A preliminary version of this paper appears in the Proceedings of CCS 2018. This is the full version.

The Multi-user Security of GCM, Revisited: Tight Bounds for Nonce Randomization

Viet Tung Hoang[1] Stefano Tessaro[2] Aishwarya Thiruvengadam[3]

August 2, 2021

Abstract

Multi-user (mu) security considers large-scale attackers (e.g., state actors) that given access to a number of sessions, attempt to compromise *at least* one of them. Mu security of authenticated encryption (AE) was explicitly considered in the development of TLS 1.3.

This paper revisits the mu security of GCM, which remains to date the most widely used dedicated AE mode. We provide new concrete security bounds which improve upon previous work by adopting a refined parameterization of adversarial resources that highlights the impact on security of (1) nonce re-use across users and of (2) re-keying.

As one of the main applications, we give tight security bounds for the nonce-randomization mechanism adopted in the record protocol of TLS 1.3 as a mitigation of large-scale multi-user attacks. We provide tight security bounds that yield the first validation of this method. In particular, we solve the main open question of Bellare and Tackmann (CRYPTO '16), who only considered restricted attackers which do not attempt to violate integrity, and only gave non-tight bounds.

[1] Department of Computer Science, Florida State University, Tallahassee, Florida 32304, USA. Email: tvhoang@cs.fsu.edu. URL: http://www.cs.fsu.edu/~tvhoang/. Supported in part by NSF grants CICI-1738912 and CRII-1755539.

[2] Department of Computer Science, University of California Santa Barbara, Santa Barbara, California 93106, USA. Email: tessaro@cs.ucsb.edu. URL: http://www.cs.ucsb.edu/~tessaro/. Supported in part by NSF grants CNS-1553758 (CAREER), CNS-1423566, CNS-1719146, CNS-1528178, and IIS-1528041, and by a Sloan Research Fellowship.

[3] Department of Computer Science, University of California Santa Barbara, Santa Barbara, California 93106, USA. Email: aish@cs.ucsb.edu. Supported in part by the Defense Advanced Research Projects Agency (DARPA) and Army Research Office (ARO) under Contract No. W911NF-15-C-0236, and a subcontract No. 2017-002 through Galois.

Contents

1 Introduction **3**
1.1 Mu Security and Nonce Randomization 3
1.2 Our Results 4

2 Preliminaries **6**
2.1 Authenticated Encryption 7
2.2 The H-coefficient Technique 7

3 Multi-security of GCM **8**
3.1 The CAU Scheme 8
3.2 Security of CAU 9

4 RGCM and the XOR transform **17**

5 The Concatenation transform **19**

6 Security with Public Salting **22**

References **22**

1 Introduction

Authenticated Encryption (AE) is symmetric encryption that protects both *confidentiality* and *integrity*, and is arguably the most widely used primitive in applied cryptography–in particular, it protects data transmission in most in-use secure communication protocols like TLS, IPSec, SSH, WPA-2, SRTP, etc.

We consider an emerging concern in the Internet-wide adoption of AE, namely *large-scale adversaries*, like state actors, which can launch coordinated attacks against a large number u of sessions (e.g., $u = 2^{20}$ or 2^{30}), which all use the same cryptographic algorithms with independent keys. The setting of *multi-user (mu) security*, introduced by Biham [6] in symmetric cryptanalysis and by Bellare, Boldyreva, and Micali [3] in public-key cryptography, deals with such attacks. More precisely, it considers attackers who succeed as long as they can compromise *at least one* out of u sessions (referred to as "users"). As made evident in a series of recent works [23, 18, 2, 10, 11, 13, 7], estimating *how* security degrades as u grows is a challenging technical problem that affects the real world: Indeed, the goal of mitigating mu attacks explicitly influenced design choices in the record protocol of TLS 1.3 [20, Appendix E.2], which have however been adopted without full validation, as we explain below.

Overview. This paper revisits AE, and more specifically the widely adopted Galois Counter-Mode (GCM) scheme [17], in the mu setting. We prove new tight bounds for GCM which improve upon existing ones [5, 13] by considering a fine-grained setting that assumes both (1) a bound d on the number of users re-using any particular nonce, and (2) a bound B on the amount of data encrypted by each user.

This allows us to analyze some deployment practices for GCM that have a positive impact on mu security. On the one hand, we show that frequent re-keying improves AE mu security. On the other hand, we show how mu security is affected by policies adopted to choose *nonces*, e.g., combining (secret) pseudorandom values and counters. We refer to such techniques as *nonce randomization*. We show, with precise tight bounds, that nonce randomization increases the mu security of AE, and apply this insight to GCM-based AE, confirming an intuition initially put forward in the design of TLS 1.3.[1] We also show that already in-place nonce selection strategies in TLS 1.2 effectively improve mu security.

Prior to this work, Bellare and Tackmann (BT) [5] were the only ones to rigorously study the specific GCM-based approach adopted by TLS 1.3. As we discuss below, their analysis is non-tight and only considers adversaries attempting to break confidentiality. Here, we complete the picture with tight bounds and full AE security, and resolve their main open question.

1.1 Mu Security and Nonce Randomization

Here, we follow the conventional AEAD interface which allows us to (deterministically) encrypt a *plaintext* M, with a *nonce* N and *associated data* A as a *ciphertext* $\mathcal{E}_K(N, A, M)$. Security is meant to hold as long as no two pairs (M, A) are encrypted with the same N. (We will not discuss nonce-misuse resistance [21] in this paper.)

The mu security of AE. One question is what is the best we can expect from an AE scheme in terms of its mu security. To this end, BT adapt a well-known generic key-recovery attack by Biham [6] to AEAD. First, fix N^*, A^* and M^*, and obtain their encryption with respect to u

[1] As we detail below, such approaches were used before, but never was mu security suggested as an explicit motivation for nonce randomization before TLS 1.3.

different users, which yields ciphertexts

$$C_i = \mathcal{E}_{K_i}(N^*, A^*, M^*) \ , \quad i = 1, \ldots, u \ ,$$

where K_i is the key of the i-th user. The attacker's goal is to recover *at least one* of the K_i's. To do so, it makes p key-guesses K (e.g., random ones), and for each guess, computes $C = \mathcal{E}_K(N^*, A^*, M^*)$. If $C = C_i$ for some i, then $K = K_i$. It is not hard to see that the probability that this attack succeeds is roughly $u \cdot p/2^k$, where k is the key-length (e.g., $k = 128$ in GCM based on 128-bit AES). Therefore, the effort to succeed is only $p \approx 2^{k-\log(u)}$.

Nonce randomization. The above generic attack is not always as threatening in practice, as in-place policies for choosing nonces limit its impact. Typically, an AE scheme would be invoked with a nonce N which combines a (usually public) part like a counter, to be sent along with the ciphertext, and an *implicit part*, often secret and already known by the endpoints (this could be generated as part of a prior handshake). McGrew [15] gives an overview of such methods in an Internet Draft, and we refer to them as "nonce randomization" techniques.

For example, RFC 5288 [22]–which describes the GCM ciphersuites for TLS 1.2–mandates nonces whose implicit part is a session-dependent (pseudo)random salt generated as part of the handshake. Thus, with u users, each nonce is re-used by (on average) $u/2^{32}$ users, and in the above attack, each ciphertext C can thus be checked against at most $u/2^{32}$ ciphertexts (rather than u), reducing the success probability to roughly $u \cdot p/2^{160}$ for a 128-bit key.

An even more effective approach (at least with respect to preventing the above attack) are so-called "unpredictable nonces", and this is the approach taken by TLS 1.3 [20] and previously used within the SRTP protocol [1]. Here, a secret random offset mask J is chosen, and then, whenever we need to encrypt a message with nonce N, it is encrypted with nonce $N \oplus J$ instead. BT [5] analyzed this method in the specific case of GCM used by TLS 1.3, casting it as a standalone AE scheme called RGCM. They fall short of a full analysis, however, giving merely non-tight bounds that confirm better-than-average passive (i.e., IND-CPA) security. We stress that integrity is even more fundamental in the mu setting – indeed, while a single session can abort after a failed verification attempt, mu attackers can spread forgery attempts across different users, making uncoordinated attack detection much harder.

We note that with the exception of the standardization of TLS 1.3 [20, Appendix E.2], the treatment of mu attacks has not been explicitly mentioned as a motivation, even though some of the published motivating work [16] considered key-collision attacks arising from two users having the same key, which are of course special cases of mu attacks.

1.2 Our Results

In this paper, we complete the picture for the security of GCM in the multi-user setting with tight and more refined bounds. These will allow us to give precise bounds when nonce randomization policies are applied to GCM.

The d-bounded model and RGCM. Here, we consider the mu version of AE security from [5], which requires indistinguishability from random ciphertexts in presence of a verification oracle. In addition, we adopt the model by Bose, Hoang, and Tessaro (BHT) [7], which we refer to as the *d-bounded model*: it postulates that each nonce can be re-used by at most d users for encryption. More formally, the attacker gets to ask encryption queries of the form (i, N, A, M), which produce an encryption $\mathcal{E}_{K_i}(N, A, M)$ under the key K_i of the i-th user. Here, the constraints are that (1) for every i, no two queries with the same N are asked, and (2) for every N, there are at most d i's for which a query (i, N, A, M) is asked for *some* M, A. However, its queries to the verification

oracle are *unrestricted*, and take the form (i, N, A, C) and return true if and only if C is a valid ciphertext under K_i for N and A.

The goal is to give security bounds which are parameterized by d. Jumping ahead, this model allows us to see a nonce-randomization policy as part of the adversary which ensures a certain d when picking nonces.

Although we rely on the model proposed by BHT [7], we emphasize that our security goal is different - we consider only nonce-respecting and do not consider misuse-resistance. While there is some conceptual overlap due to the settings, apart from relying on some balls-and-bins lemma from BHT our proofs proceed differently.

GCM IN THE d-BOUNDED MODEL. Our main technical result is an analysis of CAU – a generalization of GCM presented by BT – in the d-bounded model, assuming the underlying blockcipher is ideal. We show that for every adversary making q encryption/verification queries, p ideal-cipher queries, and encrypting/verifying overall σ blocks of data, the advantage of breaking CAU's AE security is of the order[2]

$$\frac{d(p+q)+n(p+q+\sigma)}{2^k}+\frac{\sigma B}{2^n}$$

where k and n are the blockcipher key and block length, respectively, and B is a bound on the number of blocks encrypted per user. We stress that our bound does *not* depend on the number of users u, which can depend on adaptive choice of the adversary, and can be as high as q.

In comparison, BT [5] show a bound for the case where d is unbounded (i.e., $d = u$) of order[3]

$$\frac{u(u+p)}{2^k}+\frac{uB^2}{2^n}$$

For the case where B is unbounded (i.e., $\sigma = B$), the second term in the bound was later improved by [13] to $\frac{\sigma^2}{2^n}$, which is consistent with the (tight) single-user bound [12].

