q_{\mathsf{d},1}$ be the number of decryption queries. Let σ_1 be the number of primitive calls of AE in all queries made by $\mathbf{B}$. Let p_1 be the number of offline queries. Let $Q_1 := (u_1, \sigma_1, q_{\mathsf{d},1}, \ell_1, p_1)$ be the query resources of adversaries. Then, for all possible computationally-unbounded adversaries with the resource Q_1, the maximum of the advantage function is denoted by $\mathbf{Adv}^{\mathsf{muae}}_{\mathsf{AE}}(Q_1) := \max_{\mathbf{B}} \mathbf{Adv}^{\mathsf{muae}}_{\mathsf{AE}}(\mathbf{B})$.

In addition to the mu-AE assumption, we assume that TagGen is regular and almost universal. The definitions are given below.

Definition 10.1 (Regular and Almost Universal (AU)). For an input tuple $\mathcal{D}$ to TagGen_K, let $N_{\mathcal{D}}$ be the number of primitive calls (such as BC calls and n-bit field multiplications) of $\mathsf{TagGen}_K(\mathcal{D})$. Let δ be a function that takes the number of primitive calls of TagGen_K with $\mathcal{D}_1$ and returns a probability for regular and AU. TagGen is said to be δ-regular if for any $Y \in \{0,1\}^t$ and any input tuple $\mathcal{D}$, $\Pr[K \xleftarrow{\$} \mathcal{K}; \mathsf{TagGen}_K(\mathcal{D}) = Y] \le \delta(N_{\mathcal{D}}, 0)$. TagGen is said to be δ-AU if for any distinct tuples $\mathcal{D}_1, \mathcal{D}_2$, $\Pr[K \xleftarrow{\$} \mathcal{K}; \mathsf{TagGen}_K(\mathcal{D}_1) = \mathsf{TagGen}_K(\mathcal{D}_2)] \le \delta(N_{\mathcal{D}_1}, N_{\mathcal{D}_2})$.

10.3 mu-AE Security of AE_NTKD

The following theorem shows the mu-AE-security bound of AE_NTKD with the assumptions that F is mu-PRF secure, AE is mu-AE secure, and TagGen is regular and AU. The proof is given in Section 10.5.

Theorem 10.2. *Let b_s be the maximum number of BC calls in* AE.Enc *with a pair of s-bit AD and the empty plaintext or a pair of empty AD and an s-bit plaintext. Let* TagGen *be δ-regular and δ-AU such that for the numbers of primitive calls N_1, N_2 and a positive integer c, $\delta(N_1, N_2) = \frac{c(N_1+N_2)}{2^t}$.*

$$\forall \mathbf{A} \in \mathcal{A} : \mathbf{Adv}^{\mathsf{muae}}_{\mathsf{AE\_NTKD}}(\mathbf{A}) \le \mathbf{Adv}^{\mathsf{muprf}}_{\mathsf{F}}(u, \sigma, \tau + O(\sigma)) + \frac{c\sigma_{\mathsf{n}}\sigma}{b_s 2^t} + \mathbf{Adv}^{\mathsf{muae}}_{\mathsf{AE}}(\boldsymbol{Q}_1) \ ,$$

where $\boldsymbol{Q}_1$ $(= (u_1, \sigma_1, q_{\mathsf{d},1}, \ell_1, p_1)) = (\lfloor \sigma/b_s \rfloor + q, \sigma, q_{\mathsf{d}}, b_s, p)$.

10.4 Applications to CCM and GCM

We first consider AE_NTKD with CCM in the IC model, i.e., AE = CCM. Note that the parameter c of CCM is a constant [1]. Assume that $\mathbf{Adv}^{\mathsf{muprf}}_{\mathsf{F}}(u, \sigma, \tau + O(\sigma))$ is negligible compared with the other terms, which can be realized by using highly secure KDFs, such as BC-based PRFs [6, 8, 14, 26] and SHA-2/3-based KDFs [29, 30].

We evaluate the term $\mathbf{Adv}^{\mathsf{muae}}_{\mathsf{CCM}}(\boldsymbol{Q}_1)$ with Theorem 7.2. Let $\mathbf{B}$ be an adversary with the resource $\boldsymbol{Q}_1$. Let v be the maximum number of decryption queries per user. For $w \in [u_1]$, let σ_w be the number of BC calls in queries to the w-th user. Hence, $\sigma = \sum_{w \in [u_1]} \sigma_w$. By Theorem 7.2, for any adversary $\mathbf{B}$, we have

$$\mathbf{Adv}^{\mathsf{muae}}_{\mathsf{CCM}}(\mathbf{B}) \le \frac{q_{\mathsf{d}}}{2^t} + \sum_{w \in [u_1]} \frac{\sigma_w^2}{2^n} + \frac{\left(d + \frac{n}{\log_2 n}\right)(p+\sigma)}{2^k} + \left(\frac{3(\log_2 n)\sigma}{2^n}\right)^{\frac{n}{\log_2 n}} + \frac{\sigma(p+\sigma)}{2^{k+n}} \ .$$

