Revisiting AES-GCM-SIV: Multi-user Security, Faster Key Derivation, and Better Bounds

Priyanka Bose[1], Viet Tung Hoang[2], and Stefano Tessaro[1]

[1] Dept. of Computer Science, University of California Santa Barbara.
[2] Dept. of Computer Science, Florida State University, USA.

Abstract. This paper revisits the multi-user (mu) security of symmetric encryption, from the perspective of delivering an analysis of the AES-GCM-SIV AEAD scheme. Our end result shows that its mu security is comparable to that achieved in the single-user setting, in a strong sense. In particular, even when instantiated with short keys (e.g., 128 bits), the security of AES-GCM-SIV is not impacted by the collisions of two user keys, as long as each individual nonce is not re-used by too many users. This is the first example of a scheme shown to have this property in the mu setting. Our bounds also substantially improve upon existing analyses (Gueron and Lindell, CCS '17; Iwata and Seurin, ePrint '17) *even* in the single-user setting, in particular when messages of variable lengths are encrypted. We also validate security when adopting an optimization of the key-derivation function of AES-GCM-SIV that *halves* its complexity (over the latest proposal).

On the way, we provide a number of results of independent interest. In particular, we advocate analyzing mu security in a setting where the data processed by every user is bounded, and where user keys are generated according to arbitrary, possibly correlated distributions. This model refines works on mu security by providing a more meaningful parameter regime, but also allows to unify this with a treatment of re-keying. In this model, we analyze the security of counter-mode encryption and Wegman-Carter MACs. We then lift these analyses to GCM-SIV and AES-GCM-SIV.

Keywords: Multi-user security, AES-GCM-SIV, authenticated encryption, concrete security

1 Introduction

This work continues the study of the *multi-user (mu) security* of symmetric cryptography, the setting where the adversary distributes its resources to attack multiple instances of a cryptosystem, with the end goal of compromising at least one of them. This attack model was recently the object of extensive scrutiny [2, 9, 17, 18, 23, 26, 32], and its relevance stems from the en masse deployment of symmetric cryptographic schemes, e.g., within billions of daily TLS connections.

The main goal is to study the degradation in security as the number of users increases.

Our contributions. This paper will extend this line of work in different ways. The most tangible contribution is a complete analysis in the mu setting of the AES-GCM-SIV [14] scheme by Gueron, Langley, and Lindell, a scheme for authenticated encryption with associated data (AEAD) which is undergoing evaluation to become the object of an RFC. Our main result will show that the scheme's security does not degrade in the mu setting, in a sense much stronger than what was claimed in previous mu analyses. Also, we abstract the requirement needed for AES-GCM-SIV's key-derivation step, and show that a very simple KDF is sufficient for high security. Beyond this, our analysis also delivers conceptual and technical insights of wider interest.

Concretely, our result will highlight the benefit of ensuring limited nonce re-use across different users (e.g., by choosing nonces randomly). We show that in this setting AES-GCM-SIV does *not* suffer any impact from key-collisions, in particular allowing security to go beyond the Birthday barrier (wrt the key length) even in the multi-user setting. The resulting analysis is particularly involved, and calls for a precise understanding of the power of verification queries (for which nonce re-use *cannot* be restricted). Previous analyses of AE schemes (specifically, those of [9]) do not ensure security when two users have the same key, thus forcing either an increase of key length or a worse security guarantee.

On the way, we advocate a refined model of mu security where the amount of data processed by each user is bounded. This is consistent with the fact that either users would re-key after a certain amount of data has been processed, or, even without explicit re-keying, the lifespan of a key is generically short. The model also naturally lends itself to a modular treatment of nonce-based re-keying in AES-GCM-SIV. This will, in particular, rely on analyses in this regime of (randomized) counter-mode encryption, as well as of the universal-hash based PRF underlying AES-GCM-SIV.

We now continue with a more detailed overview of our results.

Multi-user security. The notion of *multi-user* (mu) security was introduced by Bellare, Boldyreva and Micali [3] for public-key encryption, although in the symmetric setting the notion already appeared implicitly earlier [4].

For example, in the mu definition of encryption security under chosen plaintext attacks, each user i is assigned a (secret) key K_i, and the attacker can make encryption queries $\textsc{Enc}(i, M)$, which result in either an encryption of M under K_i (in the real world), or an equally long random ciphertext (in the ideal world). The goal is to distinguish what is the case.

Assessing security in this model is interesting and non-trivial. Take for example randomized counter-mode encryption (CTR), based on a block cipher with key length k and block length n, which we model as an ideal cipher. For any *single-user* adversary encrypting, in total, L blocks of data and making p queries to the ideal cipher, the advantage is bounded by $\epsilon_{su}(L, p) \leq \frac{L^2}{2^n} + \frac{p}{2^k}$ (a proof was given e.g. in [5]). If the attacker now adaptively distributes its queries across up to u users, a simple hybrid argument shows that there can be at most a multiplicative-factor loss u in the bound, or more precisely, the bound is now $\epsilon_{mu}(L,p,u) \leq u \cdot \epsilon_{su}(L,p+L) \leq \frac{2uL^2}{2^n} + \frac{u(p+L)}{2^k}$. Usually, we do not want to force any particular u, and would like the adversary to encrypt its budget of L blocks adaptively across as many users as it sees fit. And a-priori, we cannot prevent the adversary to pick one of the following worst-case choices: (1) Querying one message only with length L, or (2) Querying L messages with length 1. Thus, in the worst case, the bound becomes $\epsilon_{mu}(L,p) \leq \frac{2L^3}{2^n} + \frac{Lp+L^2}{2^k}$. A number of recent works [2,17,18,26,32] have shown (mostly, in the context of PRF security) that this is an overly pessimistic point of view: the security loss can be much smaller, and often $\epsilon_{mu}(L,p) \approx \epsilon_{su}(L,p)$ holds.

A fresh look at mu security. Paradoxically, this ideal-outcome scenario could still be somewhat pessimistic. Indeed, it is well possible that for CTR (or any other scheme) in the mu setting, even if $\epsilon_{mu}(L,p) \approx \epsilon_{su}(L,p)$, the matching attack is a single-user attack, requiring a single honest user to encrypt $L \approx 2^{n/2}$ blocks under the same key. For $k = n = 128$, this boils down to multiple exabytes of data to be encrypted with the same key. This is not likely, regardless of the computational power of the adversary, because users may re-key, or simply because a user could not be brought to encrypt that much. If we assume instead an upper bound B on the number of blocks encrypted by each user, an L-block adversary for $L > B$ would be forced to spread its effort across at least L/B users. As one of our first results, we show that for CTR, the advantage of such an attacker is upper bounded by

$$\frac{LB}{2^n} + \frac{L^2}{2^{n+k}} + \frac{ap}{2^k} \; .$$

for some constant a. This bound already shows some themes we will lift to AES-GCM-SIV later such as: (1) Beyond-birthday security is possible, e.g., for $k = n = 128$ and $B = 2^{32}$, the bound is of the order $L/2^{96} + p/2^{128}$; (2) The role of local computation – captured here by the number of ideal-cipher queries p – is independent of L, and the number of users, and very similar to the single-user case, and (3) The bound is independent of the number of users. Previous results on mu security target deterministic security games, such as PRFs/PRPs [2, 17, 18, 26, 32] or deterministic AE [9, 23], and security falls apart when more than $2^{k/2}$ users are present, and their keys collide. Here, key-collisions are irrelevant, and security well beyond $2^{k/2}$ users is possible. This viewpoint generalizes that of Abdalla and Bellare [1], who were first to observe, in a simpler model, that re-keying after encrypting B blocks increases security. In a way, what we argue here is that analyses of schemes under re-keying, and under multi-user attacks, are essentially addressing two interpretations of the same technical problem once the right modeling is adopted.

In this model, we will also derive a bound for the PRF security of GMAC^+, the PRF underlying AES-GCM-SIV. Eventually, we will use these analyses for our final result on AES-GCM-SIV.

AES-GCM-SIV: Prior work and new mu bounds. AES-GCM-SIV pushes the re-keying philosophy a bit further, making it *nonce based* – i.e., to encrypt a message M with a nonce N, we first derive nonce-key K_N from the master key and N, using a key-derivation function KD, and then encrypt the message M with nonce N under key K_N using a base AE scheme AE. The intuition is that here nonces play the roles of users, and a security bound in the mu setting for AE should become a bound on the end scheme in the *single-user* setting, where now B is a bound on the amount of data encrypted *per nonce*, rather than per user. In particular, existing analyses [16, 20] exploit this, and for the single-user security as a nonce-misuse resistant scheme (so-called mrae security) of AES-GCM-SIV with key length 128 bits, they show an advantage bound of order

$$\frac{Q}{2^{96}} + \frac{QB^2}{2^{128}} + \frac{\ell_{\max} QR}{2^{128}} + \frac{p}{2^{128}} ,$$

for any adversary that makes at most p ideal-cipher queries, encrypts at most B blocks *per nonce*, uses at most $Q < 2^{64}$ nonces in encryption/verification queries, where R is the maximum number of repetition of a nonce, and $\ell_{\max}$ is the maximal length of a verification query.

Here, for suitable instantiations of the key-derivation function (which includes in particular simpler ones than those currently under consideration) we will show that the upper bound for the *multi*-user mrae security of AES-GCM-SIV is of order

$$\frac{LB}{2^{128}} + \frac{d(p+L)}{2^{128}} ,$$

where L is an upper bound on the overall number of encrypted/verified blocks, B is a bound on the number of blocks encrypted per user-nonce pair, and d is a bound on the the number of users across which any nonce is re-used. This shows a number of surprising things: First off, our bound is an improvement even in the single-user case, as $d = 1$ vacuously holds, and even if we use the KDF considered in previous works. The term $\frac{LB}{2^{128}}$ can be much smaller than $\frac{QB^2}{2^{128}}$, as in many settings Q and L can be quite close (e.g., if most messages are very short). In fact, the point is slightly more subtle, and to our advantage, and we elaborate on it at the end of the introduction. Second, if d is constant (which we can safely assume if nonces are randomly chosen), security does not degrade as the number of users increases. In particular, the security is unaffected by key collisions. If d cannot be bounded, we necessarily need to increase the key to 256 bits, and in this case the second term becomes $\frac{d(p+L)}{2^{256}}$. Finally, we have no assumption on the data amount of verification queries per user-nonce pair (other than the overall bound L), whereas the bounds in prior works can become weak if there is a very long verification query, and the adversary uses only a single nonce among verification queries.[3]

[3] Since the penalty of verification queries can be very high, [16] has to *assume* that the term $\ell_{\max} QR/2^n$ is smaller than $QB^2/2^n$; this assumption might hold in settings where servers adopt DDOS countermeasures to terminate connections of too many invalid verification queries.

This shows that AES-GCM-SIV enjoys great quantitative security in the mu setting, and in fact presents a number of unique features. The rest of the introduction will explain how we get to our result.

Review of AES-GCM-SIV. Let us first however review AES-GCM-SIV [14] a little more in detail. The scheme is based on GCM-SIV$^+$, a slight modification of GCM-SIV, proposed in [15]. This relies in turn on SIV ("synthetic IV") [31], an AEAD scheme which combines a PRF F and an encryption scheme SE (only meant to be CPA secure) to achieve nonce-misuse resistance. For message M, nonce N, and associated data A, the encryption of SIV results into a ciphertext C obtained as

$$\mathsf{IV} \leftarrow \mathsf{F}(K_{\mathsf{F}}, (M, N, A)) \ , \ \ C \leftarrow \mathsf{SE.E}(K_{\mathsf{E}}, M; \mathsf{IV}) \ ,$$

where K_{F} and K_{E} are the two components of the secret key, and $\mathsf{SE.E}(K_{\mathsf{E}}, M; \mathsf{IV})$ is the deterministic encryption function of SE run with IV IV.

In GCM-SIV$^+$, SE is counter mode, and F is what we call GMAC^+, a Wegman-Carter MAC [34] similar to, but different from, the one used in GCM [25]. It composes an xor-universal hash function with n-bit key, with a block cipher with block length n and key length k. GMAC^+'s total key length is hence $k + n$ bits. (As we target AES, $n = 128$ and $k \in \{128, 256\}$.) A difference from the original SIV scheme is that the same block cipher key is used across GMAC^+ and counter-mode, but appropriate domain separation is used.

Finally, AES-GCM-SIV enhances GCM-SIV$^+$ via *nonce-based key derivation.* That is, now we have a single key K, and to encrypt with a nonce N, we *first* derive a $(n + k)$-bit key $K_N \leftarrow \mathsf{KD}(K, N)$ using a key-derivation function KD, and then encrypt using GCM-SIV$^+$ with the key K_N.

Mu-security and nonce-based key derivation. We will be able to combine our analyses for CTR and GMAC^+ into an analysis for GCM-SIV$^+$ in the bounded mu model. This is less obvious than it would initially seem, because due to the key re-use, the technique for generic composition used in the original SIV scheme fails. The leading terms here are of order similar to the above bound for CTR.

The question is now whether nonce-based key derivation achieves its purpose in the mu setting, where B is now a bound on the number of blocks encrypted per nonce-user pair. Indeed, say the master secret key K has length $k = 128$. Then, should the number of users exceed $2^{k/2} = 2^{64}$, with high probability two users will end up with *identical* keys. If we treat KD as a PRF, like [16, 20] do, all security will vanish at this point. Indeed, the existing mu analysis of GCM succumbs to this problem [9], and the problem seem unavoidable here too, since we are considering a deterministic security game.

Bounded nonce re-use. The way out from this problem leverages the setting where nonces are not re-used too often, which is what we are concerned with, and in particular we assume a nonce is re-used by at most d users. Consider the canonical attack to break privacy of the scheme: Fix a sufficiently long message M and nonce N, and encrypt them over and over for different users, and if the same ciphertext appears twice after roughly $2^{k/2}$ queries, we know we are likely to be in the real world, as this is not likely to happen in the ideal world, where ciphertexts are random. This however requires us to *re-use* the same nonce roughly $2^{k/2}$ times for different users. A first interesting point we observe is that KD need not become insecure after $2^{k/2}$ queries as a PRF if the number of re-uses of a nonce is sufficiently small. We will confirm this to be true for a large class of KDF constructions we consider (see below).

Unfortunately, this is not enough. The catch is that the above argument only applies to privacy. We need to guarantee authenticity, too, and even though we are bounding the number of re-uses of a nonce in encryption queries, it is not meaningful to do the same for verification queries. It seems that key collisions are back to haunt us. However, we will show that this is not necessarily the case.

To get some intuition, consider the security of KD as a MAC, i.e., the adversary issues, in a first stage, queries (i, N), getting $\mathsf{KD}(K_i, N)$, but respecting the constraint that no nonce is used more than d times across different i's where d is relatively small. Then, in a second stage, the adversary gets to ask unrestricted verification queries with input (i, N, T), except for the obvious requirement that (i, N) must be previously un-queried. The adversary wins if $\mathsf{KD}(K_i, N) = T$ for one of these verification queries. At first glance, a collision $K_i = K_j$ could help if we have queried (i, N) in the first stage, learnt T, and now can submit (j, N, T) in the second. Unfortunately, however, we need to be able to have detected such collisions. This is hard to do during the first stage, even with many queries, due to the constraint of reusing N only d times. Thus, the only obvious way to exploit this would be to try, for each of the q first-stage queries (i, N) with corresponding output T, to query (j, N, T) for many $j \neq i$. This would however require roughly 2^k trials to succeed. Finally, note that while it may be that we ask two verification queries (i, N, T) and (j', N', T') where $K_i = K_j$, this does not seem to give any help in succeeding, because a verification query does not reveal the actual output of KD on that input.

Confirming this intuition is *not* simple. We will however do so for a specific class of natural KD constructions outlined below, and point out that the setting of AE is even harder than studying the security of KD itself as a MAC. Indeed, our KD is used to derive keys for GMAC^+ and CTR *at the same time*, and we need to prove unpredictability of the overall encryption scheme on a new pair (N, i) which was previously unqueried. This is the most technically involved part of the paper.

A SIMPLER KDF. Finally, let us address *how* we instantiate KD. The construction of KD from [14] is truncation based, and makes 4 (for $k = 128$), respectively 6 (for $k = 256$) calls to a block cipher to derive a key. A recent proposal [20] suggests using the so-called XOR construction to achieve higher security, as multiple analyses [7, 11, 22, 28, 30] confirm better bounds than for truncation [12]. Still, the resulting KD would need 4 resp. 6 calls. They also consider a faster construction, based on CENC [19], which would require 3 resp. 4 calls.

Rather than following the route of analyzing these concrete constructions, we apply our result to a general class of KDFs which includes in particular all of these proposals, but also simpler ones. For instance, our bounds apply to the following simple KDF, a variant of which was in the initial AES-GCM-SIV proposal, but was discarded due to security concerns.[4] Namely, given the underlying block cipher E, the KDF outputs

$$\mathsf{KD}(K, N) = E(K, \mathsf{pad}(N, 0)) \parallel E(K, \mathsf{pad}(N, 1)) \tag{1}$$

for $k = n$ and N an nl-bit string, with $\mathsf{nl} \leq n - 2$, and, analogously, for $k = 2n$, one can extend this by additionally concatenating $E(K, \mathsf{pad}(N, 2))$. Here, pad is a mapping with the property that the sets $\{\mathsf{pad}(N, 0), \mathsf{pad}(N, 1), \mathsf{pad}(N, 2)\}$ defined by each N are disjoint. This approach seems to contradict common sense which was adopted in the new KDF variants for AES-GCM-SIV, because the derived keys are not truly random. The point is that existing proofs (implicitly or explicitly) leverage the multi-user security of E, in a setting with independent random secret keys. However, we observe (validating this in the ideal-cipher model) that mu security bounds are robust to deviation of the key distribution, and in particular additional constraints of sub-keys and part of them being distinct generally only make matters better, rather than worse.[5]

We note that a crucial point of our analyses is that we do not prove PRF security of these KDFs. Rather, we study the distributions on keys they induce, and then (implicitly) rely on the security of the underlying components using keys obtained from (slightly) non-uniform distributions.

In platforms that support AES hardware acceleration, the difference in performance between the KDF in Equation (1) and the current one in AES-GCM-SIV is not important, as demonstrated via the experiments in [14]. Still, we believe it is important for schemes to be minimal, and thus to understand the security of simplest possible instantiations of the KDF.

Sub-optimality of POLYVAL. We also observe that the universal hash POLYVAL of AES-GCM-SIV suffers from a suboptimal design issue. That is, if both the message and the associated data are the empty string, then their hash image under POLYVAL is always 0^{128}, regardless of the hash key. This does not create any issue in the single-user setting, and thus has not received any attention so far. However, in the multi-user setting, it substantially weakens the security of GCM-SIV$^+$ and GMAC^+ to $\frac{LB}{2^{128}} + \frac{d(p+L)}{2^{128}}$, despite their use of 256-bit keys. Had the padding in POLYVAL been done properly so that the hash image of empty strings under a random key has a uniform distribution, the security of GCM-SIV$^+$ and GMAC^+ could be improved to $\frac{LB}{2^{128}} + \frac{Lp}{2^{256}}$, meaning this bound is independent of the number d of users that reuse any particular nonce. While this issue does not affect the concrete security bound of AES-GCM-SIV,

[4] Thus, our analysis shows that this proposal would have been a good and more efficient choice.

[5] A similar key-derivation scheme has been used to derive sub-keys from tweaks in the setting of FPE within the DFF construction [33]. This was then formalized and studied in [6], though we stress that their analysis is quite different from ours, and considers a much less demanding setting.

this change is recommended especially if GCM-SIV$^+$ or GMAC^+ are used as standalone schemes.

Relation to existing works. As mentioned above, two recent papers [16,20] have analyzed AES-GCM-SIV in the su setting, and we want to elaborate further on our improvement in this special case. As we argue above, their bound contains a term of the order $QB^2/2^n$, which we improve to $LB/2^n$. The fact that the latter is better is not quite obvious. Indeed, it is not hard to improve the term $QB^2/2^n$ in [16,20] to $\sum_{i=1}^{Q} B_i^2/2^n$, where B_i is a bound on the number of blocks encrypted with the i-th nonce. This seems to address the point that different amounts of data can be encrypted for different nonces.