WHY THIS BOUND MATTERS. Our bound is interesting for its parameterization: It shows that when d is small, the security increases substantially, and this will enable an analysis of nonce randomization techniques. Even for the $u = d$ case, the parameterization with B shows important insights: First off, if we have u users, all transmitting roughly the same amount of data $B := \sigma/u$, the term becomes $\sigma^2/(u2^n)$–much better than $\sigma^2/2^n$ as u grows. Moreover, users normally re-key, ensuring no session transmits too much data, and thus generally B can be fixed independently of σ–moreover, the smaller (i.e., we re-key more often), the better. If for instance, $B = 2^{33}$, $n = 128$, then this allows each user/session to encrypt up to 2^{40} bits $= 2^{37}$ bytes $= 137.4$ GB, yet the term becomes $\sigma/2^{95}$.

NONCE RANDOMIZATION METHODS: GENERIC TRANSFORMS. We cast both nonce-randomization schemes discussed above as generic transformations building an AE scheme with longer keys from one with shorter ones. The first one implements unpredictable nonces, as in TLS 1.3 and SRTP, and we refer to it as XN. If the underlying scheme uses a key K of length k and nonce length r, the resulting scheme uses a key $K \,\|\, J$ of length $k+r$. If we denote by $\mathcal{E}$ the encryption of the given AE scheme, the encryption $\mathcal{E}^*$ is such that

$$\mathcal{E}^*(K \,\|\, J, N, A, M) = \mathcal{E}(K, N \oplus J, A, M)\ .$$

Note that XN still has r-bit nonces. For the specific case where $\mathcal{E}$ comes from GCM, BT refer to this construction as RGCM.

[2] We omit lower-order terms, and small constant factors.

[3] An earlier version of their work had stated a bound where the second term is of the order $u\sigma^2/2^n$.

An alternative construction, which reflects what is adopted in TLS 1.2, for example, is what we refer to as CN. Here, for a parameter $t < r$, the key is $(k+t)$-bits long (and has form $K \| J$), and the resulting nonce length is $r - t$. Then, we let

$$\mathcal{E}^*(K \| J, N, A, M) = \mathcal{E}(K, J \| N, A, M) \ .$$

We are not aware of this construction having been studied explicitly.

We prove generic results that relate the security of the XN and CN constructions to the d-bounded security of the underlying AE. The intuition why this is possible is quite clear: For XN, for example, every time an encryption query (i, N, A, M) is made, this reflects itself to encrypting with $\mathcal{E}$ using nonce $N' = N \oplus J_i$, where J_i is user i's J-component of the key. This ensures that no N' is re-used across too many users–a fact that relies on the J_i's being secret, and a balls-into-bins argument. In fact, because the J_i's are secret, it turns out that a bound in a weaker version of the model, where nonces are re-used for at most d users *also in verification queries*, is enough. We briefly discuss below applications that require the stronger model.

Bounds for RGCM: Old and new. One main consequence when instantiating XN with GCM and our analysis in the d-bounded model is that we can provide a complete and tight analysis of RGCM which substantially improves upon [5]. Their analysis only shows RGCM is no less secure than GCM, and give an improved bound which only considers attackers not making verification queries. BT's bound is of the order (parameters as above, and r is the nonce length)

$$\frac{u^2 + 40p}{2^k} + \frac{u\sigma^2}{2^n} + \frac{up\sigma}{2^{k+n}} + \frac{upq}{2^{k+r}} \ .$$

For example, if $k = 128$, then u can be at most 2^{64}.

We show a much stronger bound, crucially also taking into account verification queries. As long as $q \leq 2^{r(1-\epsilon)}$ for some small constant $\epsilon > 0$, our bound is of the order

$$\frac{n(p+\sigma)}{2^k} + \frac{\sigma B}{2^n} + \frac{\sigma^2 + pq}{2^{k+n}} \ . \tag{1}$$

Public Salting. The XN and CN constructions reflect practical usage, and keep the value J secret. However, as we discuss briefly below in the paper, our result in the d-bounded model enables us to give a much stronger result which does not require J to be secret at all, as long as nonces are not chosen arbitrarily by the adversary.

For example, if each user encrypts using nonces $J_i \oplus 0$, $J_i \oplus 1$, $J_i \oplus 2$, ... *and makes nonces public* (thus J_i is known to the adversary for each i), we can think of this as a particular adversary attacking AE security of GCM and using such nonces. Then, our bound on GCM implies similar security as that of RGCM *without* making J secret (thus saving on key length). In particular, our bound holds *even* if the attacker attempts verification queries with arbitrary repeating nonces.

The ideal-cipher model. This paper relies on ideal models, and in particular, the ideal-cipher model, for its analyses. This is a common trait of most analyses in the mu regime–one issue is that we are particularly concerned here with how local computation (approximated by the p ideal cipher queries) affects security, and classical assumptions on blockciphers (PRP security) are not helpful in making this type of statements.

2 Preliminaries

Notation. Let ε denote the empty string. For a finite set S, we let $x \leftarrow\!\!\$\ S$ denote the uniform sampling from S and assigning the value to x. Let $|x|$ denote the length of the string x, and

for $1 \le i < j \le |x|$, let $x[i,j]$ (and also $x[i:j]$) denote the substring from the ith bit to the jth bit (inclusive) of x. If A is an algorithm, we let $y \leftarrow A(x_1, \ldots; r)$ denote running A with randomness r on inputs $x_1, \ldots$ and assigning the output to y. We let $y \leftarrow\!\!\$\ A(x_1, \ldots)$ be the result of picking r at random and letting $y \leftarrow A(x_1, \ldots; r)$. In the context that we use a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, the block length of a string x, denoted $|x|_n$, is $\max\{1, \lceil |x|/n \rceil\}$.

2.1 Authenticated Encryption

An AE scheme Π is a triple $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ with message space $\mathcal{M}$ and nonce space $\mathcal{N}$. The encryption scheme $\mathcal{E}$ takes as input a key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, associated data $A \in \{0,1\}^*$, a message $M \in \mathcal{M}$, and deterministically returns a ciphertext $C \leftarrow \mathcal{E}_K(N, A, M)$. The decryption scheme $\mathcal{D}$ takes as input a key K, a nonce N, associated data A, a ciphertext C, and returns either a message $M \in \mathcal{M}$, or the error symbol $\perp$. We require that, if $C \leftarrow \mathcal{E}_K(N, A, M)$ then $M \leftarrow \mathcal{D}_K(N, A, C)$, for correctness.

MU SECURITY OF AE. Let $\Pi[E] = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AE scheme on top of an ideal cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. Let $\mathcal{A}$ be an adversary. Define

$$\mathbf{Adv}^{\text{mu-ae}}_{\Pi[E]}(\mathcal{A}) = \Pr[\text{Real}^{\mathcal{A}}_{\Pi[E]} \Rightarrow 1] - \Pr[\text{Rand}^{\mathcal{A}}_{\Pi[E]} \Rightarrow 1],$$

where games $\text{Real}^{\mathcal{A}}_{\Pi[E]}$ and $\text{Rand}^{\mathcal{A}}_{\Pi[E]}$ are defined in Fig. 1. Under each game, the adversary $\mathcal{A}$ is given access to three oracles ENC, VF, and PRIM. For encryption queries $\text{ENC}(i, N, A, M)$, we require that the adversary must not repeat the pairs (i, N). The adversary can repeat nonces in the verification queries $\text{VF}(i, N, A, C)$, but to avoid trivial wins, once the adversary queries $\text{ENC}(i, N, A, M)$ to receive C, it is prohibited from querying $\text{VF}(i, N, A, C)$.

We say that an adversary is *d-repeating* if it never uses the same nonce for more than d users in encryption queries. We stress that a d-repeating adversary can still repeat nonces across different users in verification queries as often as it wishes. The single-user setting corresponds to $d = 1$.

We say that an adversary is *strongly d-repeating* if for both encryption and verification queries, it never uses the same nonce for more than d users. While this restriction on verification queries seems impossible to enforce, we shall see later that the mu-security of RGCM against a generic adversary can be reduced to the mu-security of GCM against a strongly d-repeating adversary, for some small constant d. Similarly, the mu-security of the GCM scheme used in TLS 1.2 can be reduced to the mu-security of GCM against a strongly d-repeating adversary for an appropriate choice of d.

When we consider security against (strongly) d-repeating adversaries, we informally refer to this as the d-bounded model.

2.2 The H-coefficient Technique

SYSTEMS AND TRANSCRIPTS. Following the notation from [10] (which was in turn inspired by Maurer's framework [14]), it is convenient to consider interactions of a distinguisher $\mathcal{A}$ with an abstract system $\mathbf{S}$ which answers $\mathcal{A}$'s queries. The resulting interaction then generates a transcript $\tau = ((X_1, Y_1), \ldots, (X_q, Y_q))$ of query-answer pairs. It is known that $\mathbf{S}$ is entirely described by the probabilities $\mathsf{p}_{\mathbf{S}}(\tau)$ that correspond to the system $\mathbf{S}$ responding with answers as indicated by τ when the queries in τ are made.

We will generally describe systems informally, or more formally in terms of a set of oracles they provide, and only use the fact that they define corresponding probabilities $\mathsf{p}_{\mathbf{S}}(\tau)$ without explicitly giving these probabilities. We say that a transcript τ is valid for system $\mathbf{S}$ if $\mathsf{p}_{\mathbf{S}}(\tau) > 0$.

Game $\mathrm{Real}^{\mathcal{A}}_{\Pi[E]}$	**Game** $\mathrm{Rand}^{\mathcal{A}}_{\Pi[E]}$
$K_1, K_2, \cdots \leftarrow\!\!\$\ \mathcal{K}$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Enc},\textsc{Vf},\textsc{Prim}}$; **return** b'	$b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Enc},\textsc{Vf},\textsc{Prim}}$; **return** $(b' = 1)$
procedure $\textsc{Enc}(i, N, A, M)$ **return** $\mathcal{E}_{K_i}(N, A, M)$	**procedure** $\textsc{Enc}(i, N, A, M)$ $C \leftarrow\!\!\$\ \{0,1\}^{\lvert M\rvert+\lambda}$; **return** C
procedure $\textsc{Vf}(i, N, A, C)$ $V \leftarrow \mathcal{D}_{K_i}(N, A, C)$; **return** $(V \neq \perp)$	**procedure** $\textsc{Vf}(i, N, A, C)$ **return** false
procedure $\textsc{Prim}(J, X)$ **if** $X = (+, x)$ **then return** $E_J(x)$ **if** $X = (-, y)$ **then return** $E_J^{-1}(y)$	**procedure** $\textsc{Prim}(J, X)$ **if** $X = (+, x)$ **then return** $E_J(x)$ **if** $X = (-, y)$ **then return** $E_J^{-1}(y)$

Figure 1: **Games defining the multi-user security of an AE scheme Π.** This scheme is based on a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. We assume that under the scheme Π, the ciphertext is always λ-bit longer than the message.

The H-coefficient technique. We now describe the H-coefficient technique of Patarin [19, 8]. Generically, it considers a deterministic distinguisher $\mathcal{A}$ that tries to distinguish a "real" system $\mathbf{S}_{\text{real}}$ from an "ideal" system $\mathbf{S}_{\text{ideal}}$. The adversary's interactions with those systems define transcripts $\mathcal{T}_{\text{real}}$ and $\mathcal{T}_{\text{ideal}}$, respectively, and a bound on the distinguishing advantage of $\mathcal{A}$ is given by the statistical distance $\mathsf{SD}(\mathcal{T}_{\text{real}}, \mathcal{T}_{\text{ideal}})$.

