The proceedings version is in CRYPTO 2016. This is the full version.

The Multi-User Security of Authenticated Encryption: AES-GCM in TLS 1.3

Mihir Bellare[1] Björn Tackmann[2]

June 3, 2016

Abstract

We initiate the study of multi-user (mu) security of authenticated encryption (AE) schemes as a way to rigorously formulate, and answer, questions about the "randomized nonce" mechanism proposed for the use of the AE scheme GCM in TLS 1.3. We (1) Give definitions of mu ind (indistinguishability) and mu kr (key recovery) security for AE (2) Characterize the intent of nonce randomization as being improved mu security as a defense against mass surveillance (3) Cast the method as a (new) AE scheme RGCM (4) Analyze and compare the mu security of both GCM and RGCM in the model where the underlying block cipher is ideal, showing that the mu security of the latter is indeed superior in many practical contexts to that of the former, and (5) Propose an alternative AE scheme XGCM having the same efficiency as RGCM but better mu security and a more simple and modular design.

[1] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: mihir@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~mihir/. Supported in part by NSF grants CNS-1526801 and CNS-1228890, ERC Project ERCC FP7/615074 and a gift from Microsoft.

[2] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: btackmann@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~btackmann. Supported in part by the Swiss National Science Foundation (SNF) via Fellowship No. P2EZP2_155566 and NSF grant CNS-1228890.

Contents

1 Introduction

Traditionally, security definitions were single-user, meaning there was a single target key. Consideration of the multi-user setting began with public-key encryption [3]. In this setting, there are many users, each with their own key, and the target is to violate security under *some* key. This is, first, simply more realistic, reflecting real usage, but is now even more relevant from the mass-surveillance perspective. This paper initiates a study of the multi-user security of authenticated encryption. Our motivation comes from TLS 1.3.

AE. The form of authenticated encryption (AE) we consider is nonce-based [27]. The encryption algorithm $\mathsf{AE.Enc}$ takes key K, nonce N, message M and header H to deterministically return a ciphertext $C \leftarrow \mathsf{AE.Enc}(K, N, M, H)$. The requirement formalized in [27] is to provide privacy of M, and authenticity of both M and H, as long as a nonce is not re-used. The formalization refers to only one target key, meaning is in the single user (su) setting.

There are many AE schemes (provably) meeting this security requirement. One simple way to obtain them is via generic composition of privacy-only encryption schemes with MACs [5, 25]. There are also dedicated schemes such as OCB [30, 28, 21], CCM [10] and GCM [23, 11]. The last, with AES, is used in TLS 1.3.

Multi-User Security of AE. We formalize multi-user (mu) security of an authenticated-encryption scheme AE. The game picks an adversary-determined number u of independent target keys $K_1, \ldots, K_u$. The adversary gets an encryption oracle that takes an index $i \in [1..u]$, a message, nonce and header, and returns either an encryption of these under K_i or a random string of the same length. It also gets a verification oracle that takes i, a ciphertext, nonce and header, and indicates whether or not decryption is valid. Security is required as long as the adversary does not re-use a nonce *for a particular user*. That is, it is fine to obtain encryptions under the same nonce for different keys, just not under the same key. When $u = 1$, we get a definition equivalent to (but formulated slightly differently from) the (single-user) definition of [27].

Besides this usual goal (which we call indistinguishability), we also formalize a mu key-recovery goal. Again the game picks target keys $K_1, \ldots, K_u$ and gives the adversary an encryption oracle. This time time the latter is always true, meaning it takes an index $i \in [1..u]$, a message, nonce and header, and returns an encryption of these under K_i. The adversary also gets a verification oracle, and, to win, must find one of the target keys. A key-recovery attack is much more damaging than a distinguishing attack, and is the threat of greatest concern to practioners. Key recovery security is usually dismissed by theoreticians as being implied by indistinguishability, but this view misses the fact that the quantitative security of a scheme, in terms of bounds on adversary advantage, can be very different for the two metrics, making it worthwhile to consider key recovery security separately and additionally.

We give our definitions in the ideal-cipher model. (Standard-model definitions follow because this is just the special case where scheme algorithms and adversaries make no queries to the ideal cipher.) For all the schemes we consider, the assumption that the underlying blockcipher is a PRP suffices to prove security. The reason we use the ideal-cipher model is that adversary queries to the ideal cipher give a clear and rigorous way to measure the offline computation being performed in an attack. Also in some cases we get better bounds.

Multi-user security is not qualitatively different from single-user security. A hybrid argument shows that the latter implies the former. But the two could be quantitatively quite different, and this has important practical implications. In the hybrid reduction, there is a loss of a factor u in adversary advantage. Thus, the mu advantage of an adversary could be as much as u times its su advantage. This is the worst case. But it could be a lot less, degrading much more slowly with u.

This would be better.

AE IN TLS 1.3. As the protocol underlying `https`, TLS is the basis for secure communication on the Internet, used millions of times a day. The existing versions up to TLS 1.2 have however been subject to many attacks. The effort to create a new and (hopefully) better version, TLS 1.3, is currently underway. TLS (of whatever version) begins with a *handshake*. This is an authenticated key exchange that establishes a shared session key, called the traffic secret, between client and server. This step will not be our concern. After the handshake, data is authenticated and encrypted within the so-called *record layer*, using an authenticated encryption scheme AE that is keyed by a key K derived from the traffic secret. The currently proposed choice of AE is AES-GCM.

The most natural way to use AE in the record layer is directly, meaning the data message M is simply encrypted via $C \leftarrow \mathsf{AE.Enc}(K, N, M, H)$, where N is a nonce (in TLS 1.3 this is a sequence number that is known to the receiver) and H is the header. This is not what TLS 1.3 proposes. Instead, they randomize the nonce, computing $C \leftarrow \mathsf{AE.Enc}(K, N \oplus L, M, H)$, where the randomizer L is also derived from the traffic secret. (It is thus known to the receiver, enabling decryption.) Why do this? Brian Smith gave the following motivation on the TLS 1.3 mailing list [32]:

> ... massively parallel attacks on many keys at once seem like the most promising way to break AES-128. It seems bad to have popular endpoints encrypting the same plaintext block with the same nonce with different keys. That seems like exactly the recipe for making such attacks succeed. It seems like it would be better, instead, to require that the initial nonces to be calculated from the key block established during key agreement ... This ... should prevent any such massively-parallel attack from working.

In this paper, we aim to understand and formalize the threat alluded to here, and then assess to what extent one can prove that nonce-randomization guarantees security. In particular, we suggest that the formal cryptographic goal underlying nonce randomization and Smith's comment is improved multi-user security. In our model, the "massively parallel attack" is a key-search attack that finds the GCM key of some user out of u target users —here we are referring to the basic GCM scheme, in the absence of nonce randomization— in time $2^\kappa/u$ where κ is the key length of the underlying block cipher, $\kappa = 128$ for AES. The attack picks some N, M, H and for each $i \in [1..u]$ obtains from its encryption oracle the encryption C_i of these quantities under K_i. Now, it goes through all possible κ-bit keys L, for each computing $C_L \leftarrow \mathsf{AE.Enc}(L, N, M, H)$, and returning L if $C_L = C_i$ for some i. Note that the attack needs a single computation of $\mathsf{AE.Enc}$ for each L, not one per user, which is why the running time is $2^\kappa/u$. Given NSA computing capabilities, the fear of the TLS 1.3 designers is that this attack may be feasible for them for large u, and thus a mass-surveillance threat. Nonce randomization is a candidate way to circumvent the attack. The question this raises is whether nonce randomization works. To answer this in a rigorous way, we abstract out a (new) AE scheme and then use our definitions of mu security.

RGCM. In TLS 1.3, nonce randomization is viewed as a way to use GCM in the record layer. We take a different perspective. We view the method as defining a new AE scheme that we call RGCM. In this scheme, the randomizer is part of the key. This view is appropriate because the randomizer was derived from the traffic secret just like the base key, and has the security necessary to be used as a key, and the randomizer is also static across the session, just like the base key. While GCM has a key whose length is the key length κ of the underlying block cipher ($\kappa = 128$ for AES), RGCM has a key of length $\kappa + \nu$, where ν is the length of the randomizer ($\nu = 96$ for GCM in TLS 1.3). Nonces are assumed to also have length ν so that xoring the nonce with the randomizer makes sense.

RESULTS. With this perspective, we are looking at two AE schemes, GCM and RGCM. We can now divorce ourselves of TLS details and analyze them as AE schemes to determine the quantitative mu security of both. The number p of adversary queries to the ideal cipher is the central parameter, capturing the offline computational effort of the adversary. As before u is the number of users, and we let m denote the total number of bits encrypted, meaning the sum of the lengths of all messages in queries.

Let us first discuss mu security under key recovery, where the picture is clearer. Roughly, we show that key recovery for GCM needs $p = 2^{\kappa}/u$ while for RGCM it needs $p = 2^{\kappa+\nu}/um$. We expect m to be quite a bit less than 2^{ν} —in the current schemes, $\nu = 96$— so the effort to recover a key is significantly higher for RGCM than for GCM. This says that nonce randomization works, meaning it does increase mu security as targeted by the TLS 1.3 designers, at least for key recovery.

For mu-ind security, the picture is more complex. We distinguish the case of passive attacks, where the adversary does not query its verification oracle, and active attacks, where it does. In the passive case, RGCM still emerges as superior, but in the active case, the two schemes become comparable. Also, our bounds in the ind case are complex, and interesting terms get swamped by collision terms. We stress that the bounds here may not be tight, so the picture we are seeing could reflect limitations of our analysis techniques rather than the inherent security of the schemes. Obtaining better (and ideally tight) bounds is an interesting open question.

XGCM. Even if under some metrics superior to GCM, RGCM performs considerably worse than expected from an AE with key length $\kappa+\nu$, and the natural question is, why not use some standard scheme or construction paradigm rather than "roll your own" with RGCM? The most obvious choice is AES256-GCM. Our analysis of GCM shows that AES256-GCM has good enough mu security, simply due to the larger key size. However, AES256-GCM is slower than AES-RGCM, and a scheme using AES itself would be preferable. We suggest and analyze XGCM, derived simply as GCM with the blockcipher E: $\{0,1\}^{\kappa} \times \{0,1\}^{\lambda} \to \{0,1\}^{\lambda}$ replaced by EX: $\{0,1\}^{\kappa+\lambda} \times \{0,1\}^{\lambda} \to \{0,1\}^{\lambda}$, defined by $\mathrm{EX}(K\|L, X) = L \oplus \mathrm{E}(K, L \oplus X)$. This transform of a blockcipher uses the Even-Mansour technique [12]. It was suggested by Rivest as a key-extension method for DES and first analyzed by Kilian and Rogaway [18]. Our analysis implies that, with AES parameters, the mu security of XGCM is better than that of RGCM. Its performance is however essentially the same as that of GCM or RGCM. While it would be a viable alternative for AES-RGCM in TLS 1.3, it does require non-black-box changes to the implementation of AES-GCM, whereas for AES-RGCM the change is only the randomization of the nonce input.

RELATED WORK. GCM was proposed by McGrew and Viega (MV) [23] and standardized by NIST as [11]. MV [23] prove single-user security assuming PRP-security of the underlying blockcipher. While the original scheme allows variable-length nonces [23], IOM [17] showed that the security proof of MV was flawed in this case and the claimed security bounds did not hold. They provide a corrected proof, which was later improved by NOMI [26]. In this paper we only consider fixed-length nonces. We prove security in the mu setting in the ideal cipher model.

Key-recovery security of symmetric encryption schemes was defined in [29] for the single-user, privacy-only setting. We extend their definition to the mu, authenticated encryption setting.

BMMRT [1] and FGMP [13] analyze the record layer of TLS 1.3 relative to the goal of providing a secure channel, under an appropriate formalization of the latter. These works assume that AES-GCM is a secure AE scheme. Our work is not attempting to analyze the record layer. It is analyzing the security of GCM and RGCM as stand-alone AE schemes, with emphasis on their mu security.

We are seeing increased interest in multi-user security, further reflected in this paper. BCK [4] considered mu security for PRFs as an intermediate step in the analysis of the cascade construction. Multi-user security of PRFs and PRPs (blockciphers) has been further considered in [24, 33, 2]. The

first work that highlighted mu security as a goal and targeted quantitative security improvements seems to have been BBM [3], the primitive here being public-key encryption. Multi-user security for signatures was considered by GMS [15] and has been the subject of renewed interest in [19, 7]. Further works involving multi-user security include [8, 9, 16], and, in the cryptanalytic context, [14].

2 Preliminaries

We let ε denote the empty string. If Z is a string then $|Z|$ denotes its length and $Z[1..i]$ denotes bits 1 through i of Z. If X is a finite set, we let $x \leftarrow\!\!\$\, X$ denote picking an element of X uniformly at random and assigning it to x. Algorithms may be randomized unless otherwise indicated. Running time is worst case. If A is an algorithm, we let $y \leftarrow A(x_1, \ldots; r)$ denote running A with random coins 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 $y \leftarrow A(x_1, \ldots; r)$. We let $[A(x_1, \ldots)]$ denote the set of all possible outputs of A when invoked with inputs $x_1, \ldots$.

We use the code-based game-playing framework of BR [6]. (See Fig. 1 for an example.) By $\Pr[\mathrm{G}]$ we denote the probability that the execution of game G results in the game returning true. In games, integer variables, set variables and boolean variables are assumed initialized, respectively, to 0, the empty set, and false.

A family of functions $\mathsf{F}\colon \mathsf{F.Keys} \times \mathsf{F.Dom} \to \mathsf{F.Rng}$ is a two-argument function that takes a key K in the key space $\mathsf{F.Keys}$, an input x in the domain $\mathsf{F.Dom}$ and returns an output $\mathsf{F}(K, x)$ in the range $\mathsf{F.Rng}$. In the ROM, F takes an oracle RO. We say F has key length $\mathsf{F.kl}$ if $\mathsf{F.Keys} = \{0,1\}^{\mathsf{F.kl}}$; output length $\mathsf{F.ol}$ if $\mathsf{F.Rng} = \{0,1\}^{\mathsf{F.ol}}$; and input length $\mathsf{F.il}$ if $\mathsf{F.Dom} = \{0,1\}^{\mathsf{F.il}}$.

We say that $\mathsf{F}\colon \{0,1\}^{\mathsf{F.kl}} \times \{0,1\}^{\mathsf{F.il}} \to \{0,1\}^{\mathsf{F.ol}}$ is a *block cipher* if $\mathsf{F.il} = \mathsf{F.ol}$ and $\mathsf{F}(K, \cdot)\colon \{0,1\}^{\mathsf{F.il}} \to \{0,1\}^{\mathsf{F.ol}}$ is a permutation for each K in $\{0,1\}^{\mathsf{F.kl}}$. We denote by $\mathsf{F}^{-1}(K, \cdot)$ the inverse of $\mathsf{F}(K, \cdot)$.

Let $\mathsf{H}\colon \mathsf{H.Keys} \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^{\mathsf{H.ol}}$ be a family of functions with domain $\mathsf{H.Dom} = \{0,1\}^* \times \{0,1\}^*$. Let $\epsilon\colon \mathbb{N} \times \mathbb{N} \to [0,1]$ be a function. Somewhat extending [20], we say that H is *ϵ-almost XOR-universal* if for all distinct $(M_1, H_1), (M_2, H_2) \in \mathsf{H.Dom}$ and all $s \in \{0,1\}^{\mathsf{H.ol}}$, we have

$$\begin{aligned}\Pr[\mathsf{H}(hk, (M_1, H_1)) \oplus \mathsf{H}(hk, (M_2, H_2)) = s \;:\; hk \leftarrow\!\!\$\, \mathsf{H.Keys}] \\ \leq \epsilon(\max(|M_1|, |M_2|), \max(|H_1|, |H_2|))\ .\end{aligned}$$

3 Multi-User Security of Symmetric Encryption

We consider symmetric encryption in a multi-user setting. We give two definitions of security. The first, an indistinguishability-style definition, extends Rogaway's single-user definition [27] to the multi-user setting, and represents a very strong requirement. We also define security against key recovery, representing the goal the attacker would most like to achieve and the most common target of cryptanalysis. We will see that the security bounds for these notions can differ. Since our analyses will be in the ideal-cipher model, the definitions are given directly in that model.