Regarding the term $\sum_{w \in [u_1]} \frac{\sigma_w^2}{2^n}$, since the number of encryption queries to each user is at most 1 and the number of BC calls in each query is at most b_s, the term is maximum when for each of some $\lfloor q_{\mathsf{d}}/v \rfloor + 1$ users, $\mathbf{B}$ makes v decryption queries that require b_s BC calls per user. Without loss of generality, assume that the user indexes with the decryption queries is from 1 to $\lfloor q_{\mathsf{d}}/v \rfloor + 1$. In this case, for $w_1 \in [\lfloor q_{\mathsf{d}}/v \rfloor + 1]$, $\sigma_{w_1} = b_s + b_s v$, and for $w_2 \in [\lfloor q_{\mathsf{d}}/v \rfloor + 2, u_1]$, $\sigma_{w_2} \le b_s$. We thus have

$$\begin{aligned} \sum_{w \in [u_1]} \frac{\sigma_w^2}{2^n} &\le (\lfloor q_{\mathsf{d}}/v \rfloor + 1) \cdot \frac{(b_s + b_s v)^2}{2^n} + \sum_{w_2 \in [\lfloor q_{\mathsf{d}}/v \rfloor + 2, u_1]} \frac{b_s \sigma_{w_2}}{2^n} \\ &\le \frac{8 b_s^2 v q_{\mathsf{d}}}{2^n} + \frac{b_s \sigma}{2^n} \ . \end{aligned}$$

Regarding the number of forgery attempts v, it can be limited by rekeying. We thus assume that $vq_{\mathrm{d}} \leq \sigma$, and the above bound is at most $\frac{9b_s\sigma}{2^n}$. We then use the parameter $b_s = \sqrt{2^{n-t}\sigma_{\mathrm{n}}}$ that ensures $\frac{b_s\sigma}{2^n} = \frac{\sigma_{\mathrm{n}}\sigma}{b_s 2^t}$. Putting the bound of $\mathbf{Adv}^{\mathrm{muae}}_{\mathrm{CCM}}(\mathbf{B})$ with $b_s = \sqrt{2^{n-t}\sigma_{\mathrm{n}}}$ into Theorem 10.2, the mu-AE-security bound is about

$$\frac{q_{\mathrm{d}}}{2^t} + \frac{\sqrt{2^{n-t}\sigma_{\mathrm{n}}}\sigma}{2^n} + \frac{dp}{2^k} \ .$$

When $t = n$, the bound ensures that AE_NTKD with CCM is mu-AE secure as long as $\sigma \leq 2^n/\sqrt{\sigma_{\mathrm{n}}}$ and $p \leq 2^k$.

Regarding GCM [11], Hoang et al. [16] derive the same bound for $\mathbf{Adv}^{\mathrm{muae}}_{\mathrm{GCM}}(Q_1)$ as our CCM's bound in the d-bound model. Note that GCM uses the MAC algorithm GMAC and the parameter c of GMAC is a constant. Hence, AE_NTKD with GCM is as secure as AE_NTKD with CCM regarding mu-AE security. By using $b_s = \sqrt{\sigma_{\mathrm{n}}}$ and assuming that $\mathbf{Adv}^{\mathrm{muprf}}_{\mathsf{F}}(u, \sigma, \tau + O(\sigma))$ is negligible and $vq_{\mathrm{d}} \leq \sigma$, the mu-AE-security bound is about $\frac{q_{\mathrm{d}}}{2^t} + \frac{\sqrt{2^{n-t}\sigma_{\mathrm{n}}}\sigma}{2^n} + \frac{dp}{2^k}$. When $t = n$, AE_NTKD with GCM is mu-AE secure as long as $\sigma \leq 2^n/\sqrt{\sigma_{\mathrm{n}}}$ and $p \leq 2^k$.

10.5 Proof of Theorem 10.2

We first modify the real world, where $\mathcal{R}_\omega$ is replaced with $\mathsf{F}_{K[\omega]}$ for each $\omega \in [u]$. The modified world is called "middle world." Hence, an adversary $\mathbf{A}$ interacts with the middle-world oracles $\mathcal{O}_{\mathrm{middle}} := ((\mathsf{AE\_NTKD}[\mathcal{R}_\omega])_{\omega\in[u]}, E^{\pm})$, where $E^{\pm}$ is an IC and $\mathsf{AE\_NTKD}[\mathcal{R}_\omega]$ is $\mathsf{AE\_NTKD}_{K[\omega]}$ with $\mathcal{R}_\omega$. We then have

$$\mathbf{Adv}^{\mathrm{muae}}_{\mathsf{AE\_NTKD}}(\mathbf{A}) = \left(\Pr\left[\mathbf{A}^{\mathcal{O}_{\mathrm{real}}} = 1\right] - \Pr\left[\mathbf{A}^{\mathcal{O}_{\mathrm{middle}}} = 1\right]\right) + \left(\Pr\left[\mathbf{A}^{\mathcal{O}_{\mathrm{middle}}} = 1\right] - \Pr\left[\mathbf{A}^{\mathcal{O}_{\mathrm{ideal}}} = 1\right]\right) \ .$$

We evaluate each difference in the followings.