The crucial point is that we capture a far more general class of attacks by only limiting the adversary in terms of L, p, and d. For instance, for a parameter L, consider the following single-user adversary using $Q = L/2$ nonces. It will select a random subset of the Q nonces, of size $L/(2B)$, for which it encrypts B blocks of data, and for the remaining $L/2 - L/(2B)$ nonces, it only encrypts *one* block of data. In our bound, we still get a term $LB/2^n$. In contrast, with the parametrization adopted by [16,20], we can only set $Q = L/2$ and $B_i = B$ for all $i \in [Q]$, because *any* of the nonces can, a priori, be used to encrypt B blocks. This ends up giving a term of magnitude $LB^2/2^n$, however, which is much larger. For $B = 2^{32}$, the difference between $L/2^{64}$ and $L/2^{96}$ is enormous.

Switching to the type of bounds we consider is not just aesthetics: The adversary can of course make even more convoluted, adaptive choices in its attack pattern. The analysis needs to handle these, and this is non-trivial. This type of question was the object of several recent works in the mu regime [2,17,18,23,26,32].

Standard vs ideal-model. We also note that the bound of [20] is expressed in the standard model, and contains a term $Q\epsilon$, where ϵ is the advantage of a PRF adversary $\mathcal{A}'$ against the cipher E, making B queries. The catch is that ϵ is very sensitive to the *time* complexity of $\mathcal{A}'$, which we approximate with the number of ideal-cipher queries p. Thus, $Q\epsilon$ is of order $Q(B^2/2^n + p/2^k)$. While [20] argues that $QB^2/2^n$ is the largest term, the ideal model makes it evident that the hidden term $Qp/2^k$ is likely to be far more problematic in the case $n = k$. Indeed, $p \geq Q$ and $B^2 \leq Q$ are both plausible (the attacker can more easily invest local computation than obtain honest encryptions under equal nonces), and this becomes $\frac{Q^2}{2^k}$. This shows security is bounded by $2^{k/2}$. The work of [23] on classical GCM also seemingly focuses on the standard model and thus seems to fail to capture such hidden terms. In contrast, [16] handles this properly.

We stress that we share the sentiment that ideal-model analysis may oversimplify some security issues. However, we find them a necessary evil when trying to capture the influence of local computation in multi-user attacks, which is a fundamental part of the analysis.

Outline of this paper. We introduce basic notions and security definitions in the multi-user setting in Section 3. Then, in Section 4, we study the security of our basic building blocks, CTR and GMAC^+, in the multi-user setting. In

Section 5, we analyze SIV composition when keys are re-used across encryption and PRF, and observe this to work in particular for the setting of GCM-SIV. Finally, Section 6 studies our variant of AES-GCM-SIV with more general key derivation.

2 Preliminaries

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

Systems and Transcripts. Following the notation from [17] (which was in turn inspired by Maurer's framework [24]), it is convenient to consider interactions of a distinguisher A with an abstract system $\mathbf{S}$ which answers A's queries. The resulting interaction then generates a transcript $\tau = ((X_1, Y_1), \ldots, (X_q, Y_q))$ of query-answer pairs. It is well known that $\mathbf{S}$ is entirely described by the probabilities $\mathsf{p}_{\mathbf{S}}(\tau)$ that if we make queries in τ to system $\mathbf{S}$, we will receive the answers as indicated in τ.

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

The H-coefficient technique. We now describe the H-coefficient technique of Patarin [10, 29]. Generically, it considers a deterministic distinguisher $\mathcal{A}$, interacting with system $\mathbf{S}_0$ or with system $\mathbf{S}_1$. Let $\mathcal{X}_0$ and $\mathcal{X}_1$ be random variables for the transcripts defined by these interactions with $\mathbf{S}_0$ and $\mathbf{S}_1$, and a bound on the distinguishing advantage of $\mathcal{A}$ is given by the statistical distance $\mathsf{SD}(\mathcal{X}_0, \mathcal{X}_1)$.

Lemma 1. [10, 29] *Supposed we can partition transcripts into good and bad transcripts. Further, suppose that there exists $\epsilon \ge 0$ such that $1 - \frac{\mathsf{p}_{\mathbf{S}_0}(\tau)}{\mathsf{p}_{\mathbf{S}_1}(\tau)} \le \epsilon$ for every good transcript τ such that $\mathsf{p}_{\mathbf{S}_1}(\tau) > 0$. Then,*

$$\mathsf{SD}(\mathcal{X}_1, \mathcal{X}_0) \le \epsilon + \Pr[\mathcal{X}_1 \text{ is bad}] \ .$$

3 Multi-user Security of Symmetric Primitives

We revisit security definitions for basic symmetric primitives in the multi-user setting. We will in particular extend existing security definitions to impose overall bounds on the volume of data processed by each user, however we will relegate this matter to theorem statements restricting the considered adversaries, rather than hard-coding these bounds in the definitions.

3.1 Symmetric and Authenticated Encryption

We define AE syntax here, as well as natural multi-user generalizations of classical security notions for confidentiality and integrity. Since this paper will deal both with probabilistic and deterministic schemes, we define both, following the treatment of Namprempre, Rogaway, and Shrimpton [27]. Our notational conventions are similar to those from [9].

IV-BASED ENCRYPTION. An *IV-based symmetric encryption* scheme SE consists of two algorithms, the *randomized* encryption algorithm $\mathsf{SE.E}$ and the deterministic *decryption algorithm* $\mathsf{SE.D}$, and is associated with a corresponding key length $\mathsf{SE.kl} \in \mathbb{N}$ and initialization-vector (IV) length $\mathsf{SE.vl} \in \mathbb{N}$. Here, $\mathsf{SE.E}$ takes as input a secret key $K \in \{0,1\}^{\mathsf{SE.kl}}$ and a plaintext $M \in \{0,1\}^*$. It then samples $\mathsf{IV} \leftarrow\$ \{0,1\}^{\mathsf{SE.vl}}$, deterministically computes a ciphertext core C' from K, M and IV, and returns $C \leftarrow \mathsf{IV} \parallel C'$. We often write $C \leftarrow\$ \mathsf{SE.E}_K(M)$ or $C \leftarrow\$ \mathsf{SE.E}(K, M)$. If we want to force the encryption scheme to run on a specific initialization vector IV, then we write $\mathsf{SE.E}(K, M; \mathsf{IV})$. The corresponding decryption algorithm $\mathsf{SE.D}$ takes as input a key $K \in \{0,1\}^{\mathsf{SE.kl}}$ and a ciphertext $C \in \{0,1\}^*$, returns either a plaintext $M \in \{0,1\}^*$, or an error symbol $\perp$. For correctness, we require that if C is output by $\mathsf{SE.E}_K(M)$, then $\mathsf{SE.D}_K(C)$ returns M. We allow all algorithms to make queries to an ideal primitive Π, in which case this will be made explicit when not clear from the context, e.g., by writing $\mathsf{SE}[\Pi]$ in lieu of SE.

CHOSEN-PLAINTEXT SECURITY FOR IV-BASED ENCRYPTION. We re-define the traditional security notion of ind-security for the multi-user setting. Our definition will however incorporate a general, stateful *key-generation* algorithm KeyGen which is invoked every time a new user is spawned via a call to the NEW oracle. KeyGen is a parameter of the game, and it takes additionally some input string aux which is supplied by the adversary. The traditional mu security setting would have KeyGen simply output a random string, and ignore aux, but we will consider a more general setting to lift mu bounds to the key-derivation setting. The game is further generalized to handle an arbitrary ideal primitive (an ideal cipher, a random oracle, or a combination thereof) via an oracle PRIM.[6] Also note that the oracle PRIM can simply trivially provide no functionality, in which case we revert to the standard-model definition. We note that the key-generation algorithm KeyGen does not have access to the oracle PRIM.

[6] If PRIM is meant to handle multiple primitives, we assume they can be accessed through the same interface by pre-pending to the query a prefix indicating which primitive is meant to be queried.

Game $\mathbf{G}^{\text{mu-ind}}_{\mathsf{SE},\mathsf{KeyGen},\Pi}(\mathcal{A})$

$\mathsf{st}_0 \leftarrow \varepsilon$; $v \leftarrow 0$; $b \leftarrow\!\!\$\ \{0,1\}$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Prim}}$
Return $(b' = b)$

$\textsc{New}(\mathsf{aux})$

$v \leftarrow v + 1$
$(K_v, \mathsf{st}_v) \leftarrow\!\!\$\ \mathsf{KeyGen}(\mathsf{st}_{v-1}, \mathsf{aux})$

$\textsc{Enc}(i, M)$

If $i \notin \{1, \ldots, v\}$ then return $\perp$
$C_1 \leftarrow\!\!\$\ \mathsf{SE.E}^{\textsc{Prim}}(K_i, M)$
$C_0 \leftarrow\!\!\$\ \{0,1\}^{|C_1|}$
Return C_b

Game $\mathbf{G}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A})$

$\mathsf{st}_0 \leftarrow \varepsilon$; $v \leftarrow 0$; $b \leftarrow\!\!\$\ \{0,1\}$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Vf},\textsc{Prim}}$
Return $(b' = b)$

$\textsc{Vf}(i, N, C, A)$

If $i \notin \{1, \ldots, v\}$ then return $\perp$
If $(i, N, C, A) \in V[i]$ then return true
If $b = 0$ then return false
$M \leftarrow \mathsf{AE.D}^{\textsc{Prim}}(K_i, N, C, A)$
Return $(M \neq \perp)$

$\textsc{New}(\mathsf{aux})$

$v \leftarrow v + 1$
$(K_v, \mathsf{st}_v) \leftarrow\!\!\$\ \mathsf{KeyGen}(\mathsf{st}_{v-1}, \mathsf{aux})$

$\textsc{Enc}(i, N, M, A)$

If $i \notin \{1, \ldots, v\}$ then return $\perp$
If $(i, N, M, A) \in U[i]$ then return $\perp$
$C_1 \leftarrow \mathsf{AE.E}^{\textsc{Prim}}(K_i, N, M, A)$
$C_0 \leftarrow\!\!\$\ \{0,1\}^{|C_1|}$
$U[i] \leftarrow U[i] \cup \{(i, N, M, A)\}$
$V[i] \leftarrow V[i] \cup \{(i, N, C_b, A)\}$
Return C_b

Fig. 1: **Security definitions for chosen-plaintext security of IV-based encryption (top), as well as nonce-misuse resistance for authenticated encryption (bottom).** We assume (without making this explicit) that PRIM implements the ideal-primitive Π.

Given an adversary $\mathcal{A}$, the resulting game is $\mathbf{G}^{\text{mu-ind}}_{\mathsf{SE},\mathsf{KeyGen},\Pi}(\mathcal{A})$, and is depicted on the left of Figure 1. The associated advantage is

$$\mathsf{Adv}^{\text{mu-ind}}_{\mathsf{SE},\mathsf{KeyGen},\Pi}(\mathcal{A}) = 2 \cdot \Pr\left[\mathbf{G}^{\text{mu-ind}}_{\mathsf{SE},\mathsf{KeyGen},\Pi}(\mathcal{A})\right] - 1 .$$

Whenever we use the canonical KeyGen which outputs a random string regardless of its input, we will often omit it, and just write $\mathsf{Adv}^{\text{mu-ind}}_{\mathsf{SE},\Pi}(\mathcal{A})$ instead.

AUTHENTICATED ENCRYPTION SCHEME. An authenticated encryption scheme AE with associated data (also referred to as an AEAD scheme), the algorithms $\mathsf{AE.E}$ and $\mathsf{AE.D}$ are both deterministic. In particular, $\mathsf{AE.E}$ takes as input a secret key $K \in \{0,1\}^{\mathsf{AE.kl}}$, a *nonce* $N \in \{0,1\}^{\mathsf{AE.nl}}$, a plaintext $M \in \{0,1\}^*$, and the *associated data* A, and returns the ciphertext $C \leftarrow \mathsf{AE.E}(K, N, M, A)$. The corresponding decryption algorithm $\mathsf{AE.D}$ takes as input a key $K \in \{0,1\}^{\mathsf{AE.kl}}$, the nonce N, the ciphertext $C \in \{0,1\}^*$, and the associated data A, and returns either a plaintext $M \in \{0,1\}^*$, or an error symbol $\perp$. We require that if C is output by $\mathsf{AE.E}_K(M, N, A)$, then $\mathsf{AE.D}_K(C, N, A)$ returns M.

Our security notion for AE is nonce-misuse-resistant: Ciphertexts produced by encryptions with the same nonce are pseudorandom *as long as* the encryptions are on different messages or associated data, even if they are for the same nonce. Our formalization of AE multi-user security in terms of $\mathbf{G}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A})$ is that of Bellare and Tackmann [9], with the addition of a KeyGen algorithm to handle arbitrary correlated key distributions. It is depicted in Figure 1, at the bottom.

Given an adversary $\mathcal{A}$ and a key-generation algorithm KeyGen, we are then going to define

$$\mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}) = 2 \cdot \Pr\left[\mathbf{G}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A})\right] - 1 \ .$$

As above, KeyGen is omitted if it is the canonical one.

We say that an adversary is *d-repeating* if among the encryption queries, an adversary only uses each nonce for at most d users. We stress that we make no assumption on how the adversary picks nonces for the verification queries, and for each individual user, the adversary can repeat nonces in encryption queries as often as it wishes. If nonces are chosen arbitrarily then d can be as big as the number of encryption queries. If nonces are picked at random then d is a small constant.

A KEY-COLLISION ATTACK. We now show that for any AE scheme AE that uses the canonical KeyGen, if an adversary can choose nonces arbitrarily then there is an attack, using q encryption queries and no verification query, that achieves advantage $q(q-1)/2^{\mathsf{AE.kl}+3}$.

Suppose that under AE, a ciphertext is always at least as long as the corresponding plaintext. Fix an arbitrary message M such that $|M| \geq \mathsf{AE.kl} + 2$. Fix a nonce N and associated data A. The adversary $\mathcal{A}$ attacks q users, and for each user i, it queries $\textsc{Enc}(i, N, M, A)$ to get answer C_i. If there are distinct i and j such that $C_i = C_j$ then it outputs 1, hoping that users i and j have the same key. For analysis, we need the following well-known result; see, for example, [13, Chapter 5.8] for a proof.

Lemma 2 (Lower bound for birthday attack). *Let $q, N \geq 1$ be integers such that $q \leq \sqrt{2N}$. Suppose that we throw q balls at random into N bins. Then the chance that there is a bin of at least two balls is at least $\frac{q(q-1)}{4N}$.*

From Lemma 2 above, in the real world, the adversary will output 1 if two users have the same key, which happens with probability at least $q(q-1)/2^{\mathsf{AE.kl}+2}$. In contrast, since the ciphertexts are at least $|M|$-bit long, in the ideal world, it outputs 1 with probability at most $q(q-1)/2^{|M|+1} \leq q(q-1)/2^{\mathsf{AE.kl}+3}$. Hence

$$\mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\Pi}(\mathcal{A}) \geq \frac{q(q-1)}{2^{\mathsf{AE.kl}+2}} - \frac{q(q-1)}{2^{\mathsf{AE.kl}+3}} = \frac{q(q-1)}{2^{\mathsf{AE.kl}+3}} \ .$$

3.2 Multi-user PRF Security

We consider keyed functions $\mathsf{F} : \{0,1\}^{\mathsf{F.kl}} \times \{0,1\}^{\mathsf{F.il}} \to \{0,1\}^{\mathsf{F.ol}}$, possibly making queries to an ideal primitive Π. Here, note that we allow $\mathsf{F.il} = *$, indicating a

Game $\mathbf{G}^{\text{mu-prf}}_{\mathsf{F},\mathsf{KeyGen},\Pi}(\mathcal{A})$	$\textsc{New}(\mathsf{aux})$	$\textsc{Eval}(i, M)$
$v \leftarrow 0$; $\mathsf{st}_0 \leftarrow \emptyset$	$v \leftarrow v + 1$	If $i \notin \{1, \ldots, v\}$ return $\perp$
$b \leftarrow\!\!\$\ \{0, 1\}$	$(K_v, \mathsf{st}_v) \leftarrow\!\!\$\ \mathsf{KeyGen}(\mathsf{st}_{v-1}, \mathsf{aux})$	$Y_1 \leftarrow \mathsf{F}^{\textsc{Prim}}(K_i, M)$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\textsc{New},\textsc{Eval},\textsc{Prim}}$	$\rho_v \leftarrow\!\!\$\ \mathsf{Func}(\mathsf{F.il}, \mathsf{F.ol})$	$Y_0 \leftarrow\!\!\$\ \rho_i(M)$
Return $(b' = b)$	$K_v \leftarrow\!\!\$\ \{0, 1\}^{\mathsf{F.kl}}$	Return Y_b

Fig. 2: Definitions of multi-user PRF security. Again, Prim implements the ideal primitive Π.

variable-input-length function. We define a variant of the standard multi-user version of PRF security from [4] using (as in the previous section) a general algorithm KeyGen to sample possibly correlated keys.

Concretely, let $\mathsf{Func}(\mathsf{il}, \mathsf{ol})$ be the set of all functions $\{0, 1\}^{\mathsf{il}} \to \{0, 1\}^{\mathsf{ol}}$, where, once again, $\mathsf{il} = *$ is allowed. We give the multi-user PRF security game in Figure 2, on the left. There, F's access to Π is modeled by having oracle access to Prim, here. For any adversary $\mathcal{A}$, and key generation algorithm KeyGen, we define

$$\mathsf{Adv}^{\text{mu-prf}}_{\mathsf{F},\mathsf{KeyGen},\Pi}(\mathcal{A}) = 2 \cdot \Pr\left[\mathbf{G}^{\text{mu-prf}}_{\mathsf{F},\mathsf{KeyGen},\Pi}(\mathcal{A})\right] - 1 \; .$$

As usual, we will omit KeyGen when it is the canonical key generator outputting independent random keys.

3.3 Decomposing AE Security

While the notion mu-mrae is very strong, it might be difficult to prove that an AE scheme, say AES-GCM-SIV meets this notion, if one aims for beyond-birthday bounds. We therefore decompose this notion into separate privacy and authenticity notions, as defined below.

Privacy. Consider the game $\mathbf{G}^{\text{mu-priv}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A})$ in Fig. 3 that defines the (misuse-resistant) privacy of an AE scheme AE, with respect to a key-generation algorithm KeyGen, and an ideal primitive Π. Define

$$\mathsf{Adv}^{\text{mu-priv}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}) = 2\Pr[\mathbf{G}^{\text{mu-priv}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A})] - 1 \; .$$

Under this notion, the adversary is given access to an encryption oracle that either implements the true encryption or returns a random string of appropriate length, but there is no decryption oracle. If the adversary repeats a prior encryption query then this query will be ignored.

Authenticity. Consider the game $\mathbf{G}^{\text{mu-auth}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A})$ in Fig. 3 that defines the (misuse-resistant) authenticity of an AE scheme AE, with respect to a key-generation algorithm KeyGen, and an ideal primitive Π. Define

$$\mathsf{Adv}^{\text{mu-auth}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}) = 2\Pr[\mathbf{G}^{\text{mu-auth}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A})] - 1 \; .$$

Under this notion, initially a bit b is set to 0 and the adversary is given an encryption oracle that always implements the true encryption, and a verification

Game $\mathbf{G}^{\text{mu-priv}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A})$	Game $\mathbf{G}^{\text{mu-auth}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A})$
$v \leftarrow 0; \mathsf{st}_0 \leftarrow \varepsilon;\ b \leftarrow\!\!\$\ \{0,1\}$	$v \leftarrow 0; \mathsf{st}_0 \leftarrow \varepsilon;\ b \leftarrow 0$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{New},\text{Enc},\text{Prim}}$	$\mathcal{A}^{\text{New},\text{Enc},\text{Vf},\text{Prim}}$
Return $(b' = b)$	Return $(b = 1)$
$\underline{\text{New}(\mathsf{aux})}$	$\underline{\text{New}(\mathsf{aux})}$
$v \leftarrow v + 1$	$v \leftarrow v + 1$
$(K_v, \mathsf{st}_v) \leftarrow\!\!\$\ \mathsf{KeyGen}(\mathsf{st}_{v-1}, \mathsf{aux})$	$(K_v, \mathsf{st}_v) \leftarrow\!\!\$\ \mathsf{KeyGen}(\mathsf{st}_{v-1}, \mathsf{aux})$
$\underline{\text{Enc}(i, N, M, A)}$	$\underline{\text{Enc}(i, N, M, A)}$
If $i \notin \{1, \ldots, v\}$ then return $\perp$	If $i \notin \{1, \ldots, v\}$ then return $\perp$
If $(i, N, M, A) \in U[i]$ then return $\perp$	$C \leftarrow \mathsf{AE.E}^{\text{Prim}}(K_i, N, M, A)$
$C_1 \leftarrow \mathsf{AE.E}^{\text{Prim}}(K_i, N, M, A)$	$V[i] \leftarrow V[i] \cup \{(i, N, C, A)\}$
$C_0 \leftarrow\!\!\$\ \{0,1\}^{\lvert C_1 \rvert}$	Return C
$U[i] \leftarrow U[i] \cup \{(i, N, M, A)\}$	$\underline{\text{Vf}(i, N, C, A)}$
Return C_b	If $i \notin \{1, \ldots, v\}$ then return $\perp$
	If $(i, N, C, A) \notin V[i]$ then
	$\quad M \leftarrow \mathsf{AE.D}^{\text{Prim}}(K_i, N, C, A)$
	$\quad$ If $(M \neq \perp)$ then $b \leftarrow 1$

Fig. 3: **Games to define privacy(left), and authenticity (right) of an AE scheme AE with respect to a key-generation algorithm $\mathsf{KeyGen} : \mathcal{K} \times \mathcal{N} \to \{0,1\}^{\mathsf{AE.kl}}$.** The oracle Prim implements the ideal primitive Π. In the authenticity notion, queries to Vf must be performed *after* all queries to Enc.

oracle. We require that verification queries be made *after* all evaluation queries. On a verification (i, N, C, A), if there is a prior encryption query (i, N, M, A) for an answer C, then the oracle ignores this query. Otherwise, the oracle sets $b \leftarrow 1$ if $\mathsf{AE.D}^{\text{Prim}}(K_i, N, C, A)$ returns a non-$\perp$ answer. The goal of the adversary is to set $b = 1$.