Syntax. A symmetric encryption scheme AE specifies a deterministic encryption algorithm $\mathsf{AE.Enc}\colon \{0,1\}^{\mathsf{AE.kl}} \times \mathsf{AE.NS} \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ that takes a key $K \in \{0,1\}^{\mathsf{AE.kl}}$, a nonce $N \in \mathsf{AE.NS}$, a message $M \in \{0,1\}^*$ and a header $H \in \{0,1\}^*$ to return a ciphertext $C \leftarrow \mathsf{AE.Enc}^{\mathrm{E},\mathrm{E}^{-1}}(K, N, M, H) \in \{0,1\}^{\mathsf{AE.cl}(|M|)}$. Here $\mathsf{AE.kl} \in \mathbb{N}$ is the key length of the scheme, $\mathsf{AE.NS}$ is the nonce space and $\mathsf{AE.cl}\colon \mathbb{N} \to \mathbb{N}$ is the ciphertext length function. The oracles represent a cipher $\mathrm{E}\colon \{0,1\}^{\mathsf{AE.ckl}} \times \{0,1\}^{\mathsf{AE.bl}} \to \{0,1\}^{\mathsf{AE.bl}}$ and its inverse E^{-1}. In the security games this

Game $\mathbf{G}^{\text{mu-ind}}_{\mathsf{AE}}(A)$	$\underline{\mathrm{E}(L,x)}$
$b \leftarrow\!\!\$\ \{0,1\}$; $b' \leftarrow\!\!\$\ A^{\textsc{New},\textsc{Enc},\textsc{Vf},\mathrm{E},\mathrm{E}^{-1}}$	If $T[L,x] = \perp$ then
Return $(b' = b)$	$\quad T[L,x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[L,\cdot]}$
$\underline{\textsc{New}()}$	$\quad T^{-1}[L,T[L,x]] \leftarrow x$
$v \leftarrow v+1$; $K_v \leftarrow\!\!\$\ \{0,1\}^{\mathsf{AE.kl}}$	Return $T[L,x]$
$\underline{\textsc{Enc}(i,N,M,H)}$	$\underline{\mathrm{E}^{-1}(L,y)}$
If not $(1 \le i \le v)$ then return $\perp$	If $T^{-1}[L,y] = \perp$ then
If $((i,N) \in U)$ then return $\perp$	$\quad T^{-1}[L,y] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T^{-1}[L,\cdot]}$
$C_1 \leftarrow \mathsf{AE.Enc}^{\mathrm{E},\mathrm{E}^{-1}}(K_i,N,M,H)$	$\quad T[L,T^{-1}[L,y]] \leftarrow y$
$C_0 \leftarrow\!\!\$\ \{0,1\}^{\mathsf{AE.cl}(\lvert M\rvert)}$	Return $T^{-1}[L,y]$
$U \leftarrow U \cup \{(i,N)\}$; $V \leftarrow V \cup \{(i,N,C_b,H)\}$	
Return C_b	
$\underline{\textsc{Vf}(i,N,C,H)}$	
If not $(1 \le i \le v)$ then return $\perp$	
If $((i,N,C,H) \in V)$ then return true	
If $(b = 0)$ then return false	
$M \leftarrow \mathsf{AE.Dec}^{\mathrm{E},\mathrm{E}^{-1}}(K_i,N,C,H)$	
Return $(M \neq \perp)$	

Figure 1: Game defining multi-user indistinguishability security of symmetric encryption scheme AE in the ideal-cipher model.

cipher will be chosen at random, meaning be ideal. We view the key length $\mathsf{AE.ckl}$ and block length $\mathsf{AE.bl}$ of the cipher as further parameters of AE itself. Also specified is a deterministic decryption algorithm $\mathsf{AE.Dec}$: $\{0,1\}^{\mathsf{AE.kl}} \times \mathsf{AE.NS} \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \cup \{\perp\}$ that takes K, N, C, H and returns $M \leftarrow \mathsf{AE.Dec}^{\mathrm{E},\mathrm{E}^{-1}}(K,N,C,H) \in \{0,1\}^* \cup \{\perp\}$. Correctness requires that $\mathsf{AE.Dec}(K,N,\mathsf{AE.Enc}(K,N,M,H),H) = M$ for all $M, H \in \{0,1\}^*$, all $N \in \mathsf{AE.NS}$ and all $K \in \{0,1\}^{\mathsf{AE.kl}}$.

Indistinguishability Security. We extend Rogaway's definition of indistinguishability security for authenticated encryption [27], which is in the single-user setting, to the multi-user setting. The formalization is based on game $\mathbf{G}^{\text{mu-ind}}_{\mathsf{AE}}(A)$ of Fig. 1, associated to encryption scheme AE and adversary A. The game initially samples a random bit challenge b, with $b = 1$ indicating it is in "real" mode and $b = 0$ that it is in "ideal" mode. As per our conventions noted in Section 2, the sets U, V are assumed initialized to the empty set, and the integer v is assumed initialized to 0. Now the adversary A has access to an oracle New that creates new user instances. A also has access to an encryption oracle Enc that takes a user instance identifier i, a nonce $N \in \mathsf{AE.NS}$, a message M, and a header H. The oracle either returns a uniformly random bit string of length $\mathsf{AE.cl}$ that depends only on the length of M (for $b = 0$), or an encryption under $\mathsf{AE.Enc}$ using the key of user i (for $b = 1$). The oracle checks that A does not re-use nonces for a user instance, and that it is invoked only for user instances that exist. Analogously, there is a verification oracle Vf that takes user instance i, nonce $N \in \mathsf{AE.NS}$, ciphertext C, and header H. Oracle Vf always accepts ciphertexts generated by Enc for the same i, N, and H, rejects all other ciphertexts for $b = 0$, and uses the decryption algorithm $\mathsf{AE.Dec}$ to check the validity of the ciphertext for $b = 1$. As a last step, the adversary outputs a bit b' that can be viewed as a guess for b. The advantage of adversary

Game $\mathbf{G}^{\text{mu-kr}}_{\mathsf{AE}}(A)$	$\underline{\mathrm{E}(L,x)}$
$\bar{K} \leftarrow\!\!\$\ A^{\textsc{New},\textsc{Enc},\textsc{Vf},\mathrm{E},\mathrm{E}^{-1}}$	If $T[L,x] = \perp$ then
Return $(\bar{K} \in \{K_1,\ldots,K_v\})$	$\quad T[L,x] \leftarrow\!\!\$\ \overline{\operatorname{im} T[L,\cdot]}$
$\underline{\textsc{New}()}$	$\quad T^{-1}[L,T[L,x]] \leftarrow x$
$v \leftarrow v+1$; $K_v \leftarrow\!\!\$\ \{0,1\}^{\mathsf{AE.kl}}$	Return $T[L,x]$
$\underline{\textsc{Enc}(i,N,M,H)}$	$\underline{\mathrm{E}^{-1}(L,y)}$
If not $(1 \le i \le v)$ then return $\perp$	If $T^{-1}[L,y] = \perp$ then
If $((i,N) \in U)$ then return $\perp$	$\quad T^{-1}[L,y] \leftarrow\!\!\$\ \overline{\operatorname{im} T^{-1}[L,\cdot]}$
$C \leftarrow \mathsf{AE.Enc}^{\mathrm{E},\mathrm{E}^{-1}}(K_i,N,M,H)$	$\quad T[L,T^{-1}[L,y]] \leftarrow y$
$U \leftarrow U \cup \{(i,N)\}$	Return $T^{-1}[L,y]$
Return C	
$\underline{\textsc{Vf}(i,N,C,H)}$	
If not $(1 \le i \le v)$ then return $\perp$	
$M \leftarrow \mathsf{AE.Dec}^{\mathrm{E},\mathrm{E}^{-1}}(K_i,N,C,H)$	
Return $(M \neq \perp)$	

Figure 2: Game defining multi-user key-recovery security of symmetric encryption scheme AE in the ideal-cipher model.

A in breaking the mu-ind security of AE is defined as $\mathsf{Adv}^{\text{mu-ind}}_{\mathsf{AE}}(A) = 2\Pr[\mathbf{G}^{\text{mu-ind}}_{\mathsf{AE}}(A)] - 1$.

The ideal-cipher oracles E and E^{-1} are given to the adversary, the encryption algorithm and the decryption algorithm, where the inputs are $L \in \{0,1\}^{\mathsf{AE.ckl}}$ and $x, y \in \{0,1\}^{\mathsf{AE.bl}}$. The oracles are defined using lazy sampling. The description of game $\mathbf{G}^{\text{mu-ind}}_{\mathsf{AE}}$ in Fig. 1 uses some notation that we introduce here and use also elsewhere. First of all, $T[\cdot,\cdot]$ describes a map $\{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ that is initially $\perp$ everywhere, with new values defined during the game. By $\operatorname{im} T[\cdot,\cdot]$ we denote the set $\{z \in \{0,1\}^* : \exists x,y \in \{0,1\}^* \text{ with } T[x,y] = z\}$ and by $\operatorname{supp} T[\cdot,\cdot]$ the set $\{(x,y) \in \{0,1\}^* \times \{0,1\}^* : T[x,y] \neq \perp\}$. Both terms are also used in the obvious sense in settings where one of the inputs is fixed. (In Fig. 1, this input is L.) Finally, for a subset $A \subset B$, the notation $\overline{A}$ refers to the complement $B \setminus A$ in B. We use this notation in places when the superset B is clear from the context. (In Fig. 1, the set B is $\{0,1\}^{\mathsf{AE.bl}}$.)

Definitions of mu security for authenticated encryption in the standard model are obtained as a special case, namely by restricting attention to schemes and adversaries that do not make use of the E and E^{-1} oracles.

One can further strengthen the security of the above ind definition by considering *nonce-misuse resistance* as defined by Rogaway and Shrimpton [31]. This requires changing the condition $(i,N) \in U$ in oracle Enc to only prevent queries where nonce *and* message (or even nonce, message, and header) are repeated. We do not use such a stronger definition in this work because GCM does not achieve it.

We say that an adversary is passive if it makes no queries to its Vf oracle. In some cases we will get better bounds for passive adversaries.

Rogaway's definition of indistinguishability security for authenticated encryption (in the su setting) [27] gives the adversary a decryption oracle, while we give it a verification oracle. The latter is simpler and our definition can be shown equivalent to one with a decryption oracle by the technique of BN [5].

$\underline{\mathsf{CAU.Enc}^{\mathrm{E},\mathrm{E}^{-1}}(K, N, M, H)}$	$\underline{\mathsf{CAU.Dec}^{\mathrm{E},\mathrm{E}^{-1}}(K, N, T\|\|C, H)}$
$\ell \leftarrow \lceil \lvert M\rvert/\lambda \rceil$	$\ell \leftarrow \lceil \lvert M\rvert/\lambda \rceil - 1$
$M_1\| \ldots \|M_\ell \leftarrow M$ // block length λ	$C_1\| \ldots \|C_\ell \leftarrow C$ // block length λ
$r \leftarrow \lvert M_\ell\rvert$ // last block length	$r \leftarrow \lvert C_\ell\rvert$ // last block length
$G \leftarrow \mathrm{E}(K, 0^\lambda)$; $Y \leftarrow N\|0^{\lambda-\nu-1}1$	$G \leftarrow \mathrm{E}(K, 0^\lambda)$; $Y \leftarrow N\|0^{\lambda-\nu-1}1$
For $i = 1$ to $\ell - 1$	$T' \leftarrow \mathsf{H}(G, H, C) \oplus \mathrm{E}(K, Y)$
$\quad C_i \leftarrow M_i \oplus \mathrm{E}(K, Y + i)$	If $T \neq T'$ then return $\perp$
$C_\ell \leftarrow M_\ell \oplus \mathrm{msb}_r(\mathrm{E}(K, Y + \ell))$	For $i = 1$ to $\ell - 1$
$C \leftarrow C_1\| \ldots \|C_\ell$	$\quad M_i \leftarrow C_i \oplus \mathrm{E}(K, Y + i)$
$T \leftarrow \mathsf{H}(G, H, C) \oplus \mathrm{E}(K, Y)$	$M_\ell \leftarrow C_\ell \oplus \mathrm{msb}_r(\mathrm{E}(K, Y + \ell))$
Return $T\|C$	Return $M_1\| \ldots \|M_\ell$

Figure 3: Encryption scheme $\mathsf{CAU} = \mathbf{CAU}[\mathsf{H}, \kappa, \lambda, \nu]$. **Left:** Encryption algorithm CAU.Enc. **Right:** Decryption algorithm CAU.Dec.

Key-Recovery Security. The qualitatively weaker requirement of key-recovery security can sometimes be established with better bounds than ind, which is of practical importance since violating key recovery is much more damaging that violating ind. The formalization is based on game $\mathbf{G}^{\text{mu-kr}}_{\mathsf{AE}}(A)$ of Fig. 2, associated to encryption scheme AE and adversary A. The goal of the adversary A is simply to output the key of any honest user. It again has access to oracles New, Enc, Vf, E, and E^{-1}. Oracles Enc and Vf are defined to always return the values as determined by the scheme AE. Adversary A wins if it outputs any one of the keys that was generated using the New oracle. The advantage of A in breaking the mu-kr security of AE is defined as $\mathsf{Adv}^{\text{mu-kr}}_{\mathsf{AE}}(A) = \Pr[\mathbf{G}^{\text{mu-kr}}_{\mathsf{AE}}(A)]$.

4 The Schemes

We present a symmetric encryption scheme we call CAU, for Counter-Mode with a AXU hash function. GCM is a special case. This allows us to divorce our results and analyses from some details of GCM (namely, the particular, polynomial-evaluation based hash function) making them both simpler and more general.

The TLS Working Group introduced a specific usage mode of GCM in recent draft versions of TLS 1.3 in which material, obtained in the handshake key derivation phase, is used to mask the nonce. We take a different perspective and view this as a new symmetric encryption scheme whose generalized version we specify here as RCAU. Finally we specify XCAU, our own variant that better achieves the same goals.

CAU. Let $\kappa, \lambda, \nu \geq 1$ be integers such that $\nu \leq \lambda - 2$, where κ is referred to as the *cipher key length*, λ as the *block length* and ν as the *nonce length*. Let $\mathsf{H}\colon \{0,1\}^\lambda \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^\lambda$ be an ϵ-XOR universal hash function. We associate to these the symmetric encryption scheme $\mathsf{CAU} = \mathbf{CAU}[\mathsf{H}, \kappa, \lambda, \nu]$ —here **CAU** is a transform taking $\mathsf{H}, \kappa, \lambda, \nu$ and returning a symmetric encryption scheme that we are denoting CAU— whose encryption and decryption algorithms are specified in Fig. 3. The scheme has key length $\mathsf{CAU.kl} = \kappa$, cipher key length $\mathsf{CAU.ckl} = \kappa$ and block length $\mathsf{CAU.bl} = \lambda$. It has nonce space $\mathsf{CAU.NS} = \{0,1\}^\nu$ and ciphertext length function $\mathsf{CAU.cl}(\cdot)$ defined by $\mathsf{CAU.cl}(m) = m + \lambda$. Explanations follow.

The algorithms CAU.Enc and CAU.Dec are given access to oracles that represent a cipher $\mathrm{E}\colon \{0,1\}^\kappa \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ and its inverse E^{-1}. In the security games the cipher will be chosen

$\underline{\mathsf{RCAU.Enc}^{\mathrm{E},\mathrm{E}^{-1}}(K\|L, N, M, H)}$	$\underline{\mathsf{RCAU.Dec}^{\mathrm{E},\mathrm{E}^{-1}}(K\|L, N, T\|C, H)}$
$\ell \leftarrow \lceil \|M\|/\lambda \rceil$	$\ell \leftarrow \lceil \|M\|/\lambda \rceil - 1$
$M_1\| \ldots \|M_\ell \leftarrow M$ // block length λ	$C_1\| \ldots \|C_\ell \leftarrow C$ // block length λ
$r \leftarrow \|M_\ell\|$ // last block length	$r \leftarrow \|C_\ell\|$ // last block length
$G \leftarrow \mathrm{E}(K, 0^\lambda)$; $Y \leftarrow (N \oplus L)\|0^{\lambda-\nu-1}1$	$G \leftarrow \mathrm{E}(K, 0^\lambda)$; $Y \leftarrow (N \oplus L)\|0^{\lambda-\nu-1}1$
For $i = 1$ to $\ell - 1$	$T' \leftarrow \mathsf{H}(G, H, C) \oplus \mathrm{E}(K, Y)$
$\quad C_i \leftarrow M_i \oplus \mathrm{E}(K, Y + i)$	If $T \neq T'$ then return $\perp$
$C_\ell \leftarrow M_\ell \oplus \mathrm{msb}_r(\mathrm{E}(K, Y + \ell))$	For $i = 1$ to $\ell - 1$
$C \leftarrow C_1\| \ldots \|C_\ell$	$\quad M_i \leftarrow C_i \oplus \mathrm{E}(K, Y + i)$
$T \leftarrow \mathsf{H}(G, H, C) \oplus \mathrm{E}(K, Y)$	$M_\ell \leftarrow C_\ell \oplus \mathrm{msb}_r(\mathrm{E}(K, Y + \ell))$
Return $T\|C$	Return $M_1\| \ldots \|M_\ell$

Figure 4: Encryption scheme $\mathsf{RCAU} = \mathbf{RCAU}[\mathsf{H}, \kappa, \lambda, \nu]$. **Left:** Encryption algorithm RCAU.Enc. **Right:** Decryption algorithm RCAU.Dec.

---

at random, meaning be ideal. In practice, it will be instantiated by a block cipher, usually AES.

CAU is an encrypt-then-mac scheme [5]. Encryption is counter-mode of the block cipher. The MAC is a Carter-Wegman MAC based on the AXU function family H. Some optimizations are performed over and above generic encrypt-then-mac to use the same key for both parts. The name stands for "Counter Almost Universal."

In the description of Fig. 3, the plaintext M is first partitioned into $\ell = \lceil |M|/\lambda \rceil$ plaintext blocks $M_1, \ldots, M_\ell$. The first $\ell - 1$ blocks have length λ. The final block M_ℓ has length $1 \leq r \leq \lambda$. The value G defined as $\mathrm{E}(K, 0^\lambda)$ is later used as a key for the hash function H. The loop then computes the counter mode encryption. Here and in the rest of the paper we use the following notation. If Z is a λ bit string and $j \geq 0$ is an integer then we let

$$Z + j = Z[1..\nu]\|\langle 1 + j \rangle \tag{1}$$

where $\langle 1 + j \rangle$ is the representation of the integer $(1 + j) \bmod 2^{\lambda-\nu}$ as a $(\lambda - \nu)$-bit string. Thus, in the scheme, $Y + i = N\|\langle 1 + i \rangle$. Function msb_n, which is needed to compute the final and possibly incomplete ciphertext block C_ℓ, maps a string of length $\geq n$ to its n-bit prefix. The final step in the scheme is then to compute the function H on H and $C = C_1\| \ldots \|C_\ell$ and xor it to the output of the block cipher on input Y. To simplify the technical descriptions in our proofs, we define the ciphertext as consisting of the tag prepended to the output of the counter-mode encryption.