10.5.1 Upper-bounding $\Pr\left[\mathbf{A}^{\mathcal{O}_{\mathrm{real}}} = 1\right] - \Pr\left[\mathbf{A}^{\mathcal{O}_{\mathrm{middle}}} = 1\right]$. By the replacement from $\mathcal{O}_{\mathrm{real}}$ to $\mathcal{O}_{\mathrm{middle}}$, the difference is bounded by the mu-PRF advantage, i.e., $\Pr[\mathbf{A}^{\mathcal{O}_{\mathrm{real}}} = 1] - \Pr[\mathbf{A}^{\mathcal{O}_{\mathrm{middle}}} = 1] \leq \mathbf{Adv}^{\mathrm{muprf}}_{\mathsf{F}}(u, \sigma, \tau + O(\sigma))$.

10.5.2 Upper-bounding $\Pr\left[\mathbf{A}^{\mathcal{O}_{\mathrm{middle}}} = 1\right] - \Pr\left[\mathbf{A}^{\mathcal{O}_{\mathrm{ideal}}} = 1\right]$. We use the following notations. For $\alpha \in [q]$, let m_α and a_α be the lengths m and a of the plaintext and AD in the α-th online query. For $\alpha \in [q]$, values regarding the α-th online query are denoted by using the superscript symbol of (α), e.g., $M^{(\alpha)}, C^{(\alpha)}$, etc. Let u_α be the user index of the α-th online query.

We next define the following collision events in the middle world. For $\alpha \in [q]$ and $i \in [a_\alpha + m_\alpha]$, let $\mathcal{D}_i^{(\alpha)}$ be a pair of an AD sector and a ciphertext sector at the i-th AE call of the α-th online query. If $i \leq a_\alpha$, then $\mathcal{D}_i^{(\alpha)} = (A_i^{(\alpha)}, \varepsilon)$; if $i > a_\alpha$, then $\mathcal{D}_i^{(\alpha)} = (\varepsilon, M_{i-a_\alpha}^{(\alpha)})$.

$$\begin{aligned}\mathrm{coll} \Leftrightarrow \exists \alpha, \beta \in [q], i \in [a_\alpha + m_\alpha], j \in [a_\beta + m_\beta] \text{ s.t. } (\alpha, i) \neq (\beta, j)\\ \wedge\ \mathsf{u}_\alpha = \mathsf{u}_\beta \wedge N^{(\alpha)} = N^{(\beta)}\\ \wedge\ (T_{i-1}^{(\alpha)}, \mathcal{D}_i^{(\alpha)}) \neq (T_{j-1}^{(\beta)}, \mathcal{D}_j^{(\beta)}) \wedge T_i^{(\alpha)} = T_j^{(\beta)}.\end{aligned}$$

The collision event considers a tag collision of some two distinct sectors or keys, yielding a key collision of the underlying AE. In other worlds, each key of the AE is independently chosen as long as coll does not occur. With the event, we have

$$\Pr\left[\mathbf{A}^{\mathcal{O}_{\mathrm{middle}}} = 1\right] - \Pr\left[\mathbf{A}^{\mathcal{O}_{\mathrm{ideal}}} = 1\right] \leq \Pr\left[\mathbf{A}^{\mathcal{O}_{\mathrm{middle}}} = 1 \mid \neg\mathrm{coll}\right] - \Pr\left[\mathbf{A}^{\mathcal{O}_{\mathrm{ideal}}} = 1\right] + \Pr[\mathrm{coll}] \ .$$

10.5.3 Upper-bounding $\Pr\left[\mathbf{A}^{O_{\mathrm{middle}}} = 1 \mid \neg\mathrm{coll}\right] - \Pr\left[\mathbf{A}^{O_{\mathrm{ideal}}} = 1\right]$. We give an overview of the evaluation. The detail evaluation is given in Section 11.

Assume that coll does not occur. In the nonce-respecting setting, all keys of AE.Enc in the middle world are independently chosen by RFs $(\mathcal{R}_\omega)_{\omega\in[u]}$. Hence, $\{\mathsf{AE\_NTKD.Enc}[\mathcal{R}_\omega]\}_{\omega\in[u]}$ behave as random-bit oracle up to the advantage $\mathbf{Adv}^{\mathrm{muae}}_{\mathsf{AE}}(Q_1)$, where $Q_1 = (\lceil\sigma/b_s\rceil+q, \sigma, q_{\mathrm{d}}, b_s, p)$. Moreover, each tag of AE_NTKD.Dec in the middle world is defined by using AE.Dec. Hence, the probability of forging a tag of $\mathsf{AE\_NTKD.Dec}[\mathcal{R}_\omega]$ is bounded by the advantage $\mathbf{Adv}^{\mathrm{muae}}_{\mathsf{AE}}(Q_1)$. We thus have $\Pr\left[\mathbf{A}^{O_{\mathrm{middle}}} = 1 \mid \neg\mathrm{coll}\right] - \Pr\left[\mathbf{A}^{O_{\mathrm{ideal}}} = 1\right] \le \mathbf{Adv}^{\mathrm{muae}}_{\mathsf{AE}}(Q_1)$.