Lemma 2.1 [19, 8] *Suppose we can partition the set of valid transcripts for the ideal system into good and bad ones. Further, suppose that there exists $\epsilon \geq 0$ such that $1 - \frac{\mathsf{ps}_{\text{real}}(\tau)}{\mathsf{ps}_{\text{ideal}}(\tau)} \leq \epsilon$ for every good transcript τ. Then,*

$$\mathsf{SD}(\mathcal{T}_{\text{ideal}}, \mathcal{T}_{\text{real}}) \leq \epsilon + \Pr[\mathcal{T}_{\text{ideal}} \textit{ is bad}] \ .$$

3 Multi-security of GCM

In this section, we consider the mu security of authenticated encryption (AE) construction CAU [5], which includes GCM as a special case. CAU loosely follows the encrypt-then-MAC paradigm, where the encryption scheme is the CTR mode on a blockcipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, and the MAC is the Carter-Wegman construction via an almost XOR-universal (AXU) hash. We begin by recalling the definition of AXU hash functions.

AXU hash. Recall that for a string x, the *block length* $|x|_n$ of x is defined as $\max\{1, \lceil |x|/n \rceil\}$. We call $H : \mathcal{K} \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ a *c-AXU hash* if for any $(M, A) \neq (M', A')$ in $\{0,1\}^* \times \{0,1\}^*$, and any $z \in \{0,1\}^n$,

$$\Pr_{K \leftarrow\!\!\$\ \mathcal{K}}[H_K(M, A) \oplus H_K(M', A') = z] \leq \frac{c \cdot \max\{|M|_n + |A|_n, |M'|_n + |A'|_n\}}{2^n} \ .$$

3.1 The CAU Scheme

Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Let $H : \{0,1\}^n \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^n$ be a c-AXU hash. The nonce space $\mathcal{N}$ of CAU is $\{0,1\}^r$, for $r < n$, and its key space is $\{0,1\}^k$. For a string $Z \in \mathcal{N}$, we write $\mathrm{pad}(Z)$ to refer to the string $Z0^{n-r-1}1$. The message space is the set of binary strings whose block length is strictly less than $2^{n-r} - 1$.

procedure $\mathsf{CAU.Enc}(K, N, A, M)$	**procedure** $\mathsf{CAU.Dec}(K, N, A, T \,\|\, C)$
// $0 \le \|M_\ell\| < n$, and $\|M_i\| = n$ otherwise	$L \leftarrow E_K(0^n)$; $Y \leftarrow \mathrm{pad}(N)$
$Y \leftarrow \mathrm{pad}(N)$; $M_1 \cdots M_\ell \leftarrow M$	// $0 \le \|C_\ell\| < n$, and $\|C_i\| = n$ otherwise
// Encrypt with CTR mode and IV $Y+1$	$C_1 \cdots C_\ell \leftarrow C$; $T' \leftarrow H_L(A, C) \oplus E_K(Y)$
for $i = 1$ **to** $\ell - 1$ **do** $C_i \leftarrow M_i \oplus E_K(Y + i)$	**if** $T \neq T'$ **then return** $\perp$
$V \leftarrow E_K(Y + \ell)$; $C_\ell \leftarrow M_\ell \oplus V[1 : \|M_\ell\|]$	// Decrypt with CTR mode and IV $Y+1$
$C \leftarrow C_1 \cdots C_\ell$	**for** $i = 1$ **to** $\ell - 1$ **do** $M_i \leftarrow C_i \oplus E_K(Y + i)$
// Use Carter-Wegman with H	$V \leftarrow E_K(Y + \ell)$; $M_\ell \leftarrow C_\ell \oplus V[1 : \|C_\ell\|]$
$L \leftarrow E_K(0^n)$; $T \leftarrow H_L(A, C) \oplus E_K(Y)$	$M \leftarrow M_1 \cdots M_\ell$
return $T \,\|\, C$	**return** M

Figure 2: **The encryption (top) and decryption (bottom) of the authenticated encryption scheme CAU.** The scheme is based on a blockcipher E and an AXU hash H.

On input (K, N, A, M), the encryption scheme first encrypts M via the CTR mode of E_K with IV $\mathrm{pad}(N) + 1$, to get a ciphertext core C (that does not include the IV). It then computes a hash key $L \leftarrow E_K(0^n)$, produces a tag $T \leftarrow H_L(A, C) \oplus E_K(\mathrm{pad}(N))$ and then outputs $T \,\|\, C$ as the ciphertext. On input $(K, N, A, T \,\|\, C)$, the decryption scheme first computes the hash key $L \leftarrow E_K(0^n)$. Next, if $T \neq H_L(A, C) \oplus E_K(\mathrm{pad}(N))$, it outputs $\perp$. Otherwise, it uses the decryption of CTR on E_K with IV $\mathrm{pad}(N) + 1$ to decrypt C, and outputs the corresponding message M.

See Fig. 2 for the code of CAU. For GCM, the blockcipher E is instantiated by AES, and thus $n = 128$ and $k \in \{128, 256\}$. The nonce length r is 96 bits. The hash H is instantiated by the polynomial-based hash function GHASH, and thus one can pick $c = 1.5$. To see why, recall that in the original GCM document [17], McGrew and Viega showed that for any two distinct pairs (M, A) and (M', A'), and for any $z \in \{0,1\}^n$,

$$\begin{aligned}
\Pr_{K \xleftarrow{\$} \{0,1\}^n}[\mathsf{GHASH}_K(M, A) \oplus \mathsf{GHASH}_K(M', A') = z] &\le \frac{\lceil (1 + \max\{|M| + |A|, |M'| + |A'|\})/n \rceil}{2^n} \\
&\le \frac{1 + \max\{|M|_n + |A|_n, |M'|_n + |A'|_n\}}{2^n} \\
&\le \frac{1.5 \cdot \max\{|M|_n + |A|_n, |M'|_n + |A'|_n\}}{2^n} \ .
\end{aligned}$$

3.2 Security of CAU

Theorem 3.1 below gives a tight mu-security bound of CAU against a d-repeating adversary. We stress that the bound σ in the theorem takes into account the block length of both the message and the associated data of an encryption/verification query.

Theorem 3.1 (Mu-security of CAU/GCM) *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher, with $k \ge n \ge 128$. Let H be a c-AXU hash function. Let $\mathcal{A}$ be a d-repeating adversary attacking $\mathsf{CAU}[H, E]$ using at most $p \le 2^{n-2}$ ideal-cipher queries, q encryption/verification queries of total block length at most σ, and the total number of blocks in encryption queries of each user is at most B. Then*

$$\mathbf{Adv}^{\text{mu-ae}}_{\mathsf{CAU}[H,E]}(\mathcal{A}) \le \frac{d(p + 2q) + n(q + \sigma + p)}{2^k} + \frac{\sigma(2B + 3) + nc(\sigma + qB)}{2^n} + \frac{2q + 1}{2^{2n}} + \frac{\sigma^2 + 2pq}{2^{k+n}} \ .$$

DISCUSSION. It is important to note that the bound in Theorem 3.1 does not depend explicitly on the number of users, which can become as large as q. The only dependence on users is through the parameter d, which can be (but generally is not) as large as q. The bound in Theorem 3.1 contains three important factors, $\frac{pd}{2^k}$, $\frac{n\sigma}{2^k}$, and $\frac{\sigma B}{2^n}$ that correspond to actual attacks. We discuss them here, which will be instrumental for understanding the proof below.

First, for the term $\frac{pd}{2^k}$, consider the following attack. The adversary picks an arbitrary nonce N, a long enough message M, and makes d encryption queries $(1, N, A, M), \ldots (d, N, A, M)$, where A is the empty string, to get answers $C_1, \ldots, C_d$ respectively. (Recall that the adversary is d-repeating, so it cannot repeat a nonce N in encryption queries for more than d users.) By picking p distinct candidate keys $K_1, \ldots, K_p$ and comparing C_i with $\mathsf{CAU.Enc}(K_j, N, A, M)$ for all $1 \le i \le d$ and $1 \le j \le p$, the adversary can recover one key with probability about $\frac{pd}{2^k}$.

For the term $\frac{n\sigma}{2^k}$, consider the following attack. The adversary first picks an arbitrary nonce N and p distinct candidate keys $K_1, \ldots, K_p$, and makes $2p$ ideal-cipher queries $(K_i, (\mathrm{pad}(N), +))$, $(K_i, (0^n, +))$. The goal of the adversary is to make q verification queries $(j, N, A, T \,\|\, C)$, for $j = 1, \ldots, q$ for associated data A and ciphertext $T \,\|\, C$ of ℓ blocks total that it will determine later. To maximize its chance of winning, the adversary will iterate through all possible tuples $(A^*, T^* \,\|\, C^*)$ of ℓ blocks total and compute $\mathit{count}(A^*, T^* \,\|\, C^*)$, the number of ideal-cipher queries $(K_i, (\mathrm{pad}(N), +))$ whose answer is $H_{L_i}(A^*, C^*) \oplus T^*$, where $L_i \leftarrow E_{K_i}(0^n)$. It then picks $(A, T \,\|\, C)$ to maximize $\mathit{count}(A, T \,\|\, C)$. Then the adversary wins with advantage about $\mathbf{E}[\mathit{count}] \cdot q/2^k$. The proof of Theorem 3.1 shows that $\mathbf{E}[\mathit{count}] \le n\ell = \frac{n\sigma}{q}$ with very high probability, and thus the advantage of the adversary is at most $n\sigma/2^k$.

For the term $\sigma B/2^n$, consider the following distinguishing attack. The adversary will target u users, where $u = \lfloor \sigma/B \rfloor$. Let M be an arbitrary message of B blocks. Pick an arbitrary nonce N, and let A be the empty string. The adversary then calls $\textsc{Enc}(i, N, A, M)$ to receive $T_i \,\|\, C_i$, for every $i = 1, \ldots, u$. If some ciphertext core C_i contains two identical blocks then the adversary outputs 0, otherwise it outputs 1. By using appropriate data structure, one can implement this attack using $O(B)$ space and $O(\sigma)$ time. To analyze the adversary's advantage, we need the following technical Lemma 3.2 and Lemma 3.3. The first result states a well-known lower bound for the birthday bound; see, for example, [9, Appendix A] for a proof. The second result is a useful inequality whose proof can be found in [4].

Lemma 3.2 (Lower bound for birthday bound) *Let $N > 0$ be an integer. Suppose that we throw $1 \le q \le \sqrt{2N}$ balls into N bins uniformly at random. Then the chance that there are two balls that fall into the same bin is at least $\frac{q(q-1)}{4N}$.*

Lemma 3.3 [4] *Let $p \ge 1$ be an integer and $a \ge 0$ a real number. Assume $ap \le 1$. Then $(1-a)^p \le 1 - ap/2$.*

Back to the analysis, in the ideal world, each C_i is a truly random $B-$block string, and thus from Lemma 3.2, the chance that it contains two identical blocks is at least $\frac{B(B-1)}{4 \cdot 2^n}$. Hence in the ideal world, the chance that the adversary outputs 1 is at most

$$\left(1 - \frac{B(B-1)}{2^{n+2}}\right)^u \le 1 - \frac{B(B-1)u}{2^{n+3}} \approx 1 - \frac{\sigma B}{2^{n+3}}$$

where the inequality is due to Lemma 3.3. In contrast, in the real world, the adversary will always output 1. Hence the adversary wins with advantage about $\frac{\sigma B}{2^{n+3}}$.