GCM, as proposed by McGrew and Viega [23] and standardized by NIST [11], is obtained by instantiating the block cipher with AES, so that $\lambda = \kappa = 128$. The nonce length (in the standardized version) is $\nu = 96$. The hash function H is based on polynomial evaluation. The specifics do not matter for us. For our security analysis, all we need is that H is an ϵ-almost XOR-universal hash function (according to our definition of Section 2) for some $\epsilon\colon \mathbb{N} \times \mathbb{N} \to [0, 1]$. McGrew and Viega [23, Lemma 2] show that H has this property for $\epsilon(m, n) = (\lceil m/\lambda \rceil + \lceil n/\lambda \rceil + 1)/2^\lambda$.

CAU has fixed-length nonces, reflecting the standardized version of GCM in which $\nu = 96$. While the original scheme allows variable-length nonces [23], IOM [17] showed that the original security proof was flawed for variable-length nonces and the claimed security bounds did not hold.

RCAU. The TLS Working Group introduced a specific usage mode of GCM in recent draft versions of TLS 1.3 to prevent the scheme from evaluating the block cipher on the same inputs in each session. This countermeasure is described as computing an additional ν bits of key material in the key derivation phase, and using these to mask the ν-bit nonce given to GCM.

$\underline{\mathsf{XCAU.Enc}^{\mathrm{E},\mathrm{E}^{-1}}(K\|L,N,M,H)}$	$\underline{\mathsf{XCAU.Dec}^{\mathrm{E},\mathrm{E}^{-1}}(K\|L,N,T\|C,H)}$
$\ell \leftarrow \lceil \|M\|/\lambda \rceil$	$\ell \leftarrow \lceil \|M\|/\lambda \rceil - 1$
$M_1\| \ldots \|M_\ell \leftarrow M$ // block length λ	$C_1\| \ldots \|C_\ell \leftarrow C$ // block length λ
$r \leftarrow \|M_\ell\|$ // last block length	$r \leftarrow \|C_\ell\|$ // last block length
$G \leftarrow L \oplus \mathrm{E}(K,L)$; $Y \leftarrow N\|0^{\lambda-\nu-1}1$	$G \leftarrow L \oplus \mathrm{E}(K,L)$; $Y \leftarrow N\|0^{\lambda-\nu-1}1$
For $i = 1$ to $\ell - 1$	$T' \leftarrow \mathsf{H}(G,H,C) \oplus L \oplus \mathrm{E}(K, L \oplus Y)$
$\quad C_i = M_i \oplus L \oplus \mathrm{E}(K, L \oplus (Y+i))$	If $T \neq T'$ then return $\perp$
$C_\ell \leftarrow M_\ell \oplus \mathrm{msb}_r(L \oplus \mathrm{E}(K, L \oplus (Y+\ell)))$	For $i = 1$ to $\ell - 1$
$C \leftarrow C_1\| \ldots \|C_\ell$	$\quad M_i \leftarrow C_i \oplus L \oplus \mathrm{E}(K, L \oplus (Y+i))$
$T \leftarrow \mathsf{H}(G,H,C) \oplus L \oplus \mathrm{E}(K, L \oplus Y)$	$M_\ell \leftarrow C_\ell \oplus \mathrm{msb}_r(L \oplus \mathrm{E}(K, L \oplus (Y+\ell)))$
Return $T\|C$	Return $M_1\| \ldots \|M_\ell$

Figure 5: Encryption scheme $\mathsf{XCAU} = \mathbf{XCAU}[\mathsf{H}, \kappa, \lambda, \nu]$. **Left:** Encryption algorithm XCAU.Enc. **Right:** Decryption algorithm XCAU.Dec.

In order to analyze the effectiveness of this countermeasure, we take a different perspective, casting the method as specifying a new symmetric encryption scheme in which the mask becomes part of the key. Formally, as before, let $\kappa, \lambda, \nu \geq 1$ be integers representing the cipher key length, block length and nonce length, where $\nu \leq \lambda - 2$. Let $\mathsf{H}\colon \{0,1\}^\lambda \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^\lambda$ be an ϵ-XOR universal hash function. We associate to these the symmetric encryption scheme $\mathsf{RCAU} = \mathbf{RCAU}[\mathsf{H}, \kappa, \lambda, \nu]$ whose encryption and decryption algorithms are specified in Fig. 4. The scheme has key length $\mathsf{RCAU.kl} = \kappa + \nu$, cipher key length $\mathsf{RCAU.ckl} = \kappa$ and block length $\mathsf{RCAU.bl} = \lambda$. It has nonce space $\mathsf{RCAU.NS} = \{0,1\}^\nu$ and ciphertext length function $\mathsf{RCAU.cl}(\cdot)$ defined by $\mathsf{RCAU.cl}(m) = m + \lambda$. Note that the key length is $\kappa + \nu$, while that of CAU was κ. The definition of $Y + i$ is as per Equation (1), so $Y + i = (N \oplus L)\|\langle 1 + i \rangle$.

XCAU. We suggest a different scheme to achieve the multi-user security goal targeted by RCAU. Recall that if $\mathrm{E}\colon \{0,1\}^\kappa \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ is a block cipher than $\mathrm{EX}\colon \{0,1\}^{\kappa+\lambda} \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ is the block cipher defined by $\mathrm{EX}(K\|L, X) = L \oplus \mathrm{E}(K, L \oplus X)$. This can be viewed as strengthening E using an Even-Mansour technique [12]. This was suggested by Rivest as a key-extension method for DES and first analyzed by Kilian and Rogaway [18]. We then simply use EX in place of E in the basic CAU. Formally, as before, let $\kappa, \lambda, \nu \geq 1$ be integers representing the cipher key length, block length and nonce length, where $\nu \leq \lambda - 2$. Let $\mathsf{H}\colon \{0,1\}^\lambda \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^\lambda$ be an ϵ-XOR universal hash function. We associate to these the symmetric encryption scheme $\mathsf{XCAU} = \mathbf{XCAU}[\mathsf{H}, \kappa, \lambda, \nu]$ whose encryption and decryption algorithms are specified in Fig. 5. The scheme has key length $\mathsf{XCAU.kl} = \kappa + \lambda$, cipher key length $\mathsf{XCAU.ckl} = \kappa$ and block length $\mathsf{XCAU.bl} = \lambda$. It has nonce space $\mathsf{XCAU.NS} = \{0,1\}^\nu$ and ciphertext length function $\mathsf{XCAU.cl}(\cdot)$ defined by $\mathsf{XCAU.cl}(m) = m + \lambda$. Note that the key length is $\kappa + \lambda$, while that of RCAU was $\kappa + \nu$. The definition of $Y + i$ is as per Equation (1), so $Y + i = N\|\langle 1 + i \rangle$.

Our analysis of this scheme builds on the work of Kilian and Rogaway, but analyzes the construction directly in the multi-user setting. We believe that the bounds can be further improved along the lines of Mouha and Luykx's work [24], but this does not affect the terms we are most interested in.

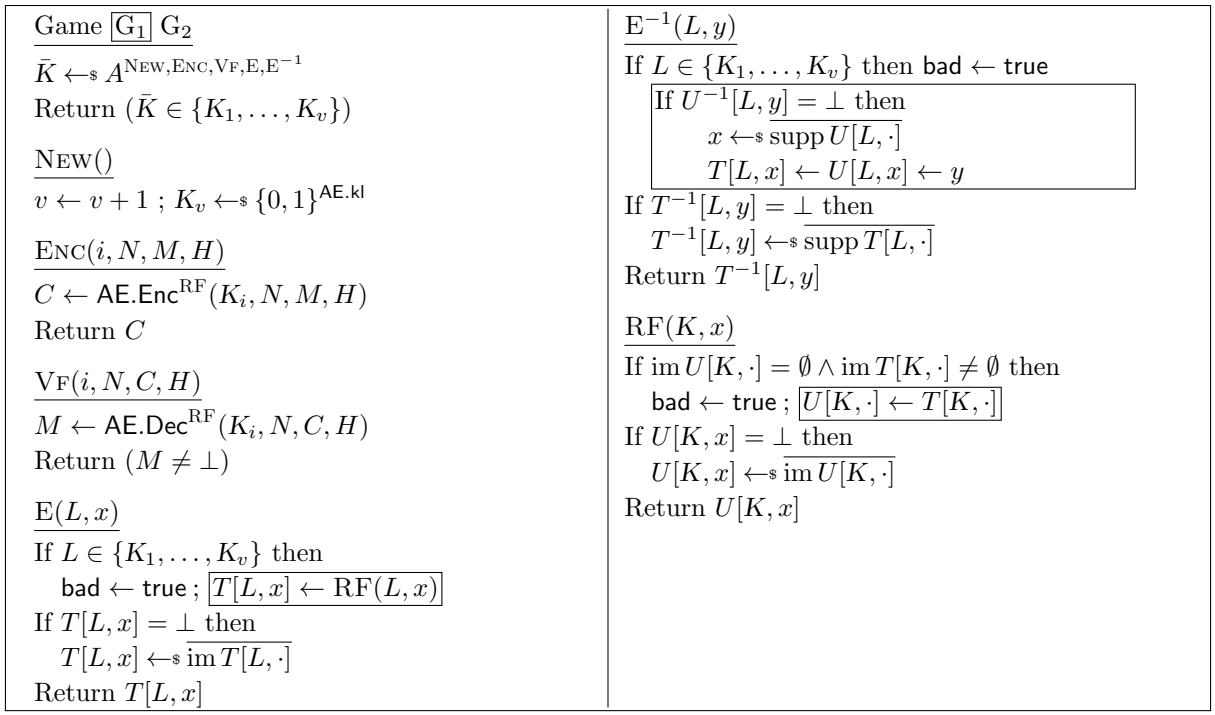

Figure 6: Intermediate games for decoupling the oracles $\mathrm{E}/\mathrm{E}^{-1}$ and RF in the proof of Theorem 5.1.

5 Key-Recovery Security

The multi-user security differences between the schemes are most easily seen in the case of security against key recovery, so we start there.

5.1 Security of CAU

We show that the multi-user kr advantage scales linearly in the number of adversarial evaluations of the ideal cipher (corresponding to offline evaluations of the blockcipher in practice) and the number of user instances. We give both an upper bound (security proof) and lower bound (attack) on the kr-advantage to show this, beginning with the former.

Theorem 5.1 *Let $\kappa, \lambda, \nu \geq 1$ be such that $\nu \leq \lambda - 2$. Let $\mathsf{H}\colon \{0,1\}^\lambda \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^\lambda$ be a family of functions. Let $\mathsf{CAU} = \mathbf{CAU}[\mathsf{H}, \kappa, \lambda, \nu]$. Let A be an adversary that makes at most u queries to its* New *oracle and p queries to its* E *and* E^{-1} *oracles. Then*

$$\mathsf{Adv}^{\text{mu-kr}}_{\mathsf{CAU}}(A) \leq \frac{u(p+1)}{2^\kappa}.$$

Proof: We use the code-based game-playing technique of BR [6]. Without loss of generality, we assume that the adversary A does not input invalid user identifiers $i \notin \{1, \ldots, v\}$ to Enc or Vf, and does not re-use nonces in encryption queries. We also assume that A does not verify correct ciphertexts they obtained from Enc at its Vf oracle. These restrictions allow us to simplify the descriptions of the games, and any arbitrary adversary A can be translated into an adversary A' that adheres to these restrictions and makes at most the same number of queries as A. Our proof proceeds in a sequence of games.

Adversary A_{u,p_e}

For $i = 1$ to u do

  NEW ; $C_i \leftarrow \text{ENC}(i, 0^\nu, 0^{2\lambda}, \varepsilon)$

For $j = 1$ to $p_e/2$ do

  $y \leftarrow \text{E}([j]_\lambda, 0^\nu \| 0^{\lambda-\nu-2} \| 10)$ // $[j]_\lambda$ is the encoding of integer j as a λ-bit string

  $y' \leftarrow \text{E}([j]_\lambda, 0^\nu \| 0^{\lambda-\nu-2} \| 11)$

  If $(\exists i : C_i[(\lambda+1)..2\lambda] = y$ and $C_i[(2\lambda+1)..3\lambda] = y')$ then return $[j]_\lambda$

Figure 7: Adversary A_{u,p_e} used in Theorem 5.2.

The first step in the proof is to rewrite game $\mathbf{G}^{\text{mu-kr}}_{\mathsf{CAU}}(A)$ syntactically by introducing an additional oracle RF that implements the forward evaluation of the ideal cipher for the algorithms CAU.Enc and CAU.Dec. This is sufficient as encryption and decryption in CAU never query E^{-1}. We call this game G_0, but do not explicitly describe as it is obtained easily from $\mathbf{G}^{\text{mu-kr}}_{\mathsf{CAU}}(A)$.

We then rewrite the game in the form of G_1, which is described in Fig. 6 and basically obtained by a syntactic modification of the oracles E, E^{-1}, and RF. In more detail, oracle RF samples the ideal cipher for the keys used in the encryption using the map $U[\cdot,\cdot]$. The oracles E and E^{-1} are adapted such that, for keys used in the game, they sample the map $T[\cdot,\cdot]$ consistently with $U[\cdot,\cdot]$. We introduce a flag bad that is set when the adversary A queries one of the oracles E or E^{-1} with a key that is also used in the oracle RF.

The next game G_2 modifies the way in which the responses for the E, E^{-1}, and RF oracles are determined. In particular, we break the consistency between E and E^{-1} on the one hand, and RF on the other hand, by sampling the oracle responses independently. Since all changes appear only after bad has been set, we can relate the games using the Fundamental Lemma from Bellare and Rogaway [6] and proceed by bounding the probability of setting bad. This probability is in fact bounded by $up/2^\kappa$. As all computations while bad is not set are independent of the values $K_1, \ldots, K_u$, the maximal probability of the adversary to guess one of these uniformly random values is $u/2^\kappa$ in each of its p queries to E and E^{-1}.

The keys in G_2 only serve as labels, the game is independent of their actual values. The only remaining step is to compute the probability of guessing any one of the u keys that are chosen at random without collision, which is also incorporated into the advantage. In more detail:

$$\begin{aligned}\mathsf{Adv}^{\text{mu-kr}}_{\mathsf{CAU}}(A) &= \Pr\left[\mathbf{G}^{\text{mu-kr}}_{\mathsf{CAU}}(A)\right] = \Pr\left[\text{G}_0\right] = \Pr\left[\text{G}_1\right] \\ &\leq \Pr\left[\text{G}_2\right] + \frac{up}{2^\kappa} \leq \frac{u}{2^\kappa} + \frac{up}{2^\kappa} = \frac{u(p+1)}{2^\kappa},\end{aligned}$$

which concludes the proof. ∎

Next we show that the security bound proven in Theorem 5.1 is (almost) tight. We describe an attack (adversary) that achieves the described bound up to a (for realistic parameters small) factor. The adversary is shown in Fig. 7. It is parameterized by a number u of users and an (even) number p_e of queries to E. It first encrypts a short message $0^{2\lambda}$ for each of the u users. Next, it queries the E oracle on the value $0^{\lambda-2}10$, the first block that is used for masking actual plaintext, for up to p_e different keys. As soon as it finds a matching key L for the first block, it simply evaluates $\text{E}(L, 0^{\lambda-2}11)$ and checks for consistency with the second block. If the check succeeds, the adversary outputs the key L, otherwise it tries further keys.

The described attack strategy extends to any application of CAU in which the nonces used in the scheme are the same in each session. As TLS 1.3 uses the sequence number to compute the nonces, a version without the nonce randomization technique would be susceptible to this attack.

Theorem 5.2 *Let $\kappa, \lambda, \nu \geq 1$ be such that $\nu \leq \lambda - 2$. Let $\mathsf{H}\colon \{0,1\}^\lambda \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^\lambda$ be a family of functions. Let $\mathsf{CAU} = \mathbf{CAU}[\mathsf{H}, \kappa, \lambda, \nu]$. Let $u \geq 1$ be an integer and $p_e \geq 2$ an even integer. Associate to them the adversary A_{u,p_e} described in Fig. 7, which makes u queries to* New*, $q_e = u$ queries to* Enc *of length 2λ bits, no queries to* Vf*, p_e queries to* E*, and no queries to* E^{-1}*. Then*

$$\mathsf{Adv}^{\text{mu-kr}}_{\mathsf{CAU}}(A_{u,p_e}) \geq \mu \cdot up_e/2^{\kappa+1},$$

where

$$\mu = \left(1 - \frac{u(u-1)}{2^{\kappa+1}}\right) \cdot \left(1 - \frac{u(2^\kappa - u)}{2^\lambda(2^\lambda - 1)}\right) .$$

This means that the success probability of A_{u,p_e} scales (almost) linearly with the number of users. The proof we give can be improved in terms of tightness, for instance, we allow the attack to completely fail if only a single collision occurs between honest users' keys. In particular the factor $(1 - u(u-1)/2^{\kappa+1})$ could be improved especially for large u.