10.5.4 Upper-bounding $\Pr[\mathrm{coll}]$. For each $\alpha,\beta \in [q], i \in [a_\alpha + m_\alpha], j \in [a_\beta + m_\beta]$ such that $(\alpha, i) \neq (\beta, j)$, $\mathsf{u}_\alpha = \mathsf{u}_\beta$ and $N^{(\alpha)} = N^{(\beta)}$, if $T^{(\alpha)}_{i-1} \neq T^{(\beta)}_{j-1}$, then the keys $\hat{K}^{(\alpha)}_i$ and $\hat{K}^{(\beta)}_j$ are independently chosen, and thus by the regular property of TagGen, the probability of the tag collision $T^{(\alpha)}_i = T^{(\beta)}_j$ is at most $\frac{cb_s}{2^n}$. If $T^{(\alpha)}_{i-1} = T^{(\beta)}_{j-1}$ and $\mathcal{D}^{(\alpha)}_i \neq \mathcal{D}^{(\beta)}_j$, then by the AXU or regular property of TagGen, the probability of the tag collision is at most $\frac{2cb_s}{2^n}$.

For $\omega \in [u]$, let N_ω be the number of distinct nonces in queries to the ω-th user. For $i \in [N_\omega]$, let $\sigma_{\omega,i}$ be the number of BC calls in online queries with the i-th nonce to the ω-th user. Then, for each $i \in [N_\omega]$, there are at most $\lceil\frac{\sigma_{\omega,i}}{b_s}\rceil$ keys of AE. By using the above bounds, we have

$$\begin{aligned}\Pr[\mathrm{coll}] &\le \sum_{\substack{\omega\in[u]\\ i\in[N_\omega]}} \binom{\lceil\frac{\sigma_{\omega,i}}{b_s}\rceil}{2} \cdot \frac{2cb_s}{2^t} \le \sum_{\substack{\omega\in[u]\\ i\in[N_\omega]}} \frac{0.5(\lceil\frac{\sigma_{\omega,i}}{b_s}\rceil)^2 \cdot 2cb_s}{2^t} \\ &\le \sum_{\omega\in[u], i\in[N_\omega]} \frac{4c(\sigma_{\omega,i}/b_s)^2 \cdot b_s}{2^t} \le \frac{4c\sigma_{\mathsf{n}}\sigma}{b_s 2^t} \ .\end{aligned}$$

10.5.5 Conclusion of the Proof. Combining the above bounds, we obtain the bound in Theorem 10.2.

11 Detail Evaluation of Section 10.5.3

Assume that coll does not occur in the middle world. Let $\mathbf{A}$ be an mu-AE-adversary against AE_NTKD that makes p offline queries and online queries such that the number of BC calls is at most σ and q_{d} decryption queries.

We construct an mu-AE adversary $\mathbf{B}$ against the AE scheme AE that has access to u_1 $(= \lceil\sigma/b_s\rceil+q)$ users and an IC $E^{\pm} = (E, E^{-1})$. For $w \in [u_1]$, let Enc_w (resp. Dec_w) be the encryption (resp. decryption) oracle of the w-th user. In the middle world, these oracles are $(\mathsf{AE\_NTKD}[\mathcal{R}_w])_{w\in[u_1]}$. In the ideal world, these oracles are ideal AEs $\{(\$_w, \perp_w)\}_{w\in[u_1]}$.

We define the adversary $\mathbf{B}$ in Algorithm 6 that simulates the $\mathbf{A}$'s environment by using the $\mathbf{B}$'s oracles $(\mathsf{Enc}_w, \mathsf{Dec}_w)_{w\in[u_1]}$. In this simulation, for encryption queries by $\mathbf{A}$, each sector of AD and plaintexts is processed by $\mathbf{B}$'s encryption oracles. In the middle world, each key of Enc_w is regarded as an output of $\mathcal{R}_w$. In the process of the i-th sector, the key of Enc_w (or user index) is determined by the nonce and sectors processed so far that we call "prefix data sequence." If $i \le a+1$, then the prefix data sequence is $(N, A_1\|\dots\|A_{i-1})$ and if $i > a+1$, then it is $(N, A, C_1\|\dots\|C_{i-1})$. Note that for prefix data sequences, we consider ciphertext sectors instead of plaintext sectors. For decryption queries by $\mathbf{A}$, the responses are defined by using $\mathbf{B}$'s decryption oracles Dec_w. For prefix data sequences of the decryption procedure, we consider only the last $(a+m)$-th Dec_w calls, since the decryption procedure performs Dec_w for the last sector. $\mathbf{B}$ has three tables $\mathcal{E}$, $\mathcal{N}$ and $\mathcal{U}$, and two variables $\mathsf{u}_{\mathsf{max}}$ and b. For offline queries by $\mathbf{A}$, the responses are defined by using $\mathbf{B}$'s IC $E^{\pm} = (E, E^{-1})$. Before $\mathbf{A}$ makes the first query, $\mathbf{B}$ performs the initialization that initializes these

Algorithm 6 Adversary **B**

Initialization

$\mathcal{E} \leftarrow \emptyset$; $\mathcal{N} \leftarrow \emptyset$; $\mathcal{U} \leftarrow \emptyset$; $\mathsf{u}_{\max} \leftarrow 0$; $b \leftarrow 0$

Encryption procedure for **A**'s query to the ω-th user (N, A, M)