The term $\sigma B/2^n$ also deserves some further discussion. It conveys an important message, and namely that as B becomes smaller, the term becomes closer to $\sigma/2^n$. A small B could be enforced,

for example, by ensuring that a session in a protocol only transfers a bounded amount of data before a re-keying operation is issued. In other words, *re-keying only improves multi-user security.* This is important, when compared to the single-user security analysis, which gives a bound of the order $\sigma^2/2^n$. (Of course, if we have one single user, then $B = \sigma$.)

PROOF IDEAS. The proof examines several cases but here we discuss two illustrative ones that correspond to the two attacks above. First, consider the event that the adversary can query $\textsc{Prim}(K, (x, +))$ and query $\textsc{Enc}(i, N, A, M)$ such that $K = K_i$ and $x \in \{\mathrm{pad}(N), \ldots, \mathrm{pad}(N) + \ell\}$, where $\ell = |M|_n$. This case includes the first attack above. Note that for any query $\textsc{Prim}(K, (x, +))$, since the adversary is d-repeating, there are at most d queries $\textsc{Enc}(i, N, A, M)$ such that $x \in \{\mathrm{pad}(N), \ldots, \mathrm{pad}(N) + \ell\}$, where $\ell = |M|_n$, and the chance that some of these d latter queries satisfies $K_i = K$ is at most $d/2^k$. Hence, this case happens with probability at most $dp/2^k$.

On the other hand, in GCM, every user i derives the hash key L_i via $E_{K_i}(0^n)$. Thus by querying $\textsc{Prim}(K, (0^n, +))$ for p keys K, the adversary may accidentally obtain some blockcipher key K_i and its associated hash key L_i with probability about $pu/2^k$, where u is the number of users, and in the worst case, u can be as large as q. This creates a problem in using the AXU-property of the hash function H, since we can no longer treat the hash keys as independent of the queries. This is exactly the issue in the second attack above, where the adversary *adaptively* picks verification queries *after* seeing the hash keys.

To make the analysis simpler, at the beginning, we will even grant the adversary all pairs $(K, E_K(0^n))$ for every $K \in \{0,1\}^k$, and this can only help the adversary. However, now when we pick $K_i \leftarrow\!\!\$\ \{0,1\}^k$, the corresponding key $L_i \leftarrow E_{K_i}(0^n)$ is no longer uniformly random. To understand the distribution of the key L_i, we need the following balls-into-bins result of Bose, Hoang, and Tessaro [7].

Lemma 3.4 ([7]) *Fix integers $n \geq 128, \ell \geq 2$, and $a \geq 1$. Suppose that we throw $q \leq a \cdot 2^n$ balls into 2^n bins. The throws may be inter-dependent, but for each i-th throw, conditioning on the result of the prior throws, the conditional probability that the i-th ball falls into any particular bin is at most 2^{1-n}. Then the chance that the heaviest bin contains $\lceil a\ell n/2 \rceil$ or more balls is at most $2^{-(3\ell+2)n}$ if $a \geq 2$, and is at most $2^{-1.5\ell n}$ if $a = 1$.*

Now, view each granted pair $(K, E_K(0^n))$ as throwing a ball into bin $E_K(0^n)$. Thus we throw 2^k balls uniformly at random into 2^n bins. Thus using Lemma 3.4 with $a = 2^{k-n}$ and $\ell = 2$, with probability at least $1 - 2^{-3n}$, each bin contains at most $n \cdot 2^{k-n}$ balls. Thus for any $L \in \{0,1\}^n$, there are at most $n \cdot 2^{k-n}$ keys K such that $E_K(0^n) = L$. In other words, when we pick $K_i \leftarrow\!\!\$\ \{0,1\}^k$, the conditional min-entropy of L_i is at least $-\lg(n \cdot 2^{k-n}/2^k) = n - \lg(n)$.

Going back to the dependency issue of the hash keys and its inputs, a particularly tough case is to analyze the probability that the adversary can first make a query $\textsc{Prim}(K, (\mathrm{pad}(N), +))$ and obtain answer y and then query $\textsc{Vf}(i, N, A, T \,\|\, C)$, and it happens that $K = K_i$ and $H_{L_i}(A, C) \oplus T = y$, where K_i is the blockcipher key of user i, and $L_i \leftarrow E_K(0^n)$. This case includes the second attack above. To deal with this case, we employ a trick from [7]. Specifically, consider a *fixed* tuple (N^*, A^*, C^*) and let $\ell = |A^*|_n + |C^*|_n$. View each query $\textsc{Prim}(K, (\mathrm{pad}(N^*), +))$ of answer y as throwing a ball into bin $H_L(A^*, C^*) \oplus y$, where $L \leftarrow E_K(0^n)$. By Lemma 3.4 above, with probability at least $1 - 2^{-(3\ell+2)n}$, each bin contains at most ℓn balls. Thus for an adaptive T, the number $count^*$ of matching ideal-cipher queries is at most $\ell n = (|A^*|_n + |C^*|_n)n$, with probability at least $1 - 2^{-(3\ell+2)n}$. Then for any adaptive choice $(N, A, T \,\|\, C)$, the chance that there are at most $(|A|_n + |C|_n) \cdot n$ matching ideal-cipher queries is at least

$$1 - \sum_{\ell=2}^{\infty} \sum_{(i^*,N^*,A^*,C^*):|A^*|_n+|C^*|_n=\ell} 2^{-(3\ell+2)n} \geq 1 - \sum_{\ell=2}^{\infty} 2^{2n+2\ell n} \cdot 2^{-(3\ell+2)n} \geq 1 - \frac{2}{2^{2n}} \ .$$

Hence, the chance that the case above happens is at most $n\sigma/2^k + 2q/2^{2n}$.

Proof of Theorem 3.1 : Without loss of generality, assume that $\sigma \leq 2^n/n$; otherwise the bound is moot. As mentioned earlier, at the beginning, we will give the adversary $(K, E_K(0^n))$ for every $K \in \{0,1\}^k$, and this can only help the adversary. Because we consider computationally unbounded adversaries, without loss of generality, assume that $\mathcal{A}$ is deterministic, and never repeats a prior query. Assume that if the adversary queries $\textsc{Prim}(K,(x,+))$ to get an answer y then it will not subsequently query $\textsc{Prim}(K,(y,-))$, since the answer would be x. Likewise, assume that if the adversary queries $\textsc{Prim}(K,(y,-))$ to get an answer x then it will not later query $\textsc{Prim}(K,(x,+)$. Our proof is based on the H-coefficient technique.

Defining bad transcripts. In the real world, after the adversary finishes querying, we will give it the blockcipher keys K_i of all users i. In the ideal world, we instead give the adversary truly random strings $K_i \xleftarrow{\$} \{0,1\}^k$, independent of the transcript. Thus the transcript implicitly includes the hash keys $L_i \leftarrow E_{K_i}(0^n)$. This key revealing only helps the adversary. Thus a transcript consists of the revealed keys, the granted ideal-cipher queries, and the following information:

- **Ideal-cipher queries:** For each query $\textsc{Prim}(K,(x,+))$ with answer y, we associate it with an entry $(\mathsf{prim}, K, x, y, +)$. Likewise, for each query $\textsc{Prim}(K,(y,-))$ with answer x, we associate it with an entry $(\mathsf{prim}, K, x, y, -)$. We stress that we do not create prim entries for the granted ideal-cipher queries, and thus there are at most p prim entries.

- **Encryption queries:** For each query $\textsc{Enc}(i,N,A,M)$ with answer $T \,\|\, C$, let $M = M_1 \cdots M_\ell$ and $C = C_1 \cdots C_\ell$, with $0 \leq |M_\ell| = |C_\ell| < n$, and $|M_j| = |C_j| = n$ for every $j < \ell$. For each $j < \ell$, let $V_j = M_i \oplus C_j$. Let $V_0 = H_{L_i}(A,C) \oplus T$. If $|M_\ell| = 0$ then let $V \leftarrow V_0 \cdots V_{\ell-1}$, otherwise let $V \leftarrow V_0 \cdots V_\ell$, where $V_\ell \leftarrow E_{K_i}(\mathrm{pad}(N) + \ell)$ in the real world, and $V_\ell \leftarrow (C_\ell \oplus M_\ell) \,\|\, Z$ in the ideal world, with $Z \xleftarrow{\$} \{0,1\}^{n-|M_\ell|}$. The string V is revealed to the adversary when it finishes querying, which can only improve its advantage. Associate the query above with the entry $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V)$.

- **Verification queries:** For each query $\textsc{Vf}(i,N,A,T \,\|\, C)$, associate it with the corresponding entry $(\mathsf{vf}, i, N, A, T \,\|\, C)$. Note that we do not need to keep track of the answers of the verification queries, since for any valid transcript in the ideal world, the answers of all verification queries must be false.

We say that a transcript is *bad* if one of the following happens:

1. There are two entries $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V)$ and $(\mathsf{enc}, j, N, A', M', T' \,\|\, C', V')$ with $i \neq j$ but $K_i = K_j$. Eliminating this case removes potential inconsistency due to the nonce reuse.

2. There is an entry $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$ and some indices $0 \leq s < t \leq \ell$ such that $V_s = V_t$. Recall that in the real world, V_s and V_t are outputs of E_{K_i} on different inputs $\mathrm{pad}(N) + s$ and $\mathrm{pad}(N) + t$. Thus in the real world, the strings V_s and V_t can't be the same.

3. There are two entries $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$ and $(\mathsf{enc}, j, N', A', M', T' \,\|\, C', V'_0 \cdots V'_u)$ with $N \neq N'$ and with some indices s and t such that $K_i = K_j$, and $V_s = V'_t$. Again, in the real world, V_s and V'_t are outputs of E_{K_i} on different inputs $\mathrm{pad}(N) + s$ and $\mathrm{pad}(N') + t$. Thus in the real world, the strings V_s and V'_t can't be the same.

4. There is an entry $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$ and an index t such that $V_t = L_i$. Recall that in the real world, $L_i = E_{K_i}(0^n)$ whereas V_t is the output of E_{K_i} on input $\mathrm{pad}(N)+t \neq 0^n$. Thus in the real world, the strings L_i and V_t must be different.

5. There are two entries $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$ and $(\mathsf{prim}, K, x, y, \cdot)$ such that $K = K_i$ and $x \in \{\mathrm{pad}(N), \ldots, \mathrm{pad}(N)+\ell\}$. Eliminating this case removes the potential inconsistency due to the adversary's accidental query of a correct key.

6. There are two entries $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$ and $(\mathsf{prim}, K, x, y, \cdot)$ such that $K = K_i$ and $y \in \{V_0, \ldots, V_\ell\}$. Again, eliminating this case removes the potential inconsistency due to the adversary's accidental query of a correct key.