Proof of Theorem 5.2: The probability for any of the $u = q_e$ keys to collide is at most $u(u-1)/2^{\kappa+1}$. In the subsequent steps we compute the probabilities based on the assumption that no user keys generated within New collide, which is correct with probability at least $1 - u(u-1)/2^{\kappa+1}$. In more detail, given that we have no collisions of user keys, the adversary uses at least $p_e/2$ attempts to guess any one of $u = q_e$ (uniformly random, without collision) keys from a set of size 2^κ. We still need to bound the probability of false positives, that is, keys that were not sampled in a New oracle but coincide with the block cipher outputs, and therefore lead to a wrong guess: The probability that the ideal cipher for a specific "wrong" key (out of $2^\kappa - u$) coincides with the ideal cipher for each of the u "correct" keys on both inputs $0^\nu \| 0^{\lambda-\nu-2} \| 10$ and $0^\nu \| 0^{\lambda-\nu-2} \| 11$ is $2^{-\lambda}(2^\lambda - 1)^{-1}$. The existence of such a colliding key can be bounded using the Union Bound to be at most $u(2^\kappa - u)/(2^\lambda(2^\lambda - 1))$, so the probability that no such collision exists is at least $1 - u(2^\kappa - u)/(2^\lambda(2^\lambda - 1))$. Overall, we obtain the stated bound. ∎

Evaluating the formula for realistic values for GCM in TLS 1.3, we set $\kappa = 128$. We allow the adversary to make $p_e = 2^{64}$ evaluations of the block cipher. We estimate the number of TLS sessions per day as 2^{40}, which leaves us a security margin of roughly 2^{24}. While this means that on expectation the attack still needs 2^{24} days to recover a single key, it is important to recall that this estimate is obtained under the strong assumption that AES behaves like an ideal cipher.

5.2 Security of RCAU

RCAU aims to avoid the attack strategy described in Section 5.1 by randomizing the nonce before it is used in the block cipher. Here we assess whether the measure succeeds, again first upper bounding adversary advantage via a proof, then lower bounding it via an attack.

In contrast to the bound for CAU, the bound for RCAU depends on more parameters. This is caused by the more intricate "decoupling" of the $\mathrm{E}/\mathrm{E}^{-1}$ and RF oracles.

Theorem 5.3 *Let $\kappa, \lambda, \nu \geq 1$ be such that $\nu \leq \lambda - 2$. Let $\mathsf{H}\colon \{0,1\}^\lambda \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^\lambda$ be a family of functions. Let $\mathsf{RCAU} = \mathbf{RCAU}[\mathsf{H}, \kappa, \lambda, \nu]$. Let A be an adversary that makes at most*

u queries to its NEW *oracle, q_e queries to its* ENC *oracle with messages of length at most ℓ_{bit} bits, q_v queries to its* VF *oracle of length at most $\ell_{\text{bit}} + \lambda$ bits, p_e queries to its* E *oracle, and p_i queries to its* E^{-1} *oracle. Then*

$$\mathsf{Adv}^{\text{mu-kr}}_{\mathsf{RCAU}}(A) \leq \frac{2up(\ell_{\text{blk}}(q_e+q_v)+1)}{2^{\kappa+\nu}} + \frac{up(\ell_{\text{blk}}(q_e+q_v)+1)}{2^{\kappa}(2^{\lambda}-p)} + \frac{up(\ell_{\text{blk}}(q_e+q_v)+1)}{2^{\kappa}(2^{\lambda}-q_e-q_v)} + \frac{p_i+u}{2^{\kappa}}, \quad (2)$$

where $\ell_{\text{blk}} = \lceil \ell_{\text{bit}}/\lambda \rceil + 1$.

Proof: As in Theorem 5.1, we restrict our attention to adversaries A that do not use invalid user identifiers, that do not re-use nonces, and that do not verify ciphertexts obtained from the ENC oracle. As in the proof of Theorem 5.1, we now aim at "decoupling" the oracles E/E^{-1} and RF, but this time we have to be cautious: we cannot just "give up" when the adversary "guesses" one of the users keys in calls to E/E^{-1}; this would ruin our bound. The first step is as above to introduce an auxiliary map $U[\cdot,\cdot]$ in addition to $T[\cdot,\cdot]$, but keep the maps synchronized. The change from $\mathbf{G}^{\text{mu-kr}}_{\mathsf{RCAU}}(A)$ to G_0 is therefore only syntactic. Intuitively, the lazy sampling of the block cipher is now performed using both maps, where $T[\cdot,\cdot]$ is filled in calls to E and E^{-1}, and $U[\cdot,\cdot]$ is filled in RF. The oracles make sure that the maps stay consistent.

In game G_1, described in detail in Fig. 8, we first change the way in which the responses are sampled, but still in an equivalent way, namely we first attempt to sample consistently only for $T[\cdot,\cdot]$ and then check for consistency with $U[\cdot,\cdot]$. If this fails, we set $\mathsf{bad} \leftarrow \mathsf{true}$ and re-sample with the correct distribution. Additionally, we set $\mathsf{bad} \leftarrow \mathsf{true}$ whenever we need to answer for either $T[\cdot,\cdot]$ or $U[\cdot,\cdot]$ and the answer is already defined by the respective other map. Game G_1 is equivalent to G_0. The proof is further complicated by the fact that RCAU derives the key for H as $\text{E}(K, 0^{\lambda})$ and this query is therefore not randomized. As a consequence, we have to treat the queries with value 0^{λ} independently of the other queries, and keep the maps $T[\cdot, 0^{\lambda}]$ and $U[\cdot, 0^{\lambda}]$ consistent for the next proof steps.

In game G_2 we modify the behavior of the oracles E, E^{-1}, and RF to not re-sample to avoid inconsistencies with the other oracles. Also, we do not enforce consistency between $T[K,\cdot]$ and $U[K,\cdot]$ for values that are defined already in one of the maps; we sample a fresh value in a map independently of whether the point is already defined in the other map. As both modifications occur only after the flag bad has been set, we can use the Fundamental Lemma to relate the advantages of an adversary in games G_1 and G_2.

To bound the probability for the flag bad to be set in games G_1 or G_2, respectively, we begin with the following observation: As long as bad is not set, each row $T[K,\cdot]$ or $U[K,\cdot]$ for a specific key K is sampled without collisions within this row, but independently of any other row, and also mutually independent between $T[K,\cdot]$ and $U[K,\cdot]$. This is the case because the only other way of defining a value for those maps is either re-sampling or copying from the other map; in both cases we set the flag bad. Furthermore, we observe that all operations that occur before the flag bad is set are independent of the actual values of the keys $K_1, \ldots, K_u$. Given these insights, we now analyze the probabilities for setting the bad flag at the different code points, first for E and E^{-1}:

- The probability of enforcing re-sampling in E or E^{-1} is analyzed as follows: For a particular key $\bar{K} \in \{K_1, \ldots, K_u\}$ for which m blocks have been defined through queries to RF, the probability of sampling a value that collides is at most $m/(2^{\lambda}-p)$, as we choose uniformly from

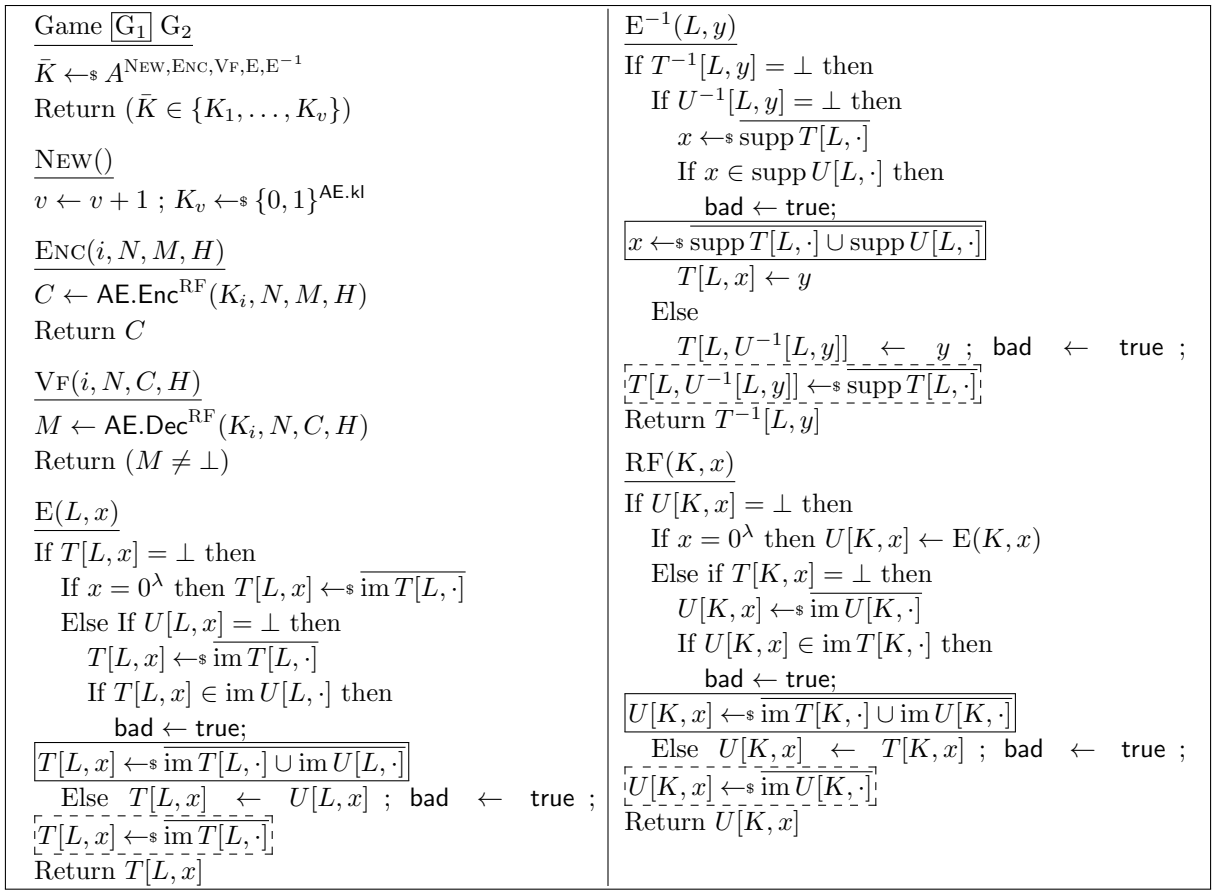

Figure 8: Intermediate games for decoupling the oracles $\mathrm{E}/\mathrm{E}^{-1}$ and RF in the proof of Theorem 5.3.

$2^\lambda - p$ values. The expected number of blocks for the key L in the query is $u(\ell_{\mathrm{blk}}(q_e+q_v)+1)/2^\kappa$, which leads to an overall probability of $u(\ell_{\mathrm{blk}}(q_e + q_v) + 1)/(2^\kappa(2^\lambda - p))$ for each query.

- The probability of enforcing that a value be copied (that is, the final "Else" statement becomes active) in E is bounded by $u(\ell_{\mathrm{blk}}(q_e+q_v)+1)/2^{\kappa+\nu}$ for each of the p queries. This is computed analogously to above: executing the "Else" statement means that the adversary guessed a combination of a κ-bit key and a ν-bit mask value.
- Finally, the probability for copying a value in E^{-1} is bounded by the term $1/2^{-\kappa}$. The reason is that it corresponds to guessing a the key for a specific user.

We obtain the bounds $up(\ell_{\mathrm{blk}}(q_e + q_v) + 1)/(2^\kappa(2^\lambda - p))$, $up_e(\ell_{\mathrm{blk}}(q_e + q_v) + 1)/2^{\kappa+\nu}$, and $p_i/2^{-\kappa}$ as the adversary makes at most p_e queries to E, p_i queries to E^{-1}, and $p = p_e + p_i$ queries accumulated.

We proceed by analyzing the probabilities for RF analogously:

- With respect to enforcing re-sampling in RF, for a key L for which m blocks have been defined, the probability of sampling a colliding value is $m/(2^\lambda - q_e - q_v)$. This leads to an overall probability of at most $p/(2^\kappa(2^\lambda - q_e - q_v))$.
- The probability of enforcing that a value be copied (that is, the final "Else" statement becomes active) in RF is bounded by $u(\ell_{\mathrm{blk}}(q_e + q_v) + 1)p/2^{\kappa+\nu}$. The reason is that for a particular

Adversary A_{p_i}

$\textsc{New}$; $C \leftarrow \textsc{Enc}(1, 0^\nu, 0^{2\lambda}, \varepsilon)$

For $j = 1$ to $p_i/2$ do

  $y \leftarrow \mathrm{E}^{-1}([j]_\lambda, C[(\lambda+1)..2\lambda])$ // $[j]_\lambda$ means encoding as λ-bit string

  If $\exists N \in \{0,1\}^\nu : y = N \| 0^{\lambda-\nu-2}10$ then

    If $\mathrm{E}^{-1}([j]_\lambda, C[(2\lambda+1)..3\lambda]) = N \| 0^{\lambda-\nu-2}10$ then

      Return $[j]_\lambda$

Figure 9: Adversary A_{p_i} used in Theorem 5.4.

key L for which m blocks have been defined through queries to E and E^{-1}, the probability that an query to RF as done by $\mathsf{CAU.Enc}^{\mathrm{RF}}$ uses the same input is bounded by $m/2^\nu$. This leads to a probability of $p/2^{\kappa+\nu}$.

Since the encryption and decryption algorithms overall make $u(\ell_{\mathrm{blk}}(q_e + q_v) + 1)$ queries to RF, we obtain the bounds $up(\ell_{\mathrm{blk}}(q_e + q_v) + 1)/(2^\kappa(2^\lambda - q_e - q_v))$ and $up(\ell_{\mathrm{blk}}(q_e + q_v) + 1)/2^{\kappa+\nu}$.

Finally, as in G_2 the oracles E and E^{-1} are independent of the oracle RF that is used in RCAU, the probability of guessing a key is $u/2^\kappa$. All these terms together comprise the bound in the theorem statement. ∎

For realistic parameters, the bound in Theorem 5.3 means that the "best" attack for passive adversaries is now the inversion of a block observed while eavesdropping. In contrast to the attack analyzed in Section 5.1, this attack does not scale in the mass surveillance scenario, because the adversary has to target one specific ciphertext block.

In more detail, the adversary strategy A analyzed in the below lemma and specified in detail in Fig. 9 proceeds as follows. First obtain an encryption of $0^{2\lambda}$ from an honest user. Then brute-force the key by decrypting the first ciphertext block using E^{-1}, checking whether the output satisfies the structure $N \| 0^{\lambda-\nu-2}10$. In case this structure is observed, verify the key by checking if the next block is consistent with an evaluation of E with the same key and plaintext $N \| 0^{\lambda-\nu-2}11$.

Since the described attack strategy applies independently of how the nonces are chosen (prior to the randomization) as long as the value is predictable, the lower bound also applies to the scheme as used in the latest draft of TLS 1.3.

Theorem 5.4 *Let $\kappa, \lambda, \nu \geq 1$ be such that $\nu \leq \lambda - 2$. Let H: $\{0,1\}^\lambda \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^\lambda$ be a family of functions. Let $\mathsf{RCAU} = \mathbf{RCAU}[\mathsf{H}, \kappa, \lambda, \nu]$. Let $p_i \geq 2$ an even integer and the adversary A_{p_i} as described in Fig. 9, which makes 1 query to each* $\textsc{New}$ *and* $\textsc{Enc}$ *(the latter of length 2λ bits), no queries to* $\textsc{Vf}$*, p_i queries to* E^{-1}*, and no queries to* E*. Then*

$$\mathsf{Adv}^{\text{mu-kr}}_{\mathsf{RCAU}}(A_{p_i}) \geq \mu \cdot p_i \cdot 2^{-\kappa-1},$$

with

$$\mu = 1 - \frac{(2^\kappa - 1)2^\nu}{2^\lambda(2^\lambda - 1)} .$$

Proof: Let K_1 be the key sampled during the invocation of $\textsc{New}$ in the game. The probability for the block cipher on a key $K \neq K_1$ to satisfy the first condition is $2^{\nu-\lambda}$, since in the first invocation of E^{-1} the value is sampled uniformly at random and $\lambda - \nu$ bits have to match. The second invocation

of E^{-1} has to lead to the correct outcome $N\|0^{\lambda-\nu-2}11$, the value is drawn uniformly at random from the remaining $2^\lambda - 1$ values not equal to the outcome of the first query. There are 2^κ keys, so by the Union Bound the probability of any key $K \neq K_1$ to lead to an admissible pattern on the first two blocks is bounded by $(2^\kappa - 1)2^\nu/(2^\lambda(2^\lambda - 1))$.

In the event that no key $K \neq K_1$ satisfies the above condition, this advantage of adversary A_{p_i} is simply the probability of guessing a uniformly random key of κ bits in $p_i/2$ attempts, as for each key A_{p_i} spends at most 2 queries. This completes the proof. ▮

The attack analyzed in Theorem 5.4 is considerably harder to mount than the one analyzed in Theorem 5.2, because the queries in the Theorem 5.2 attack can be preprocessed and apply to all observed communication sessions equally, whereas in the Theorem 5.4 attack the queries have to be made for a particular session under attack. Still, in the following Section 5.3, we show that at low computational cost for the honest parties, the Theorem 5.4 attack can be made considerably harder.

5.3 Security of XCAU

The term $p_i/2^\kappa$ in the bound for RCAU originates in the fact that only the input of the block cipher is masked, and inversion queries by the adversaries are not hindered. In the scheme XCAU, an advantage beyond the randomization of the input to derive the hash function key is that the output of the block cipher is masked, which restricts the power of inversion queries to the block cipher considerably.