$T_0 \leftarrow 0^t$; $A_1, \ldots, A_a \xleftarrow{s} A$; $M_1, \ldots, M_m \xleftarrow{s} M$
$\hat{N} \leftarrow \mathcal{N}(\omega, N)$; **if** $\mathcal{N}(\omega, N) = \varepsilon$ **then** $\hat{N} \xleftarrow{\$} \{0,1\}^\nu$ **end if**
$\mathcal{N} \xleftarrow{\cup} \{\omega, N, \hat{N}\}$; $A^* \leftarrow \varepsilon$; $C^* \leftarrow \varepsilon$
for $i \in [a+m]$ **do**
  $\hat{N}_i \leftarrow \mathsf{add}_{\mathsf{ntk}}(\hat{N}, i)$; $w \leftarrow \mathcal{U}(\omega, N, A^*, C^*)$
  if $w = \varepsilon$ **then** $\mathsf{u}_{\max} \leftarrow \mathsf{u}_{\max} + 1$; $w \leftarrow \mathsf{u}_{\max}$ **end if**
  if $i \le a$ **then** $\mathcal{D}_i \leftarrow (A_i, \varepsilon)$ **else** $\mathcal{D}_i \leftarrow (\varepsilon, M_{i-a})$ **end if**
  $(C_i, T_i) \leftarrow \mathcal{E}(w, \hat{N}_i, \mathcal{D}_i)$
  if $T_i = \varepsilon$ **then** $(C_i, T_i) \leftarrow \mathsf{Enc}_w(\hat{N}_i, \mathcal{D}_i)$ **end if**
  $\mathcal{E} \xleftarrow{\cup} (w, \hat{N}_i, \mathcal{D}_i, C_i, T_i)$; $\mathcal{U} \xleftarrow{\cup} (\omega, N, A^*, C^*, w)$
  if $i \le a$ **then** $A^* \leftarrow A^* \| A_i$ **else** $C^* \leftarrow C^* \| C_i$ **end if**
end for
$C \leftarrow C_1 \| \cdots \| C_m$; $T \leftarrow T_{a+m}$; **return** (C, T)

Decryption procedure for **A**'s query to the ω-th user (N, A, C)

$T_0 \leftarrow 0^t$; $A_1, \ldots, A_a \xleftarrow{s} A$; $C_1, \ldots, C_m \xleftarrow{s} C$
$\hat{N} \leftarrow \mathcal{N}(\omega, N)$; **if** $\mathcal{N}(\omega, N) = \varepsilon$ **then** $\hat{N} \xleftarrow{\$} \{0,1\}^\nu$ **end if**
$\mathcal{N} \xleftarrow{\cup} \{\omega, N, \hat{N}\}$
if $C = \varepsilon$ **then** $A^* \leftarrow A_1 \| \cdots \| A_{a-1}$; $C^* \leftarrow \varepsilon$ **else** $A^* \leftarrow A$; $C^* \leftarrow C_1 \| \cdots \| C_{m-1}$ **end if**
$w \leftarrow \mathcal{U}(\omega, N, A^*, C^*)$
if $w = \varepsilon$ **then** $\mathsf{u}_{\max} \leftarrow \mathsf{u}_{\max} + 1$; $w \leftarrow \mathsf{u}_{\max}$ **end if**
$\hat{N}_{a+m} \leftarrow \mathsf{add}_{\mathsf{ntk}}(\hat{N}, a+m)$
if $C = \varepsilon$ **then** $\mathcal{D}_{a+m} \leftarrow (A_a, \varepsilon)$ **else** $\mathcal{D}_{a+m} \leftarrow (\varepsilon, C_m)$ **end if**
$M \leftarrow \mathsf{Dec}_w(\hat{N}_{a+m}, \mathcal{D}_{a+m})$; $\mathcal{U} \xleftarrow{\cup} (\omega, N, A^*, C^*, w)$
if $M =$ **reject then return reject**
  else $b \leftarrow 1$; **goto** the finalization **end if**

A's offline query $(\hat{K}, \hat{X})$ to E

Makes an offline query $(\hat{K}, \hat{X})$ to the **B**'s oracle E and receive the response $\hat{Y}$
return $\hat{Y}$

A's offline query $(\hat{K}, \hat{Y})$ to E^{-1}

Makes an offline query $(\hat{K}, \hat{Y})$ to the **B**'s oracle E^{-1} and receive the response $\hat{X}$
return $\hat{X}$

Finalization

if $b = 0$ **then** $b \leftarrow$ (an output of **A**) **end if**
return b

tables and variables. After finishing all queries of **A** or a valid plaintext is returned in **B**'s decryption procedure, **B** runs the finalization.