7. There are two entries $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$ and $(\mathsf{vf}, j, N, A', T' \,\|\, C')$ such that $V_0 = H_{L_j}(A', C') \oplus T'$ and $K_i = K_j$. This means that the adversary should have received the answer true for this verification query, but recall that for valid transcripts in the ideal world, the answer must be false, leading to inconsistency.

8. There are entries $(\mathsf{vf}, i, N, A, T \,\|\, C)$ and $(\mathsf{prim}, K, x, y, \cdot)$ such that $K = K_i$ and $H_{L_i}(A, C) \oplus T = y$ and $x = \mathrm{pad}(N)$. This means that the adversary should have received the answer true for this verification query, but recall that for valid transcripts in the ideal world, the answer must be false, leading to inconsistency.

If a transcript is not bad and is valid for the ideal system then we say that it is *good*.

PROBABILITY OF BAD TRANSCRIPTS. Let $\mathcal{T}_{\mathrm{ideal}}$ be the random variable for the transcript in the ideal system. We now bound the probability that $\mathcal{T}_{\mathrm{ideal}}$ is bad. For each $j \in \{1, \ldots, 8\}$, let Bad_j be the set of transcripts that violates the j-th constraint of badness.

We first bound the probability $\Pr[\mathcal{T}_{\mathrm{ideal}} \in \mathsf{Bad}_1]$. For each entry $(\mathsf{enc}, i, N, \cdot, \cdot, \cdot, \cdot)$, there are at most d other entries $(\mathsf{enc}, j, N, \cdot, \cdot, \cdot, \cdot)$ such that $j \neq i$, and the chance that one of those d entries satisfy $K_j = K_i$ is at most $d/2^k$. Summing over at most q encryption entries,

$$\Pr[\mathcal{T}_{\mathrm{ideal}} \in \mathsf{Bad}_1] \leq \frac{dq}{2^k} \ .$$

Next, we bound the probability $\Pr[\mathcal{T}_{\mathrm{ideal}} \in \mathsf{Bad}_2]$. Consider an entry $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$. Since we are in the ideal world, the strings $V_0, \ldots, V_\ell$ are uniformly random and independent. Thus the chance that there are $0 \leq s < t \leq \ell$ such that $V_s = V_t$ is at most

$$\frac{\ell(\ell+1)}{2^{n+1}} \leq \frac{\ell B}{2^n} \leq \frac{|M|_n \cdot B}{2^n} \ .$$

Summing this over all encryption queries,

$$\Pr[\mathcal{T}_{\mathrm{ideal}} \in \mathsf{Bad}_2] \leq \frac{\sigma B}{2^n} \ .$$

Next, we bound the probability $\Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_3]$. For each entry $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$, consider another entry $(\mathsf{enc}, j, N', A', M', T' \,\|\, C', V'_0 \cdots V'_u)$. Since we are in the ideal world, the strings $V_0, \ldots, V_\ell, V'_0, \ldots, V'_u$ are uniformly random and independent, and thus the chance that there are s, t such that $V_s = V'_t$ is at most $(\ell+1)(u+1)/2^n \le (|M|_n + |A|_n)(|M'|_n + |A'|_n)/2^n$. We consider the following cases.

Case 1: $i = j$, and thus $K_i = K_j$. By summing over all encryption entries of user j, we obtain a bound $(|M|_n + |A|_n)B/2^n$ for the particular entry $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$ above. Summing this over all encryption entries, the probability corresponding to this case is at most $\sigma B/2^n$.

Case 2: $i \neq j$, and thus the conditional probability that $K_i = K_j$ is 2^{-k}. Summing over all pairs of encryption entries, we obtain a bound $\sigma^2/2^{k+n}$ for this case.

Summing up,

$$\Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_3] \le \frac{\sigma B}{2^n} + \frac{\sigma^2}{2^{k+n}} \ .$$

We now bound the probability $\Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_4]$. For each encryption entry $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$, the strings $V_0, \ldots, V_\ell$ are uniformly random and independent of L_i, and thus the chance that there is some V_s such that $V_s = L_i$ is at most $(\ell+1)/2^n \le (|M|_n + |A|_n)/2^n$. Summing this over all encryption entries,

$$\Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_4] \le \frac{\sigma}{2^n} \ .$$

Next, we bound the probability $\Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_5]$. For each entry (prim, K, x, y), there are at most d entries $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$ such that $x \in \{\text{pad}(N), \ldots, \text{pad}(N) + \ell\}$, and the chance that one of those d entries satisfies $K_i = K$ is at most $d/2^k$. Summing over all p ideal-cipher queries,

$$\Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_5] \le \frac{dp}{2^k} \ .$$

Next, we bound the probability $\Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_6]$. View each entry $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$ as throwing $\ell + 1 \le |M|_n + |A|_n$ balls into bins $V_0, \ldots, V_\ell$. Hence totally, we throw at most σ balls into 2^n bins, and the throws are uniformly random. Using Lemma 3.4, with probability at least $1 - 2^{-3n}$, each bin contains at most n balls. Thus for each entry $(\mathsf{prim}, K, x, y, \cdot)$, there are at most n entries $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$ such that $y \in \{V_0, \ldots, V_\ell\}$, and the chance that one of those n entries satisfies $K_i = K$ is at most $n/2^k$. Summing over all p ideal-cipher queries,

$$\Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_6] \le 2^{-3n} + \frac{pn}{2^k} \ .$$

We now bound the probability $\Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_7]$. View each granted query $(K, E_K(0^n))$ as throwing a ball into bin $E_K(0^n)$. Thus we throw 2^k balls into 2^n bins uniformly at random. By applying Lemma 3.4 for $a = 2^{k-n}$ and $\ell = 2$, with probability at least $1 - 2^{-3n}$, for every string $L \in \{0,1\}^n$, there are at most $n \cdot 2^{k-n}$ keys K such that $E_K(0^n) = L$. In other words, given the queries/answers that the adversary receives, the conditional min-entropy of each hash key L_i is at least $n - \lg(n)$.

Consider an entry $(\mathsf{vf}, i, N, A', T' \,\|\, C')$. Since the adversary is d-repeating, there are at most d entries $(\mathsf{enc}, j, N, A, M, T \,\|\, C, V_0 \cdots V_\ell)$ of the same nonce N. We consider the following cases.

Case 1: $j = i$ but $(A, C) \neq (A', C')$. As H is c-AXU and the conditional min-entropy of L_i is at least $n - \lg(n)$, the chance that $H_{L_i}(A', C') \oplus T' = H_{L_i}(A, C) \oplus T$ is at most

$$nc(|C'|_n + |A'|_n + |C|_n + |A|_n)/2^n \ .$$

Summing that over all verification queries, the probability corresponding to this case is at most $nc(\sigma + qB)/2^n$.

Case 2: $i = j$ and $(A, C) = (A', C')$. If $H_{L_i}(A', C') \oplus T' = H_{L_i}(A, C) \oplus T$ then we must have $T = T'$. Due to the definitional restriction to avoid trivial wins, the verification query must happen before the encryption query. In the ideal world, for each verification query $\textsc{Vf}(i, N, A', T' \,\|\, C')$, the chance that *the* subsequent encryption query of the same (i, N) that results in the same $T' \,\|\, C'$ is at most $2^{-|C'|} \leq 2^{-n}$. Summing this over all verification queries, the probability corresponding to this case is at most $q/2^n \leq \sigma/2^n$.

Case 3: $j \neq i$. The chance that $K_i = K_j$ is at most 2^{-k}. Summing this over all verification queries and all d matching encryption entries, the probability corresponding to this case is at most $dq/2^k$.

Combining all three cases,

$$\Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_7] \leq \frac{1}{2^{3n}} + \frac{dq}{2^k} + \frac{\sigma}{2^n} + \frac{nc(\sigma + qB)}{2^n} \ .$$

Finally, we bound the probability $\Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_8]$. We consider the following cases.

Case 1: The event $\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_8$ is caused by a prim entry of sign $-$. Again, view each entry $(\mathsf{prim}, K, x, y, -)$, as throwing a ball into bin y. Thus we throw at most p balls into 2^n bins, and while the throws can be inter-dependent, their distribution satisfies the requirement of Lemma 3.4 due to the hypothesis that $p \leq 2^{n-2}$. Then by Lemma 3.4, with probability at least $1 - 2^{-38n}$, the heaviest bin contains at most n balls. Hence for each entry $(\mathsf{vf}, i, N, A, T \,\|\, C)$, there are at most n entries $(\mathsf{prim}, K, x, y, -)$ such that $x = \text{pad}(N)$, and the chance that one of those prim entries satisfies the property $K = K_i$ is at most $n/2^k$. Summing over all q verification queries, the chance that this case happens is at most $2^{-3n} + qn/2^k$.

Case 2: The event $\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_8$ is caused by a prim entry of sign $+$ and a prior $\textsc{Vf}$ query. Consider an entry $(\mathsf{prim}, K, x, y, +)$. The chance that there is a prior entry $(\mathsf{vf}, i, N, A, T \,\|\, C)$ such that $H_L(A, C) \oplus T = y$, with $L \leftarrow E_K(0^n)$, and $K_i = K$ is at most $q/2^k(2^n - p) \leq 2q/2^{k+n}$. Summing over all p ideal-cipher queries, the chance that this case happens is at most $2pq/2^{k+n}$.

Case 3: The event $\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_8$ is caused by a prim entry of sign $+$ and a subsequent $\textsc{Vf}$ query. Fix (i^*, N^*, A^*, C^*) and let $\ell = |C^*|_n + |A^*|_n$. View each entry $(\mathsf{prim}, K, x, y, +)$ as throwing a ball into bin $y \oplus H_L(A^*, C^*)$, where $L \leftarrow E_K(0^n)$. Thus we throw at most p balls into 2^n bins, and while the throws can be inter-dependent, their distribution satisfies the requirement of Lemma 3.4 due to the hypothesis that $p \leq 2^{n-2}$. Then by Lemma 3.4, with probability at least $1 - 2^{-(3\ell+2)n}$, the heaviest bin contains at most ℓn balls. Thus for any adaptive choice of T, the entry $(\mathsf{vf}, i^*, N^*, A^*, T \,\|\, C^*)$ has at most $n\ell$ corresponding entries $(\mathsf{prim}, K, x, y, -)$ such that $y \oplus H_L(A^*, C^*) = T$, with $L \leftarrow E_K(0^n)$. Then for any adaptive entry $(\mathsf{vf}, i, N, A, T \,\|\, C)$, the chance that it has at most $n(|C|_n + |A|_n)$ corresponding entries $(\mathsf{prim}, K, x, y, -)$ such that $y \oplus H_L(A, C) = T$, with $L \leftarrow E_K(0^n)$, is at least

$$1 - \sum_{\ell=2}^{\infty} \sum_{(i^*, N^*, A^*, C^*) : |A^*|_n + |C^*|_n = \ell} 2^{-(3\ell+2)n} \geq 1 - \sum_{\ell=2}^{\infty} 2^{2n+2\ell n} \cdot 2^{-(3\ell+2)n} \geq 1 - \frac{2}{2^{2n}} \ .$$

Moreover, the probability that one of those $n(|C|_n+|A|_n)$ entries $(\mathsf{prim}, K, x, y, +)$ satisfies $K = K_i$ is at most $n(|C|_n + |A|_n)/2^k$. Summing over all verification queries, the chance that this case happens is at most $n\sigma/2^k + 2q/2^{2n}$.