Our analysis of XCAU is based on combining the analysis of DESX-like input and output whitening in a multi-user setting, and then prove the security of XCAU along the lines of Theorem 6.2. We first prove a multi-user bound for the DESX-like construction. The security goal is described by the games in Fig. 10.

Theorem 5.5 *Let A be an adversary that makes at most u queries to its* New*, q_2 queries to its* RF *oracle per user, p queries to its* E *oracle and* E^{-1} *oracles. Then*

$$|\Pr[\mathrm{R}(A)] - \Pr[\mathrm{S}(A)]| \;\leq\; \frac{u \cdot q_2 \cdot p}{2^{\lambda+\kappa+1}}.$$

Proof: We introduce two intermediate games G_0 and G_1 in Fig. 11. Game G_0 is equivalent to game $\mathrm{R}(A)$; the introduction of the additional map $U[\cdot,\cdot]$ is only syntactic as we make sure that it stays consistent with $T[\cdot,\cdot]$ throughout. We also modify the procedures for sampling new values for the maps $U[\cdot,\cdot]$ and $T[\cdot,\cdot]$ such that first we sample a new value such that it is consistent only with the respective map, then check whether it is consistent with the other map, and re-sample consistently if we determine that it is not. In G_1, the map $U[\cdot,\cdot]$ is completely independent of the map $T[\cdot,\cdot]$. Both G_0 and G_1 set the flag bad on occasions where the sampling creates inconsistencies between $U[\cdot,\cdot]$ and $T[\cdot,\cdot]$.

The probability of setting the bad flag in G_2 and G_3 can be bounded as follows. We first observe that besides the bad flag, G_3 is equivalent to S. For both G_2 and G_3, as long as bad is not set, all outputs are uniformly distributed among the values that are valid for the respective oracle and key. Moreover, xoring K_i' to all inputs or outputs modifies each concrete permutation; however, the distribution of a uniformly random permutation remains unchanged by this operation. Following the definition of Maurer [22], this means that both games G_2 and G_3 with the respective flags bad are *conditionally equivalent* to the game S. (In other words, *conditioned on* bad = false, the outputs of the games are distributed exactly as in S.)

Game R(A)	Game S(A)
$v \leftarrow 0$; $b \leftarrow\!\!\$\ A^{\text{New},\text{E},\text{E}^{-1},\text{RF}}$ Return b	$v \leftarrow 0$; $b \leftarrow\!\!\$\ A^{\text{New},\text{E},\text{E}^{-1},\text{RF}}$ Return b
New() $v \leftarrow v + 1$ $(K_v, K'_v) \leftarrow\!\!\$\ \{0,1\}^\kappa \times \{0,1\}^\lambda$	New() $v \leftarrow v + 1$ $K_v \leftarrow\!\!\$\ \{0,1\}^\kappa$

RF(i, x)
If $T[K_i, x \oplus K'_i] = \perp$ then
  $T[K_i, x \oplus K'_i] \leftarrow\!\!\$\ \overline{\text{im}\, T[K_i, \cdot]}$
Return $T[K_i, x \oplus K'_i] \oplus K'_i$
	RF(i, x)
	If $i \le v$ and $U[K_i, x] = \perp$ then
	  $U[K_i, x] \leftarrow\!\!\$\ \overline{\text{im}\, U[K_i, \cdot]}$
	Return $U[K_i, x]$

E(L, x)
If $T[L, x] = \perp$ then
  $T[L, x] \leftarrow\!\!\$\ \overline{\text{im}\, T[L, \cdot]}$
Return $T[L, x]$
	E(L, x)
	If $T[L, x] = \perp$ then
	  $T[L, x] \leftarrow\!\!\$\ \overline{\text{im}\, T[L, \cdot]}$
	Return $T[L, x]$

$\text{E}^{-1}(L, y)$
If $T^{-1}[L, y] = \perp$ then
  $x \leftarrow\!\!\$\ \overline{\text{supp}\, T[L, \cdot]}$
  $T[L, x] \leftarrow y$
Return $T^{-1}[L, y]$
	$\text{E}^{-1}(L, y)$
	If $T^{-1}[L, y] = \perp$ then
	  $x \leftarrow\!\!\$\ \overline{\text{supp}\, T[L, \cdot]}$
	  $T[L, x] \leftarrow y$
	Return $T^{-1}[L, y]$

Figure 10: Multi-user security for block-cipher key extension. **Left:** Game giving the adversary access to the actual construction. **Right:** Game giving the adversary access to an independent ideal cipher.

Subsequently, we can employ Maurer's result [22, Theorem 1] to bound the distinguishing advantage between G_2 and G_3 by the advantage of the best *non-adaptive* distinguisher. As the adversary makes at most p queries to its E and E^{-1} oracles, and uq_2 queries to its RF oracle, there are $u \cdot q_2 \cdot p$ possible combinations of queries that may provoke the flag bad to be set, and each case appears with probability $2^{-\lambda-\kappa-1}$. We conclude the proof via the Union Bound. ▌

Analogously to the previous results on CAU and RCAU, we now analyze the key-recovery security of XCAU.

Theorem 5.6 *Let $\kappa, \lambda, \nu \ge 1$ be such that $\nu \le \lambda - 2$. Let $\mathsf{H}\colon \{0,1\}^\lambda \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^\lambda$ be a family of functions. Let $\mathsf{XCAU} = \mathbf{XCAU}[\mathsf{H}, \kappa, \lambda, \nu]$. Let A be an adversary that makes at most u queries to its* New *oracle, q_e queries to its* Enc *oracle with messages of length at most ℓ_{bit} bits, q_v queries to its* Vf *oracle with messages of length at most $\ell_{\text{bit}} + \lambda$ bits, and p queries to its* E *and* E^{-1} *oracles. Assume furthermore that $q_e \le 2^\nu$, and $\ell_{\text{bit}} \le \lambda(2^{\lambda-\nu} - 2)$. Then, with $\ell_{\text{blk}} = \lceil \ell_{\text{bit}}/\lambda \rceil + 1$,*

$$\mathsf{Adv}^{\text{mu-kr}}_{\mathsf{XCAU}}(A) \le \frac{up(\ell_{\text{blk}}(q_e + q_v) + 1)}{2^{\lambda+\kappa+1}} + \frac{u}{2^\kappa}.$$

Proof: As in Theorem 5.1 and 5.3, we restrict our attention to adversaries A that do not use invalid user identifiers, that do not re-use nonces, and that do not verify ciphertexts obtained from

Game $\boxed{\mathrm{G}_0}$ $\lceil\mathrm{G}_1\rceil$

$v \leftarrow 0$; $b \leftarrow\!\!\$\ A^{\textsc{New},\mathrm{E},\mathrm{E}^{-1},\mathrm{RF}}$

Return b

$\underline{\textsc{New}}$

$v \leftarrow v + 1$

$(K_v, K'_v) \leftarrow\!\!\$\ \{0,1\}^\kappa \times \{0,1\}^\lambda$

$\underline{\mathrm{RF}(i,x)}$

If $U[K_i, x \oplus K'_i] = \perp$ then

  If $T[K_i, x \oplus K'_i] = \perp$ then

    $U[K_i, x \oplus K'_i] \leftarrow\!\!\$\ \overline{\mathrm{im}\, U[K_i, \cdot]}$

    If $U[K_i, x \oplus K'_i] \in \mathrm{im}\, T[K_i, \cdot]$ then

      $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{U[K_i, x \oplus K'_i] \leftarrow\!\!\$\ \overline{\mathrm{im}\, U[K_i, \cdot] \cup \mathrm{im}\, T[K_i, \cdot]}}$

  Else

    $U[K_i, x \oplus K'_i] \leftarrow T[K_i, x \oplus K'_i]$; $\mathsf{bad} \leftarrow \mathsf{true}$

    $\lceil U[K_i, x \oplus K'_i] \leftarrow\!\!\$\ \overline{\mathrm{im}\, U[K_i, \cdot]} \rceil$

Return $U[K_i, x \oplus K'_i] \oplus K'_i$

$\underline{\mathrm{E}(L,x)}$

If $T[L,x] = \perp$ then

  If $U[L,x] = \perp$ then

    $T[L,x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[L, \cdot]}$

    If $T[L,x] \in U[L, \cdot]$ then

      $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{T[L,x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[L, \cdot] \cup \mathrm{im}\, U[L, \cdot]}}$

  Else $T[L,x] \leftarrow U[L,x]$; $\mathsf{bad} \leftarrow \mathsf{true}$; $\lceil T[L,x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[L, \cdot]} \rceil$

Return $T[L,x]$

$\underline{\mathrm{E}^{-1}(L,y)}$

If $T^{-1}[L,y] = \perp$ then

  If $U^{-1}[L,y] = \perp$ then

  $x \leftarrow\!\!\$\ \overline{\mathrm{supp}\, T[L, \cdot]}$

    If $x \in \mathrm{supp}\, U[L, \cdot]$ then

      $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{x \leftarrow\!\!\$\ \overline{\mathrm{supp}\, T[L, \cdot] \cup \mathrm{supp}\, U[L, \cdot]}}$

  Else $x \leftarrow U^{-1}[L,y]$; $\mathsf{bad} \leftarrow \mathsf{true}$; $\lceil x \leftarrow\!\!\$\ \overline{\mathrm{supp}\, T[L, \cdot]} \rceil$

  $T[L,x] \leftarrow y$

Return $T^{-1}[L,y]$

Figure 11: Modification of the sampling algorithm. In G_0, the values are sampled to keep consistency between $U[\cdot,\cdot]$ and $T[\cdot,\cdot]$, with the flag bad set if attempted independent sampling leads to inconsistencies. In G_1, the maps $U[\cdot,\cdot]$ and $T[\cdot,\cdot]$ are sampled independently, making RF an independent ideal cipher.

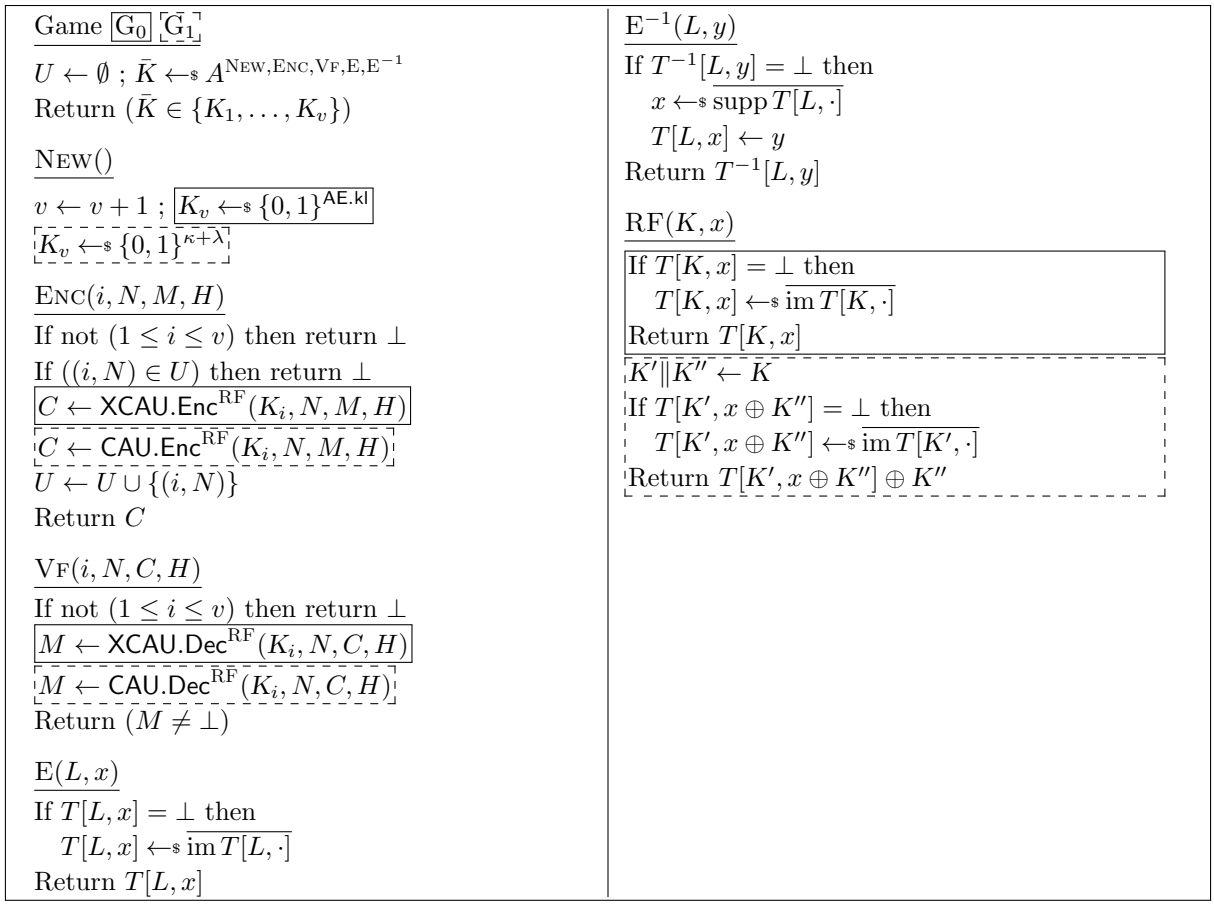

Figure 12: Games that intuitively correspond to the security of AES-XCAU (G_0) as well as AESX-CAU (G_1).

---

the Enc oracle. The first step in this proof is to rewrite the game as G_0 in the same way as in the previous proofs; the scheme is changed to use the oracle RF that is, however, kept consistent with E and E^{-1}. The game is described in Fig. 12.

The next game G_1 is again a syntactic modification from G_0. The change is that we replace XCAU, which uses the original block cipher and applies the input and output whitening for the block cipher as a part of the encryption and decryption procedures, by CAU instantiated with a block cipher with key length $\lambda + \kappa$. Consequently, we rewrite the oracle RF to perform the input and output whitening.

In the next game G_2, the oracles E and E^{-1}, and the oracle RF are based on different maps $T[\cdot, \cdot]$ (for E and E^{-1}) and $U[\cdot, \cdot]$ (for RF), but the oracles are defined to keep them consistent. This is achieved by first sampling them independently, but then re-sampling in case an inconsistency occurs. Should that be the case, the flag bad is set. Apart from this flag, games G_1 and G_2 are equivalent. We do not describe the game G_2 explicitly, but remark that it is obtained by verbatim replacement of the oracles E, E^{-1}, and RF in game G_1 by the ones described in game G_0 in Fig. 11. In the next game G_3, the re-sampling procedure keeping the oracles consistent is abandoned, which means that the oracles RF and E together with E^{-1} are independent. Like G_2, game G_3 is obtained by replacing the oracles E, E^{-1}, and RF by the ones in game G_1 in Fig. 11.

The probability of setting the bad flag in G_2 and G_3 can be bounded using Theorem 5.5. More technically, we describe an adversary $B = B(A)$ that emulates oracles to A as follows: Queries NEW, E, and E^{-1} by B are responded by B performing the same query in its game. Queries ENC and DEC are responded by B emulating the respective oracles using the oracle RF in its game to evaluate CAU.Enc and CAU.Dec. The view of A is the same in G_2 and in the game $R(B(A))$, and in G_3 and the game $S(B(A))$, respectively. The numbers of queries u to the NEW oracle and p to the E and E^{-1} oracles are preserved by B. At most q_e queries of length at most ℓ_{bit} to ENC and at most q_v queries of length at most $\ell_{\text{bit}} + \lambda$ to VF translate into at most $\ell_{\text{blk}}(q_e + q_v) + 1$ queries to RF in the game played by B. Using Theorem 5.5, this means that the probability of setting bad can be bounded by $up(\ell_{\text{blk}}(q_e + q_v) + 1)/2^{\lambda+\kappa+1}$.

All that remains to be done is bounding the probability of A guessing any key in G_3. As in this game, similarly to the previous proofs, the keys used to reference values in $U[\cdot, \cdot]$ is only used as an index to the table and is unrelated to all values that A observes in the game, the guessing probability is at most $u/2^{\kappa}$. This concludes the proof. ∎

6 Indistinguishability Security

In this section we prove the multi-user indistinguishability security bounds for CAU, RCAU, and XCAU, all in the ideal cipher model.

6.1 Preparation: A Lemma on CAU

We begin with a multi-user analysis of CAU which models the block cipher as a uniform random permutation and is useful in the subsequent proofs. The analysis is related to the ones of MV [23], IOM [17], and NOMI [26], with the main difference that they proved single-user security, while we directly prove multi-user security. We formalize the random-permutation model using our game $\mathbf{G}^{\text{mu-ind}}_{\text{CAU}}$ while considering only adversaries that do not make use of the oracles E and E^{-1}.

Lemma 6.1 *Let $\kappa, \lambda, \nu \geq 1$ be such that $\nu \leq \lambda - 2$. Let $\mathsf{H}\colon \{0,1\}^{\lambda} \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^{\lambda}$ be an ϵ-almost XOR-universal hash function, for some $\epsilon\colon \mathbb{N} \times \mathbb{N} \to [0,1]$. Let $\mathsf{CAU} = \mathbf{CAU}[\mathsf{H}, \kappa, \lambda, \nu]$. Let A be an adversary that makes at most u queries to its* NEW *oracle, q_e queries to its* ENC *oracle with messages of length at most ℓ_{bit} bits, and q_v queries to its* VF *oracle with messages of length at most $\ell_{\text{bit}} + \lambda$ bits,[1]. In particular, A does not use the* E *and* E^{-1} *oracles. Assume furthermore that $q_e \leq 2^{\nu}$ and $\ell_{\text{bit}} \leq \lambda(2^{\lambda-\nu} - 2)$. Then*

$$\mathsf{Adv}^{\text{mu-ind}}_{\mathsf{CAU}}(A) \leq \frac{u(u-1)}{2^{\kappa+1}} + \frac{u(\ell_{\text{blk}}(q_e + q_v) + 1)^{2\cdot}}{2^{\lambda+1}} + uq_v \cdot \epsilon(\ell_{\text{bit}}, \ell_{\text{head}}),$$

for $\ell_{\text{blk}} = \lceil \ell_{\text{bit}}/\lambda \rceil + 1$ and where the AEAD headers are restricted to ℓ_{head} bits.