Relations. Note that in the mrae notion, the adversary can perform encryption and verification queries in an arbitrary order. In contrast, in the authenticity notion, the adversary can only call the verification oracle after it finishes querying the encryption oracle. Still, in Proposition 1 below, we show that authenticity and privacy tightly implies mrae security. See Appendix C.

Proposition 1. *Let* AE *be an AE scheme associated with a key-generation algorithm* KeyGen *and an ideal primitive* Π*. Suppose that a ciphertext in* AE *is always at least* n*-bit longer than the corresponding plaintext. For any adversary* $\mathcal{A}_0$ *that makes* q_v *verification queries, we can construct adversaries* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *such that*

$$\mathsf{Adv}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}_0) \leq \mathsf{Adv}^{\text{mu-priv}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}_1) + \mathsf{Adv}^{\text{mu-auth}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}_2) + \frac{2q_v}{2^n} \ .$$

Any query of $\mathcal{A}_1$ or $\mathcal{A}_2$ is produced directly from $\mathcal{A}_0$. If $\mathcal{A}_0$ is d-repeating then so are $\mathcal{A}_1$ and $\mathcal{A}_2$.

4 Multi-User Security of Basic Symmetric Schemes

4.1 Security of Counter-Mode Encryption

We study the mu-security of counter mode encryption, or CTR for short. While this is interesting on its own right (we are not aware of any analysis achieving a comparable bound in the literature), we will also use Theorem 1 below to obtain security results for AES-GCM-SIV. For this reason, we introduce some extra notions to handle the degree of generality needed for our proof. Also, our result is general enough to suggest an efficient solution to the re-keying problem first studied by Abdalla and Bellare [1].

General IVs. We will consider a general IV-increasing procedure add, which is associated with some maximal message length of $L_{\max}$ blocks, and a block length n. In particular, add takes an n-bit string IV and an offset $i \in \{0, \ldots, L_{\max} - 1\}$ as inputs, and is such that $\mathsf{add}(\mathsf{IV}, i)$ returns an n-bit string, and for all IV, the strings $\mathsf{add}(\mathsf{IV}, 0), \ldots, \mathsf{add}(\mathsf{IV}, L_{\max} - 1)$ are distinct. We also say that add has *min-entropy* h if for a random n-bit IV, and every $i \in \mathbb{Z}_{L_{\max}}$, $\mathsf{add}(\mathsf{IV}, i)$ takes any value with probability at most 2^{-h}, i.e., its min-entropy is at least h.

For example, the canonical IV addition is such that $\mathsf{add}(\mathsf{IV}, i) = \mathsf{IV} + i \pmod{2^n}$, where we identify n-bit strings with integers in $\mathbb{Z}_{2^n}$. Here, $L_{\max} = 2^n$. In contrast, the AES-GCM-SIV will use CTR with $L_{\max} = 2^{32}$, $n = 128$, and $\mathsf{add}(\mathsf{IV}, i) = 1 \,\|\, \mathsf{IV}[2, 96] \,\|\, (\mathsf{IV}[97, 128] + i \pmod{2^{32}})$. Clearly, here, the min-entropy is 127 bits, due to the first bit being set to one.

CTR encryption. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher, i.e., $E(K, \cdot)$ is a permutation for all k-bit K. We denote $E(K, \cdot) = E_K(\cdot)$, and E_K^{-1} is the inverse of E_K. Further, let add be a general IV-increasing procedure with maximal block length $L_{\max}$. We define the IV-based encryption scheme $\mathsf{CTR} = \mathsf{CTR}[E, \mathsf{add}]$ with $\mathsf{CTR.kl} = k$, and where encryption operates as follows (where we use $\xleftarrow{n}$ to denote some function which pads a message M into n-bit blocks).

$\underline{\mathsf{CTR.E}(K, M)}$:
$C[0] \leftarrow \mathsf{IV} \leftarrow_{\$} \{0,1\}^n$, $M[1], \ldots, M[\ell] \xleftarrow{n} M$
If $\ell > L_{\max}$ then return $\perp$
For $i = 1$ to ℓ do $C[i] \leftarrow E_K(\mathsf{add}(\mathsf{IV}, i-1)) \oplus M[i]$
Return $C[0] \,\|\, C[1] \,\|\, \cdots \,\|\, C[\ell]$

Decryption $\mathsf{CTR.D}$ re-computes the masks $E_K(\mathsf{add}(\mathsf{IV}, i-1))$ using $C[0] = \mathsf{IV}$, and then retrieves the message blocks by xoring the masks to the ciphertext. Here, we assume without loss of generality messages are padded (e.g., PKCS#7), so that they are split uniquely into full-length n-bit blocks. Our result extends easily to the more common padding-free variant where the last block is allowed to be shorter than n bits, and the output of $E_K(\mathsf{add}(\mathsf{IV}, \ell - 1))$ is truncated accordingly, since an adversary can simulate the padding-free version by removing the appropriate number of bits from the received ciphertexts.

Security of CTR. We establish the (CPA) security of randomized CTR in the ideal-cipher model for an arbitrary key-generation algorithm KeyGen which produces keys that collide with small probability. In particular, we say that KeyGen is *α-smooth* if for a sequence of keys $(K_1, \ldots, K_u)$ output by an arbitrary interaction with New, we have $\Pr[K_i = K] \leq \alpha$ for all i and $K \in \{0,1\}^k$, and $\Pr[K_i = K_j] \leq \alpha$ for all $i \neq j$. The canonical KeyGen is α-smooth for $\alpha = 2^{-k}$. See Appendix D for the proof.

Theorem 1. *Let E be modeled as an ideal cipher,* add *have min-entropy h, and* KeyGen *be α-smooth. Further, let $L, B \geq 1$ such that $L \leq 2^{(1-\epsilon)h-1}$, for some $\epsilon \in (0,1]$, and let $\mathcal{A}$ be an adversary that queries* Enc *for at most L n-bit blocks, and at most B blocks for each user, and makes p* Prim *queries. Then,*

$$\mathsf{Adv}^{\mathsf{mu\text{-}ind}}_{\mathsf{CTR}[E,\mathsf{add}],\mathsf{KeyGen},E}(\mathcal{A}) \leq 2^{-n/2} + \left(LB + L^2\alpha\right) \cdot \left(\frac{1}{2^n} + \frac{1}{2^h}\right) + ap\alpha \ ,$$

where $a := \left\lceil \frac{1.5n}{\epsilon h} \right\rceil - 1$.

The bound highlights the benefits when each user only encrypts B blocks. In particular, assume $h = n$, $\alpha = 1/2^k$. If $B = 2^b$, then the number L of blocks encrypted overall by the scheme can be as high as 2^{n-b}. (The second term has L^2 in the numerator, but the denominator is much larger, i.e., 2^{n+k}.) Another interesting feature is that the contribution of Prim queries to the bound is independent of the number of users and L.

More on the bound. Previous works [16,20] implicitly give mu security bounds for CTR, but adopt a different model, where the adversary is a-priori constrained in (1) the number of queries q, (2) a bound B_i on the number of blocks encrypted per user $i \in [u]$. The resulting bounds contain a leading term $\sum_{i=1}^{u} B_i^2/2^n$, assuming no primitive queries are made (adding primitive queries p only degrades the bound). This is essentially what one can obtain by applying a naïve hybrid argument to the single-user analysis. We discussed the disadvantage of such a bound in the introduction already.

Re-keying, revisited. Also, in contrast to previous works, the above result holds for an arbitrary KeyGen, and only requires *very weak* randomness from it. This suggests a new and efficient solutions for the re-keying problem of [1]. Let $H : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^k$ be a hash function, and let KeyGen, on input $\mathsf{aux} \in \{0,1\}^*$, simply output $H(K, \mathsf{aux})$ for some master secret key K, and this KeyGen is α-smooth if H is for example POLYVAL from AES-GCM-SIV, where $\alpha = \ell/2^k$, and ℓ is an upper bound on the length of aux. We can assume ℓ to be fixed to something short, even 1. Indeed, aux could be a counter, or some other short string. The resulting bound (when $h = n$) would be $2^{-n/2} + \frac{2LB}{2^n} + \frac{2L^2}{2^{n+k}} + ap/2^k$. Note that this solution heavily exploits the ideal-cipher model – clearly, we are indirectly assuming some form of related-key security on E implicitly, and one should carefully assess the security of E in this setting.

The results in the model of Abdalla and Bellare [1] are weaker in that they only study more involved key-derivation methods (but with the benefit of a standard-model security reduction), in a more constrained model, where the adversary sequentially queries B blocks on a key, before moving to the next key. Our model, however, is adaptive, as the adversary can distribute queries as it pleases across users. But difference is not only qualitative, as quantitative bounds in [1] are obtained via naïve hybrid arguments.

4.2 Security of GMAC^+

This section deals with an abstraction of GMAC^+, the PRF used within the AES-GCM-SIV mode of operation. We show good mu bounds for this construction. The ideas extend similarly to various Wegman-Carter type MACs [34], but we focus here on GMAC^+.

THE GMAC^+ CONSTRUCTION. The construction relies on a hash function $H : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$, which is meant to satisfy the following properties. (We employ the shorthand $H_K(M, A) = H(K, M, A)$.)

Definition 1. *Let $H : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$. We say that H is* c-almost XOR universal *if for all $(M, A) \neq (M', A')$, and all $\Delta \in \{0,1\}^n$, and $K \leftarrow_\$ \{0,1\}^n$,*

$$\Pr[H_K(M, A) \oplus H_K(M', A') = \Delta] \leq \frac{c \cdot \max\{|M|_n + |A|_n, |M'|_n + |A'|_n\}}{2^n} ,$$

where $|X|_n = \max\{1, \lceil |X| / n \rceil\}$ is the block length of string X, as defined in Section 2. Further, we say it is c-regular *if for all $Y \in \{0,1\}^n$, $M, A \in \{0,1\}^*$, and $K \leftarrow_\$ \{0,1\}^n$,*

$$\Pr[H_K(M, A) = Y] \leq \frac{c \cdot (|M|_n + |A|_n)}{2^n} .$$

We say it is weakly c-regular *if this is only true for $(M, A) \neq (\varepsilon, \varepsilon)$, and $H_K(\varepsilon, \varepsilon) = 0^n$ for all K.*

Remark 1. Note that for POLYVAL as used in AES-GCM-SIV, we can set $c = 1.5$ *provided* that we exclude the empty string as input. This is because the empty string results in POLYVAL outputting 0^n regardless of the key, and thus POLYVAL is only *weakly* c-regular. It is easy to fix POLYVAL so that this does not happen (as the input is padded with its length, it is sufficient to ensure that the length padding of the empty string contains at least one bit with value 1). See Appendix B for more details.

We also consider a generic function $\mathsf{xor} : \{0,1\}^n \times \{0,1\}^{\mathsf{nl}} \to \{0,1\}^n$, for $\mathsf{nl} < n$, which is meant to add a nonce to a string. In particular, we require: (1) *λ-regularity:* For every $N \in \{0,1\}^{\mathsf{nl}}$ and $Z \in \{0,1\}^n$, there are at most λ strings $Y \in \{0,1\}^n$ such that $\mathsf{xor}(Y,N) = Z$, (2) *injectivity:* For every Y, $\mathsf{xor}(Y,\cdot)$ is injective, and (3) *linearity:* For every Y, Y', N, N', we have $\mathsf{xor}(Y,N) \oplus \mathsf{xor}(Y',N') = \mathsf{xor}(Y \oplus Y', N \oplus N')$.

Example 1. In GCM-SIV and AES-GCM-SIV, one uses

$$\mathsf{xor}(Y,N) = 0 \,\|\, (Y \oplus 0^{n-\mathsf{nl}} N)[2:n] \ .$$

This is clearly 2-regular, injective, and linear. Note that here it is important to prepend 0's to the nonce N; if one instead appends 0's to N then injectivity of xor will be destroyed.

Given H and xor, as well as a block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, we define $\mathsf{GMAC}^+ = \mathsf{GMAC}^+[H,E,\mathsf{xor}] : \{0,1\}^{k+n} \times (\{0,1\}^* \times \{0,1\}^* \times \{0,1\}^{\mathsf{nl}}) \to \{0,1\}^n$ such that

$$\mathsf{GMAC}^+(K_{\mathsf{in}} \,\|\, K_{\mathsf{out}}, (M,A,N)) = E_{K_{\mathsf{out}}}(\mathsf{xor}(H_{K_{\mathsf{in}}}(M,A),N)) \ . \tag{2}$$

Mu-prf security of GMAC^+. We upper bound the mu prf advantage for GMAC^+; see Appendix E for the proof. We stress here that the adversary's Eval queries have form (i,M,A,N), and the length of such queries is implicitly defined as $|M|_n + |A|_n$.

We also consider an arbitrary KeyGen algorithm, which outputs pairs of keys $(K^i_{\mathsf{in}}, K^i_{\mathsf{out}}) \in \{0,1\}^n \times \{0,1\}^k$. We will only require these keys to be pairwise-close to uniform, i.e., we say that KeyGen is *β-pairwise almost uniform (AU)* if for every $i \neq j$, the distribution of $(K^i_{\mathsf{in}}, K^i_{\mathsf{out}}), (K^j_{\mathsf{in}}, K^j_{\mathsf{out}})$ is such that very pair of $(n+k)$-bit strings appears with probability at most $\beta\frac{1}{2^{2(n+k)}}$. Clearly, the canonical KeyGen satisfies this with $\beta = 1$, but we will be for instance interested later on in cases where $\beta = 1 + \epsilon$ for some small constant $\epsilon > 0$.

Theorem 2 (Security of GMAC^+). *Let $H : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ be c-almost xor universal and c-regular, KeyGen be β-pairwise AU, xor be injective, linear, and λ-regular, and let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher, which we model as an ideal cipher. Then, for any adversary $\mathcal{A}$ making q Eval queries of at most L n-bit blocks (with at most B blocks queries per user), as well as p ideal-cipher queries,*

$$\mathsf{Adv}^{\mathsf{mu\text{-}prf}}_{\mathsf{GMAC}^+[H,E,\mathsf{xor}],B,E}(\mathcal{A}) \leq \frac{(1+C)qB}{2^n} + \frac{CL(p+q) + \beta q^2}{2^{n+k}} \ , \tag{3}$$

where $C := c \cdot \lambda \cdot \beta$.

Here, parameters are even better than in the case of counter-mode, but this is in part due to the longer key. In particular, this being PRF security, it is unavoidable that security is compromised when more than $2^{(k+n)/2}$ users are involved. The interesting fact is that *partial* key collisions (i.e., a collision in the hash keys or in the cipher keys) alone do not help.

For example, take $k = n = 128$, $C = \beta = 1$, $B = 2^{32}$, $L = qB$, $q \le 2^{95}$, then the bound becomes roughly $q/2^{95} + p/2^{128}$, and note that this is when processing up to 2^{128} blocks of data.

Weak regularity. We also provide a version of Theorem 2 for the case where H is only weakly c-regular. We stress that the security loss is substantial here (and thus if using GMAC^+ alone, one should rather make sure H is c-regular), but nonetheless the security is preserved in the case where a nonce N is reused across a sufficiently small number d of users. A proof sketch is in Appendix E.1.

Theorem 3 (Security of GMAC^+, weak regularity). *Let $H : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ be c-almost xor universal and weakly c-regular, KeyGen be β-pairwise AU, xor be injective, linear, and λ-regular, and let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher, which we model as an ideal cipher. Then, for any adversary $\mathcal{A}$ making q Eval queries of at most L n-bit blocks (with at most B blocks queries per user), as well as p ideal-cipher queries,*

$$\mathsf{Adv}^{\text{mu-prf}}_{\mathsf{GMAC}^+[H,E,\mathsf{xor}],B,E}(\mathcal{A}) \le \frac{(1+C)qB}{2^n} + \frac{CL(p+2q)+\beta q^2}{2^{n+k}} + \frac{d(p+q)}{2^k}\ , \quad (4)$$

where $C := c \cdot \lambda \cdot \beta$, and d is a bound on the number of users re-using any given nonce.

5 SIV Composition with Key Reuse

SIV with key reuse. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Let $\mathsf{F} : \{0,1\}^{\mathsf{F.kl}} \times \mathcal{N} \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be a keyed function, with $\mathsf{F.kl} \ge k$. Let $\mathsf{SE} : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^*$ be an IV-based encryption scheme of IV length n. Both F and SE are built on top of E. In a generic SIV composition, the key $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ of F and the key J of SE will be chosen independently. However, for efficiency, it would be convenient if one can reuse $K_{\mathsf{out}} = J$, which GCM-SIV does. Formally, let $\mathsf{AE} = \mathsf{SIV}[\mathsf{F}, \mathsf{SE}]$ be the AE scheme as defined in Fig. 4.

Results. We consider security of the SIV construction for $\mathsf{F} = \mathsf{GMAC}^+$ and $\mathsf{SE} = \mathsf{CTR}$. We assume that GMAC^+ and CTR use functions xor and add, respectively, such that (1) xor is 2-regular, injective, and linear, and $\mathsf{xor}(X, N) \in 0\{0,1\}^{n-1}$ for every string $X \in \{0,1\}^n$ and every nonce $N \in \{0,1\}^{\mathsf{nl}}$, and (2) add has min-entropy $n-1$, and $\mathsf{add}(\mathsf{IV}, \ell) \in 1\{0,1\}^{n-1}$ for every $\mathsf{IV} \in \{0,1\}^n$ and every $\ell \in \mathbb{N}$. (Those notions for add and xor can be found in Section 4.1 and Section 4.2 respectively.) This assumption holds for the design choice of AES-GCM-SIV. We thus only write $\mathsf{CTR}[E]$ or $\mathsf{GMAC}^+[H, E]$ instead of $\mathsf{CTR}[E, \mathsf{add}]$ or $\mathsf{GMAC}^+[H, E, \mathsf{xor}]$. Below, we show the mu-mrae security of $\mathsf{SIV}[\mathsf{GMAC}^+[H, E], \mathsf{CTR}[E]]$, with respect to a pairwise AU KeyGen, and a c-regular, c-AXU hash function H; the

$\mathsf{AE.E}(K_{\mathsf{in}} \parallel K_{\mathsf{out}}, N, M, A)$	$\mathsf{AE.D}(K_{\mathsf{in}} \parallel K_{\mathsf{out}}, N, C, A)$
$\mathsf{IV} \leftarrow \mathsf{F}(K_{\mathsf{in}} \parallel K_{\mathsf{out}}, N, M, A)$	$\mathsf{IV} \parallel C' \leftarrow C$; $M \leftarrow \mathsf{SE.D}^E(K_{\mathsf{out}}, C)$
$C \leftarrow \mathsf{SE.E}^E(K_{\mathsf{out}}, M; \mathsf{IV})$	$T \leftarrow \mathsf{F}^E(K_{\mathsf{in}} \parallel K_{\mathsf{out}}, N, M, A)$
Return C	If $T \neq \mathsf{IV}$ then return $\perp$ else return M

Fig. 4: **The SIV construction (with key reuse) $\mathsf{AE} = \mathsf{SIV}[\mathsf{F}, \mathsf{SE}]$ that is built on top of an ideal cipher E.**

notion of pairwise AU for key-generation algorithms can be found in Section 4.2. See Appendix F for the proof.