Hence by the union bound,

$$\Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_8] \le \frac{1}{2^{3n}} + \frac{2q}{2^{2n}} + \frac{2pq}{2^{k+n}} + \frac{n(q+\sigma)}{2^k} \ .$$

Thus totally,

$$\begin{aligned}\Pr[\mathcal{T}_{\text{ideal}} \text{ is bad}] &\le \sum_{i=1}^{8} \Pr[\mathcal{T}_{\text{ideal}} \in \mathsf{Bad}_i] \\ &\le \frac{d(p+2q)+n(q+\sigma+p)}{2^k} + \frac{\sigma(2B+2)+nc(\sigma+qB)}{2^n} + \frac{2q+1}{2^{2n}} + \frac{\sigma^2+2pq}{2^{k+n}} \quad (2)\end{aligned}$$

Transcript Ratio. Fix a good transcript τ. For a key $K \in \{0,1\}^k$, let the multi-set $S_1(K)$ be the union of $\{(x,y) \mid (\mathsf{prim}, K, x, y, \cdot) \in \tau\}$ and the set $\{(0^n, E_K(0^n))\}$ as indicated by τ. Next, initialize the multi-set $S_2(K)$ as the empty set, and for every entry $(\mathsf{enc}, i, N, A, M, T \,\|\, C, V_0 \cdots V_\ell) \in \tau$, if τ indicates that $K_i = K$ then add the pairs $(\mathrm{pad}(N), V_0), \ldots, (\mathrm{pad}(N)+\ell, V_\ell)$ to $S_2(K)$. Finally, initialize the multi-set $S_3(K)$ as the empty set, and for every $(\mathsf{vf}, i, N, A, T \,\|\, C) \in \tau$, if τ indicates that $K_i = K$ and there is no entry of the form $(\mathrm{pad}(N), \cdot)$ or $(\cdot, Z)$ in $S_1(K) \cup S_2(K)$, where $Z \leftarrow H_{L_i}(A, C) \oplus T$, then add the pair $(\mathrm{pad}(N), Z)$ to $S_3(K)$. Let

$$s = \sum_{K \in \{0,1\}^k} |S_3(K)|$$

which is at most the total number of verification queries. Thus $s \le q$.

Suppose that this transcript τ contains exactly u users. Then in the ideal world, since τ is good,

$$\mathsf{ps}_{\text{ideal}}(\tau) = 2^{-ku} \prod_{K \in \{0,1\}^k} \prod_{i=0}^{|S_1(K)|-1} \frac{1}{2^n - i} \cdot \prod_{j=0}^{|S_2(K)|-1} \frac{1}{2^n} \ .$$

On the other hand, in the real world, the multi-sets $S_1(K)$ and $S_2(K)$ indicate pairs (x,y) such that $E_K(x)$ must be y, and the multi-set $S_3(K)$ indicate pairs (u,v) such that $E_K(u)$ must *not* be v. Since τ is good, those multi-sets contain no conflicting information, and $S_1(K)$ and $S_2(K)$ are disjoint. Let $V(K) = |S_1(K)| + |S_2(K)|$. Note that $V(K) + |S_3(K)| \le \sigma + q + p + 1 \le 2^{n-1}$. Then

$$\mathsf{ps}_{\text{real}}(\tau) \ge 2^{-ku} \prod_{K \in \{0,1\}^k} \prod_{i=0}^{V(K)-1} \frac{1}{2^n - i} \prod_{j=0}^{|S_3(K)|-1} \left(1 - \frac{1}{2^n - V(K) - j}\right) \ .$$

Hence

$$\frac{\mathsf{ps}_{\text{real}}(\tau)}{\mathsf{ps}_{\text{ideal}}(\tau)} \ \ge \ \prod_{K \in \{0,1\}^k} \prod_{j=0}^{|S_3(K)|-1} \left(1 - \frac{1}{2^n - V(K) - j}\right)$$

$$\geq \prod_{K\in\{0,1\}^k} \prod_{j=0}^{|S_3(K)|-1} \Big(1 - \frac{1}{2^n - V(K) - |S_3(K)|}\Big)$$

$$\geq \prod_{K\in\{0,1\}^k} \prod_{j=0}^{|S_3(K)|-1} \Big(1 - \frac{1}{2^{n-1}}\Big) = \Big(1 - \frac{1}{2^{n-1}}\Big)^s \geq 1 - \frac{s}{2^{n-1}} \geq 1 - \frac{q}{2^{n-1}} \ , \qquad (3)$$

where the second last inequality is due to the fact that $(1-x)^t \geq 1 - tx$ for any $t \geq 1$ and any $0 < x < 1$. Note that $q \leq \sigma/2$, because each query consists of at least two blocks, one for the message/ciphertext, and another for the associated data. Applying this observation to Eq. (3),

$$\frac{\mathsf{ps}_{\text{real}}(\tau)}{\mathsf{ps}_{\text{ideal}}(\tau)} \geq 1 - \frac{\sigma}{2^n} \ . \qquad (4)$$

Wrapping up. From Eq. (2) and Eq. (4), by using Lemma 2.1 with $\epsilon = \sigma/2^n$,

$$\begin{aligned} & \mathbf{Adv}^{\text{mu-ae}}_{\mathsf{CAU}[H,E]}(\mathcal{A}) \\ \leq \ & \frac{d(p+2q) + n(q+\sigma+p)}{2^k} + \frac{\sigma(2B+3) + nc(\sigma + qB)}{2^n} + \frac{2q+1}{2^{2n}} + \frac{\sigma^2 + 2pq}{2^{k+n}} \ . \end{aligned}$$

as claimed. ∎

Remarks. In TLS, an adversary can attempt at most one verification query per user, because a verification failure causes termination of the connection. One might wonder if the security of GCM can be improved in this restricted setting. In other words, we are interested in security of GCM against an adversary who, for each user, must make at most a verification query, and this verification query is made after all encryption queries for that particular user. However, any bound for such an adversary will continue to contain the bottleneck terms $\frac{pd}{2^k}$ and $\frac{\sigma B}{2^n}$, as there are matching attacks that only use encryption and ideal-cipher queries. Thus in the restricted setting above, the bound can only be slightly improved at best.

4 RGCM and the XOR transform

In this section, we introduce the XN transform that turns an AE scheme Π into another AE scheme Π^* by randomizing the effective nonces via an XOR operation. The scheme RGCM can be viewed as XN(GCM). We then reduce the mu security of Π^* under a generic adversary to that of Π under a strongly d-repeating adversary, where d is a small constant.

The XN transform. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AE scheme of nonce length r and key length k. Define the AE scheme $\Pi^* = (\mathcal{K}^*, \mathcal{E}^*, \mathcal{D}^*)$ of nonce length r and key length $k + r$ as in Fig. 3. For a key $K \,\|\, J$ of Π^*, we refer to the subkey K the *encryption key*, and the subkey J as the *nonce randomizer*.

Security gain via XN. We now reduce security of $\Pi^* = \mathsf{XN}(\Pi)$ under a generic adversary to that of Π under a strongly d-repeating adversary. This seems to be just a direct corollary of a generalized balls-into-bins result, where one throws q inter-dependent balls into 2^r bins as follows: (1) the marginal distribution of each ball is uniformly random, (2) balls of the same user must fall into different bins, and (3) balls of different users are independent. This balls-into-bins phenomenon is analyzed in Lemma 4.1 below.

procedure $\mathcal{K}^*()$	**procedure** $\mathcal{E}^*(K \,\|\, J, N, A, M)$	**procedure** $\mathcal{D}^*(K \,\|\, J, N, A, C)$
$K \xleftarrow{\$} \mathcal{K}();\ J \xleftarrow{\$} \{0,1\}^r$	$N^* \leftarrow N \oplus J;\ \ C \xleftarrow{\$} \mathcal{E}(K, N^*, A, M)$	$N^* \leftarrow N \oplus J;\ \ M \leftarrow \mathcal{D}(K, N^*, A, C)$
return $K \,\|\, J$	**return** C	**return** M

Figure 3: The XN transform to turn an AE scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ to another AE scheme $\Pi^* = (\mathcal{K}^*, \mathcal{E}^*, \mathcal{D}^*)$.

Lemma 4.1 *Let $0 < \epsilon < 1$ be a number, and let $r \geq 1$ be an integer. Suppose that we throw $q \leq 2^{(1-\epsilon)r}$ balls into 2^r bins. Before each ball is thrown, it is associated with a user i. The marginal distribution of each ball is uniformly random, balls of the same user must fall into different bins, and balls of different users are independent. Let X be the random variable for the number of balls in the heaviest bin, and let $d = \lceil 1.5/\epsilon \rceil - 1$. Then*

$$\Pr[X > d] \leq 2^{-r/2} \ .$$

Proof: Let $s = d + 1 = \lceil 1.5/\epsilon \rceil$. Since we throw q balls, there are

$$\binom{q}{s} \leq \frac{q^s}{s!}$$

sets of s balls. For each set, if it contains two balls of the same user then the balls in this set cannot be in the same bin. Otherwise, the balls in this set are thrown uniformly and independently, and thus the chance that they are in the same bin is $2^{-r(s-1)}$. By the union bound, the chance that there is a bin of s or more balls is at most

$$\frac{q^s}{2^{r(s-1)}} \leq \frac{2^{(1-\epsilon)rs}}{2^{r(s-1)}} = \frac{1}{2^{r(\epsilon s-1)}} \leq \frac{1}{2^{r/2}} \ .$$

This concludes the proof. ∎

Back to the security gain via the XN transform, the analysis above however only holds if the adversary non-adaptively chooses its nonces. If the adversary is somehow able to adaptively learn the nonce randomizers via its queries, it can then repeat the effective nonces as often as it wishes. Theorem 4.2 below refines the prior naive argument to handle adaptivity.