Proof: We make the same assumptions about the validity of the queries made by adversary A as in Theorem 5.1. We describe a game G_0 in Fig. 13 that deviates from game $\mathbf{G}^{\text{mu-ind}}_{\mathsf{AE}}(A)$ in the NEW oracle, where we introduce a new flag bad that is set to true if a collision among the keys of the honest parties occurs. Game G_1 is almost the same, but we re-sample the key from the set of so-far unused keys if a freshly sampled key collides with one that was sampled before. The probability of bad to be set can easily be bounded by the collision probability $u(u-1)/2^{\kappa+1}$.

[1]The ciphertext contains an λ-bit MAC tag, so the length of the contained plaintext is ℓ_{bit} bits.

Game G_0 $\boxed{G_1}$	Game G_3 $\boxed{G_2}$
$b \leftarrow\!\!\$\ \{0,1\}$; $b' \leftarrow\!\!\$\ A^{\text{New},\text{Enc},\text{Vf}}$ Return $(b' = b)$	$b \leftarrow\!\!\$\ \{0,1\}$; $b' \leftarrow\!\!\$\ A^{\text{New},\text{Enc},\text{Vf}}$ Return $(b' = b)$

New()
$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^{\mathsf{CAU.kl}}$
If $K_v \in \{K_1, \ldots, K_{v-1}\}$ then
  $\mathsf{bad} \leftarrow \mathsf{true}$
  $\boxed{K_v \leftarrow\!\!\$\ \overline{\{K_1, \ldots, K_{v-1}\}}}$
	New()
	$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \overline{\{K_1, \ldots, K_{v-1}\}}$

Enc(i, N, M, H) $C_1 \leftarrow\!\!\$\ \mathsf{CAU.Enc}^{\mathrm{E}}(K_i, N, M, H)$ $C_0 \leftarrow\!\!\$\ \{0,1\}^{\mathsf{CAU.cl}(\lvert M\rvert)}$ Return C_b	Enc(i, N, M, H) $C_1 \leftarrow\!\!\$\ \mathsf{CAU.Enc}^{\mathrm{E}}(K_i, N, M, H)$ $C_0 \leftarrow\!\!\$\ \{0,1\}^{\mathsf{CAU.cl}(\lvert M\rvert)}$ Return C_b
Vf(i, N, C, H) If $(b = 0)$ then return false $M \leftarrow \mathsf{CAU.Dec}^{\mathrm{E}}(K_i, N, C, H)$ Return $(M \neq \perp)$	Vf(i, N, C, H) If $(b = 0)$ then return false $M \leftarrow \mathsf{CAU.Dec}^{\mathrm{E}}(K_i, N, C, H)$ Return $(M \neq \perp)$

E(K, x)
If $U[K, x] = \perp$ then
  $y \leftarrow\!\!\$\ \overline{\operatorname{im} U[K, \cdot]}$
Return $U[K, x]$
	E(K, x)
	If $U[K, x] = \perp$ then
	  $y \leftarrow\!\!\$\ \{0,1\}^{\lambda}$
	  If $y \in \operatorname{im} U[K, \cdot]$ then
	  $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{y \leftarrow\!\!\$\ \overline{\operatorname{im} U[K, \cdot]}}$
	  $U[K, x] \leftarrow y$
	Return $U[K, x]$

Figure 13: **Left:** Changing the game to prevent collisions among user keys. **Right:** Using a random function instead of a random permutation.

In game G_2 we first simplify the description of the oracle New in comparison with G_1. The next proof step is instrumental for establishing the secrecy of the scheme, the goal is to replace the ideal cipher used in CAU by an ideal random function with the same range and domain. This will allow us later to prove that the ciphertexts are uniformly distributed. We rewrite the game G_1 to the form of game G_2. The sampling procedure of the value $U[K, x]$ in E is an equivalent formulation that will allow the next proof step. The idea is, similar to proofs in [6], to first sample the output of the ideal cipher uniformly at random. In case a collision with previously sampled values occurrs, we sample uniformly from the set not containing the previous values. This does not change the distribution, since, intuitively, the probability mass associated to invalid responses is distributed uniformly over the valid responses. We additionally introduce a new flag bad that does not change the behavior of the game but detects when the re-sampling strategy is invoked.

The next step is then to switch to game G_3 in which the oracle E samples without collisions. The games G_2 and G_3 differ only for after bad has been set, and the Fundamental Lemma then allows to bound the difference in advantage by the probability of provoking bad, where each block sampled initially in E is uniformly random. Since the collisions are checked for each key $K \in \{K_1, \ldots, K_u\}$ individually, and the invocations of CAU lead to at most $\ell_{\text{blk}}(q_e + q_v) + 1$ queries per user, the Union Bound and the standard collision probability let us bound the overall probability by $u \cdot (\ell_{\text{blk}}(q_e + q_v) + 1)^2 / 2^{\lambda+1}$.

In game G_4 in Fig. 14, we introduce a new flag bad in the verification oracle Vf. The flag is set if

Game G_4 $\boxed{G_5}$	$\underline{\textsc{Vf}(i, N, C, H)}$
$b \leftarrow\!\!\$\ \{0,1\}$; $b' \leftarrow\!\!\$\ A^{\textsc{New},\textsc{Enc},\textsc{Vf}}$	If $(b = 0)$ then return false
Return $(b' = b)$	$M \leftarrow \mathsf{CAU.Dec}^{\mathrm{E}}(K_i, N, C, H)$
$\underline{\textsc{New}()}$	If $M \neq \bot$ then $\mathsf{bad} \leftarrow \mathsf{true}$
$v \leftarrow v + 1$	$\boxed{\text{Return false;}}$ Return $(M \neq \bot)$
$K_v \leftarrow\!\!\$\ \overline{\{K_1, \ldots, K_{v-1}\}}$	$\underline{\mathrm{E}(K, x)}$
$\underline{\textsc{Enc}(i, N, M, H)}$	If $U[K, x] = \bot$ then
$C_1 \leftarrow\!\!\$\ \mathsf{CAU.Enc}^{\mathrm{E}}(K_i, N, M, H)$	$\quad U[K, x] \leftarrow\!\!\$\ \{0,1\}^{\lambda}$
$C_0 \leftarrow\!\!\$\ \{0,1\}^{\mathsf{CAU.cl}(\lvert M \rvert)}$	Return $U[K, x]$
Return C_b	

Figure 14: Between the games G_4 and G_5, we change the behavior of the $\textsc{Vf}$ oracle to reject forgery attempts also for $b = 1$.

---

the decryption algorithm $\mathsf{CAU.Dec}$ returns a valid result but the ciphertext has not been obtained through $\textsc{Enc}$; in other words, the flag bad is set when the adversary successfully forges a ciphertext. In game G_5 we then modify what happens after the flag has been set: the oracle rejects anyway.

To bound the probability of bad to be set in G_4 and G_5, we use that H is an ϵ-almost XOR-universal hash function. For each query $\textsc{Vf}(i, N, C, H)$, the probability for bad to be set is bounded by $\epsilon(\ell_{\mathrm{bit}}, \ell_{\mathrm{head}})$. In more detail, we distinguish two cases:

- If N was not used in a query to $\textsc{Enc}$ before, the event occurs with probability at most $\epsilon(\ell_{\mathrm{bit}}, \ell_{\mathrm{head}})$, as
$$\Pr\left[\mathsf{H}(G, H, C^*) \oplus \mathsf{H}(G, \varepsilon, \varepsilon) = T \oplus U[K_i, N \| 0^{31}1]\right] \leq \epsilon(\ell_{\mathrm{bit}}, \ell_{\mathrm{head}})$$
holds by the ϵ-almost XOR-universality of H, for $C = T \| C^*$, so C^* is the part of the ciphertext corresponding to the counter-mode encryption. (The function H fulfills $\mathsf{H}(G, \varepsilon, \varepsilon) = 0^{\lambda}$.)

- If N was used in a query to $\textsc{Enc}$ before with header H' and leading to ciphertext $\bar{C} = \bar{T} \| \bar{C}^*$, then
$$\Pr\left[\mathsf{H}(G, H, C) \oplus \mathsf{H}(G, H, \bar{C}^*) = T \oplus \bar{T}\right] \leq \epsilon(\ell_{\mathrm{bit}}, \ell_{\mathrm{head}}),$$
where $C = T \| C^*$ is the ciphertext output by $\textsc{Enc}$ in the query with nonce N. We can use the AXU property here because, while the tag T is known to the adversary prior to the choice of $\bar{C}$, the output of H is masked by a uniformly random value obtained from E. Therefore, we can equivalently view the adversary as inputting the mask $T \oplus \bar{T}$ without any knowledge of the output of H.

Adversary A makes at most uq_v queries to the $\textsc{Vf}$ oracle, which means that we can bound the overall probability that bad is set by $uq_v \cdot \epsilon(\ell_{\mathrm{bit}}, \ell_{\mathrm{head}})$.

Given that A makes at most $q_e \leq 2^{\nu}$ queries to $\textsc{Enc}$ per user instance, and the length of each message is bounded by $\ell_{\mathrm{blk}} \leq 2^{\lambda - \nu} - 2$ blocks, and because by assumption A does not repeat nonces, then in game G_5, the ciphertexts C_0 and C_1 are both uniformly random bit strings of the same length. The game is therefore independent of the challenge bit b, which means that the adversary cannot have any advantage in guessing b.

We combine all bounds shown in the above paragraphs:

$$\begin{aligned}
\mathsf{Adv}^{\text{mu-ind}}_{\mathsf{CAU}}(A) &= 2\Pr[\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}(A)] - 1 = 2\Pr[\mathrm{G}_0] - 1 \\
&\leq 2\Pr[\mathrm{G}_1] - 1 + \frac{u(u-1)}{2^{\kappa+1}} \\
&\leq 2\Pr[\mathrm{G}_3] - 1 + \frac{u(u-1)}{2^{\kappa+1}} + \frac{u((q_e+q_v)\cdot\ell_{\text{blk}})^2}{2^{\lambda+1}} \\
&\leq 2\Pr[\mathrm{G}_5] - 1 + \frac{u(u-1)}{2^{\kappa+1}} \\
&\quad + \frac{u((q_e+q_v)\cdot\ell_{\text{blk}})^2}{2^{\lambda+1}} + uq_v\cdot\epsilon(\ell_{\text{bit}},\ell_{\text{head}}),
\end{aligned}$$

which concludes the proof ▌

6.2 Security of CAU

We now prove the multi-user indistinguishability security of plain CAU in the ideal-cipher model.

Theorem 6.2 *Let $\kappa,\lambda,\nu \geq 1$ be such that $\nu \leq \lambda-2$. Let $\mathsf{H}\colon \{0,1\}^\lambda \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^\lambda$ be an ϵ-almost XOR-universal hash function, for some $\epsilon\colon \mathbb{N}\times\mathbb{N}\to[0,1]$. Let $\mathsf{CAU} = \mathbf{CAU}[\mathsf{H},\kappa,\lambda,\nu]$. Let A be an adversary that makes at most u queries to its* New *oracle, q_e queries to its* Enc *oracle with messages of length at most ℓ_{bit} bits, q_v queries to its* Vf *oracle with messages of length at most $\ell_{\text{bit}}+\lambda$ bits, and p queries to its* E *and* E^{-1} *oracles. Assume furthermore that $q_e \leq 2^\nu$ and $\ell_{\text{bit}} \leq \lambda(2^{\lambda-\nu}-2)$. Then*

$$\mathsf{Adv}^{\text{mu-ind}}_{\mathsf{CAU}}(A) \leq \frac{up}{2^\kappa} + \frac{u(\ell_{\text{blk}}(q_e+q_v)+1)^{2\cdot}}{2^{\lambda+1}} + \frac{u(u-1)}{2^{\kappa+1}} + uq_v\cdot\epsilon(\ell_{\text{bit}},\ell_{\text{head}}),$$

for $\ell_{\text{blk}} = \lceil \ell_{\text{bit}}/\lambda\rceil + 1$ and where the AEAD headers are restricted to ℓ_{head} bits.

The first term originates from the advantage of the adversary in guessing a user's key in a query to the ideal cipher. This term grows linearly in the number of honest sessions, and it also grows linearly in the number of adversary calls to the ideal cipher. We show below in Theorem 5.2 that a term of this size is inevitable by proving the effectiveness of an attack. The second term stems from a PRF/PRP-switching in the proof of counter mode. The third term stems from a potential collision of honest-user keys, and the final term from the authentication using the AUH-based MAC.

Proof: We make the same assumptions about the validity of the queries made by adversary A as in Theorem 5.1. The first game G_0 we consider is described in Fig. 15 and is only a syntactic modification of game $\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}(A)$. We introduce an additional oracle RF that implements the forward evaluation of the ideal cipher for the algorithms CAU.Enc and CAU.Dec. This is sufficient as encryption and decryption in GCM never query E^{-1}. In more detail, oracle RF samples the ideal cipher for the keys used in the encryption using the map $U[\cdot,\cdot]$. The oracles E and E^{-1} are adapted such that, for keys used in the game, they sample the map $T[\cdot,\cdot]$ consistently with $U[\cdot,\cdot]$. We introduce a flag bad that is set when the adversary A queries one of the oracles E or E^{-1} with a key that is also used in the oracle RF. Apart from this flag, game G_0 is equivalent to $\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}(A)$.

We then rewrite the game in the form of G_1 analogously to the modifications in Theorem 5.1, which modifies the way in which the responses for the E, E^{-1}, and RF oracles are determined. In particular, we break the consistency between E and E^{-1} on the one hand, and RF on the other hand, by sampling the oracle responses independently. Since all changes appear only after bad has

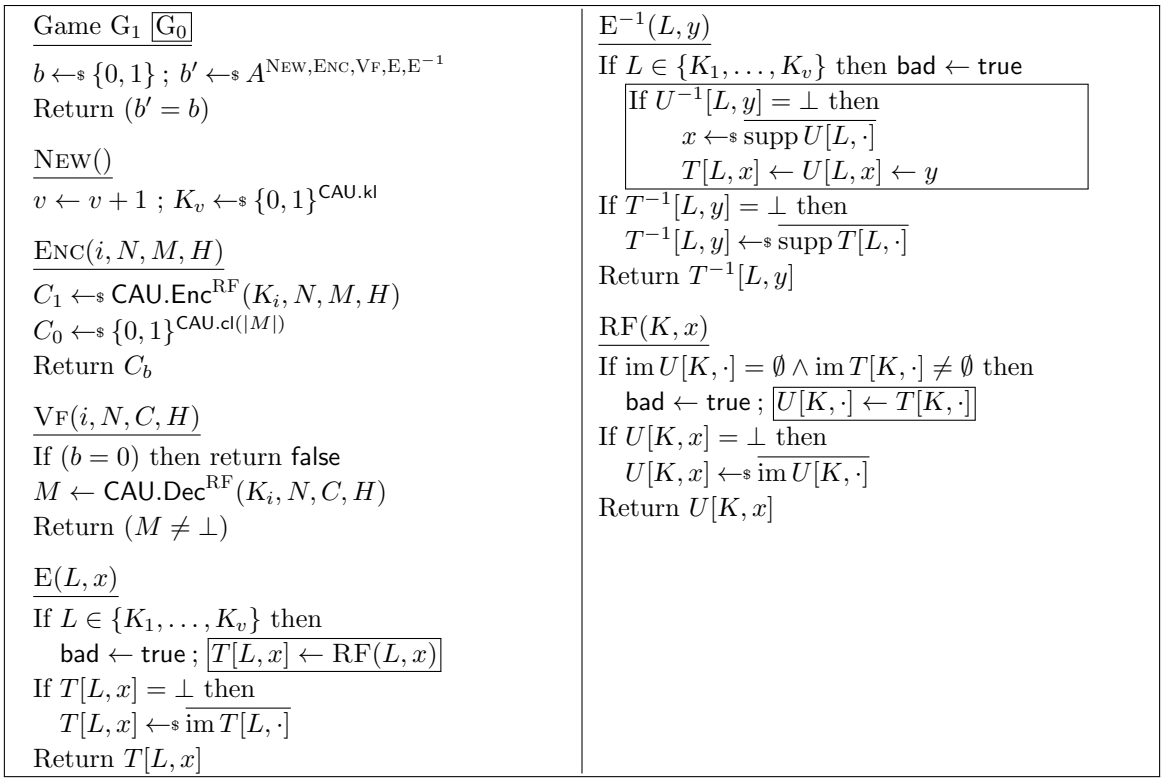

Figure 15: In game G_0 the oracles RF and E/E^{-1} are consistent, in game G_1 they are independent.

---

been set, we can relate the games using the Fundamental Lemma from Bellare and Rogaway [6] and proceed by bounding the probability of setting bad. This probability is in fact bounded by $up/2^\kappa$. As all computations while bad is not set are independent of the values $K_1, \ldots, K_u$, the maximal probability of the adversary to guess one of these uniformly random values is $u/2^\kappa$ in each of its p queries to E and E^{-1}.