Theorem 4 (Security of SIV). *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Fix $0 < \epsilon < 1$. Let $H : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be a c-regular, c-AXU hash. Let $\mathsf{AE} \leftarrow \mathsf{SIV}[\mathsf{GMAC}^+[H,E], \mathsf{CTR}[E]]$. Then for any β-pairwise AU KeyGen and for any adversary $\mathcal{A}$ that makes at most q encryption/verification queries whose total block length is at most $L \leq 2^{(1-\epsilon)n-4}$, and encryption queries of at most B blocks per user, and $p \leq 2^{(1-\epsilon)n-4}$ ideal-cipher queries,*

$$\mathsf{Adv}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A}) \leq \frac{1}{2^{n/2}} + \frac{\beta a p}{2^k} + \frac{(3\beta c + 7\beta)L^2 + 4\beta c L p}{2^{n+k}} + \frac{(4c\beta + 0.5\beta + 6.5)LB}{2^n},$$

where $a = \lceil 1.5n/(n-1)\epsilon \rceil - 1$.

Remarks. The proof of Theorem 4 only needs to know that the mu-ind proof of CTR and the mu-prf proof of GMAC^+ follow some high-level structure that we will describe below. We do not need to know any other specific details about those two proofs. This saves us the burden of repeating the entire prior proofs in Section 4.1 and Section 4.2. The mu-ind proof of CTR uses the H-coefficient technique and follows this canonical structure:

(i) When the adversary finishes querying, we grant it all the keys. Note that in the ideal world, the keys are still created but not used.

(ii) For each ideal-cipher query $E_K(X)$ for answer Y, the transcript correspondingly stores an entry $(\texttt{prim}, K, X, Y, +)$. Likewise, for each query $E^{-1}(K, Y)$ for answer X, the transcript stores an entry $(\texttt{prim}, K, X, Y, -)$. For each query $\textsc{Enc}(i, M)$ with answer C, we store an entry (enc, i, M, C).

(iii) When the adversary finishes querying, for each entry (enc, i, M, C), in the real world, we grant it a table that stores all triples $(K_i, X, E(K_i, X))$ for all queries $E(K_i, X)$ that $\mathsf{CTR.E}[E](K_i, M; T)$ makes, where K_i is the key of user i and T is the IV of C. In the ideal world, the proof generates a corresponding fake table as follows. If we consider the version of CTR in which messages are padded (e.g., PKCS#7), then one can first parse

$\mathsf{IV} \| C_1 \| \cdots \| C_m \xleftarrow{n} C$ and $M_1 \| \cdots \| M_m \xleftarrow{n} M$ and then return $(K_i, X_1, C_1 \oplus M_1), \ldots, (K_i, X_m, C_m \oplus M_m)$, where $X_i = \mathsf{add}(\mathsf{IV}, i-1)$ and we use $\xleftarrow{n}$ to denote some function that pads a message into n-bit blocks. If one uses the well-known padding-free version of CTR where the last block of the message is allowed to be shorter than n-bit, then one first pads C with random bits so that the last fragmentary block becomes n-bit long, and likewise pads M with 0's so that the last fragmentary block becomes n-bit long, and then proceeds as above. (This step can be optionally omitted for the padding version since the adversary can generate the table by itself.)

(iv) Consider a transcript τ. If there are two tables $\mathcal{T}_1$ and $\mathcal{T}_2$ in τ that contain triples (K, X, Y) and (K, X', Y') respectively, and either $X = X'$, or $Y = Y'$, then τ must be considered bad. If there is a table $\mathcal{T}$ that contains triples (K, X, Y) and (K, X', Y') such that either $X = X'$, or $Y = Y'$, then τ is also considered bad. In addition, if there is a table $\mathcal{T}$ that contains a triple (K, X, Y), and there is an entry $(\mathtt{prim}, K, X', Y', \cdot)$, and either $X = X'$ or $Y = Y'$, then τ is considered bad. The proof may define some other criteria for badness of transcripts.

We say that a CTR transcript is CTR-*bad* if it is bad according to the criteria defined by the proof of Theorem 1. (Note that although not all of those criteria are specified in the structure above, it is enough for our purpose, as our proof of Theorem 4 does not need to know those specific details.) The proof of GMAC^+ also follows a similar high-level structure. We say that a GMAC^+ transcript is GMAC^+-*bad* if it is bad according to the criteria defined by the proof of Theorem 2.

Weak regularity. We also provide a version of Theorem 4 for the case where H is only weakly c-regular. Again, the security loss is substantial here, but security is preserved if each nonce is reused across a sufficiently small number d of users. A proof sketch is given in Appendix F.1.

Theorem 5 (Security of SIV, weak regularity). *Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Fix $0 < \epsilon < 1$. Let $H : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be a weakly c-regular, c-AXU hash. Let $\mathsf{AE} \leftarrow \mathsf{SIV}[\mathsf{GMAC}^+[H, E], \mathsf{CTR}[E]]$. Then for any β-pairwise AU KeyGen and for any adversary $\mathcal{A}$ that makes at most q encryption/verification queries whose total block length is at most $L \le 2^{(1-\epsilon)n-4}$, and encryption queries of at most B blocks per user, and $p \le 2^{(1-\epsilon)n-4}$ ideal-cipher queries,*

$$\begin{aligned}\mathsf{Adv}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A}) \le \frac{1}{2^{n/2}} + \frac{\beta a p}{2^k} + \frac{(3\beta c + 7\beta)L^2 + 4\beta c L p}{2^{n+k}} \\ + \frac{(4c\beta + 0.5\beta + 6.5)LB}{2^n} + \frac{dp + (2d+a)L}{2^k},\end{aligned}$$

where $a = \lceil 1.5n/(n-1)\epsilon \rceil - 1$, and d is a bound on the number of users re-using any given nonce.

6 AES-GCM-SIV with A Generic Key-Derivation

In this section we consider the mu-mrae security of AES-GCM-SIV with respect to a quite generic class of key-derivation functions. This class includes the current KDF KD_0 of AES-GCM-SIV, but it contains another KDF KD_1 that is not only simpler but also twice faster. This KD_1 was the original KDF in AES-GCM-SIV, but then subsequently replaced by KD_0. Our multi-user bound is even better than the single-user bound of Gueron and Lindell [16]. In this section, we assume that GMAC^+ and CTR use functions xor and add, respectively, such that (1) xor is 2-regular, injective, and linear, and $\mathsf{xor}(X, N) \in 0\{0,1\}^{n-1}$ for every string $X \in \{0,1\}^n$ and every nonce $N \in \mathcal{N} = \{0,1\}^{\mathsf{nl}}$, and (2) add has min-entropy $n-1$, and $\mathsf{add}(\mathsf{IV}, \ell) \in 1\{0,1\}^{n-1}$ for every $\mathsf{IV} \in \{0,1\}^n$ and every $\ell \in \mathbb{N}$. (Those notions for add and xor can be found in Section 4.1 and Section 4.2 respectively.) This assumption holds for the design choice of AES-GCM-SIV. We thus only write $\mathsf{CTR}[E]$ or $\mathsf{GMAC}^+[H, E]$ instead of $\mathsf{CTR}[E, \mathsf{add}]$ or $\mathsf{GMAC}^+[H, E, \mathsf{xor}]$.

Below, we will formalize the Key-then-Encrypt transform that captures the way AES-GCM-SIV generates session keys for every encryption/decryption. We then describe our class of KDFs.

THE KtE TRANSFORM. Let AE be an AE scheme of nonce space $\mathcal{N}$ and let $\mathsf{KD} : \mathcal{K} \times \mathcal{N} \to \{0,1\}^{\mathsf{AE.kl}}$ be a key-derivation function. Given KD and AE, the Key-then-Encrypt (KtE) transform constructs another AE scheme $\overline{\mathsf{AE}} = \mathsf{KtE}[\mathsf{KD}, \mathsf{AE}]$ as shown in Fig. 5.

$\overline{\mathsf{AE}}.\mathsf{E}(K, N, M, A)$	$\overline{\mathsf{AE}}.\mathsf{D}(K, N, C, A)$
$J \leftarrow \mathsf{KD}(K, N)$; $C \leftarrow \mathsf{AE.E}(J, N, M, A)$	$J \leftarrow \mathsf{KD}(K, N)$; $M \leftarrow \mathsf{AE.D}(J, N, C, A)$
Return C	Return M

Fig. 5: **The AE scheme $\overline{\mathsf{AE}} = \mathsf{KtE}[\mathsf{KD}, \mathsf{AE}]$ constructed from an AE scheme AE and a key-derivation function KD, under the KtE transform.**

NATURAL KDFS. Let $n \geq 1$ be an integer and let $k \in \{n, 2n\}$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Let $\mathsf{pad} : \mathcal{N} \times \{0, \ldots, 5\} \to \{0,1\}^n$ be a padding mechanism such that $\mathsf{pad}(N_0, s_0) \neq \mathsf{pad}(N_1, s_1)$ for every distinct pairs $(N_0, s_0), (N_1, s_1) \in \mathcal{N} \times \{0, \ldots, 5\}$. Let $\mathsf{KD}[E] : \{0,1\}^k \times \mathcal{N} \to \{0,1\}^{n+k}$ be a KDF that is associated with a deterministic algorithm $\mathsf{KD.Map} : (\{0,1\}^n)^6 \to \{0,1\}^{n+k}$. We say that $\mathsf{KD}[E]$ is *natural* if on input (K, N), $\mathsf{KD}[E]$ first calls $R_0 \leftarrow E(K, \mathsf{pad}(N, 0)), \ldots, R_5 \leftarrow E(K, \mathsf{pad}(N, 5))$, and then returns $\mathsf{KD.Map}(R_0, \ldots, R_5)$.

It might seem arbitrary to limit the number of blockcipher calls of a natural KDF to six. However, note that since $k \leq 2n$, the block length of each $(k+n)$-bit derived key is at most three. All known good constructions, which we list

$\underline{\mathsf{KD}_0[E](K,N)}$	$\underline{\mathsf{KD}_1[E](K,N)}$
For $s = 0$ to 5 do $R_s \leftarrow E_K(\mathsf{pad}(N,s))$ For $i = 0$ to 2 do $V_i \leftarrow R_{2i}[1 : n/2] \parallel R_{2i+1}[1 : n/2]$ Return $(V_0 \parallel V_1 \parallel V_2)[1 : n+k]$	For $s = 0$ to 5 do $R_i \leftarrow E_K(\mathsf{pad}(N,s))$ Return $(R_0 \parallel R_1 \parallel R_2)[1 : n+k]$

Fig. 6: **Key-derivation functions KD_0 (left) and KD_1 (right).**

below, use at most six blockcipher calls. Using more would simply make the performance and even the bounds worse. We therefore define a natural KDF to use at most six blockcipher calls.

The current KDF $\mathsf{KD}_0[E]$ of AES-GCM-SIV, as shown in the left panel of Fig. 6, is natural; it is defined for even n only. For $k = n$, it can be implemented using four blockcipher calls, but for $k = 2n$ it needs six blockcipher calls. Consider the KDF $\mathsf{KD}_1[E]$ on the right panel of Fig. 6. For $k = n$ it can be implemented using two blockcipher calls, and $k = 2n$ it needs three blockcipher calls. This KDF is also simpler to implement than KD_0. Iwata and Seurin [21] propose to use either the XOR construction [8, 11] or the CENC construction [19]. Both XOR and CENC constructions are natural; the former uses four blockcipher calls for $k = n$ and six blockcipher calls for $k = 2n$, and the latter uses three and four blockcipher calls respectively.

For a natural key-derivation function $\mathsf{KD}[E]$, we say that it is γ-*unpredictable* if for any subset $S \subseteq \{0,1\}^n$ of size at least $\frac{15}{16} \cdot 2^n$ and any $s \in \{0,1\}^{n+k}$, if the random variables $R_0, \ldots, R_5$ are sampled uniformly without replacement from S then $\Pr[\mathsf{KD.Map}(R_0, \ldots, R_5) = s] \le \gamma/2^{n+k}$. Lemma 3 below shows that both $\mathsf{KD}_0[E]$ and $\mathsf{KD}_1[E]$ are 2-unpredictable; see Appendix G for the proof. One might also show that both the XOR and CENC constructions are 2-unpredictable. Therefore, in the remainder of this section, we only consider natural, 2-unpredictable KDFs.

Lemma 3. *Let $n \ge 128$ be an even integer and let $k \in \{n, 2n\}$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Then both $\mathsf{KD}_0[E]$ and $\mathsf{KD}_1[E]$ are 2-unpredictable.*

Ideal counterpart of natural KDF. For a natural KDF $\mathsf{KD}[E]$, consider its following ideal version $\mathsf{KD}[k]$. The key space of $\mathsf{KD}[k]$ is the entire set $\mathrm{Perm}(n)$. It takes as input a permutation $\pi \in \mathrm{Perm}(n)$ and a string $N \in \mathcal{N}$, computes $R_s \leftarrow \pi(\mathsf{pad}(N,s))$ for all $s \in \{0, \ldots, 5\}$, and returns $\mathsf{KD.Map}(R_0, \ldots, R_5)$. Of course $\mathsf{KD}[k]$ is impractical since its key length is huge, but it will be useful in studying the security of the KtE transform. The following bounds the privacy and authenticity of $\mathsf{KtE}[\mathsf{KD}[k], \mathsf{AE}]$ via the mu-mrae security of the AE scheme AE; the proof is in Appendix H. In light of that, in the subsequent subsections, we will analyze the difference between security of $\mathsf{KtE}[\mathsf{KD}[E], \mathsf{AE}]$ and that of $\mathsf{KtE}[\mathsf{KD}[k], \mathsf{AE}]$.

$\underline{\mathsf{KeyGen}(\mathsf{st}, \mathsf{aux})}$
$(N, i) \leftarrow \mathsf{aux}$; $(\pi_1, S_1, \ldots, \pi_m, S_m) \leftarrow \mathsf{st}$
If $(i \in \{1, \ldots, m\}$ and $N \in S_i)$ or $(i \notin \{1, \ldots, m+1\})$ then
  //Unexpected input, return a random key anyway
  $K \leftarrow\!\!\$\ \{0,1\}^{k+n}$; return (K, st)
If $i \in \{1, \ldots, m\}$ then $S_i \leftarrow S_i \cup \{N\}$; $\mathsf{st} \leftarrow (\pi_1, S_1, \ldots, \pi_m, S_m)$
If $i = m+1$ then $\pi_{m+1} \leftarrow\!\!\$\ \mathrm{Perm}(n)$; $S_{m+1} \leftarrow \{N\}$; $\mathsf{st} \leftarrow (\pi_1, S_1, \ldots, \pi_{m+1}, S_{m+1})$
Return $(\mathsf{KD}[k](\pi_i, N), \mathsf{st})$

Fig. 7: **Key-generation algorithm KeyGen corresponding to $\mathsf{KD}[k]$.**

Proposition 2. *Let $n \geq 8$ be an integer and let $k \in \{n, 2n\}$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Let $\mathsf{KD}[E]$ be a natural KDF. Let AE be an AE scheme of key length $k + n$. Let $\overline{\mathsf{AE}} = \mathsf{KtE}[\mathsf{KD}[k], \mathsf{AE}]$. Then for any adversaries $\overline{\mathcal{A}}_1$ and $\overline{\mathcal{A}}_2$, we can construct a key-generation algorithm $\mathsf{KD.KeyGen}$ as shown in Fig. 7, and an adversary $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\overline{\mathsf{AE}},E}(\overline{\mathcal{A}}_1) + \mathsf{Adv}^{\mathsf{mu\text{-}auth}}_{\overline{\mathsf{AE}},E}(\overline{\mathcal{A}}_2) \leq 3\, \mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A}) \ .$$

For any type of queries, the number of $\mathcal{A}$'s queries is at most the maximum of that of $\overline{\mathcal{A}}_1$ and $\overline{\mathcal{A}}_2$, and the similar claim holds for the total block length of the encryption/verification queries. Moreover, the maximum of total block length of encryption queries per user of $\mathcal{A}$ is at most the maximum of that per (user, nonce) pair of $\overline{\mathcal{A}}_1$ and $\overline{\mathcal{A}}_2$.

The following lemma says that if $\mathsf{KD}[E]$ is 2-unpredictable then the constructed KeyGen in the theorem statement of Proposition 2 is 4-pairwise AU; the notion of pairwise AU for key-generation algorithms can be found in Section 4.2. The proof is in Appendix I.

Lemma 4. *Let $n \geq 8$ be an integer and let $k \in \{n, 2n\}$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Let $\mathsf{KD}[E]$ be a natural, 2-unpredictable KDF. Then the corresponding key-generation algorithm KeyGen in Fig. 7 is 4-pairwise AU.*

Indistinguishability of $\mathsf{KD}[E]$. For an adversary $\mathcal{A}$, define

$$\mathsf{Adv}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\mathcal{A}) = 2\Pr[\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\mathcal{A})] - 1$$

as the advantage of $\mathcal{A}$ in distinguishing a natural KDF $\mathsf{KD}[E]$ and its ideal counterpart $\mathsf{KD}[k]$ in the multi-user setting, where game $\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\mathcal{A})$ is defined in Fig. 8. Under this notion, the adversary is given access to both E and E^{-1}, an oracle $\textsc{New}()$ to initialize a new user v with a truly random master key K_v and a secret ideal permutation π_v, and an evaluation oracle $\textsc{Eval}$ that either implements $\mathsf{KD}[E]$ or $\mathsf{KD}[k]$. We say that an adversary $\mathcal{A}$ is *d-repeating* if among its evaluation queries, a nonce is used for at most d users.

Game $\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\mathcal{A})$	$\underline{\text{EVAL}(i,N)}$
$v \leftarrow 0$; $b \leftarrow\!\!\$\ \{0,1\}$; $b' \leftarrow\!\!\$\ \mathcal{A}^{\text{NEW},\text{EVAL},E,E^{-1}}$	If $i > v$ then return $\perp$
Return $(b' = b)$	If $b = 1$ then return $\mathsf{KD}[E](K_i, N)$
Procedure NEW()	Else return $\mathsf{KD}[k](\pi_i, N)$
$v \leftarrow v + 1; K_v \leftarrow\!\!\$\ \{0,1\}^k$; $\pi_v \leftarrow\!\!\$\ \mathrm{Perm}(n)$	

Fig. 8: **Game to distinguish $\mathsf{KD}[E]$ and its ideal counterpart $\mathsf{KD}[k]$.**

Lemma 5 below bounds the indistinguishability advantage between $\mathsf{KD}[E]$ and $\mathsf{KD}[k]$. The proof is in Appendix J; it uses some technical balls-into-bins results in Appendix A.

Lemma 5. *Fix $0 < \epsilon < 1$. Let $n \geq 16$ be an integer and let $k \in \{n, 2n\}$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Let $\mathsf{KD}[E]$ be a natural KDF. For any d-repeating adversary $\mathcal{A}$ that makes at most $p \leq 2^{n-4}$ ideal-cipher queries, and $q \leq 2^{(1-\epsilon)n-4}$ evaluation queries,*

$$\mathsf{Adv}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\mathcal{A}) \leq \frac{1}{2^{n/2}} + \frac{24pq + 18q^2}{2^{k+n}} + \frac{ap + d(p + 3q)}{2^k}$$

where $a = \lceil 1.5/\epsilon \rceil - 1$. The theorem statement still holds if we grant the adversary the master keys when it finishes querying.

6.1 Privacy Analysis

Lemma 6 below reduces the privacy security of $\mathsf{KtE}[\mathsf{KD}[E], \mathsf{AE}]$ for a generic AE scheme AE, to that of $\mathsf{KtE}[\mathsf{KD}[k], \mathsf{AE}]$; the proof relies crucially on Lemma 5.

Lemma 6. *Fix $0 < \epsilon < 1$. Let $n \geq 16$ be an integer and let $k \in \{n, 2n\}$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Let $\mathsf{KD}[E]$ be a natural KDF. Let AE be an AE scheme of key length $k+n$, and let $\overline{\mathsf{AE}} = \mathsf{KtE}[\mathsf{KD}[E], \mathsf{AE}]$. Consider a d-repeating adversary $\mathcal{A}$ that makes $p \leq 2^{n-5}$ ideal-cipher queries and $q \leq 2^{(1-\epsilon)n-4}$ encryption queries. Suppose that using AE to encrypt $\mathcal{A}$'s encryption queries would need to make $L \leq 2^{n-5}$ ideal-cipher queries. Then*

$$\mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\overline{\mathsf{AE}},E}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\mathsf{KtE}[\mathsf{KD}[k],\mathsf{AE}],E}(\mathcal{A}) + \frac{2}{2^{n/2}} + \frac{48(L+p)q + 36q^2}{2^{k+n}} + \frac{2a(L+p) + 2d(L+p+3q)}{2^k} \ ,$$

where $a = \lceil 1.5/\epsilon \rceil - 1$.