Regarding the tables, $\mathcal{E}$ keeps tuples $(w, \hat{N}, \mathcal{D}, C, T)$ where w is a user index of an **B**'s oracle, $\hat{N}$ is a nonce-based IV, $\mathcal{D}$ is an input pair of AD and plaintext sectors to Enc_w, and C and T are the ciphertext and tag of Enc_w with these inputs. Let $\mathcal{E}(\omega, \hat{N}, \mathcal{D}) \coloneqq (C, T)$ if $(\omega, \hat{N}, \mathcal{D}, C, T) \in \mathcal{E}$; $\mathcal{E}(\omega, \hat{N}, \mathcal{D}) \coloneqq (\varepsilon, \varepsilon)$ otherwise. $\mathcal{N}$ keeps tuples $(\omega, N, \hat{N})$ where ω is a user index ω of **A**'s oracles, N is nonce in a query by **A**, and $\hat{N}$ is a nonce-based IV corresponding with N. Let $\mathcal{N}(\omega, N) \coloneqq \hat{N}$ if $(\omega, N, \hat{N}) \in \mathcal{N}$ and $\mathcal{N}(\omega, N) \coloneqq \varepsilon$ otherwise. $\mathcal{U}$ keeps tuples (ω, N, A^*, C^*, w) where ω is a user index ω of **A**'s oracles, N is nonce in a query by **A**, A^* is a concatenation of AD sectors, C^* is a concatenation of ciphertext sectors, and w is a user index of **B**'s oracles. Let $\mathcal{U}(\omega, N, A^*, C^*) \coloneqq w$ if $(\omega, N, A^*, C^*, w) \in \mathcal{U}$; $\mathcal{U}(\omega, N, A^*, C^*) \coloneqq \varepsilon$ otherwise. The table is used for extracting a user index which is performed in some previous query with the same prefix data sequence.

Regarding the variables, $\mathsf{u}_{\mathsf{max}}$ keeps the maximum user index of **B**'s oracles that **B** had access to and b is the output bit of **B**. These variables are initially set to 0. $\mathsf{u}_{\mathsf{max}}$ becomes 1 if **B** makes a decryption query such that a valid plaintext is returned or **A** returns 1; otherwise 0.

By the following justifications for the algorithm, **B** succeeds in simulating the **A**'s environment. If **A** returns 1 or **B** makes a decryption query to Dec_w such that a valid plaintext is returned, then **B** returns 1. Hence, for any adversary **A**, the adversary **B** ensures that

$$\Pr\left[\mathbf{A}^{\mathcal{O}_{\mathsf{middle}}} = 1 \mid \neg\mathsf{coll}\right] - \Pr\left[\mathbf{A}^{\mathcal{O}_{\mathsf{ideal}}} = 1\right] \leq \mathbf{Adv}^{\mathsf{muae}}_{\mathsf{AE}}(\mathbf{B}).$$

The number of offline queries by **B** is p. Since BC calls of each AE call is b_s, the number of users in **B**'s game is at most $\lceil \sigma/b_s \rceil + q$. The maximum number of BC calls in online queries by **B** is σ. The number of decryption queries by **B** is q_{d}. Hence, **B**'s resource is $\boldsymbol{Q}_1 = (\lceil \sigma/b_s \rceil + q, \sigma, q_{\mathrm{d}}, b_s, p)$.

Justification of the Encryption Procedure in Algorithm 6. In the real world, regarding the encryption procedure for the α-th online query made by **A**, **B** first defines a nonce-based IV $\hat{N}^{(\alpha)}$. If there exists $\beta \in [q]$ such that $\mathsf{u}_\alpha = \mathsf{u}_\beta$ and the β-th online query by **A** is a decryption one with $N^{(\alpha)} = N^{(\beta)}$, $\hat{N}^{(\alpha)} = \hat{N}^{(\beta)}$ must hold. This condition is realized by using the table $\mathcal{N}$. If the same nonce is not in $\mathcal{N}$, $\hat{N}^{(\alpha)}$ is chosen uniformly at random from $\{0,1\}^\nu$, since it is defined by an RF in the middle world. With the (simulated) nonce-based IV $\hat{N}^{(\alpha)}$, AD and plaintext sectors of the α-th online query are processed by iterating **B**'s encryption oracles $(\mathsf{Enc}_w)_{w\in[u_1]}$. We assume that coll does not occur, thus there is no collision occur in inputs to RFs in the encryption queries to the same user, ensuring that the keys of the underlying AE's encryptions are all independent. Thus, our simulation succeeds in simulating **A**'s encryption oracles in the middle world.

Note that **A** can make encryption and decryption queries with the same nonce to the same user. Hence, in the middle world, the keys of the underlying AEs of encryption and decryption queries must be the same if the prefix sequences of the keys are the same. This case is managed by using the table $\mathcal{U}$ in our algorithm which keeps tuples of the prefix sequence and the following user index.

In the ideal world, ciphertext sectors and tags are defined by using random-bit oracles $(\$_w)_{w\in[u_1]}$, and thus our simulation succeeds in simulating **A**'s encryption oracles in the ideal world.

Justification of the Decryption Procedure in Algorithm 6. In the real world, for each β-th online query made by **A** that is a decryption one, if there exists an α-th online query such that the query is an encryption one, $\mathsf{u}_\alpha = \mathsf{u}_\beta$, $N^{(\beta)} = N^{(\alpha)}$, and the prefix data sequence of the last sector in the β-th query is equal to some prefix sequence of the α-th query, then the output of the decryption oracle is defined by using the same user as the encryption query by the table $\mathcal{U}$. If there is no such encryption query, then **B** uses a new user's decryption oracle. Hence, our simulation succeeds in

simulating the **A**'s decryption oracles in the middle world as long as the decryption oracles do not return a plaintext.

In the ideal world, for each decryption query from **B**, our algorithm returns **reject**.

Hence, our simulation succeeds in simulating the **A**'s decryption oracles in the ideal world.