We now observe that winning game G_1 is almost equivalent to winning $\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}$ without access to the oracles E and E^{-1}, but these oracles are independent of all other oracles in the game, and so they are easy to simulate. We therefore consider the adversary $B = B(A)$ in the game $\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}$ that simulates the oracles E and E^{-1} to the adversary A and forwards all other queries to its own oracles. Adversary B therefore never uses the oracles E and E^{-1}, therefore we can apply Lemma 6.1. Clearly, $\Pr[G_1] = \Pr[\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}(B(A))]$, and therefore

$$\begin{aligned}
\mathsf{Adv}^{\text{mu-ind}}_{\mathsf{CAU}}(A) &= 2\Pr[\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}(A)] - 1 = 2\Pr[G_0] - 1 \\
&\leq 2\Pr[\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}(B(A))] - 1 + \frac{up}{2^\kappa} \\
&\leq \frac{u(u-1)}{2^{\kappa+1}} + \frac{u((q_e+q_v)\cdot\ell_{\text{blk}})^2}{2^{\lambda+1}} + uq_v \cdot \epsilon(\ell_{\text{bit}}, \ell_{\text{head}}) + \frac{up}{2^\kappa},
\end{aligned}$$

which concludes the proof. ∎

6.3 Security of RCAU

In terms of bounds for RCAU, we first show a simple corollary proving that the same bounds as for CAU also apply for RCAU. This follows immediately by a reduction that randomizes the nonces.

Corollary 6.3 *Let $\kappa, \lambda, \nu \geq 1$ be such that $\nu \leq \lambda - 2$. Let $\mathsf{H}\colon \{0,1\}^\lambda \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^\lambda$ be an ϵ-almost XOR-universal hash function, for some $\epsilon\colon \mathbb{N} \times \mathbb{N} \to [0,1]$. Let $\mathsf{RCAU} = \mathbf{RCAU}[\mathsf{H}, \kappa, \lambda, \nu]$. Let A be an adversary that makes at most u queries to its* New *oracle, q_e queries to its* Enc *oracle with messages of length at most ℓ_{bit} bits, q_v queries to its* Vf *oracle with messages of length $\ell_{\text{bit}} + \lambda$ bits, p_e queries to its* E *oracle, p_i queries to its* E^{-1} *oracle, and $p = p_e + p_i$. Assume furthermore that $q_e \leq 2^\nu$, and $\ell_{\text{bit}} \leq \lambda(2^{\lambda-\nu} - 2)$. (For brevity we write $q = q_e + q_v$.) Then*

$$\mathsf{Adv}^{\text{mu-ind}}_{\mathsf{RCAU}}(A) \leq \frac{up}{2^\kappa} + \frac{u(\ell_{\text{blk}}(q_e + q_v) + 1)^{2\cdot}}{2^{\lambda+1}} + \frac{u(u-1)}{2^{\kappa+1}} + uq_v \cdot \epsilon(\ell_{\text{bit}}, \ell_{\text{head}}),$$

for $\ell_{\text{blk}} = \lceil \ell_{\text{bit}}/\lambda \rceil + 1$ and where the AEAD headers are restricted to ℓ_{head} bits.

Proof: We prove this via a reduction as follows: adversary B within game $\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}$ emulates the oracles of game $\mathbf{G}^{\text{mu-ind}}_{\mathsf{RCAU}}$ toward the assumed adversary A as follows: queries to oracles E and E^{-1} are simply forwarded to the same oracles in game $\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}$. Queries to oracle New are handled by calling the same oracle in $\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}$, and for the ith user sampling a random masking key $K' \leftarrow\!\!\$\ \{0,1\}^\nu$. Queries $\textsc{Enc}(i, N, M, H)$ and $\textsc{Dec}(i, N, C, H)$ are handled by calling the respective oracle in $\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}$, but replacing N by $N \oplus K'_i$. The view of A in $\mathbf{G}^{\text{mu-ind}}_{\mathsf{RCAU}}(A)$ and in $\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}(B(A))$ is the same. As B makes exactly the same number of queries as A, the claimed bound follows. ▌

We prove a stronger bound for the advantage of a *passive* adversary that does not use its Vf oracle in a non-trivial way. The bound differs from the one proven above significantly: we show that for *passive* adversaries we can replace the term $up/2^\kappa$ in the bound for CAU by terms that are smaller for realistic parameters. The proof does, however, not extend to *active* adversaries that make use of the Vf oracle: In fact, RCAU evaluates the block cipher, in each session, on the fixed value 0^λ to obtain the key for H, and our analysis of the authenticity guarantee requires that this key be uniformly random. This requirement is of course not fulfilled if the adversary evaluated the block cipher on the value 0^λ for the respective key.

In the result for RCAU, we explicitly distinguish between the numbers for evaluation p_e and inversion p_i queries for the block cipher, with $p = p_e + p_i$.

Theorem 6.4 *Let $\kappa, \lambda, \nu \geq 1$ be such that $\nu \leq \lambda - 2$. Let $\mathsf{H}\colon \{0,1\}^\lambda \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^\lambda$ be a family of functions. Let $\mathsf{RCAU} = \mathbf{RCAU}[\mathsf{H}, \kappa, \lambda, \nu]$. Let A be an adversary that makes at most u queries to its* New *oracle, q_e queries to its* Enc *oracle with messages of length at most ℓ_{bit} bits, q_v queries to its* Vf *oracle with messages of length $\ell_{\text{bit}} + \lambda$ bits, p_e queries to its* E *oracle, p_i queries to its* E^{-1} *oracle, and $p = p_e + p_i$. Assume furthermore that $q_e \leq 2^\nu$, and $\ell_{\text{bit}} \leq \lambda(2^{\lambda-\nu} - 2)$. (For brevity we write $q = q_e + q_v$.) Then*

$$\mathsf{Adv}^{\text{mu-ind}}_{\mathsf{RCAU}}(A) \leq \frac{u(\ell_{\text{blk}} q_e + 1)^2}{2^{\lambda+1}} + \frac{up(\ell_{\text{blk}} q_e + 1)}{2^{\kappa+\nu-1}} + \frac{up(\ell_{\text{blk}} q_e + 1)}{2^\kappa(2^\lambda - p)} + \frac{up(\ell_{\text{blk}} q_e + 1)}{2^\kappa(2^\lambda - q_e)} + \frac{2p_i + u(u-1)}{2^{\kappa+1}}, \quad (3)$$

for $\ell_{\text{blk}} = \lceil \ell_{\text{bit}}/\lambda \rceil + 1$, and for an adversary A making $q_v = 0$ verification queries.

In comparison with the bound proven in Theorem 6.2, the major difference in Equation (3) is that the term $up/2^{\kappa}$ is replaced by the four terms $up(\ell_{\mathrm{blk}}(q_e+q_v)+1)/2^{\kappa+\nu}$, $up(\ell_{\mathrm{blk}}(q_e+q_v)+1)/2^{\kappa}(2^{\lambda}-u)$, $up(\ell_{\mathrm{blk}}(q_e+q_v)+1)/2^{\kappa}(2^{\lambda}-q_e-q_v)$ and $p_i/2^{\kappa}$. This is an improvement because for the values used in TLS 1.3 it is reasonable to assume $\ell_{\mathrm{blk}}(q_e+q_v)+1 \ll 2^{96}$ as well as $q_e+q_v, p \ll 2^{96}$, and the term $p_i/2^{\kappa}$ does not scale with u. Unfortunately, our proof does not support a similar statement for active attacks.

Proof: The proof starts along the same lines as the proof of Theorem 6.2, which means that we describe a game G_0 in which the calls of RCAU.Enc and RCAU.Dec to E are replaced by calls to a different oracle RF that performs exactly the same computation as E. This is an equivalent formulation of $\mathbf{G}^{\text{mu-ind}}_{\mathsf{RCAU}}(A)$, since the oracles E and RF are equivalent. Additionally, we introduce a flag bad that is set when a key sampled in the NEW oracle for a new user collides with a key chosen previously for another user. Game G_0 is described in Fig. 16.

In game G_1, we change the NEW oracle for the case that a collision between the keys of honest users occurs. The oracle re-samples the key in a way that avoids a collisions. The modification occurs after the flag bad is set, and analogously to the proof of Theorem 6.2, we conclude that the probability for the flag to be set is bounded by $u(u-1)/2^{\kappa+1}$.

As in the proof of Theorem 6.2, we now aim at "decoupling" the oracles $\mathrm{E}/\mathrm{E}^{-1}$ and RF, but this time we have to be cautious: we cannot just "give up" when the adversary "guesses" one of the users keys in calls to $\mathrm{E}/\mathrm{E}^{-1}$; this would ruin our bound. The first step is as above to introduce an auxiliary map $U[\cdot,\cdot]$ in addition to $T[\cdot,\cdot]$, but keep the maps synchronized. The change from G_1 to G_2 is therefore only syntactic. Intuitively, the lazy sampling of the block cipher is now performed using both maps, where $T[\cdot,\cdot]$ is filled in calls to E and E^{-1}, and $U[\cdot,\cdot]$ is filled in RF. The oracles make sure that the maps stay consistent.

In game G_3, we first change the way in which the responses are sampled, but still in an equivalent way, namely we first attempt to sample consistently only for $T[\cdot,\cdot]$ and then check for consistency with $U[\cdot,\cdot]$. If this fails, we set $\mathsf{bad} \leftarrow \mathsf{true}$ and re-sample with the correct distribution. Additionally, we set $\mathsf{bad} \leftarrow \mathsf{true}$ whenever we need to answer for either $T[\cdot,\cdot]$ or $U[\cdot,\cdot]$ and the answer is already defined by the respective other map. Game G_3 is equivalent to G_2. The proof is further complicated by the fact that RCAU derives the key for H as $\mathrm{E}(K,0^{\lambda})$ and this query is therefore not randomized. As a consequence, we have to treat the queries with value 0^{λ} independently of the other queries, and keep the maps $T[\cdot,0^{\lambda}]$ and $U[\cdot,0^{\lambda}]$ consistent for the next proof steps.

In game G_4 we modify the behavior of the oracles E, E^{-1}, and RF to not re-sample to avoid inconsistencies with the other oracles. Also, we do not enforce consistency between $T[K,\cdot]$ and $U[K,\cdot]$ for values that are defined already in one of the maps; we sample a fresh value in a map independently of whether the point is already defined in the other map. As both modifications occur only after the flag bad has been set, we can use the Fundamental Lemma to relate the advantages of an adversary in games G_3 and G_4.

To bound the probability for the flag bad to be set in games G_3 or G_4, respectively, we begin with the following observation: As long as bad is not set, each row $T[K,\cdot]$ or $U[K,\cdot]$ for a specific key K is sampled without collisions within this row, but independently of any other row, and also mutually independent between $T[K,\cdot]$ and $U[K,\cdot]$. This is the case because the only other way of defining a value for those maps is either re-sampling or copying from the other map; in both cases we set the flag bad. Furthermore, we observe that all operations that occur before the flag bad is set are independent of the actual values of the keys $K_1,\ldots,K_u$. Given these insights, we now analyze the probabilities for setting the bad flag at the different code points, first for E and E^{-1}:

Game G_0 $\boxed{\mathrm{G}_1}$

$b \leftarrow\!\!\$\ \{0,1\}$
$b' \leftarrow\!\!\$\ A^{\textsc{New},\textsc{Enc},\textsc{Vf},\mathrm{E},\mathrm{E}^{-1}}$
Return $(b' = b)$

$\underline{\textsc{New}()}$
$v \leftarrow v+1$; $K_v \leftarrow\!\!\$\ \{0,1\}^{\mathsf{RCAU.kl}}$
If $K_v \in \{K_1, \ldots, K_{v-1}\}$ then
  $\mathsf{bad} \leftarrow \mathsf{true}$
  $\boxed{K_v \leftarrow\!\!\$\ \overline{\{K_1, \ldots, K_{v-1}\}}}$

$\underline{\textsc{Enc}(i, N, M, H)}$
$C_1 \leftarrow\!\!\$\ \mathsf{RCAU.Enc}^{\mathrm{RF}}(K_i, N, M)$
$C_0 \leftarrow\!\!\$\ \{0,1\}^{\mathsf{RCAU.cl}(|M|)}$
Return C_b

$\underline{\textsc{Vf}(i, N, C, H)}$
If $(b = 0)$ then return false
$M \leftarrow \mathsf{RCAU.Dec}^{\mathrm{RF}}(K_i, N, C, H)$
Return $(M \neq \perp)$

$\underline{\mathrm{E}(L, x)}$
If $T[L, x] = \perp$ then
  $T[L, x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[L, \cdot]}$
Return $T[L, x]$

$\underline{\mathrm{E}^{-1}(L, y)}$
If $T^{-1}[L, y] = \perp$ then
  $x \leftarrow\!\!\$\ \overline{\mathrm{supp}\, T[L, \cdot]}$
  $T[L, x] \leftarrow y$
Return $T^{-1}[L, y]$

$\underline{\mathrm{RF}(K, x)}$
If $T[K, x] = \perp$ then
  $T[K, x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[K, \cdot]}$
Return $T[K, x]$

Game G_2

Initialization, New, Enc, *and* Vf *unmodified from* G_1.

$\underline{\mathrm{E}(L, x)}$
If $T[L, x] = \perp$ then
  If $x = 0^\lambda$ then
    $T[L, x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[L, \cdot]}$
  Else if $U[L, x] = \perp$ then
    $T[L, x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[L, \cdot] \cup \mathrm{im}\, U[L, \cdot]}$
  Else $T[L, x] \leftarrow U[L, x]$
Return $T[L, x]$

$\underline{\mathrm{E}^{-1}(L, y)}$
If $T^{-1}[L, y] = \perp$ then
  If $U^{-1}[L, y] = \perp$ then
    $x \leftarrow\!\!\$\ \overline{\mathrm{supp}\, T[L, \cdot] \cup \mathrm{supp}\, U[K, \cdot]}$
    $T[L, x] \leftarrow y$
  Else $T[L, U^{-1}[L, y]] \leftarrow y$
Return $T^{-1}[L, y]$

$\underline{\mathrm{RF}(K, x)}$
If $U[K, x] = \perp$ then
  If $x = 0^\lambda$ then
    $U[K, x] \leftarrow \mathrm{E}(K, x)$
  Else if $T[K, x] = \perp$ then
    $U[K, x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[K, \cdot] \cup \mathrm{im}\, U[K, \cdot]}$
  Else $U[K, x] \leftarrow T[K, x]$
Return $U[K, x]$

Figure 16: **Left:** The multi-user game in which the calls from AE.Enc and AE.Dec to E are replaced by calls to RF, which implements the same function. **Right:** Equivalent formulation of the same game, in which the management of the lazy sampling is achieved by different but equivalent code.

- The probability of enforcing re-sampling in E or E^{-1} is analyzed as follows: For a particular key $\bar{K} \in \{K_1, \ldots, K_u\}$ for which m blocks have been defined through queries to RF, the probability of sampling a value that collides is at most $m/(2^\lambda - p)$, as we choose uniformly from $2^\lambda - p$ values. The expected number of blocks for the key L in the query is $u(\ell_{\mathrm{blk}}(q_e + q_v) + 1)/2^\kappa$, which leads to an overall probability of $u(\ell_{\mathrm{blk}}(q_e + q_v) + 1)/(2^\kappa(2^\lambda - p))$ for each query.

- The probability of enforcing that a value be copied (that is, the final "Else" statement becomes active) in E is bounded by $u(\ell_{\mathrm{blk}}(q_e + q_v) + 1)/2^{\kappa+\nu}$ for each of the p queries. This is computed analogously to above: executing the "Else" statement means that the adversary guessed a

Game $\boxed{\mathrm{G}_3}$ $\boxed{\mathrm{G}_4}$

Initialization, NEW, ENC, *and* VF *unmodified from* G_1.

$\underline{\mathrm{E}(L,x)}$

If $T[L,x] = \bot$ then
  If $x = 0^\lambda$ then $T[L,x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[L,\cdot]}$
  Else If $U[L,x] = \bot$ then
    $T[L,x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[L,\cdot]}$
    If $T[L,x] \in \mathrm{im}\, U[L,\cdot]$ then
      $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{T[L,x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[L,\cdot] \cup \mathrm{im}\, U[L,\cdot]}}$
  Else $T[L,x] \leftarrow U[L,x]$; $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{T[L,x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[L,\cdot]}}$
Return $T[L,x]$

$\underline{\mathrm{E}^{-1}(L,y)}$

If $T^{-1}[L,y] = \bot$ then
  If $U^{-1}[L,y] = \bot$ then
    $x \leftarrow\!\!\$\ \overline{\mathrm{supp}\, T[L,\cdot]}$
    If $x \in \mathrm{supp}\, U[L,\cdot]$ then
      $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{x \leftarrow\!\!\$\ \overline{\mathrm{supp}\, T[L,\cdot] \cup \mathrm{supp}\, U[L,\cdot]}}$
    $T[L,x] \leftarrow y$
  Else
    $T[L, U^{-1}[L,y]] \leftarrow y$; $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{T[L, U^{-1}[L,y]] \leftarrow\!\!\$\ \overline{\mathrm{supp}\, T[L,\cdot]}}$
Return $T^{-1}[L,y]$

$\underline{\mathrm{RF}(K,x)}$

If $U[K,x] = \bot$ then
  If $x = 0^\lambda$ then $U[K,x] \leftarrow \mathrm{E}(K,x)$
  Else if $T[K,x] = \bot$ then
    $U[K,x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, U[K,\cdot]}$
    If $U[K,x] \in \mathrm{im}\, T[K,\cdot]$ then
      $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{U[K,x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[K,\cdot] \cup \mathrm{im}\, U[K,\cdot]}}$
  Else $U[K,x] \leftarrow T[K,x]$; $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{U[K,x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, U[K,\cdot]}}$
Return $U[K,x]$

Figure 17: Modifying the oracles such that the maps $T[\cdot,\cdot]$ and $U[\cdot,\cdot]$ become independent.

combination of a κ-bit key and a ν-bit mask value.