Proof. We first construct an adversary $\overline{\mathcal{A}}$ that tries to distinguish $\mathsf{KD}[E]$ and $\mathsf{KD}[k]$. Adversary $\overline{\mathcal{A}}$ simulates game $\mathbf{G}^{\mathsf{mu\text{-}priv}}_{\overline{\mathsf{AE}},E}(\mathcal{A})$, but each time it needs to generate a session key, it uses its EVAL oracle instead of $\mathsf{KD}[E]$. However, if $\overline{\mathcal{A}}$

previously queried $\textsc{Eval}(i, N)$ for an answer K, next time it simply uses K without querying. Finally, adversary $\overline{\mathcal{A}}$ outputs 1 only if the simulated game returns true. Let b be the challenge bit in game $\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\overline{\mathcal{A}})$. Then

$$\Pr[\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\overline{\mathcal{A}}) \Rightarrow \mathsf{true} \mid b = 1] = \Pr[\mathbf{G}^{\mathsf{mu\text{-}priv}}_{\overline{\mathsf{AE}},E}(\mathcal{A})], \text{ and}$$
$$\Pr[\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\overline{\mathcal{A}}) \Rightarrow \mathsf{false} \mid b = 0] = \Pr[\mathbf{G}^{\mathsf{mu\text{-}priv}}_{\mathsf{KtE}[\mathsf{KD}[k],\mathsf{AE}],E}(\mathcal{A})] \ .$$

Subtracting, we get

$$\mathsf{Adv}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\overline{\mathcal{A}}) = \frac{1}{2}\big(\mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_1) - \mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\mathsf{KtE}[\mathsf{KD}[k],\mathsf{AE}],E}(\mathcal{A}_1)\big) \ .$$

Note that $\overline{\mathcal{A}}$ makes at most $p + L \le 2^{n-4}$ ideal-cipher queries, and q $\textsc{Eval}$ queries. Moreover, $\overline{\mathcal{A}}$ is also d-repeating. Hence using Lemma 5,

$$\mathsf{Adv}^{\mathsf{dist}}_{\mathsf{KD}[E],\mathsf{KD}[k]}(\overline{\mathcal{A}}) \le \frac{1}{2^{n/2}} + \frac{24(L+p)q + 18q^2}{2^{k+n}} + \frac{a(L+p) + d(L+p+3q)}{2^k} \ .$$

Putting this all together,

$$\mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\overline{\mathsf{AE}},E}(\mathcal{A}) \le \mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\mathsf{KtE}[\mathsf{KD}[k],\mathsf{AE}],E}(\mathcal{A}) + \frac{2}{2^{n/2}} + \frac{48(L+p)q + 36q^2}{2^{k+n}} + \frac{2a(L+p) + 2d(L+p+3q)}{2^k} \ .$$

This concludes the proof. ☐

6.2 Authenticity Analysis

In Section 6.1, we bound the privacy advantage by constructing a d-repeating adversary distinguishing $\mathsf{KD}[E]$ and $\mathsf{KD}[k]$, and then using Lemma 5. This method does not work for authenticity: the constructed adversary might be q-repeating, because there is no restriction of the nonces in verification queries, and one would end up with an inferior term $q(L+p+q)/2^k$. We instead give a dedicated analysis.

Restricting to simple adversaries. We say that an adversary is *simple* if for any nonce N and user i, if the adversary uses N for an encryption query of user i, then it will never use nonce N on verification queries for user i. Lemma 7 below reduces the authenticity advantage of a general adversary against $\mathsf{KtE}[\mathsf{KD}[E],\mathsf{AE}]$ to that of a simple adversary; the proof is in Appendix K, and is based on the idea of splitting the cases of where the adversary forges on a fresh (N, i) pair and where it does not, and the latter can be handled using Lemma 5 above. Handling the former is the harder part, which we deal with below. We discuss the bound however below, and give an overview of the proof.

Lemma 7. *Let $n \geq 16$ be an integer and let $k \in \{n, 2n\}$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Let $\mathsf{KD}[E]$ be a natural KDF. Let AE be an AE scheme of key length $n + k$, and let $\overline{\mathsf{AE}} = \mathsf{KtE}[\mathsf{KD}[E], \mathsf{AE}]$. Let $\mathcal{A}_0$ be a d-repeating adversary that makes at most $q \leq 2^{(1-\epsilon)n-4}$ encryption/verification queries and $p \leq 2^{n-5}$ ideal-cipher queries. Suppose that using AE to encrypt $\mathcal{A}_0$'s encryption queries and decrypt its verification queries would need to make $L \leq 2^{n-5}$ ideal-cipher queries. Then, we can construct an adversary $\mathcal{A}_1$ and a simple adversary $\mathcal{A}_2$, both d-repeating, such that*

$$\mathsf{Adv}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_0) \leq \mathsf{Adv}^{\text{mu-auth}}_{\mathsf{KtE}[\mathsf{KD}[k],\mathsf{AE}],E}(\mathcal{A}_1) + \mathsf{Adv}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_2) + \frac{2}{2^{n/2}} + \frac{48(L+p)q + 36q^2}{2^{n+k}} + \frac{2(a+d)L + 2(a+d)p + 6dq}{2^k} \enspace ,$$

where $a = \lceil 1.5/\epsilon \rceil - 1$. Any query of $\mathcal{A}_1$ or $\mathcal{A}_2$ is also a query of $\mathcal{A}_0$.

Handling simple adversaries. Lemma 8 below shows that the AE scheme $\mathsf{KtE}\big[\mathsf{KD}[E], \mathsf{SIV}[\mathsf{GMAC}^+[H,E], \mathsf{CTR}[E]]\big]$ has good authenticity against simple adversaries, for any 2-unpredictable, natural KDF $\mathsf{KD}[E]$. The proof is in Appendix L; it also uses some technical balls-into-bins results in Appendix A. Note that here we can handle both regular and weakly regular hash functions. (If we instead consider just regular hash functions, we can slightly improve the bound, but the difference is inconsequential.)

Lemma 8. *Fix $0 < \epsilon < 1$ and let $a = \lceil 1.5/\epsilon \rceil - 1$. Let $n \geq 128$ be an integer, and let $k \in \{n, 2n\}$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Let $H : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ be a hash function that is either c-regular or weakly c-regular. Let $\mathsf{KD}[E]$ be a natural, 2-unpredictable KDF. Let $\mathsf{AE} = \mathsf{SIV}[\mathsf{GMAC}^+[H,E], \mathsf{CTR}[E]]$ and $\overline{\mathsf{AE}} = \mathsf{KtE}[\mathsf{KD}[E], \mathsf{AE}]$. Let $\mathcal{A}$ be a d-repeating, simple adversary that makes at most $p \leq 2^{(1-\epsilon)n-8}$ ideal-cipher queries, and $q \leq 2^{(1-\epsilon)n-8}$ encryption/verification queries whose total block length is at most $L \leq 2^{(1-\epsilon)n-8}$. Then*

$$\mathsf{Adv}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}) \leq \frac{3}{2^{n/2}} + \frac{11q}{2^n} + \frac{288(L+p)q + 36q^2 + 48c(L+p+q)L}{2^{n+k}} + \frac{(8a + 7a^2 + 3d)q}{2^k} + \frac{(na + 6a + 6d)L + 6(a+d)p}{2^k} \enspace .$$

Discussion. The bound in Lemma 8 consists of three important terms $\frac{q}{2^n}$, $\frac{pd}{2^k}$, and $\frac{naL}{2^k}$, each corresponding to an actual attack. Let us revisit these, as this will be helpful in explaining the proof below. First, since the IV length is only n-bit long, even if an adversary simply outputs q verification queries in a random fashion, it would get an advantage about $\frac{q}{2^n}$. Next, for the term $\frac{pd}{2^k}$, consider an adversary that picks a long enough message M and then makes encryption queries $(1, N, M, A), \ldots, (d, N, M, A)$ of the same nonce N and associated data, for answers $C_1, \ldots, C_d$ respectively. (Recall that the adversary is d-repeating, so it cannot use the nonce N in encryption queries for more than d users.) By picking

p candidate master keys $K_1, \ldots, K_p$ and comparing C_i with $\overline{\mathsf{AE}}.\mathsf{E}(K_j, N, M, A)$ for all $i \le d$ and $j \le p$, the adversary can recover one master key with probability about $\frac{pd}{2^k}$.

Finally, for the term $\frac{naL}{2^k}$, consider the following attack. The adversary first picks a nonce N and p candidate keys $K_1, \ldots, K_p$, and then queries $R_{0,j} \leftarrow E_K(K_j, \mathsf{pad}(N, 0)), \ldots, R_{5,j} \leftarrow E(K_j, \mathsf{pad}(N, 5))$ for every $j \le p$. Let $K_{\mathsf{in}}^j \| K_{\mathsf{out}}^j \leftarrow \mathsf{KD.Map}(R_{0,j}, \ldots, R_{5,j})$. Now, if some K_j is the master key of some user i then $K_{\mathsf{in}}^j \parallel K_{\mathsf{out}}^j$ will be the session key of that user i for nonce N. The adversary then picks an arbitrary ciphertext C, and then computes $M_j \leftarrow \mathsf{CTR}[E].\mathsf{D}(K_j, C)$ and $V_j \leftarrow E^{-1}(K_{\mathsf{out}}^j, T)$ for each $j \le p$, where T is the IV of C. The goal of the adversary is to make a sequence of q verification queries $(1, N, C, A), \ldots, (q, N, C, A)$, for an ℓ-block associated data A that it will determine later. (Recall that in verification queries, the adversary can reuse a nonce across as many users as it likes.) To maximize its chance of winning, the adversary will iterate through every possible string A^* of block length ℓ, and let $\mathit{count}(A^*)$ denote the number of j's that $\mathsf{xor}(H(K_{\mathsf{in}}^j, M_j, A^*), N) = V_j$. Then it picks A as the string to maximize $\mathit{count}(A)$. The proof of Lemma 8 essentially shows that with very high probability, we have $\mathit{count}(A) \le na(\ell + |C|_n) \le \frac{naL}{q}$, and thus the advantage of this attack is bounded by $\frac{naL}{2^k}$.

Proof ideas. We now sketch some ideas in the proof of Lemma 8. First consider an adversary that does not use the encryption oracle. Assume that the adversary does not repeat a prior ideal-cipher query, or make redundant ideal-cipher queries. For each query $E_K(Y)$ of answer Y, create an entry $(\mathtt{prim}, K, X, Y, +)$. Likewise, for each query $E_K^{-1}(Y)$ of answer X, create an entry $(\mathtt{prim}, K, X, Y, -)$. Consider a verification query (i, N, C, A). Let K_i be the secret master key of user i, and let $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ be the session key of user i for nonce N. Let T be the IV of C. The proof examines several cases, but here we only discuss a few selective ones. If there is no entry $(\mathtt{prim}, K_i, X, Y, \cdot)$ such that $X \in \{\mathsf{pad}(N, 0), \ldots, \mathsf{pad}(N, 5)\}$ then given the view of the adversary, the session key $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ still has at least $k + n - 1$ bits of (conditional) min-entropy. In this case, the chance that $\mathsf{AE.D}(K_{\mathsf{in}} \parallel K_{\mathsf{out}}, N, C, M)$ returns a non-$\perp$ answer is roughly $1/2^n$. Next, suppose that there is an entry $(\mathtt{prim}, K, X, Y, -)$ such that $K = K_i$ and $X \in \{\mathsf{pad}(N, 0), \ldots, \mathsf{pad}(N, 5)\}$. By using some balls-into-bins analysis,[7] we can argue that it is very likely that there are at most $6a$ entries $(\mathtt{prim}, K^*, X^*, Y^*, -)$ such that $X^* \in \{\mathsf{pad}(N, 0), \ldots, \mathsf{pad}(N, 5)\}$. Hence the chance this case happens is at most $6a/2^k$.

Now consider the case that there are entries $(\mathtt{prim}, K_i, \mathsf{pad}(N, 0), R_0, +), \ldots, (\mathtt{prim}, K_i, \mathsf{pad}(N, 5), R_5, +)$, and $(\mathtt{prim}, K_{\mathsf{out}}, V, T, -)$, with $V \in 0\{0, 1\}^{n-1}$ and $K_{\mathsf{in}} \| K_{\mathsf{out}} \leftarrow \mathsf{KD.Map}(R_0, \ldots, R_5)$. This corresponds to the last attack in the discussion above. We need to bound $\Pr[\mathsf{Bad}]$, where Bad is the the event (i) this case happens, and (ii) $V = \mathsf{xor}(H(K_{\mathsf{in}}, M, A), N)$, where $M \leftarrow \mathsf{CTR}[E].\mathsf{D}(K_{\mathsf{out}}, C)$.

[7] We note that this is not the classic balls-into-bins setting, because the balls are thrown in an inter-dependent way. In Appendix A we analyze this biased balls-into-bins setting.

This is highly non-trivial because somehow the adversary already sees the keys K_i and $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$, and can *adaptively* pick (C, A), as shown in the third attack above.

To deal with this, we consider a *fixed* (i^*, N^*, C^*, A^*). There are at most p septets $\mathcal{T}$ of entries $(\texttt{prim}, K, \mathsf{pad}(N^*, 0), R_0^*, +), \ldots, (\texttt{prim}, K, \mathsf{pad}(N^*, 5), R_5^*, +)$ and $(\texttt{prim}, J, U, T^*, -)$, with $U \in 0\{0,1\}^{n-1}$ and $J' \parallel J \leftarrow \mathsf{KD.Map}(R_0^*, \ldots, R_5^*)$. We then show that the chance that there are $n\ell a$ such septets $\mathcal{T}$ such that $\mathsf{xor}(H(J'(\mathcal{T}), M^*(\mathcal{T}), A^*), N^*) = U(\mathcal{T})$ is at most $2^{1-(3\ell n+2n)}$, where $\ell = |C^*|_n + |A^*|_n \geq 2$ and $M^*(\mathcal{T}) \leftarrow \mathsf{CTR}[E].\mathsf{D}(J(\mathcal{T}), C^*)$. Hence, regardless of how the adversary picks (i, N, C, A) from all possible choices of (i^*, N^*, C^*, A^*), the chance that there are $na(|C|_n + |A|_n)$ septets $\mathcal{T}$ such that $\mathsf{xor}(H(J'(\mathcal{T}), M(\mathcal{T}), A), N) = U(\mathcal{T})$, where $M(\mathcal{T}) \leftarrow \mathsf{CTR}[E].\mathsf{D}(J(\mathcal{T}), C)$, is at most

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

Thus $\Pr[\mathsf{Bad}] \leq \frac{1}{2^n} + \frac{na \cdot \mathbf{E}[|A|_n + |C|_n]}{2^k}$.

Now we consider the general case where the adversary $\mathcal{A}$ might use the encryption oracle. Clearly if for each encryption query (i, N, M, A), we grant the adversary the session key $\mathsf{KD}[E](K_i, N)$, where K_i is the master key of user i, then it only helps the adversary. Recall that here the adversary is simple, so it cannot query $\textsc{Enc}(i, N, M, A)$ and later query $\textsc{Vf}(i, N, C', A')$. We also let the adversary compute up to $L + p$ ideal-cipher queries, so that the encryption oracle does not have to give the ciphertexts to the adversary. Effectively, we can view that $\mathcal{A}$ is in the following game G_0. It is given access to E/E^{-1} and an oracle $\textsc{Eval}(i, N)$ that generates $\mathsf{KD}[E](i, N)$. Then it has to generate a list of verification queries. The game then tries to decrypt those, and returns true only if some gives a non-$\perp$ answer.

To remove the use of the $\textsc{Eval}$ oracle, it is tempting to consider the variant G_1 of game G_0 where $\textsc{Eval}$ instead implements $\mathsf{KD}[k]$, and then bound the gap between G_0 and G_1 by constructing a d-repeating adversary $\overline{\mathcal{A}}$ distinguishing $\mathsf{KD}[E]$ and $\mathsf{KD}[k]$. However, this approach does not work because it is impossible for $\overline{\mathcal{A}}$ to correctly simulate the processing of the verification queries. Instead, we define game G_1 as follows. Its $\textsc{Eval}$ again implements $\mathsf{KD}[k]$, but after the adversary produces its verification queries, the game tries to *program* E so that the outputs of $\textsc{Eval}$ are consistent with $\mathsf{KD}[E]$ on random master keys $K_1, K_2, \cdots \leftarrow_{\$} \{0,1\}^{n+k}$. (But E still has to remain consistent with its past ideal-cipher queries.) Of course it is not always possible, because the fake $\textsc{Eval}$ might have generated some inconsistency. In this case, the game returns false, meaning that the adversary *loses*. If there is no inconsistency, then after the programming, the game processes the verification queries as in G_0.

To bound the gap between G_0 and G_1, we will construct a d-repeating adversary $\overline{\mathcal{A}}$ distinguishing $\mathsf{KD}[E]$ and $\mathsf{KD}[k]$, but additionally, it wants to be granted the master keys after it finishes querying. Note that Lemma 5 applies to this key-revealing setting. Now, after the adversary $\overline{\mathcal{A}}$ finishes querying, it is granted the master keys and checks for inconsistency between the outputs of EVAL and the ideal-cipher queries. If there is inconsistency then $\overline{\mathcal{A}}$ outputs 0, indicating that it has been dealing with $\mathsf{KD}[k]$. Otherwise, it has to simulate the processing of the verification queries. However, although it knows the keys now, it can no longer queries E. Instead, $\overline{\mathcal{A}}$ tries to sample an *independent* blockcipher $\tilde{E}$, subject to (1) $\tilde{E}$ and E agree on the outputs of the past ideal-cipher queries, and the outputs of EVAL are consistent with $\mathsf{KD}[\tilde{E}]$ on master keys $K_1, K_2, \ldots$. It then processes the verification queries using this blockcipher $\tilde{E}$ instead of E.

Although the game G_1 above does not completely remove the use of the EVAL oracle, it still creates some sort of independence between the sampling of the master keys, and the outputs that the adversary $\mathcal{A}$ receives, allowing us to repeat several proof ideas above.

HANDLING GENERAL ADVERSARIES. Combining Lemmas 7 and 8, we immediately obtain the following result.

Lemma 9. *Fix $0 < \epsilon < 1$ and let $a = \lceil 1.5/\epsilon \rceil - 1$. Let $n \geq 128$ be an integer, and let $k \in \{n, 2n\}$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Let $H : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ be a hash function that is either c-regular hash or weakly c-regular. Let $\mathsf{KD}[E]$ be a natural, 2-unpredictable KDF. Let $\mathsf{AE} = \mathsf{SIV}[\mathsf{GMAC}^+[H,E], \mathsf{CTR}[E]]$ and $\overline{\mathsf{AE}} = \mathsf{KtE}[\mathsf{KD}[E], \mathsf{AE}]$. Let $\mathcal{A}$ be a d-repeating adversary that makes at most $p \leq 2^{(1-\epsilon)n-8}$ ideal-cipher queries, and $q \leq 2^{(1-\epsilon)n-8}$ encryption/verification queries whose total block length is at most $L \leq 2^{(1-\epsilon)n-8}$. Then we can construct a d-repeating adversary $\overline{\mathcal{A}}$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{mu\text{-}auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}) &\leq \mathsf{Adv}^{\mathsf{mu\text{-}auth}}_{\mathsf{KtE}[\mathsf{KD}[k],\mathsf{AE}],E}(\overline{\mathcal{A}}) + \frac{5}{2^{n/2}} + \frac{11q}{2^n} + \frac{336(L+p)q + 72q^2}{2^{n+k}} \\ &\quad + \frac{48c(L+p+q)L}{2^{n+k}} + \frac{(8a+7a^2+9d)q + (na+8a+8d)L + 8(a+d)p}{2^k} \ .\end{aligned}$$

Moreover, any query of $\overline{\mathcal{A}}$ is also a query of $\mathcal{A}$.

6.3 Unwinding Mu-Mrae Security

The following Theorem 6 concludes the mu-mrae security of AE scheme $\overline{\mathsf{AE}} = \mathsf{KtE}[\mathsf{KD}[E], \mathsf{SIV}[\mathsf{GMAC}^+[H,E], \mathsf{CTR}[E]]]$; the proof is in Appendix M. Note that here we can handle both regular and weakly regular hash functions. (If we instead consider just regular hash functions, we can slightly improve the bound, but the difference is inconsequential.)