12 Conclusion

This work studied mu-security of CCM, which has received less attention compared to GCM. We presented the improved mu-bound of CCM given by $\frac{\sigma_u\sigma}{2^n} + \frac{q_d}{2^t} + \frac{up+u^2}{2^k}$. Our bound improves the online term of the previous bound from $\min\{\frac{u\sigma_u^2}{2^n}, \frac{\sigma^2}{2^n}\}$ to $\frac{\sigma_u\sigma}{2^n}$, and this matches the mu-bound of GCM. We further studied how the existing enhancing methods developed for GCM, nonce randomization (NR) and nonce-based key derivation (NKD), could be applied to CCM. NR enhances the third term representing offline security, yielding the enhanced bound of $\frac{\sigma_u\sigma}{2^n} + \frac{q_d}{2^t} + \frac{dp}{2^k}$ in the d-bound model. NKD further replaces the first term to $\frac{\sigma_n\sigma}{2^n}$. As a result, the mu-security of CCM becomes as good as those based on GCM. Given that some real-world applications would require even higher mu-bound, we proposed a new enhancement method nonce-based and tag-based key derivation (NTKD) that is applied to both GCM and CCM. NTKD enhances the mu-bound to $\frac{\sqrt{\sigma_n}\sigma}{2^n} + \frac{q_d}{2^t} + \frac{dp}{2^k}$, which meets such real-world needs when BC is AES.

Dedicated mu-security analysis of symmetric-key schemes is still emerging, and there are several targets for further research, including the conventional CCM variants such as CCM-SIV [23] and EAX [2], and the newer schemes such as GCM-SST [7, 18].

References

[1] Mihir Bellare, Krzysztof Pietrzak, and Phillip Rogaway. 2005. Improved Security Analyses for CBC MACs. In *CRYPTO 2005 (LNCS)*, Vol. 3621. Springer, 527–545.

[2] Mihir Bellare, Phillip Rogaway, and David A. Wagner. 2004. The EAX Mode of Operation. In *FSE 2004 (LNCS)*, Vol. 3017. Springer, 389–407.

[3] Mihir Bellare and Björn Tackmann. 2016. The Multi-user Security of Authenticated Encryption: AES-GCM in TLS 1.3. In *CRYPTO 2016 (LNCS)*, Vol. 9814. Springer, 247–276.

[4] Eli Biham. 2002. How to Decrypt or Even Substitute DES-Encrypted Messages in 2^{28} Steps. *Inf. Process. Lett.* 84, 3 (2002), 117–124.

[5] David L. Black and David A. McGrew. 2008. Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol. *RFC* 5282 (2008), 1–19.

[6] Priyanka Bose, Viet Tung Hoang, and Stefano Tessaro. 2018. Revisiting AES-GCM-SIV: Multi-user Security, Faster Key Derivation, and Better Bounds. In *EUROCRYPT 2018 (LNCS)*, Vol. 10820. 468–499.

[7] Matt Campagna, Alexander Maximov, and John P. Mattsson. 2023. Galois Counter Mode with Secure Short Tags (GCM-SST). https://datatracker.ietf.org/doc/draft-mattsson-cfrg-aes-gcm-sst/03/. (2023).

[8] Benoît Cogliati, Ashwin Jha, and Mridul Nandi. 2020. How to Build Optimally Secure PRFs Using Block Ciphers. In *ASIACRYPT 2020 (LNCS)*, Shiho Moriai and Huaxiong Wang (Eds.), Vol. 12491. Springer, 754–784.

[9] Jean Paul Degabriele, Jérôme Govinden, Felix Günther, and Kenneth G. Paterson. 2021. The Security of ChaCha20-Poly1305 in the Multi-User Setting. In *CCS 2021*. ACM, 1981––2003.

[10] Morris Dworkin. 2007. NIST Special Publication 800-38C: Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality. https://csrc.nist.gov/pubs/sp/800/38/c/upd1/final. (2007).

[11] Morris Dworkin. 2007. NIST Special Publication 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. https://csrc.nist.gov/pubs/sp/800/38/d/final. (2007).

[12] Morris Dworkin. 2016. NIST Special Publication 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. https://csrc.nist.gov/pubs/sp/800/38/b/upd1/final. (2016).

[13] Shay Gueron, Adam Langley, and Yehuda Lindell. 2019. AES-GCM-SIV: Nonce Misuse-Resistant Authenticated Encryption. *RFC* 8452 (2019), 1–42.

[14] Shay Gueron and Yehuda Lindell. 2017. Better Bounds for Block Cipher Modes of Operation via Nonce-Based Key Derivation. In *CCS 2017*. ACM, 1019–1036.

[15] Felix Günther, Martin Thomson, and Christopher A. Wood. 2023. Usage Limits on AEAD Algorithms. https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-aead-limits-07. (2023).

[16] Viet Tung Hoang, Stefano Tessaro, and Aishwarya Thiruvengadam. 2018. The Multi-user Security of GCM, Revisited: Tight Bounds for Nonce Randomization. In *CCS 2018*. ACM, 1429–1440.

[17] Russell Housley. 2005. Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP). *RFC* 4309 (2005), 1–13.