Theorem 4.2 *Let E be a blockcipher that we will model as an ideal cipher. Let $\Pi[E] = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AE scheme of nonce length r, and let $\Pi^*[E] = \mathsf{XN}(\Pi[E])$. Fix $0 < \epsilon < 1$. Let $\mathcal{A}$ be an adversary attacking Π^* using at most $q \leq 2^{(1-\epsilon)r}$ queries. Then we can construct a strongly d-repeating adversary $\mathcal{B}$ of the same concrete query complexity as $\mathcal{A}$, where $d = \lceil 1.5/\epsilon \rceil - 1$, such that*

$$\mathbf{Adv}^{\text{mu-ae}}_{\Pi^*[E]}(\mathcal{A}) \leq \mathbf{Adv}^{\text{mu-ae}}_{\Pi[E]}(\mathcal{B}) + \frac{1}{2^{r/2}} \ .$$

Proof: Adversary $\mathcal{B}$ initializes a flag $\mathsf{bad} \leftarrow \mathsf{false}$ and runs $\mathcal{A}$ with direct access to its ideal cipher. For each encryption query (i, N, A, M) (respectively, verification query (i, N, A, C)) of the latter, $\mathcal{B}$ initializes $J_i \xleftarrow{\$} \{0,1\}^r$ if the string J_i is not defined, otherwise it uses the existing J_i, and then creates an effective nonce $N^* \leftarrow N \oplus J_i$. If $\mathcal{B}$ did use N^* for d other users previously, it'll set $\mathsf{bad} \leftarrow \mathsf{true}$, terminate $\mathcal{A}$, and output 1. Otherwise, $\mathcal{B}$ queries $C \leftarrow \textsc{Enc}(i, N^*, A, M)$ (respectively,

$\text{VF}(i, N^*, A, C)$), and returns the answer to $\mathcal{A}$. When $\mathcal{A}$ finishes (without being terminated prematurely) and outputs a bit b', adversary $\mathcal{B}$ will output the same bit. Note that $\mathcal{B}$ is strongly d-repeating, and for each individual user, if $\mathcal{A}$ does not repeat a nonce among encryption queries then $\mathcal{B}$ also does not repeat an effective nonce among encryption queries. Moreover,

$$\Pr[\text{Real}^{\mathcal{B}}_{\Pi[E]} \Rightarrow 1] \geq \Pr[\text{Real}^{\mathcal{A}}_{\Pi^*[E]} \Rightarrow 1] \ , \tag{5}$$

because $\mathcal{B}$ either outputs 1, or agrees with $\mathcal{A}$. Since game $\text{Rand}^{\mathcal{A}}_{\Pi^*[E]}$ and the game that $\mathcal{B}$ simulates in its ideal world are identical until bad is set,

$$\Pr[\text{Rand}^{\mathcal{B}}_{\Pi[E]} \Rightarrow 1] \leq \Pr[\text{Rand}^{\mathcal{B}}_{\Pi[E]} \text{ sets } \mathsf{bad}] + \Pr[\text{Rand}^{\mathcal{A}}_{\Pi^*[E]} \Rightarrow 1] \ . \tag{6}$$

Subtracting Eq. (6) from Eq. (5) side by side, we obtain

$$\mathbf{Adv}^{\text{mu-ae}}_{\Pi[E]}(\mathcal{B}) \geq \mathbf{Adv}^{\text{mu-ae}}_{\Pi^*[E]}(\mathcal{A}) - \Pr[\text{Rand}^{\mathcal{B}}_{\Pi[E]} \text{ sets } \mathsf{bad}] \ .$$

It now suffices to show that $\Pr[\text{Rand}^{\mathcal{B}}_{\Pi[E]} \text{ sets } \mathsf{bad}] \leq 2^{-r/2}$. Recall that $\mathcal{B}$ sets bad to true only if adversary $\mathcal{A}$ can force $\mathcal{B}$ to use some effective nonce across more than d users. However, in game $\text{Rand}^{\mathcal{B}}_{\Pi[E]}$, the oracle answers are completely independent of the nonce randomizers J_i that $\mathcal{B}$ chooses. Hence one can view $\text{Rand}^{\mathcal{B}}_{\Pi[E]}$ as $\mathcal{A}$'s throwing q balls into 2^r bins where the throwing distribution is specified in Lemma 4.1, and bad is set only if some bin contains d or more balls. From Lemma 4.1, $\text{Rand}^{\mathcal{B}}_{\Pi[E]}$ sets bad with probability at most $2^{-r/2}$ as claimed. ∎

Security of RGCM. Combining Theorem 4.2 above with Theorem 3.1, we immediately obtain a strong security bound for $\mathsf{RCAU} = \mathsf{XN}(\mathsf{CAU})$, which includes RGCM as a special case for $c = 1.5$, $r = 96$, $n = 128$ and $k \in \{128, 256\}$.

Theorem 4.3 (Mu-security of RCAU/RGCM) *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher, with $k \geq n \geq 128$. Let H be a c-AXU hash function, and let r be the nonce length. Fix a number $0 < \epsilon < 1$, and let $d = \lceil 1.5/\epsilon \rceil - 1$. Let $\mathcal{A}$ be an adversary attacking $\mathsf{RCAU}[H, E]$ using at most $p \leq 2^{n-2}$ ideal-cipher queries, $q \leq 2^{(1-\epsilon)r}$ encryption/verification queries of total block length at most σ, and the encryption queries are of at most B blocks per user. Then*

$$\begin{aligned} & \mathbf{Adv}^{\text{mu-ae}}_{\mathsf{RCAU}[H,E]}(\mathcal{A}) \\ \leq \ & \frac{d(p+2q)+n(q+\sigma+p)}{2^k} + \frac{\sigma(2B+3)+nc(\sigma+qB)}{2^n} + \frac{2q+1}{2^{2n}} + \frac{\sigma^2+2pq}{2^{k+n}} + 2^{-r/2} \ . \end{aligned}$$

5 The Concatenation transform

In this section, we introduce the CN transform that turns an AE scheme Π into another AE scheme Π^* by randomizing the effective nonces by concatenating a random string. We then reduce the mu security of Π^* under a generic adversary to that of Π under a strongly d-repeating adversary. This transformation is used, for example, in the GCM scheme in TLS 1.2 and IPSec.

The CN transform. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AE scheme of nonce length r and key length k. For a parameter $t < r$, define the AE scheme $\Pi^* = (\mathcal{K}^*, \mathcal{E}^*, \mathcal{D}^*)$ of nonce length $r - t$ and key length $k + t$ as in Fig. 4. For a key $K \,\|\, J$ of Π^*, we refer to the subkey K the *encryption key*, and the subkey J as the *nonce randomizer*.

procedure $\mathcal{K}^*()$	**procedure** $\mathcal{E}^*(K \,\|\, J, N, A, M)$	**procedure** $\mathcal{D}^*(K \,\|\, J, N, A, C)$
$K \leftarrow\!\!\$\, \mathcal{K}();\ J \leftarrow\!\!\$\, \{0,1\}^t$	$N^* \leftarrow J \,\|\, N;\ C \leftarrow\!\!\$\, \mathcal{E}(K, N^*, A, M)$	$N^* \leftarrow J \,\|\, N;\ M \leftarrow \mathcal{D}(K, N^*, A, C)$
return $K \,\|\, J$	**return** C	**return** M

Figure 4: The CN transform to turn an AE scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ to another AE scheme $\Pi^* = (\mathcal{K}^*, \mathcal{E}^*, \mathcal{D}^*)$.

SECURITY GAIN VIA CN. We now reduce security of $\Pi^* = \mathsf{CN}(\Pi)$ under a generic adversary to that of Π under a strongly d-repeating adversary. To prove this theorem, we need the following lemma whose proof is along the same lines as that of Lemma 3.4 [7] except with the assumption that $t \geq 32$ instead of $t \geq 128$. We briefly recall the proof for completeness.

Lemma 5.1 *Fix integers $t \geq 32$, and $a, \ell \geq 2$. Suppose that we throw $q \leq a \cdot 2^t$ balls into 2^t bins. The throws may be inter-dependent, but for each i-th throw, conditioning on the result of the prior throws, the conditional probability that the i-th ball falls into any particular bin is at most 2^{1-t}. Then the chance that the heaviest bin contains $\lceil a\ell t/2 \rceil$ or more balls is at most $2^{-t(2\ell-1)}$.*

Proof: Let $s = t - 1$ and $r = \lceil a\ell t/2 \rceil \geq 32a$. There are

$$\binom{q}{r} \leq \frac{q^r}{r!}$$

sets of r balls out of the thrown q balls. For each set, the chance that all the balls are in the same bin is $2^{-s(r-1)}$. By the union bound, the chance that there is a bin of r or more balls is at most

$$\frac{q^r}{r! \cdot 2^{s(r-1)}} \leq \frac{(2a)^r 2^{rs}}{r! \cdot 2^{s(r-1)}} \leq \frac{(2a)^r 2^s}{(r/e)^r} \ .$$

The second inequality is due to the fact that $n! \geq (n/e)^n$ for every integer $n \geq 1$. Further,

$$\frac{(2a)^r 2^s}{(r/e)^r} \leq \frac{2^t}{(16/e)^{\ell t}} \leq 2^{-t(2\ell-1)} \ .$$

The last inequality relies on the assumption that $\ell \geq 2$. This concludes the proof. ∎

We are now ready to prove Theorem 5.2.

Theorem 5.2 *Let E be a blockcipher that we will model as an ideal cipher. Let $\Pi[E] = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AE scheme of nonce length r. Let $\Pi^*[E] = \mathsf{CN}(\Pi[E])$ have nonce length $r - t$ where $r > t \geq 32$. Let $d = 4t \cdot \lceil q/2^t \rceil$. Let $\mathcal{A}$ be an adversary attacking Π^* using at most q queries. Then we can construct a strongly d-repeating adversary $\mathcal{B}$ of the same concrete query complexity as $\mathcal{A}$, such that*

$$\mathbf{Adv}^{\text{mu-ae}}_{\Pi^*[E]}(\mathcal{A}) \leq \mathbf{Adv}^{\text{mu-ae}}_{\Pi[E]}(\mathcal{B}) + 2^{-7t} \ .$$

Proof of Theorem 5.2: Adversary $\mathcal{B}$ initializes a flag $\mathsf{bad} \leftarrow \mathsf{false}$ and runs $\mathcal{A}$ with direct access to its ideal cipher. For each encryption query (i, N, A, M) (respectively, verification query (i, N, A, C)) of the latter, $\mathcal{B}$ initializes $J_i \leftarrow\!\!\$\, \{0,1\}^t$ if the string J_i is not defined, otherwise it uses the existing J_i, and then creates an effective nonce $N^* \leftarrow J_i \,\|\, N$. If $\mathcal{B}$ did use N^* for d other users previously,

it'll set $\mathsf{bad} \leftarrow \mathsf{true}$, terminate $\mathcal{A}$, and output 1. Otherwise, $\mathcal{B}$ queries $C \leftarrow \textsc{Enc}(i, N^*, A, M)$ (respectively, $\textsc{Vf}(i, N^*, A, C)$), and returns the answer to $\mathcal{A}$. When $\mathcal{A}$ finishes (without being terminated prematurely) and outputs a bit b', adversary $\mathcal{B}$ will output the same bit. Note that $\mathcal{B}$ is strongly d-repeating, and for each individual user, if $\mathcal{A}$ does not repeat a nonce among encryption queries then $\mathcal{B}$ also does not repeat an effective nonce among encryption queries. Moreover,

$$\Pr[\mathrm{Real}^{\mathcal{B}}_{\Pi[E]} \Rightarrow 1] \geq \Pr[\mathrm{Real}^{\mathcal{A}}_{\Pi^*[E]} \Rightarrow 1] \ , \tag{7}$$

because $\mathcal{B}$ either outputs 1, or agrees with $\mathcal{A}$. Since game $\mathrm{Rand}^{\mathcal{A}}_{\Pi^*[E]}$ and the game that $\mathcal{B}$ simulates in its ideal world are identical until bad is set,

$$\Pr[\mathrm{Rand}^{\mathcal{B}}_{\Pi[E]} \Rightarrow 1] \leq \Pr[\mathrm{Rand}^{\mathcal{B}}_{\Pi[E]} \text{ sets } \mathsf{bad}] + \Pr[\mathrm{Rand}^{\mathcal{A}}_{\Pi^*[E]} \Rightarrow 1] \ . \tag{8}$$

Subtracting Eq. (8) from Eq. (7) side by side, we obtain

$$\mathbf{Adv}^{\text{mu-ae}}_{\Pi[E]}(\mathcal{B}) \geq \mathbf{Adv}^{\text{mu-ae}}_{\Pi^*[E]}(\mathcal{A}) - \Pr[\mathrm{Rand}^{\mathcal{B}}_{\Pi[E]} \text{ sets } \mathsf{bad}] \ .$$

It now suffices to show that $\Pr[\mathrm{Rand}^{\mathcal{B}}_{\Pi[E]} \text{ sets } \mathsf{bad}] \leq 2^{-7t}$. Recall that $\mathcal{B}$ sets bad to true only if adversary $\mathcal{A}$ can force $\mathcal{B}$ to use some effective nonce across more than d users. However, in game $\mathrm{Rand}^{\mathcal{B}}_{\Pi[E]}$, the oracle answers are completely independent of the nonce randomizers J_i that $\mathcal{B}$ chooses. Hence one can view $\mathrm{Rand}^{\mathcal{B}}_{\Pi[E]}$ as $\mathcal{A}$'s throwing q balls into 2^t bins and bad is set only if some bin contains d or more balls. Then, $\mathrm{Rand}^{\mathcal{B}}_{\Pi[E]}$ sets bad with probability at most 2^{-7t} by Lemma 5.1 by setting $\ell = 4$ and $a = 2 \cdot \lceil q/2^t \rceil$ in the lemma. ∎

Combining Theorem 5.2 above with Theorem 3.1, we immediately obtain a strong security bound for CGCM which we define as $\mathsf{CGCM} = \mathsf{CN}(\mathsf{CAU})$.