Theorem 6 (Security of AES-GCM-SIV). *Let $n \geq 128$ be an integer, and let $k \in \{n, 2n\}$. Fix $0 < \epsilon < 1$ and let $a = \lceil 1.5n/(n-1)\epsilon \rceil - 1$. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher that we will model as an ideal cipher. Let $H : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ be a c-AXU hash function. Moreover,* *either H is c-regular, or weakly c-regular. Let* $\mathsf{KD}[E]$ *be a natural, 2-unpredictable KDF. Let* $\mathsf{AE} = \mathsf{SIV}[\mathsf{GMAC}^+[H, E], \mathsf{CTR}[E]]$ *and* $\overline{\mathsf{AE}} = \mathsf{KtE}[\mathsf{KD}[E], \mathsf{AE}]$. *Let* $\mathcal{A}$ *be a d-repeating adversary that makes at most* $p \le 2^{(1-\epsilon)n-8}$ *ideal-cipher queries, and* $q \le 2^{(1-\epsilon)n-8}$ *encryption/verification queries whose total block length is at most* $L \le 2^{(1-\epsilon)n-8}$ *and encryption queries of at most B blocks per (user, nonce) pair. Then,*

$$\mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\overline{\mathsf{AE}},E}(\mathcal{A}) \le \frac{10}{2^{n/2}} + \frac{(17a + 4a^2 + 24d + na)L + (22a + 13d)p}{2^k} + \frac{(48c + 30)LB}{2^n} + \frac{(303 + 108c)L^2 + (192 + 96c)Lp}{2^{n+k}} .$$

We note that one way that d can be kept small is by choosing nonces randomly, or at least with sufficient entropy. Then, by a classical balls-into-bins analysis, if q is quite smaller than 2^{nl}, where nl is the nonce length, which holds in practice for $\mathsf{nl} = 96$, then the value d is bounded by a constant with high probability. We also point out that if d cannot be bounded, then our security bound still gives very meaningful security guarantees if $k = 2n$ (i.e., this would have us use AES-256). As there is a matching attack in the unbounded d case, which just exploits key collisions, this suggests the need to increase keys to 256 in the multi-user case. However, many uses in practice will have d bounded, and for these 128-bit keys will suffice.

Acknowledgments. We thank Shay Gueron and Yehuda Lindell for insightful feedback.

Priyanka Bose and Stefano Tessaro were supported by NSF grants CNS-1553758 (CAREER), CNS-1423566, CNS-1719146, CNS-1528178, and IIS-1528041, and by a Sloan Research Fellowship. Viet Tung Hoang was supported in part by NSF grant CICI-1738912 and the First Year Assistant Professor Award of Florida State University.

References

1. M. Abdalla and M. Bellare. Increasing the lifetime of a key: a comparative analysis of the security of re-keying techniques. In T. Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 546–559. Springer, Heidelberg, Dec. 2000.
2. M. Bellare, D. J. Bernstein, and S. Tessaro. Hash-function based PRFs: AMAC and its multi-user security. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 566–595. Springer, Heidelberg, May 2016.
3. 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.
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. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *38th FOCS*, pages 394–403. IEEE Computer Society Press, Oct. 1997.
6. M. Bellare and V. T. Hoang. Identity-based Format-Preserving Encryption. In *CCS 2017*, 2017.
7. M. Bellare and R. Impagliazzo. A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP to PRF conversion. Cryptology ePrint Archive, Report 1999/024, 1999. http://eprint.iacr.org/1999/024.
8. M. Bellare, T. Krovetz, and P. Rogaway. Luby-Rackoff backwards: Increasing security by making block ciphers non-invertible. In K. Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 266–280. Springer, Heidelberg, May / June 1998.
9. M. Bellare and B. Tackmann. The multi-user security of authenticated encryption: AES-GCM in TLS 1.3. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 247–276. Springer, Heidelberg, Aug. 2016.
10. S. Chen and J. P. Steinberger. Tight security bounds for key-alternating ciphers. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 327–350. Springer, Heidelberg, May 2014.
11. W. Dai, V. T. Hoang, and S. Tessaro. Information-theoretic indistinguishability via the chi-squared method. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 497–523. Springer, Heidelberg, Aug. 2017.
12. S. Gilboa and S. Gueron. Distinguishing a truncated random permutation from a random function. Cryptology ePrint Archive, Report 2015/773, 2015. http://eprint.iacr.org/2015/773.
13. S. Goldwasser and M. Bellare. Lecture notes on cryptography. Summer Course "Cryptography and Computer Security" at MIT, 1999.
14. S. Gueron, A. Langley, and Y. Lindell. AES-GCM-SIV: Specification and analysis. Cryptology ePrint Archive, Report 2017/168, 2017. http://eprint.iacr.org/2017/168.
15. S. Gueron and Y. Lindell. GCM-SIV: Full nonce misuse-resistant authenticated encryption at under one cycle per byte. In I. Ray, N. Li, and C. Kruegel:, editors, *ACM CCS 15*, pages 109–119. ACM Press, Oct. 2015.
16. S. Gueron and Y. Lindell. Better bounds for block cipher modes of operation via nonce-based key derivation. In *CCS 2017*, 2017.
17. V. T. Hoang and S. Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 3–32. Springer, Heidelberg, Aug. 2016.
18. V. T. Hoang and S. Tessaro. The multi-user security of double encryption. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 381–411. Springer, Heidelberg, May 2017.
19. T. Iwata. New blockcipher modes of operation with beyond the birthday bound security. In M. J. B. Robshaw, editor, *FSE 2006*, volume 4047 of *LNCS*, pages 310–327. Springer, Heidelberg, Mar. 2006.
20. T. Iwata and Y. Seurin. Reconsidering the security bound of AES-GCM-SIV. Cryptology ePrint Archive, Report 2017/708, 2017. http://eprint.iacr.org/2017/708.
21. T. Iwata and Y. Seurin. Reconsidering the security bound of aes-gcm-siv. Cryptology ePrint Archive, Report 2017/708, 2017. https://eprint.iacr.org/2017/708.

22. S. Lucks. The sum of PRPs is a secure PRF. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 470–484. Springer, Heidelberg, May 2000.
23. A. Luykx, B. Mennink, and K. G. Paterson. Analyzing multi-key security degradation. Cryptology ePrint Archive, Report 2017/435, 2017. http://eprint.iacr.org/2017/435.
24. 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.
25. 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.
26. 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.
27. 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.
28. J. Patarin. A proof of security in O(2n) for the xor of two random permutations. In R. Safavi-Naini, editor, *ICITS 08*, volume 5155 of *LNCS*, pages 232–248. Springer, Heidelberg, Aug. 2008.
29. J. Patarin. The “coefficients H” technique (invited talk). In R. M. Avanzi, L. Keliher, and F. Sica, editors, *SAC 2008*, volume 5381 of *LNCS*, pages 328–345. Springer, Heidelberg, Aug. 2009.
30. J. Patarin. Introduction to mirror theory: Analysis of systems of linear equalities and linear non equalities for cryptography. Cryptology ePrint Archive, Report 2010/287, 2010. http://eprint.iacr.org/2010/287.
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.
32. 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.
33. J. Vance and M. Bellare. Delegatable Feistel-based Format Preserving Encryption mode. Submission to NIST, Nov 2015.
34. M. N. Wegman and L. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22:265–279, 1981.

A Biased Balls-Into-Bins

Consider the following game in which we throw q balls into 2^m bins. The throws can be inter-dependent, but for each i-th throw, conditioning on the result of the prior throws, the conditional probability that the i-th ball falls into any particular bin is at most 2^{1-m}. Let $\mathrm{Balls}(q, m)$ denote the random variable for the number of balls in the heaviest bin in this game. The following result gives a strong concentration bound on $\mathrm{Balls}(q, m)$ when q is quite smaller than 2^m.

Lemma 10. *Fix $0 < \epsilon < 1$. Let $m, q \in \mathbb{N}$ such that $q \le 2^{(1-\epsilon)m-1}$. Then*

$$\Pr\Big[\text{Balls}(q, m) \ge \lceil 1.5/\epsilon \rceil\Big] \le 2^{-m/2} \enspace .$$

Proof. Let $s = m - 1$ and $r = \lceil 1.5/\epsilon \rceil$. Since the adversary throws at most q balls, there are

$$\binom{q}{r} \le \frac{q^r}{r!} \le \frac{q^r}{2}$$

sets of r balls. For each set, the chance that all balls in the set land in the same bin is at most $2^{-(r-1)s}$. Hence the chance that there are r balls landing in the same bin is at most

$$\frac{q^r}{2 \cdot 2^{(r-1)s}} \le \frac{2^{r(1-\epsilon)s}}{2 \cdot 2^{(r-1)s}} = 2^{-1-(\epsilon r - 1)s} \le 2^{-(s/2+1)} \le 2^{-m/2} \enspace .$$

This concludes the proof. □

Next, we give a concentration bound on $\text{Balls}(q, m)$ when q might be quite bigger than 2^m.

Lemma 11. *Fix $0 < \epsilon < 1$ and $m \in \mathbb{N}$ such that $m \ge 128$. Let $m, \ell, c, q \in \mathbb{N}$ such that $\ell \ge 2$ and $q \le c \cdot 2^m$. Then*

$$\Pr\Big[\text{Balls}(q, m) \ge \lceil c\ell m/2 \rceil\Big] \le 2^{-(3\ell+2)m} \enspace .$$

Proof. Let $s = m - 1$ and $r = \lceil c\ell m/2 \rceil \ge 128c$. Since the adversary throws at most q balls, there are

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

sets of r balls. For each set, the chance that all balls in the set land in the same bin is at most $2^{-(r-1)s}$. Hence the chance that there are r balls landing in the same bin is at most

$$\begin{aligned} \frac{q^r}{r! \cdot 2^{(r-1)s}} \le \frac{(2c)^r 2^{rs}}{r! \cdot 2^{(r-1)s}} = \frac{(2c)^r 2^s}{r!} \le \frac{(2c)^r 2^m}{(r/e)^r} = \frac{2^m}{(r/2ec)^r} \\ \le \frac{2^m}{(64/e)^{\ell m}} \le \frac{2^m}{(8/e)^{2m} \cdot 8^{\ell m}} \le 2^{-(3\ell+2)m} \enspace , \end{aligned}$$

where the second inequality is due to the fact that $n! \ge (n/e)^n$ for every integer $n \ge 1$, and the second last inequality is due to the hypothesis that $\ell \ge 2$. This concludes the proof. □

$\underline{\mathsf{PolyVal}[\mathbb{F}](K, M, A)}$
$X \leftarrow A0^* \parallel M0^* \parallel [|A|]_{n/2} \parallel [|M|]_{n/2}$
$X_1 \cdots X_m \leftarrow X$ // Each $|X_i| = n$
// Interpret K and $X_1, \ldots, X_m$ as elements in $\mathbb{F}$
$Y \leftarrow X_1 \bullet K^m \oplus X_2 \bullet K^{m-1} \oplus \cdots \oplus X_m \bullet K$
Return Y

Fig. 9: **The POLYVAL hash function.** For a string Z, we write $Z0^*$ to denote the string obtained by padding 0's to Z until the next n-bit boundary. In particular, if $|Z|$ is divisible by n then $Z0^* = Z \parallel 0^n$. For a number $t \in \{0, \ldots, 2^{n/2} - 1\}$, we write $[t]_r$ to denote an r-bit representation of t.

B The POLYVAL Hash Function

Let $n \geq 2$ be an even integer. Let $\mathbb{F}$ be a finite field of 2^n elements, meaning that we can interpret a string in $\{0,1\}^n$ as an element in $\mathbb{F}$, and vice versa. Assume that the string 0^n is interpreted as the zero element of $\mathbb{F}$, and the addition operator in $\mathbb{F}$ is equivalent to xor in $\{0,1\}^n$. Let $\bullet$ denote the multiplication operator of $\mathbb{F}$. The POLYVAL hash function $\mathsf{PolyVal}[\mathbb{F}] : \{0,1\}^n \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ is defined as in Fig. 9. Note that if $M = A = \varepsilon$ then $\mathsf{PolyVal}[\mathbb{F}](K, M, A) = 0^n$ for any key K.

Weak regularity of POLYVAL. We first show that PolyVal is weakly 1.5-regular. Consider arbitrary $(M, A) \in (\{0,1\}^*)^2 \backslash (\varepsilon, \varepsilon)$ and $Z \in \{0,1\}^n$. Let $X \leftarrow A0^* \parallel M0^* \parallel [|A|]_{n/2} \parallel [|M|]_{n/2}$, where $Z0^*$ denotes the string obtained by padding 0's to Z until the next n-bit boundary, and $[t]_{n/2}$ denotes an $n/2$-bit representation of the number t. Note that

$$m = |X|_n = |A|_n + |M|_n + 1 \leq 1.5(|A|_n + |M|_n),$$

since $|A|_n, |M|_n \geq 1$. Let $X_1 \cdots X_m \leftarrow X$, where each $|X_i| = n$. Let

$$f(x) = X_1 \bullet x^m \oplus X_2 \bullet x^{m-1} \oplus \cdots \oplus X_m \bullet x \oplus Z \enspace .$$

Note that f is a polynomial of degree at most m, and since $(M, A) \neq (\varepsilon, \varepsilon)$, f is non-zero. Hence f has at most m roots. If we pick $K \leftarrow\!\!\$\ \{0,1\}^n$, the chance that K is one of those m roots is at most $m/2^n \leq 1.5(|M|_n + |A|_n)/2^n$. Hence Hence

$$\begin{aligned}\Pr_{K \leftarrow\$ \{0,1\}^n}[\mathsf{PolyVal}[\mathbb{F}](K, M, A) = Z] &= \Pr_{K \leftarrow\$ \{0,1\}^n}[f(K) = 0^n] \\ &\leq \frac{1.5(|M|_n + |A|_n)}{2^n}\end{aligned}$$

and thus PolyVal is weakly 1.5-regular.

XOR universality of POLYVAL. Next, we show that PolyVal is 1.5-AXU. Consider distinct (M, A) and (M', A') in $(\{0,1\}^*)^2$, and fix $Z \in \{0,1\}^n$. Let

$X \leftarrow A0^* \parallel M0^* \parallel [|A|]_{n/2} \parallel [|M|]_{n/2}$, and $X' \leftarrow A'0^* \parallel M'0^* \parallel [|A'|]_{n/2} \parallel [|M'|]_{n/2}$. Without loss of generality, assume that $m = |X|_n \geq |X'|_n = \ell$. Note that

$$m = |X|_n = |A|_n + |M|_n + 1 \leq 1.5(|A|_n + |M|_n),$$

since $|A|_n, |M|_n \geq 1$. Let $X_1 \cdots X_m \leftarrow X$ and $X'_1 \cdots X'_\ell \leftarrow X'$, where $|X_i| = |X'_j| = n$. Let

$$g(x) = (X_1 \bullet x^m \oplus \cdots \oplus X_m \bullet x) \oplus (X'_1 \bullet x^\ell \oplus \cdots \oplus X'_\ell \bullet x) \oplus Z \ .$$

Note that $g(x)$ is a polynomial of degree at most m, and since $(M, A) \neq (M', A')$, g is non-zero. Hence g has at most m roots. If we pick $K \leftarrow\!\!\$\ \{0,1\}^n$, the chance that K is one of those m roots is at most $m/2^n \leq 1.5(|M|_n + |A|_n)/2^n$. Hence

$$\Pr_{K \leftarrow\!\$\ \{0,1\}^n}\big[\mathsf{PolyVal}[\mathbb{F}](K, M, A) \oplus \mathsf{PolyVal}[\mathbb{F}](K, M', A') = Z\big]$$
$$= \Pr_{K \leftarrow\!\$\ \{0,1\}^n}[g(K) = 0^n] \leq \frac{1.5(|M|_n + |A|_n)}{2^n}$$

and thus $\mathsf{PolyVal}$ is 1.5-AXU.

Fixing the weak regularity of POLYVAL. As shown above, $\mathsf{PolyVal}$ is just weakly regular. There are several ways to make $\mathsf{PolyVal}$ regular. For example, instead of padding M and A with 0's, one can pad them with 1's. The resulting construction would be 1.5-regular and 1.5-AXU.

C Proof of Proposition 1

Without loss of generality, assume that if a verification query returns true then the adversary $\mathcal{A}_0$ will simply terminate and return 1. This can only increase its advantage. Assume that $\mathcal{A}_0$ never repeats an encryption query, and if it queries (i, N, M, A) to Enc for a ciphertext C, then subsequently, it will not query (i, N, C, A) to Vf.

We now construct an adversary $\overline{\mathcal{A}}$ attacking the mrae security of AE, but it only calls Vf after finishing querying Enc. Adversary $\overline{\mathcal{A}}$ runs $\mathcal{A}_0$, and uses its Enc and New oracles to respond to the latter's queries accordingly. For each verification query of $\mathcal{A}_0$, adversary $\overline{\mathcal{A}}$ simply returns false, but stores the query in a set S. When $\mathcal{A}_0$ terminates and outputs a bit b', adversary $\overline{\mathcal{A}}$ will iterate over queries in its set S. For each query (i, N, C, A) in S, if there is an encryption query (i, N, M, A) with answer C (that is made after $\mathcal{A}_0$ queries Vf(i, N, C, A)), then $\overline{\mathcal{A}}$ will terminate and output 1. Otherwise, it will query Vf(i, N, C, A), and if the answer is true, it will again terminate and output 1. If all verification queries return false then $\overline{\mathcal{A}}$ outputs b'. Let a and b be the challenge bits of game $\mathbf{G}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\overline{\mathcal{A}})$ and $\mathbf{G}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}_0)$ respectively. Then

$$\Pr[\mathbf{G}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\overline{\mathcal{A}}) \mid a = 1] = \Pr[\mathbf{G}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}_0) \mid b = 1] \ .$$

Indeed, in the real world, if some verification query (i, N, C, A) of $\mathcal{A}_0$ can return true then $\mathcal{A}_0$ will output 1, and so does $\overline{\mathcal{A}}$: either $\overline{\mathcal{A}}$ will eventually query (i, N, C, A) to get answer true, or later there is an encryption query (i, N, M, A) of answer C that makes $\overline{\mathcal{A}}$ outputs 1. If no verification query of $\mathcal{A}_0$ can return true then $\overline{\mathcal{A}}$ correctly simulates the verification oracle for $\mathcal{A}_0$, and both will give the same answer b'. On the other hand,

$$\Pr[\mathbf{G}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\overline{\mathcal{A}}) \mid a = 0] \ge \Pr[\mathbf{G}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}_0) \mid b = 0] - \frac{2q_v}{2^n} \ .$$

Indeed, in the ideal world, adversary $\overline{\mathcal{A}}$ correctly simulates the verification oracle for $\mathcal{A}_0$. The answer of the two adversaries will be different only if there is a verification query (i, N, C, A) and a subsequent encryption query (i, N, M, A) with the same answer C. For each verification query (i, N, C, A), it can be "targeted" by at most 2^{s+1} encryption queries, where $s = |C| - n$, but the chance that some such encryption query can result in the same ciphertext C is at most $2^{s+1}/2^{|C|} = 2/2^n$. Summing this over q_v verification queries gives us the bound $2q_v/2^n$. Hence

$$\mathsf{Adv}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\overline{\mathcal{A}}) \ge \mathsf{Adv}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}_0) - \frac{2q_v}{2^n} \ .$$

Recall that $\overline{\mathcal{A}}$ always makes verification queries *after* all encryption queries. We now construct adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$. Adversary $\mathcal{A}_1$ runs $\overline{\mathcal{A}}$, and uses its Enc and New oracles to respond to the latter's queries accordingly. For the verification queries of $\overline{\mathcal{A}}$, adversary $\mathcal{A}_1$ simply answers false. When $\overline{\mathcal{A}}$ outputs a bit a', $\mathcal{A}_1$ also output a'. Adversary $\mathcal{A}_2$ runs $\overline{\mathcal{A}}$ and uses its oracles New and Enc to respond to the latter's queries accordingly. For each verification query of $\overline{\mathcal{A}}$, adversary $\mathcal{A}_2$ queries it to its Vf oracle, but always returns false to $\overline{\mathcal{A}}$. Let game G_1 correspond to game $\mathbf{G}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\overline{\mathcal{A}})$ with challenge bit $a = 1$. Let G_2 be identical to game G_1, except that the verification oracle will always return false. Let G_3 be identical to game G_2, except that now the encryption oracle will always return a fresh random answer of appropriate length. Then

$$\mathsf{Adv}^{\text{mu-auth}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}_2) \ge \Pr[G_1] - \Pr[G_2]$$

because it is impossible for $\overline{\mathcal{A}}$ to distinguish G_1 and G_2, unless it manages to trigger the verification oracle to return a true answer in game G_1. On the other hand,

$$\mathsf{Adv}^{\text{mu-priv}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}_1) = \Pr[G_2] - \Pr[G_3] \ .$$

Summing up,

$$\begin{aligned}
\mathsf{Adv}^{\text{mu-priv}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}_1) + \mathsf{Adv}^{\text{mu-auth}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}_2) &\ge \Pr[G_1] - \Pr[G_3] \\
&= \mathsf{Adv}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\overline{\mathcal{A}}) \\
&\ge \mathsf{Adv}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},\Pi}(\mathcal{A}_0) - \frac{2q_v}{2^n} \ .
\end{aligned}$$

This concludes the proof.