[18] Akiko Inoue, Ashwin Jha, Bart Mennink, and Kazuhiko Minematsu. 2024. Generic Security of GCM-SST. *IACR Cryptol. ePrint Arch.* (2024), 1928. https://eprint.iacr.org/2024/1928

[19] ISO. 2020. ISO/IEC 19772:2020 Information Security—Authenticated Encryption. (2020).

[20] Tetsu Iwata. 2006. New Blockcipher Modes of Operation with Beyond the Birthday Bound Security. In *FSE 2006 (LNCS)*, Vol. 5665. Springer, 67–83.

[21] Jakob Jonsson. 2002. On the Security of CTR + CBC-MAC. In *Selected Areas in Cryptography, SAC 2002 (LNCS)*, Vol. 2595. Springer, 76–93.

[22] Panos Kampanakis, Matt Campagna, Eric Crocket, and Adam Petcher. 2023. Practical Challenges with AES-GCM and the Need for a New Mode and Wide-Block Cipher. presented at NIST The Third NIST Workshop on Block Cipher Modes of Operation 2023, https://csrc.nist.gov/Presentations/2023/practical-challenges-with-aes-gcm. (2023).

[23] Patrick Kresmer and Alexander Zeh. 2019. CCM-SIV: Single-PRF Nonce-Misuse-Resistant Authenticated Encryption. *IACR Cryptol. ePrint Arch.* (2019), 892. https://eprint.iacr.org/2019/892

[24] Atul Luykx, Bart Mennink, and Kenneth G. Paterson. 2017. Analyzing Multi-key Security Degradation. In *ASIACRYPT 2017 (LNCS)*, Vol. 10625. Springer, 575–605.

[25] David A. McGrew and Daniel V. Bailey. 2012. AES-CCM Cipher Suites for Transport Layer Security (TLS). *RFC* 6655 (2012), 1–8.

[26] Yusuke Naito. 2017. Tweakable Blockciphers for Efficient Authenticated Encryptions with Beyond the Birthday-Bound Security. *IACR Trans. Symmetric Cryptol.* 2017, 2 (2017), 1–26.

[27] National Institute of Standards and Technology. 2025. Pre-Draft Call for Comments: GCM and GMAC Block Cipher Modes of Operation. https://csrc.nist.gov/pubs/sp/800/38/d/r1/iprd. (2025).

[28] Yoav Nir. 2015. ChaCha20, Poly1305, and Their Use in the Internet Key Exchange Protocol (IKE) and IPsec. *RFC* 7634 (2015), 1–13.

[29] NIST. 2008. FIPS Pub. 198-1: The Keyed-Hash Message Authentication Code (HMAC). https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.198-1.pdf. (2008).

[30] NIST. 2015. FIPS Pub. 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions). https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf. (2015).

[31] Jacques Patarin. 2008. The "Coefficients H" Technique. In *Selected Areas in Cryptography, SAC 2008 (LNCS)*, Vol. 5381. Springer, 328–345.

[32] Eric Rescorla. 2018. The Transport Layer Security (TLS) Protocol Version 1.3. *RFC* 8446 (2018), 1–160.

[33] Eric Rescorla, Hannes Tschofenig, and Nagendra Modadugu. 2021. The Datagram Transport Layer Security (DTLS) Protocol Version 1.3 – draft-ietf-tls-dtls13-43. https://tools.ietf.org/html/draft-ietf-tls-dtls13-43. (2021).

[34] Phillip Rogaway and David A. Wagner. 2003. A Critique of CCM. Cryptology ePrint Archive, Paper 2003/070. (2003).

[35] Joseph A. Salowey, David McGrew, and Abhijit Choudhury. 2008. AES Galois Counter Mode (GCM) Cipher Suites for TLS. *RFC* 5288 (2008), 1–8.

[36] Mick Seaman. 2018. IEEE Standard for Local and Metropolitan Area Networks—Media Access Control (MAC) Security. https://ieeexplore.ieee.org/document/8585421. (2018).

[37] Martin Thomson and Sean Turner. 2021. Using TLS to Secure QUIC. *RFC* 9001 (2021), 1–52.

[38] John Viega and David McGrew. 2006. The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH. *RFC* 4543 (2006), 1–14.

[39] Doug Whiting, Russell Housley, and Niels Ferguson. 2002. IEEE P802.11 Wireless LANs: AES Encryption & Authentication Using CTR Mode & CBC-MAC. https://mentor.ieee.org/802.11/dcn/02/11-02-0001-02-000i-aes-encryption-authentication-using-ctr-mode-with-cbc-mac.doc. (2002).

[40] Doug Whiting, Russell Housley, and Niels Ferguson. 2003. Counter with CBC-MAC (CCM). *RFC* 3610 (2003), 1–26.

[41] Wi-Fi Alliance. 2023. WPA3 Specification Version 3.2. https://www.wi-fi.org/system/files/WPA3%20Specification%20v3.2.pdf. (2023).

[42] Martin Woolley. 2023. Bluetooth Core Specification Version 5.4. (2023).

[43] Xiangyang Zhang, Yaobin Shen, and Lei Wang. 2024. Multi-User Security of CCM Authenticated Encryption Mode. In *CCS 2024*. ACM, 4331–4345.

[44] ZigBee Alliance, Inc. 2015. ZigBee Specification. https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf. (2015).