- Finally, the probability for copying a value in E^{-1} is bounded by the term $1/2^{-\kappa}$. The reason is that it corresponds to guessing a the key for a specific user.

We obtain the bounds $u(\ell_{\mathrm{blk}}(q_e + q_v) + 1) \cdot p/2^{\kappa+\lambda}$, $u(\ell_{\mathrm{blk}}(q_e + q_v) + 1)p_e/2^{\kappa+\nu}$, and $p_i/2^{-\kappa}$ as the adversary makes at most p_e queries to E, p_i queries to E^{-1}, and $p = p_e + p_i$ queries accumulated.

We proceed by analyzing the probabilities for RF analogously:

- With respect to enforcing re-sampling in RF, for a key L for which m blocks have been defined, the probability of sampling a colliding value is $m/(2^\lambda - q_e - q_v)$. This leads to an

Game $\boxed{\mathrm{G}_5}$ G_6	$\underline{\mathrm{RF}(K,x)}$
Initialization, NEW, ENC, *and* VF *unmodified from* G_1.	If $U[K,x] = \bot$ then
$\underline{\mathrm{E}(L,x)}$	If $x = 0^\lambda$ then
If $T[L,x] = \bot$ then	$U[K,x] \leftarrow \mathrm{E}(K,x)$
$T[L,x] \leftarrow\!\!\$\ \overline{\mathrm{im}\, T[L,\cdot]}$	Else
Return $T[L,x]$	$y \leftarrow\!\!\$\ \{0,1\}^\lambda$
$\underline{\mathrm{E}^{-1}(L,y)}$	If $y \in \mathrm{im}\, U[K,\cdot]$ then
If $T^{-1}[L,y] = \bot$ then	$\mathsf{bad} \leftarrow \mathsf{true}$
$x \leftarrow\!\!\$\ \overline{\mathrm{supp}\, T[L,\cdot]}$	$\boxed{y \leftarrow\!\!\$\ \overline{\mathrm{im}\, U[K,\cdot]}}$
$T[L,x] \leftarrow y$	$U[K,x] \leftarrow y$
Return $T^{-1}[L,y]$	Return $U[K,x]$

Figure 18: Moving to a version of the game where RF is a random function.

overall probability of at most $p/(2^\kappa(2^\lambda - q_e - q_v))$.

- The probability of enforcing that a value be copied (that is, the final "Else" statement becomes active) in RF is bounded by $u(\ell_{\mathrm{blk}}(q_e + q_v) + 1)p/2^{\kappa+\nu}$. The reason is that for a particular key L for which m blocks have been defined through queries to E and E^{-1}, the probability that an query to RF as done by $\mathsf{CAU.Enc}^{\mathrm{RF}}$ uses the same input is bounded by $m/2^\nu$. This leads to a probability of $p/2^{\kappa+\nu}$.

Since the encryption and decryption algorithms overall make $u(\ell_{\mathrm{blk}}(q_e + q_v) + 1)$ queries to RF, we obtain the bounds $up(\ell_{\mathrm{blk}}(q_e + q_v) + 1)/(2^\kappa(2^\lambda - q_e - q_v))$ and $up(\ell_{\mathrm{blk}}(q_e + q_v) + 1)/2^{\kappa+\nu}$.

The next step of the proof modifies the way in which RF samples its outputs and is almost as in the proof of Lemma 6.1. Game G_5 is equivalent to G_4, the changes are only syntactic. In particular, we change all oracles E, E^{-1}, and RF to remove redundant code and unify statements where the "If" and "Else" branches are equivalent. Additionally, we change the way in which the oracle RF samples its outputs to a sample-check-resample strategy. Game G_6 behaves similarly, but does not resample. As in Lemma 6.1, the probability for bad to be set can be bounded by $u(\ell_{\mathrm{blk}}(q_e + q_v) + 1)^2/2^\lambda$.

The then proof concludes by combining all the terms to obtain the bound stated in the Theorem. ∎

We stress that the term $up/2^\kappa$ in Equation (6.3) does, unlike the one in Theorem 6.2, not immediately corresponds to a matching attack on the use of the scheme within the TLS protocol. The reason is that such an attack would require sending a great amount of crafted ciphertexts within the TLS session, but TLS tears down a session and discards the keys after the first failure in MAC verification. Therefore, it is conceivable that the scheme as used within TLS achieves considerably better security against active attacks than our above bound suggests. Moreover, such an attack would be inherently *active* and not suitable for mass surveillance.

6.4 Security of XCAU

To analyze the indistinguishability security of XCAU, we combine the results of Theorem 5.5 and Lemma 6.1. The proof is almost the same as the one for Theorem 6.2, but the step of "decoupling"

the $\mathrm{E}/\mathrm{E}^{-1}$ and RF oracles makes use of the results in Theorem 5.5. Most notably and in contrast to RCAU, the bound does not contain a term of the type $p_i/2^\kappa$, and applies to active adversaries as well.

Theorem 6.5 *Let $\kappa, \lambda, \nu \geq 1$ be such that $\nu \leq \lambda - 2$. Let $\mathsf{H}\colon \{0,1\}^\lambda \times (\{0,1\}^* \times \{0,1\}^*) \to \{0,1\}^\lambda$ be an ϵ-almost XOR-universal hash function, for some $\epsilon\colon \mathbb{N} \times \mathbb{N} \to [0,1]$. Let $\mathsf{XCAU} = \mathbf{XCAU}[\mathsf{H}, \kappa, \lambda, \nu]$. Let A be an adversary that makes at most u queries to its* New *oracle, q_e queries to its* Enc *oracle with messages of length at most ℓ_{bit} bits, q_v queries to its* Vf *oracle with messages of length at most $\ell_{\mathrm{bit}} + \lambda$ bits, and p queries to its* E *and* E^{-1} *oracles. Assume furthermore that $q_e \leq 2^\nu$ and $\ell_{\mathrm{bit}} \leq \lambda(2^{\lambda-\nu} - 2)$. Then*

$$\mathsf{Adv}^{\text{mu-ind}}_{\mathsf{XCAU}}(A) \leq \frac{up(\ell_{\mathrm{blk}}(q_e + q_v) + 1)}{2^{\lambda+\kappa+1}} + \frac{up(\ell_{\mathrm{blk}}(q_e + q_v) + 1)^2}{2^{\lambda+1}} + uq_v \cdot \epsilon(\ell_{\mathrm{bit}}, \ell_{\mathrm{head}}) + \frac{u(u-1)}{2^{\kappa+1}},$$

for $\ell_{\mathrm{blk}} = \lceil \ell_{\mathrm{bit}}/\lambda \rceil + 1$, and with headers of length at most $\lambda \ell_{\mathrm{head}}$ bits.

Proof: As in the previous proofs, we restrict ourselves to adversaries A that do not make trivially invalid queries. The first step, as in the proof of Theorem 5.6, is to change the game $\mathbf{G}^{\text{mu-ind}}_{\mathsf{XCAU}}(A)$ in an equivalent way by introducing an additional oracle RF that implements the same ideal cipher as E and E^{-1} and is used in the algorithms XCAU.Enc and XCAU.Dec, call this game G_0. The next step, still as in Theorem 5.6, is to re-write the oracles Enc, Dec, and RF such that the input and output whitening of the block cipher is performed within RF, this game is called G_1 and the changes are analogous to the modifications in G_1 in Theorem 5.6. The next game G_2 is then obtained by replacing RF with an independent instance of an ideal cipher, this is the same as G_1 in Theorem 6.2. Both games have a flag bad that is set when the oracles RF and $\mathrm{E}/\mathrm{E}^{-1}$ diverge.

The probability of setting the bad flag in G_1 and G_2 can be bounded using Theorem 5.5. More technically, we describe an adversary $B = B(A)$ that emulates oracles to A as follows: Queries New, E, and E^{-1} by B are responded by B performing the same query in its game. Queries Enc and Dec are responded by B emulating the respective oracles using the oracle RF in its game to evaluate CAU.Enc and CAU.Dec. The view of A is the same in G_1 and in the game $\mathrm{R}(B(A))$, and in G_2 and the game $\mathrm{S}(B(A))$, respectively. The numbers of queries u to the New oracle and p to the E and E^{-1} oracles are preserved by B. At most q_e queries of length at most ℓ_{bit} to Enc and at most q_v queries of length at most $\ell_{\mathrm{bit}} + \lambda$ to Vf translate into at most $\ell_{\mathrm{blk}}(q_e + q_v) + 1$ queries to RF in the game played by B. Using the claim, this means that the probability of setting bad can be bounded by $up(\ell_{\mathrm{blk}}(q_e + q_v) + 1)/2^{\lambda+\kappa+1}$.

We now conclude the proof similarly to the one in Theorem 6.2. We observe that game G_2 is almost equivalent to $\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}$ without access to the oracle E and E^{-1}, because these oracles are independent of all other oracles in the game, which makes them easy to simulate. We therefore describe an adversary $B = B(A)$ in the game $\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}$ that simulates the oracles E and E^{-1} to the adversary A and forwards all other queries to its own oracles. Clearly, $\Pr[\mathrm{G}_2] = \Pr[\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}(B(A))]$, and therefore

$$\begin{aligned} \mathsf{Adv}^{\text{mu-ind}}_{\mathsf{XCAU}}(A) &= 2\Pr[\mathbf{G}^{\text{mu-ind}}_{\mathsf{XCAU}}(A)] - 1 = 2\Pr[\mathrm{G}_0] - 1 = 2\Pr[\mathrm{G}_1] - 1 \\ &\leq 2\Pr[\mathbf{G}^{\text{mu-ind}}_{\mathsf{CAU}}(B(A))] - 1 + \frac{up(\ell_{\mathrm{blk}}(q_e + q_v) + 1)}{2^{\lambda+\kappa+1}} \end{aligned}$$

$$\leq \frac{u(u-1)}{2^{\kappa+1}} + \frac{u((q_e+q_v)\cdot\ell_{\mathrm{blk}})^2}{2^{\lambda+1}} + uq_v\cdot\epsilon(\ell_{\mathrm{bit}},\ell_{\mathrm{head}}) + \frac{up(\ell_{\mathrm{blk}}(q_e+q_v)+1)}{2^{\lambda+\kappa+1}},$$

which concludes the proof. ∎

7 Conclusion

TLS 1.2 is the most widely used cryptographic protocol in the Internet, but due to issues with both performance and security, it will soon be replaced by its successor, TLS 1.3. Given that the bulk of Internet traffic will likely be protected by TLS 1.3 in the next years, it is extremely important that the security of the protocol is well-understood. Facing the threat of mass surveillance and the expected great number of TLS 1.3 sessions, the TLS Working Group has introduced a nonce-randomization technique to improve the resilience of TLS 1.3 against such attacks.

We show that the proposed technique can be understood as a key-length extension for AE; it essentially extends the 128-bit key of AES-GCM to a 224-bit key. We first describe the authenticated encryption CAU (Counter mode Almost Universal) as an abstraction of GCM. We then describe the scheme with randomized nonces as its variant RCAU and analyze it in the multi-user setting, where we show that it improves the resilience against (passive) mass surveillance as intended by the designers. We also show, however, that the AE does not perform as well as one might expect from an AE with a 224-bit key, especially in presence of active attacks. One alternative would be to simply increase the key size by, e.g., switching to an AES-256-based mode; this achieves better security but also impacts performance.

We suggest a new encryption mode that we call XCAU. The mode uses an additional 128-bit key (256 bits in total) to randomize the inputs and outputs of the block cipher (here AES) as in DESX. The mode is almost as efficient as the mode RCAU used in TLS 1.3, only adding two 128-bit xor operations for each call to the block cipher over plain CAU, our abstraction for GCM. We show that, still, its security is improved over RCAU in two ways. The security bounds we prove for security of XCAU *against active attacks* scale significantly better in the number u of users than those for RCAU, this stems mostly from the fact that *all* inputs to the block cipher are randomized. Furthermore, the whitening of the block-cipher output allows to remove the (for realistic parameters largest) term $p_i/2^{\kappa}$ from the security bound. (It should be noted, however, that this term is not worrisome for realistic parameters.) The fact that the implementation of XCAU, in contrast to that of RCAU, requires non-black-box changes to the libraries implementing CAU, however, makes adoption in the currently developed standard TLS 1.3 difficult.

References

[1] C. Badertscher, C. Matt, U. Maurer, P. Rogaway, and B. Tackmann. Augmented secure channels and the goal of the TLS 1.3 record layer. In M. H. Au and A. Miyaji, editors, *ProvSec 2015*, volume 9451 of *LNCS*, pages 85–104. Springer, Heidelberg, Nov. 2015. 5

[2] M. Bellare, D. J. Bernstein, and S. Tessaro. Hash-function based PRFs: AMAC and its multi-user security. In *Advances in Cryptology–EUROCRYPT 2016*, pages 566–595. Springer, 2016. 5

[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, 6

[4] M. Bellare, R. Canetti, and H. Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th FOCS*, pages 514–523. IEEE Computer Society Press, Oct. 1996. 5

[5] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. Springer, Heidelberg, Dec. 2000. 3, 8, 10

[6] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Heidelberg, May / June 2006. 6, 12, 13, 23, 26

[7] D. J. Bernstein. Multi-user Schnorr security, revisited. Cryptology ePrint Archive, Report 2015/996, 2015. `http://eprint.iacr.org/2015/996`. 6

[8] M. K. Boyarsky. Public-key cryptography and password protocols: The multi-user case. In *ACM CCS 99*, pages 63–72. ACM Press, Nov. 1999. 6

[9] Y. Dodis, P. J. Lee, and D. H. Yum. Optimistic fair exchange in a multi-user setting. In T. Okamoto and X. Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 118–133. Springer, Heidelberg, Apr. 2007. 6

[10] M. Dworkin. Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality. NIST Special Publication 800-38C, May 2004. 3

[11] M. Dworkin. Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D, November 2007. 3, 5, 10

[12] S. Even and Y. Mansour. A construction of a cipher from a single pseudorandom permutation. *Journal of Cryptology*, 10(3):151–162, 1997. 5, 11

[13] M. Fischlin, F. Günther, G. A. Marson, and K. G. Paterson. Data is a stream: Security of stream-based channels. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 545–564. Springer, Heidelberg, Aug. 2015. 5

[14] P.-A. Fouque, A. Joux, and C. Mavromati. Multi-user collisions: Applications to discrete logarithm, Even-Mansour and PRINCE. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 420–438. Springer, Heidelberg, Dec. 2014. 6

[15] S. Galbraith, J. Malone-Lee, and N. P. Smart. Public key signatures in the multi-user setting. *Information Processing Letters*, 83(5):263–266, 2002. 6

[16] Q. Huang, G. Yang, D. S. Wong, and W. Susilo. Efficient optimistic fair exchange secure in the multi-user setting and chosen-key model without random oracles. In T. Malkin, editor, *CT-RSA 2008*, volume 4964 of *LNCS*, pages 106–120. Springer, Heidelberg, Apr. 2008. 6

[17] 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, 10, 22

[18] J. Kilian and P. Rogaway. How to protect DES against exhaustive key search (an analysis of DESX). *Journal of Cryptology*, 14(1):17–35, 2001. 5, 11

[19] E. Kiltz, D. Masny, and J. Pan. Optimal security proofs for signatures from identification schemes. Cryptology ePrint Archive, Report 2016/191, 2016. `http://eprint.iacr.org/`. 6

[20] H. Krawczyk. LFSR-based hashing and authentication. In Y. Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 129–139. Springer, Heidelberg, Aug. 1994. 6

[21] T. Krovetz and P. Rogaway. The software performance of authenticated-encryption modes. In A. Joux, editor, *FSE 2011*, volume 6733 of *LNCS*, pages 306–327. Springer, Heidelberg, Feb. 2011. 3

[22] 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. 18, 19

[23] 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, 5, 10, 22

[24] 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. 5, 11

[25] C. Namprempre, P. Rogaway, and T. Shrimpton. Reconsidering generic composition. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 257–274. Springer, Heidelberg, May 2014. 3

[26] Y. Niwa, K. Ohashi, K. Minematsu, and T. Iwata. GCM security bounds reconsidered. In G. Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 385–407. Springer, Heidelberg, Mar. 2015. 5, 22

[27] P. Rogaway. Authenticated-encryption with associated-data. In V. Atluri, editor, *ACM CCS 02*, pages 98–107. ACM Press, Nov. 2002. 3, 6, 7, 8

[28] P. Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In P. J. Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 16–31. Springer, Heidelberg, Dec. 2004. 3

[29] P. Rogaway and M. Bellare. Robust computational secret sharing and a unified account of classical secret-sharing goals. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM CCS 07*, pages 172–184. ACM Press, Oct. 2007. 5

[30] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In *ACM CCS 01*, pages 196–205. ACM Press, Nov. 2001. 3

[31] 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. 8

[32] B. Smith. Pull request: Removing the AEAD explicit IV. Mail to IETF TLS Working Group, March 2015. 4

[33] 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. 5