D Proof of Theorem 1

Our proof uses the H-coefficient method. We let $\mathbf{S}_0$ and $\mathbf{S}_1$ be two systems which models the oracles accessed by $\mathcal{A}$ in the game $\mathbf{G}^{\mathsf{mu\text{-}ind}}_{\mathsf{AE},B}(\mathcal{A})$ in the cases where ciphertexts are real ($b = 1$) or random ($b = 0$), respectively. Here, $\mathcal{A}$ is deterministic without loss of generality, and transcripts only contain two types of queries:

1. **Encryption queries** have form $(\mathtt{enc}, i, M, (\mathsf{IV}, C))$, where i indicates the user for which the query has been made, $M \in \{0,1\}^*$ is the plaintext, $\mathsf{IV} \in \{0,1\}^n$ is the first block of the ciphertext, whereas C are the remaining blocks. We will normally think of C as made of n-bit blocks $C[1], \ldots, C[\ell]$ and M of blocks $M[1], \ldots, M[\ell]$.
2. **Ideal-cipher queries** have form $(\mathtt{prim}, K, u, v)$, and correspond to the adversary making a query to the ideal cipher, either (K, u) (in the forward direction) or (K, v) in the backward direction, returning u and v, respectively.[8]

We do not record $\mathcal{A}$'s New queries explicitly, but add the resulting keys $K_1, \ldots, K_u$ to the transcript (note that such keys are generated even in the ideal case, just never used). We also assume without loss of generality that if an encryption query for user i appears, then v has been previously increased beyond i using New queries.

Further, let us fix a transcript τ be a transcript with u keys $K_1, \ldots, K_u$, and let $\mathcal{K} = \mathcal{K}(\tau) = \{K_1, \ldots, K_u\}$.[9] Also, let $\mathsf{q} = (\mathtt{enc}, i, M, (\mathsf{IV}, C)) \in \tau$ such that M and C are made of the n-bit blocks $M[1], \ldots, M[\ell]$ and $C[1], \ldots, C[\ell]$. Then, we define the following multi-sets (i.e., elements are allowed to be repeated)

$$U(\mathsf{q}) = \{\mathsf{IV} + 1, \ldots, \mathsf{IV} + \ell\} ,$$
$$V(\mathsf{q}) = \{C[1] \oplus M[1], \ldots, C[\ell] \oplus M[\ell]\} ,$$

as well as $K(\mathsf{q}) = K_i$. Then, for any $K \in \mathcal{K}$, we let

$$V(K) = \bigcup_{\mathsf{q}: K(\mathsf{q}) = K} V(\mathsf{q}) , \quad U(K) = \bigcup_{\mathsf{q}: K(\mathsf{q}) = K} U(\mathsf{q}) .$$

Here, union is on multisets. Finally, for each $K \in \{0,1\}^k$, we also define $P(K)$ as the set of inputs U such that there exists V with $(\mathtt{prim}, K, U, V) \in \tau$.

Good transcripts, and ratio analysis. With this notation, we can give a definition of good/bad transcripts.

Definition 2 (Good and bad transcripts). *We say that τ is* good *if the following conditions are satisfied, for all $K \in \mathcal{K}$:*

---

[8] It will not be necessary for the transcript to record the direction of ideal-cipher queries, i.e., whether the query is in the forward or backward direction.

[9] Note that as keys may be repeated, we can only ensure $|\mathcal{K}| \le u$.

(a) Each element in $U(K)$ appears once, i.e., there are no repetitions.
(b) Each element in $V(K)$ appears once, i.e., there are no repetitions.
(c) $P(K) \cap U(K) = \emptyset$.

If τ is not good, then it is bad.

In the following, we prove that for all good transcript τ, we have $\mathsf{p}_{\mathbf{S}_0}(\tau) \geq \mathsf{p}_{\mathbf{S}_1}(\tau)$. First off, define by $\mathsf{p}_{\mathsf{KeyGen}}(K_1, \ldots, K_u)$ the probability that New query asked indeed generate these keys. Now, with $N = 2^n$, and q the number of encryption queries,

$$\mathsf{p}_{\mathbf{S}_0}(\tau) = \frac{1}{N^q} \cdot \mathsf{p}_{\mathsf{KeyGen}}(K_1, \ldots, K_u) \cdot \left[\prod_{K \in \{0,1\}^k} \prod_{i=0}^{|P(K)|+|U(K)|-1} \frac{1}{N-i} \right] .$$

where the first term takes into account the random choice of the IVs, the second the choice of the keys, the third the ideal-cipher evaluations within the encryption and direct primitive queries. Note that $\sum_{K \in \{0,1\}^k} |P(K)| = p$ and $\sum_{K \in \mathcal{K}} |U(K)| = L$. On the other hand, in the ideal world,

$$\mathsf{p}_{\mathbf{S}_1}(\tau) = \frac{1}{N^q} \cdot \mathsf{p}_{\mathsf{KeyGen}}(K_1, \ldots, K_u) \cdot \frac{1}{N^L} \cdot \left[\prod_{K \in \{0,1\}^k} \prod_{i=0}^{|P(K)|-1} \frac{1}{N-i} \right] ,$$

since ciphertexts are random. Therefore, $\mathsf{p}_{\mathbf{S}_0}(\tau)/\mathsf{p}_{\mathbf{S}_1}(\tau) \geq 1$, and we can use the H-coefficient technique with $\epsilon = 0$, and only need the probability that a transcript generated in an ideal execution is bad.

Probability of a bad transcript. We now turn to computing the probability of a transcript being bad in $\mathbf{S}_1$. In particular, denote by $\mathcal{X}_1$ the transcript generated by $\mathcal{A}$'s interaction, and let $\mathcal{B}_a, \mathcal{B}_b$ and $\mathcal{B}_c$ be the sets of transcripts which violate (a), (b), or (c) in Definition 2. Then, by the union bound,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}] \leq \Pr[\mathcal{X}_1 \in \mathcal{B}_a] + \Pr[\mathcal{X}_1 \in \mathcal{B}_b] + \Pr[\mathcal{X}_1 \in \mathcal{B}_c] .$$

We now upper bound the three probabilities separately. Note that because we are in the ideal world, and New queries do not return any output and the generated keys are not used, we can think of KeyGen without loss of generality being run at the end of the execution, and generating the resulting keys.

Case a). We let $L_1, L_2, L_3, \ldots$ be the individual lengths of each query performed by the attacker, which can be chosen adaptively. Also let $\mathcal{B}_{a,i}$ for $i \in [q]$ the set of transcripts where the i-th query, of length L_i, generates an input to the ideal cipher which is used in one of the previous queries using the same key. Then,

$$\begin{aligned}
\Pr[\mathcal{X}_1 \in \mathcal{B}_a] &\le \sum_{i=1}^{q} \Pr[\mathcal{X}_1 \in \mathcal{B}_{a,i}] \\
&\le \sum_{i=1}^{q} \sum_{\ell_i \ge 1} \Pr[L_i = \ell_i] \cdot \ell_i \cdot \left[\frac{B}{2^h} + \frac{L}{2^h}\alpha\right] \\
&= \left[\frac{B}{2^n} + \frac{L}{2^h}\alpha\right] \sum_{i=1}^{q} \mathbf{E}[L_i] \\
&= \left[\frac{B}{2^h} + \frac{L}{2^h}\alpha\right] \mathbf{E}\left[\sum_{i=1}^{q} L_i\right] \le \frac{LB}{2^h} + \frac{L^2}{2^h}\alpha \ ,
\end{aligned}$$

because upon generating a new IV for a query encrypting ℓ_i blocks, there is probability at most $\ell_i B/2^h$ that one of the ℓ_i offsets $\mathsf{add}(\mathsf{IV}, 0), \ldots, \mathsf{add}(\mathsf{IV}, \ell_i - 1)$ will collide with one of the offsets to encrypt a previous message for the *same* user, and probability $\ell_i \cdot L/2^h$ that there is a collision with one of the offsets used by some other user. In the latter case, however, such a collision only contributes to the bad event if the keys associated with the two users also collide, thus incurring an additional α multiplicative factor.

Case b). The argument is very similar to the one for Case a). Instead of looking at the offsets $\mathsf{add}(\mathsf{IV}, i)$ generated during an encryption, and checking collisions with previously used offsets, we look at the actual ciphertext blocks which are output (independently and randomly), and make sure they do not provoke the transcript to be in $\mathcal{B}_b$. This gives us a bound of

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_b] \le \frac{LB}{2^n} + \frac{L^2}{2^n}\alpha \ . \tag{5}$$

Case c). Recall that when sampling $\mathcal{X}_1$, the keys $K_1, \ldots, K_u$ are sampled at the end of the execution, independently of it. For a transcript τ, we define by $Z(\tau, U)$ to be the maximal number of encryption queries $\mathsf{q} \in \tau$ such that $U \in U(\mathsf{q})$. Also, let $Z = \max_x Z(\mathcal{X}_1, x)$. Then,

$$\begin{aligned}
\Pr[\mathcal{X}_1 \in \mathcal{B}_c] &\le \sum_{z \le a} \Pr[Z = z] \cdot zp\alpha + \Pr\left[Z > a\right] \\
&\le ap\alpha + \Pr\left[Z > a\right]
\end{aligned}$$

because for every query $(\mathtt{prim}, K, U, V)$, out of p potential ones, there are at most $Z(\mathcal{X}_1, U) \le Z$ queries $\mathsf{q} = (\mathtt{enc}, i, M, (\mathsf{IV}, C))$ such that $U \in U(\mathsf{q})$. Thus, the probability (over the choice of $K_1, \ldots, K_u$) that $K = K_i$ is α. We have then decided to cut the sum at $z = a = \left\lceil \frac{1.5n}{\epsilon h} \right\rceil - 1$, as we are going to justify next that the probability that Z exceeds this is negligible.

This follows from the following lemma, whose proof is found below, and which follows a classical balls-into-bins approach, with some extra care needed to handle the adaptivity of the adversary.

Lemma 12. *Let $L \leq 2^{(1-\epsilon)h-1}$. Then,*

$$\Pr\left[Z \geq \left\lceil \frac{1.5n}{\epsilon h} \right\rceil\right] < 2^{-n/2} \; .$$

This concludes the proof.

Proof (Of Lemma 12). One can without loss of generality consider the following adaptive balls-into-bins game. There are $N = 2^n$ bins, corresponding to the block-cipher inputs. At each query, the attacker chooses *adaptively* a length ℓ, then a random $\mathsf{IV} \leftarrow\!\!\$\ \{0,1\}^n$ is chosen, and a ball is placed into the bins $\mathsf{add}(\mathsf{IV}, i)$ for $i = 0, \ldots, \ell - 1$. We assume without loss of generality the attacker's lengths always sum up to L, as this only increases Z. Then, we let Z_i be the load of bin i, and $Z = \max_i Z_i$.

Fix some bin $i \in [N]$. We will now upper bound Z_i, for an arbitrary strategy. Note that with respect to the goal of maximizing Z_i, without loss of generality, the adversary needs only to know whether a ball was thrown into bin i or not after each move. (It can simulate the rest consistently.) Note that when the adversary chooses a random IV and some length ℓ, we have

$$\Pr[i \in \{\mathsf{add}(\mathsf{IV}, 0), \ldots, \mathsf{add}(\mathsf{IV}, \ell - 1)\}] \leq \frac{\ell}{2^h} \; ,$$

because each individual item has min-entropy h, and is equal i with probability at most $1/2^h$. Imagine we modify the game so that when the adversary selects ℓ, we make $2 \times \ell$ independent attempts to throw a ball into bin i, each of them succeeding with probability $1/2^h$, and if any of this lands into bin i, the adversary learns this (for now, we will not reveal how many balls land in bin i, only none, or at least one). Then, the probability that one ball lands in i is

$$1 - \left(1 - \frac{1}{2^h}\right)^{2\ell} \geq 1 - e^{-\frac{2\ell}{2^h}} \geq \frac{1}{2} \cdot \frac{2\ell}{2^h} = \frac{\ell}{2^h}$$

where we have used the fact that $e^{-x} \leq 1 - \frac{x}{2}$ whenever $x \leq 1$, and $\frac{\ell}{2^h} \leq 1$. Therefore, let Z_i' be load of i in the modified game, we clearly have $\Pr[Z_i \geq m] \leq \Pr[Z_i' \geq m]$ for every m.

But note that because all balls are thrown independently, the latter probability does not become smaller if we consider the setting where $2L$ balls are thrown independently, each hitting i with probability $1/2^h$. Call the resulting value Z_i'' denoting the load of i, we thus have $\Pr[Z_i \geq m] \leq \Pr[Z_i'' \geq m]$. By repeatedly applying the union bound, we have

$$\Pr[Z \geq m] \leq \sum_{i=1}^{N} \Pr[Z_i'' \geq m] \leq N \binom{2L}{m} \left(\frac{1}{2^h}\right)^m < N \left(\frac{2L}{2^h}\right)^m \leq 2^{n - \epsilon h m} \; ,$$

where we have used the fact that $\binom{a}{b} < a^b$ and $L \leq 2^{(1-\epsilon)h-1}$. Now, set $m = \lceil \frac{1.5n}{\epsilon h} \rceil$. Then, the above is upper bounded by $2^{-n/2}$. □

E Proof of Theorem 2

We shall use H-coefficient technique to prove the claimed bound. We define two systems $\mathbf{S}_0$ and $\mathbf{S}_1$ that represents the real game ($b = 1$) and ideal ($b = 0$) game of $\mathbf{G}^{\mathsf{mu\text{-}prf}}_{\mathsf{GMAC^+}[H,E],B,E}(\mathcal{A})$. Also, without loss of generality we assume that our adversary $\mathcal{A}$ is deterministic and it does not repeat queries. After the adversary finishes querying, we grant the adversary the key pairs $\{K^i_{\mathsf{in}}, K^i_{\mathsf{out}}\}_{i=1,\ldots,u}$ for all u users spawned by calls to NEW. (Note that in the ideal world, these keys do not influence the behavior of the system and can be thought as being generated at the end, consistent with the earlier inputs to the NEW queries.) In addition to the key pairs granted to the adversary, a mu-prf transcript τ contains the following two types of queries:

- **Evaluation queries** are the entries of type $(\mathsf{eval}, i, M, A, N, T)$, where i indicates the user that this query targets, M is message, A the associated data, N is the nonce, and T is corresponding tag.
- **Primitive queries** are of type $(\mathtt{prim}, K, U, V)$, which result from a forward ideal-cipher query (K, U) returning V, or a backward ideal-cipher query (K, V) returning U.

Again, for a query q, we write $\mathsf{q} \in \tau$ to denote its appearance in the transcript. Also, for each $K \in \{0,1\}^k$ we define the following numbers:

$$q(K) = \left|\{(\mathsf{eval}, i, M, A, N, T) \in \tau \mid K^i_{\mathsf{out}} = K\}\right|$$

$$p(K) = |\{(\mathtt{prim}, K', U, V) \in \tau \mid K' = K\}|$$

DEFINING BAD TRANSCRIPTS. We say a transcript τ is *bad* if it satisfies one of the following constraints (it is called *good* otherwise).

1. There exist two entries $(\mathsf{eval}, i, M_1, A_1, N_1, T_1)$ and $(\mathsf{eval}, j, M_2, A_2, N_2, T_2)$ such that $(i, M_1, A_1, N_1) \neq (j, M_2, A_2, N_2)$, $K^i_{\mathsf{out}} = K^j_{\mathsf{out}}$ and
$$\mathsf{xor}(H_{K^i_{\mathsf{in}}}(M_1, A_1), N_1) = \mathsf{xor}(H_{K^j_{\mathsf{in}}}(M_2, A_2), N_2)\ . \tag{6}$$
2. There exist two entries $(\mathsf{eval}, i, M_1, A_1, N_1, T_1)$ and $(\mathsf{eval}, j, M_2, A_2, N_2, T_2)$ such that $T_1 = T_2$ and $K^i_{\mathsf{out}} = K^j_{\mathsf{out}}$.
3. There exist entries $(\mathtt{prim}, K, U, V)$ and $(\mathsf{eval}, i, M, A, N, T)$ such that $K^i_{\mathsf{out}} = K$ and $\mathsf{xor}(H_{K^i_{\mathsf{in}}}(M, A), N) = U$.

TRANSCRIPT RATIO. We now need to compare $\mathsf{p}_{\mathbf{S}_1}(\tau)$ and $\mathsf{p}_{\mathbf{S}_0}(\tau)$ for a good transcript τ. First off, note that in the ideal world, all queries are replied randomly and independently, and moreover, keys are also chosen independently of the rest of the transcript, with a certain probability $p^* = \mathsf{p}_{\mathsf{KeyGen}}(\{K^i_{\mathsf{in}}, K^i_{\mathsf{out}}\}_{i=1,\ldots,u})$. Further, ideal-cipher queries are also answered independently of evaluation queries. Therefore,

$$\mathsf{p}_{\mathbf{S}_1}(\tau) = p^* \cdot 2^{-nq} \cdot \prod_{K \in \{0,1\}^k} \prod_{i=0}^{p(K)-1} \frac{1}{2^n - i}\ .$$

In the real world, note that because the transcript τ is good, no two queries to the ideal cipher within EVAL queries with the same outer key K are on the same input, and moreover, such inputs do not appear as inputs of direct PRIM queries (even though the outer key itself might). The keys are also generated with probability p^*. For this reason,

$$\mathsf{p_{S_0}}(\tau) = p^* \cdot \prod_{K \in \{0,1\}^k} \prod_{i=0}^{p(K)+q(K)-1} \frac{1}{2^n - i} ,$$

and thus in particular, because $\sum_K q(K) = q$, we see that $\mathsf{p_{S_0}}(\tau)$ replaces the q factors 2^{-n} in the product in $\mathsf{p_{S_1}}(\tau)$ with other factors of form $\frac{1}{2^n - i} \geq \frac{1}{2^n}$ for some $i \geq 0$, and therefore, $\mathsf{p_{S_0}}(\tau) \geq \mathsf{p_{S_1}}(\tau)$. Thus, we can use the H-coefficient technique with $\epsilon = 0$, and only need to upper bound the probability that an ideal transcript is bad, which we do next.

PROBABILITY OF BAD TRANSCRIPTS. Let $\mathcal{X}_1$ be the random variable for the transcript in the ideal system. Let $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$ be the sets of transcripts that satisfies (1), (2) and (3) according to the definition of bad transcripts. Then by the union bound,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}] \leq \Pr[\mathcal{X}_1 \in \mathcal{B}_1] + \Pr[\mathcal{X}_1 \in \mathcal{B}_2] + \Pr[\mathcal{X}_1 \in \mathcal{B}_3]$$

We now upper bound the three probabilities on the RHS separately.

For (1) and (3), we can assume wlog that the execution has terminated, and the transcript so far is fixed, and the keys $K^1_{\mathsf{in}}, K^1_{\mathsf{out}}, K^2_{\mathsf{in}}, K^2_{\mathsf{out}}, \ldots$ are generated, independently of the execution, and are the only random variables. Assume in particular the execution has involved u users and there are q_i evaluation queries for user i, and thus $\sum_{i=1}^{u} q_i \leq q$. Also, assume that the q_i queries intended for user i have each lengths $\ell_{i,1}, \ell_{i,2}, \ldots, \ell_{i,q_i}$ blocks where we intentionally arrange them to be sorted as $\ell_{i,1} \leq \ell_{i,2} \leq ... \leq \ell_{i,q_i}$. Clearly, for all i, $\sum_{j=1}^{q_i} \ell_{i,j} \leq B$.