Theorem 5.3 (mu-security of $\mathsf{CN}(\mathsf{CAU})$) *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher, with $k \geq n \geq 128$. Let H be a c-AXU hash function. Let the key length of CGCM be $k + t$ and nonce length be $r - t$ with $r > t \geq 32$. Let $\mathcal{A}$ be an adversary attacking $\mathsf{CGCM}[H, E]$ using at most $p \leq 2^{n-2}$ ideal-cipher queries, q encryption/verification queries of total block length at most σ, and the encryption queries are of at most B blocks per user. Let $d = 4t \cdot \lceil q/2^t \rceil$. Then*

$$\mathbf{Adv}^{\text{mu-ae}}_{\mathsf{CGCM}[H,E]}(\mathcal{A}) \leq \frac{d(p+2q) + n(q+\sigma+p)}{2^k} + \frac{\sigma(2B+3) + nc(\sigma + qB)}{2^n} + \frac{2q+1}{2^{2n}} + \frac{\sigma^2 + 2pq}{2^{k+n}} + 2^{-7t} \ .$$

Comparing RGCM and CGCM. For concreteness, consider the setting in which an adversary can encrypt at most $B = 2^{40}$ blocks per user. Under TLS 1.2 and IPSec, 4 bytes of a nonce would be a random salt and remain fixed for an entire session, whereas the remaining 8 bytes would be implemented as a counter. Thus for CGCM in TLS 1.2 and IPSec, $d = 128 \cdot \lceil q/2^{32} \rceil \approx q/2^{25}$ and $n = 128$. The resulting bound is about $\frac{pq+q^2}{2^{153}} + \frac{\sigma}{2^{86}} + \frac{p+q}{2^{120}}$, which is much stronger than the bound $\frac{pq+\sigma^2}{2^{128}}$ in prior works [5, 13]. Still, in the same setting, RGCM is much better: using $d = 14$ (meaning that q is required to be smaller than 2^{72}), the advantage of the adversary is about $\frac{p}{2^{120}} + \frac{\sigma}{2^{86}}$.

Discussion. Recall that our security definition requires that nonces for each individual must be distinct. In TLS 1.2, however, one might implement nonces as 64-bit random strings. To capture security for this setting, one can relax our security definition by allowing adversaries to repeat nonces for some users, with probability at most ε. In the case of TLS 1.2, one can pick $\varepsilon = \frac{\sigma B}{2^{64}}$. Next, for any

AE scheme Π and for an adversary $\mathcal{A}$ who repeats nonces for some user with probability ε, one can easily construct an adversary $\mathcal{B}$ who is nonce-respecting, such that $\mathbf{Adv}^{\text{mu-ae}}_{\Pi}(\mathcal{A}) \leq \mathbf{Adv}^{\text{mu-ae}}_{\Pi}(\mathcal{B})+\varepsilon$. Thus the security of CGCM with random nonces can be bounded by the formula in Theorem 5.3 plus an additional term $\frac{\sigma B}{2^{64}}$.

6 Security with Public Salting

In both the XN and CN transforms in the previous sections, the nonce randomizer (or salt) J is part of the secret key. This reflects transformations actually used in practice. However, in general, the secrecy of the nonce randomized is unnecessary for mu security. We observe here that as long as the nonces are not chosen arbitrarily by the adversary one can guarantee mu security even when the nonces are made public, and security of such schemes can be described in terms of an appropriate d-repeating adversary.

An example. Somewhat informally, imagine that we are in a scenario where each user picks a nonce randomizer $J_i \in \{0,1\}^r$. Then, the nonce of the c-th message sent by user i is in particular $J_i \oplus c$, and is sent along with the message. More generally, the XOR can be replaced by any operator $\boxplus$ such that $(\{0,1\}^r, \boxplus)$ is an abelian group.

Our formalism allows us to capture this scenario by restricting ourselves to d-repeating adversaries $\mathcal{A}$ (for an appropriate d) which only invoke the encryption oracle with queries of the form $\textsc{Enc}(i, J_i \boxplus c_i, A, M)$, where c_i is a counter increased each time a message is encrypted for user i. Note that we allow here $\mathcal{A}$ to make unrestricted verification queries, exploiting the full power of our model – this is consistent with the fact that a person-in-the-middle attacker may attempt to inject ciphertexts with arbitrary nonces.

In particular, by an argument similar to that of Lemma 4.1, if such $\mathcal{A}$ makes at most $q \leq 2^{(1-\epsilon)r}$ encryption/verification queries, it is d-repeating for $d = \lceil 1.5/\epsilon \rceil - 1$ except with probability $2^{-r/2}$. Thus, for CAU, the mu security with respect to such $\mathcal{A}$ follows from Theorem 3.1, and is such that

$$\mathbf{Adv}^{\text{mu-ae}}_{\Pi[H,E]}(\mathcal{A}) \leq \frac{d(p+2q)+n(q+\sigma+p)}{2^k} + \frac{\sigma(2B+3)+nc(\sigma+qB)}{2^n} + \frac{2q+1}{2^{2n}} + \frac{\sigma^2+2pq}{2^{k+n}} + 2^{-r/2} \quad .$$

This is the same bounds as that obtained for RGCM in Theorem 7. In particular, this means that we can obtain the same security for GCM without keeping any part of the nonces secret and hence potentially saving on key length.

Acknowledgments

We thank the CCS reviewers for their insightful comments, and Kenny Paterson and Eric Rescorla for providing very useful information on the TLS protocol. We also thank Jean Paul Degabriele, Jérôme Govinden, Felix Günther, and Kenny Paterson for extensive feedback and for pointing out glitches in an earlier version of our proofs.

References

[1] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The Secure Real-time Transport Protocol (SRTP). Internet-draft, Internet Engineering Task Force, 2004. 4

[2] M. Bellare, D. J. Bernstein, and S. Tessaro. Hash-function based PRFs: AMAC and its multi-user security. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 566–595. Springer, Heidelberg, May 2016. 3

[3] M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000. 3

[4] M. Bellare and V. T. Hoang. Identity-based format-preserving encryption. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 17*, pages 1515–1532. ACM Press, Oct. / Nov. 2017. 10

[5] M. Bellare and B. Tackmann. The multi-user security of authenticated encryption: AES-GCM in TLS 1.3. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 247–276. Springer, Heidelberg, Aug. 2016. 3, 4, 5, 6, 8, 21

[6] E. Biham. How to decrypt or even substitute DES-encrypted messages in 2^{28} steps. *Inf. Process. Lett.*, pages 117–124, 2002. 3

[7] P. Bose, V. T. Hoang, and S. Tessaro. Revisiting AES-GCM-SIV: Multi-user security, faster key derivation, and better bounds. In *EUROCRYPT 2018*, 2018. 3, 4, 5, 11, 20

[8] S. Chen and J. P. Steinberger. Tight security bounds for key-alternating ciphers. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 327–350. Springer, Heidelberg, May 2014. 8

[9] S. Goldwasser and M. Bellare. Lecture notes on cryptography. Summer Course "Cryptography and Computer Security" at MIT, 1999. 10

[10] V. T. Hoang and S. Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 3–32. Springer, Heidelberg, Aug. 2016. 3, 7

[11] V. T. Hoang and S. Tessaro. The multi-user security of double encryption. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 381–411. Springer, Heidelberg, May 2017. 3

[12] T. Iwata, K. Ohashi, and K. Minematsu. Breaking and repairing GCM security proofs. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 31–49. Springer, Heidelberg, Aug. 2012. 5

[13] A. Luykx, B. Mennink, and K. G. Paterson. Analyzing multi-key security degradation. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 575–605. Springer, Heidelberg, Dec. 2017. 3, 5, 21

[14] U. M. Maurer. Indistinguishability of random systems. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 110–132. Springer, Heidelberg, Apr. / May 2002. 7

[15] D. A. McGrew. Generation of Deterministic Initialization Vectors (IVs) and Nonces. Internet-Draft draft-mcgrew-iv-gen-03, Internet Engineering Task Force, Oct. 2013. Work in Progress. 4

[16] D. A. McGrew and S. R. Fluhrer. Attacks on additive encryption of redundant plaintext and implications on internet security. In D. R. Stinson and S. E. Tavares, editors, *SAC 2000*, volume 2012 of *LNCS*, pages 14–28. Springer, Heidelberg, Aug. 2001. 4

[17] D. A. McGrew and J. Viega. The security and performance of the Galois/counter mode (GCM) of operation. In A. Canteaut and K. Viswanathan, editors, *INDOCRYPT 2004*, volume 3348 of *LNCS*, pages 343–355. Springer, Heidelberg, Dec. 2004. 3, 9

[18] N. Mouha and A. Luykx. Multi-key security: The Even-Mansour construction revisited. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 209–223. Springer, Heidelberg, Aug. 2015. 3

[19] J. Patarin. The "coefficients H" technique (invited talk). In R. M. Avanzi, L. Keliher, and F. Sica, editors, *SAC 2008*, volume 5381 of *LNCS*, pages 328–345. Springer, Heidelberg, Aug. 2009. 8

[20] E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. Internet-draft, Internet Engineering Task Force, 2018. Work in Progress. 3, 4

[21] P. Rogaway and T. Shrimpton. A provable-security treatment of the key-wrap problem. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer, Heidelberg, May / June 2006. 3

[22] J. Salowey, A. Choudhury, and D. A. McGrew. AES galois counter mode (GCM) cipher suites for TLS. *RFC*, 5288:1–8, 2008. 4

[23] S. Tessaro. Optimally secure block ciphers from ideal primitives. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 437–462. Springer, Heidelberg, Nov. / Dec. 2015. 3