We define two subsets $\mathcal{B}_{11}$ and $\mathcal{B}_{12}$ of $\mathcal{B}_1$. The first consists of transcripts in which there are two entries $(\mathsf{eval}, i, M_1, N_1, A_1, T_1)$ and $(\mathsf{eval}, i, M_2, N_2, A_2, T_2)$ with $(M_1, A_1, N_1) \neq (M_2, A_2, N_2)$, and

$$\mathsf{xor}(H_{K^i_{\mathsf{in}}}(M_1, A_1), N_1) = \mathsf{xor}(H_{K^i_{\mathsf{in}}}(M_2, A_2), N_2) .$$

The second set consists of the transcripts with two entries $(\mathsf{eval}, i_1, M_1, A_1, N_1, T_1)$ and $(\mathsf{eval}, i_2, M_2, A_2, N_2, T_2)$ with $i_1 \neq i_2$, $K^{i_1}_{\mathsf{out}} = K^{i_2}_{\mathsf{out}}$, and

$$\mathsf{xor}(H_{K^{i_1}_{\mathsf{in}}}(M_1, A_1), N_1) = \mathsf{xor}(H_{K^{i_2}_{\mathsf{in}}}(M_2, A_2), N_2) .$$

Note that here $(M_1, A_1, N_1) = (M_2, A_2, N_2)$ is allowed.

We start with $\mathcal{B}_{11}$. Then, for any two $(M_1, A_1, N_1) \neq (M_2, A_2, N_2)$ with $(M_1, A_1) \neq (M_2, A_2)$,

$$\begin{aligned}
&\Pr[\mathsf{xor}(H_{K_{\mathsf{in}}^i}(M_1, A_1), N_1) = \mathsf{xor}(H_{K_{\mathsf{in}}^i}(M_2, A_2), N_2)] \\
&\qquad = \Pr[\mathsf{xor}(H_{K_{\mathsf{in}}^i}(M_1, A_1) \oplus H_{K_{\mathsf{in}}^i}(M_2, A_2), N_1 \oplus N_2) = 0^n] \\
&\qquad\qquad \leq \frac{c \cdot \lambda \cdot \beta \cdot \max\{|M_1|_n + |A_1|_n, |M_2|_n + |A_2|_n\}}{2^n} ,
\end{aligned}$$

for the following reasons. The first equality follows by linearity of xor. Then, by λ-regularity of xor, there are at most λ strings Δ such that $\mathsf{xor}(\Delta, N_1 \oplus N_2) = 0^n$. By c-xor-universality, for any such Δ, the number of keys k in $\{0,1\}^n$ that make the xor of the hashes equal Δ is at most $c \cdot \max\{|M_1|_n + |A_1|_n, |M_2|_n + |A_2|_n\}$. By β-AU, the probability of each such key is at most $\beta/2^n$. Clearly, if $(M_1, A_1) = (M_2, A_2)$, but $N_1 \neq N_2$, then the upper bound also holds vacuously by injectivity of xor.

Therefore, by taking the union bound, and exploiting our ordering of queries according to their lengths (recall $C := \beta c \lambda$),

$$\begin{aligned}
\Pr[\mathcal{X}_1 \in \mathcal{B}_{11}] &\leq \sum_{i=1}^{u} \sum_{1 \leq j' < j \leq q_i} \frac{C \cdot \ell_{i,j}}{2^n} = C \sum_{i=1}^{u} \sum_{j=1}^{q_i} (j-1) \cdot \frac{\ell_{i,j}}{2^n} \\
&\leq C \sum_{i=1}^{u} q_i \sum_{j=1}^{q_i} \frac{\ell_{i,j}}{2^n} \leq C \sum_{i=1}^{u} \frac{q_i B}{2^n} \leq \frac{CqB}{2^n} .
\end{aligned}$$

We move on to $\mathcal{B}_{12}$. Note that for any two relevant entries, we have that

$$\begin{aligned}
&\Pr[\mathsf{xor}(H_{K_{\mathsf{in}}^{i_1}}(M_1, A_1), N_1) = \mathsf{xor}(H_{K_{\mathsf{in}}^{i_2}}(M_2, A_2), N_2) \wedge K_{\mathsf{out}}^{i_1} = K_{\mathsf{out}}^{i_2}] \leq \\
&\qquad\qquad \leq \frac{C \cdot \min\{|M_1|_n + |A_1|_n, |M_2|_n + |A_2|_n\}}{2^{n+k}} ,
\end{aligned}$$

because of the following reasons: Assume wlog $|M_1|_n + |A_1|_n \geq |M_2|_n + |A_2|_n$ (otherwise the argument is symmetric). Then, for every for every $K_{\mathsf{in}}^{i_1}$, there are at most λ values Y such that $\mathsf{xor}(H_{K_{\mathsf{in}}^{i_1}}(M_1, A_1), N_1) = \mathsf{xor}(Y, N_2)$ by λ-regularity of xor, and for each such Y, by c-regularity of H, at most $c \cdot (|M_2|_n + |A_2|_n)$ values of $K_{\mathsf{in}}^{i_2}$ are such that $H_{K_{\mathsf{in}}^{i_2}}(M_2, A_2) = Y$. Thus there are at most $C \cdot (|M_2|_n + |A_2|_n) \cdot 2^{n+k}$ tuples $(K_{\mathsf{in}}^{i_1}, K_{\mathsf{out}}^{i_1}, K_{\mathsf{in}}^{i_2}, K_{\mathsf{out}}^{i_2})$ of keys that provoke the event, and each one of them appears with probability at most $\beta/2^{2(n+k)}$ by β-AU. Thus, overall

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_{12}] \leq C \sum_{i=1}^{u} \sum_{j=1}^{q_i} \frac{q \cdot \ell_{i,j}}{2^{n+k}} \leq \frac{CqL}{2^{n+k}} .$$

Hence, we conclude that

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_1] \leq \frac{CqB}{2^n} + \frac{CqL}{2^{n+k}} \tag{7}$$

Now, for $\mathcal{B}_3$, we use a similar argument. For any one of the p Prim queries $(\texttt{prim}, K, U, V)$ and any of the q Eval queries $(\mathsf{eval}, i, M, A, N, T)$, we have

$$\Pr[K_{\mathsf{out}}^i = K \wedge \mathsf{xor}(H_{K_{\mathsf{in}}^i}(M, A), N) = U] \leq \frac{C(|M|_n + |A|_n)}{2^{n+k}} .$$

Taking a union bound over all p and q queries, yields

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_3] \leq C \cdot p \cdot \sum_{i=1}^{u} \sum_{j=1}^{q_i} \frac{\ell_{i,j}}{2^{n+k}} \leq \frac{CpL}{2^{n+k}} .$$

Finally, we turn to $\mathcal{B}_2$. We partition the set of transcripts into two subsets $\mathcal{B}_{21}$ and $\mathcal{B}_{22}$. The first subset $\mathcal{B}_{21}$ consists of the transcripts which contain two entries $(\mathsf{eval}, i, M_1, A_1, N_1, T_1)$ and $(\mathsf{eval}, i, M_2, A_2, N_2, T_2)$ such that $T_1 = T_2$. The second subset $\mathcal{B}_{22}$ consists of transcripts with two entries $(\mathsf{eval}, i_1, M_1, A_1, N_1, T_1)$ and $(\mathsf{eval}, i_2, M_2, A_2, N_2, T_2)$ such that $T_1 = T_2$, $i_1 \neq i_2$, and $K_{\mathsf{out}}^{i_1} = K_{\mathsf{out}}^{i_2}$. Then, for each new query, the probability that the output collides with one of the previously issued queries for the same user is at most $B/2^n$. Therefore, by the union bound,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_{21}] \leq \frac{qB}{2^n}; .$$

In contrast, to enter the second set, note that for each new query, there is probability at most $q/2^n$ that the output collides with one of the previous queries, and the probability that additionally the outer keys collide is at most $\beta/2^k$. Thus,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_{22}] \leq \frac{\beta q^2}{2^{n+k}} .$$

Summing up we get,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_2] \leq \frac{qB}{2^n} + \frac{\beta q^2}{2^{n+k}}$$

This concludes the proof.

E.1 Proof of Theorem 3

We merely discuss how to adapt the proof of Theorem 2 to accommodate the case that $H_{K_{\mathsf{in}}}(\varepsilon, \varepsilon) = 0^n$ for all keys K_{in}, where ε denotes the empty string. Further, this yields a proof for Theorem 3. The bad transcripts are exactly the same as in Theorem 2, the changes are the probabilities that these bad transcripts occur, specifically for the events $\mathcal{B}_{12}$ and $\mathcal{B}_3$. Note that we assume an upper bound d on the number of users re-using a particular nonce N, and this is going to be used below.

Analysis of $\mathcal{B}_{12}$. Recall that we are looking at the probability that there are two transcript entries $(\mathsf{eval}, i_1, M_1, A_1, N_1, T_1)$ and $(\mathsf{eval}, i_2, M_2, A_2, N_2, T_2)$ with $i_1 \neq i_2$, $K_{\mathsf{out}}^{i_1} = K_{\mathsf{out}}^{i_2}$, and

$$\mathsf{xor}(H_{K_{\mathsf{in}}^{i_1}}(M_1, A_1), N_1) = \mathsf{xor}(H_{K_{\mathsf{in}}^{i_2}}(M_2, A_2), N_2) \ .$$

Note that here $(M_1, A_1, N_1) = (M_2, A_2, N_2)$ is allowed. There are three sub-cases resulting in three different probability terms:

- If $(M_1, A_1) \neq (\varepsilon, \varepsilon)$, $(M_2, A_2) \neq (\varepsilon, \varepsilon)$, then we are in the same situation as in Theorem 2 above, and get an upper bound $\dfrac{CqL}{2^{n+k}}$.
- If $(M_1, A_1) = (\varepsilon, \varepsilon)$ and $(M_2, A_2) \neq (\varepsilon, \varepsilon)$, then

$$\Pr[\mathsf{xor}(H_{K_{\mathsf{in}}^{i_1}}(M_1, A_1), N_1) = \mathsf{xor}(H_{K_{\mathsf{in}}^{i_2}}(M_2, A_2), N_2) \wedge K_{\mathsf{out}}^{i_1} = K_{\mathsf{out}}^{i_2}] \leq \\ \leq \frac{C \cdot (|M_2|_n + |A_2|_n)}{2^{n+k}} \ ,$$

  where we have used the regularity of the function output on (M_2, A_2). Taking a union bound over all such pairs, this results in a term $\dfrac{CqL}{2^{n+k}}$.
- Finally, we consider the case of $(M_1, A_1) = (M_2, A_2) = (\varepsilon, \varepsilon)$. Here, the probability $\Pr[\mathsf{xor}(H_{K_{\mathsf{in}}^{i_1}}(M_1, A_1), N_1) = \mathsf{xor}(H_{K_{\mathsf{in}}^{i_2}}(M_2, A_2), N_2) \wedge K_{\mathsf{out}}^{i_1} = K_{\mathsf{out}}^{i_2}]$ is either zero if $N_1 \neq N_2$, or 2^{-k} if $N_1 = N_2$. (This follows from the injective property of xor.) Let now q_N be the number of queries $(\varepsilon, \varepsilon)$ with nonce N, and thus in particular $\sum_N q_N \leq q$, and further $q_N \leq d$. Then, the overall probability that $\mathcal{B}_{12}$ occurs due to such a pair is at most

$$\sum_N q_N^2 / 2^k \leq d \cdot \sum_N q_N / 2^k \leq \frac{dq}{2^k} \ .$$

Analysis of $\mathcal{B}_3$. As in Theorem 2, the probability that for one of the p Prim queries $(\texttt{prim}, K, U, V)$ and one of the q Eval queries $(\mathsf{eval}, i, M, A, N, T)$ with $(M, A) \neq (\varepsilon, \varepsilon)$ we have $K_{\mathsf{out}}^i = K$ and $\mathsf{xor}(H_{K_{\mathsf{in}}^i}(M, A), N) = U$ is at most $\frac{CpL}{2^{n+k}}$.

In contrast, for a nonce N, let $N' = \mathsf{xor}(0^n, N)$. Then, for every Prim query $(\texttt{prim}, K, N', V)$, there are at most d Eval queries $(\mathsf{eval}, i, \varepsilon, \varepsilon, N, T)$, and the probability that $K_{\mathsf{in}}^i = K$ for any of these is 2^{-k}. Therefore, the overall probability that the transcript is in $\mathcal{B}_3$ because of such a pair is at most $\frac{pd}{2^k}$.

F Proof of Theorem 4

We will use the H-coefficient technique. The real system $\mathbf{S}_0$ and ideal system $\mathbf{S}_1$ implement game $\mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A})$ with challenge bit 1 and 0 respectively. Assume that $\mathcal{A}$ does not repeat a prior query (except for New ones), and it does not make redundant ideal-cipher queries. Assume that if the adversary makes an encryption query (i, N, M, A) for an answer C then later it will not make a verification query (i, N, C, A). Since we consider computationally unbounded adversaries, without loss of generality, assume that the adversary is deterministic. When the adversary finishes querying, we grant it all the keys $K_1, K_2, \cdots$. This should only help the adversary. Beside the revealed keys and the information of the New queries, the transcript stores the following information:

– **Ideal-cipher queries**: for each query $E(K, X)$ with answer Y, create an entry $(\mathtt{prim}, K, X, Y, +)$. Likewise, for each query $E^{-1}(K, Y)$ with answer X, create an entry $(\mathtt{prim}, K, X, Y, -)$.
– **Encryption queries**: for each encryption query (i, N, M, A) with answer C, store an entry $(\mathsf{enc}, i, N, M, A, C)$. Additionally, in the real world, grant the adversary the table of triples $(K, X, E_K(X))$ for any query $E(K, X)$ that $\mathsf{CTR}[E].\mathsf{E}(J_i, M; T)$ makes, where J_i is the k-bit suffix of the key K_i of user i, and T is the IV of C. In the ideal world, generate the corresponding fake table as described in Section 5. The extra information in the table will only help the adversary.
– **Verification queries**: for each verification query (i, N, C, A) with answer b, store an entry $(\mathsf{vf}, i, N, C, A, b)$.

Now, from such a transcript τ, we can extract a transcript $\mathscr{R}_1(\tau)$ for GMAC^+, and another transcript $\mathscr{R}_2(\tau)$ for CTR as follows. The transcript $\mathscr{R}_1(\tau)$ consists of the revealed keys, information of the New queries, and all $\mathtt{prim}$ entries of τ, and for each entry $(\mathsf{enc}, i, N, M, A, C)$ of τ, we accordingly store an entry $(\mathsf{eval}, i, N, M, A, T)$ in $\mathscr{R}_1(\tau)$, where T is the IV of C. The transcript $\mathscr{R}_2(\tau)$ consists of the k-bit suffixes of the revealed keys, information of the New queries, and all $\mathtt{prim}$ entries of τ, and for each entry $(\mathsf{enc}, i, N, M, A, C)$ of τ, we accordingly store an entry (enc, i, M, C) in $\mathscr{R}_2(\tau)$. If (1) $\mathscr{R}_1(\tau)$ is GMAC^+-good and $\mathscr{R}_2(\tau)$ is CTR-good, and (2) τ contains no verification query of answer true, then we additionally grant the adversary the following information:

– In the real world, for each entry $(\mathsf{vf}, i, N, C, A, \mathsf{false})$, we run $\mathsf{CTR}[E].\mathsf{D}(J_i, C)$, where J_i is the k-bit suffix of K_i, and grant the adversary the decrypted message M. For each query $E(K, X)$ of answer Y that $\mathsf{CTR}[E].\mathsf{D}$ makes, if there is no entry $(\mathtt{prim}, K, X, Y, \cdot)$ or no triple (K, X, Y) in all tables then we grant the adversary an entry (dec, K, X, Y).
– In the ideal world, create a blockcipher $\tilde{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ as follows. For each $K \in \{0,1\}^k$, sample $\tilde{E}(K, \cdot)$ uniformly random from $\mathrm{Perm}(n)$, subject to the constraint that (i) for any entry $(\mathtt{prim}, K, X, Y, \cdot)$ in $\mathscr{R}_2(\tau)$, we must have $\tilde{E}(K, X) = Y$, and (ii) for any triple (K, X', Y') in the tables of $\mathscr{R}_2(\tau)$, we must have $\tilde{E}(K, X') = Y'$. This blockcipher can be generated, because $\mathscr{R}_2(\tau)$ is CTR-good. For each entry $(\mathsf{vf}, i, N, C, A, \mathsf{false})$, we run $\mathsf{CTR}[\tilde{E}].\mathsf{D}(J_i, C)$, where J_i is the k-bit suffix of K_i and grant the adversary the decrypted message M, and the entries (dec, K, X, Y) as above.

Defining bad transcripts. A transcript τ is *bad* if one of the following happens:

1. The GMAC^+-transcript $\mathscr{R}_1(\tau)$ of τ is GMAC^+-bad.
2. The CTR-transcript $\mathscr{R}_2(\tau)$ of τ is CTR-bad.
3. There is a table in $\mathscr{R}_2(\tau)$ that contains a triple (K, X, Y), and there is an entry $(\mathsf{eval}, i, N, M, A, T)$ in $\mathscr{R}_1(\tau)$ such that $K = K_{\mathsf{out}}$ and $Y = T$, where $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ is the key K_i of user i.
4. There is an entry (dec, K, X, Y) in τ and an entry $(\mathsf{eval}, i, N, M, A, T)$ in $\mathscr{R}_1(\tau)$ such that $K = K_{\mathsf{out}}$ and $Y = T$, where $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ is the key K_i of user i.
5. There is an entry $(\mathsf{vf}, i, N, C, A, \mathsf{false})$ in τ and an entry $(\mathsf{eval}, j, N', M', A', T)$ in $\mathscr{R}_1(\tau)$ such that T is the IV of C, $K_{\mathsf{out}} = K'_{\mathsf{out}}$, and $\mathsf{xor}(H(K_{\mathsf{in}}, M, A), N) = \mathsf{xor}(H(K'_{\mathsf{in}}, M', A'), N')$, where $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ and $K'_{\mathsf{in}} \parallel K'_{\mathsf{out}}$ are the keys K_i and K_j of users i and j respectively, and M is the decrypted message associated with the vf entry above.
6. There are entries $(\mathtt{prim}, K, X, Y)$ and $(\mathsf{vf}, i, N, C, A, \mathsf{false})$ in τ such that $K = K_{\mathsf{out}}, Y = T$, and $\mathsf{xor}(H(K_{\mathsf{in}}, M, A), N) = X$, where $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ is the key K_i of user i, and T is the IV of C, and M is the decrypted message associated with the vf entry above.

If a transcript is not bad then we say that it is *good*. Below, let ϵ_1 be the number that the GMAC^+ proof uses to upper bound the probability of bad transcripts, for any adversary $\overline{\mathcal{A}}$ that makes at most q evaluation queries whose total block length is at most L, at most B-block queries per user, and p ideal-cipher queries, and for any β-pairwise AU key-generation algorithm. Applying Theorem 2 with $\lambda = 2$, and note that $q \le L$,

$$\epsilon_1 \le \frac{(1 + 2\beta c)LB}{2^n} + \frac{2\beta c L p + (2\beta c + \beta)L^2}{2^{n+k}} \tag{8}$$

Define ϵ_2 for CTR similarly, for a $\frac{\beta}{2^k}$-smooth key-generation algorithm, where the notion of smoothness can be found in Section 4.1. Applying Theorem 1 with $\alpha = \beta/2^k$ and $h = n - 1$, and note that $q \le L$,

$$\epsilon_2 \le \frac{1}{2^{n/2}} + \frac{3LB}{2^n} + \frac{3\beta L^2}{2^{n+k}} + \frac{\beta a p}{2^k} \tag{9}$$

Probability of bad transcripts. Let $\mathcal{X}_1$ be the random variable for the transcript in the ideal system. Let $\mathcal{B}_j$ denote the set of transcripts that violates the jth constraint in badness. For the first constraint of badness, consider the following adversary $\overline{\mathcal{A}}$ attacking the mu-prf security of $\mathsf{GMAC}^+[H, E]$, with respect to the key-generation algorithm KeyGen. It runs $\mathcal{A}$ and uses its New oracle to respond to the latter's queries of the same type. For each encryption query (i, N, M, A) of $\mathcal{A}$, adversary $\overline{\mathcal{A}}$ queries $\textsc{Eval}(i, N, M, A)$ to get an answer T, generates a ciphertext core C' of appropriate length, and then returns $T \parallel C'$ to $\mathcal{A}$. For each verification query of $\mathcal{A}$, adversary $\overline{\mathcal{A}}$ simply returns false. When $\mathcal{A}$ finishes querying and asks for the keys, $\overline{\mathcal{A}}$ also finishes querying and gives $\mathcal{A}$ what it receives. Then the transcript of $\overline{\mathcal{A}}$ in the ideal world has the same distribution as $\mathscr{R}_1(\mathcal{X}_1)$. Since adversary $\overline{\mathcal{A}}$ uses at most q evaluation queries whose total block length is at most L, at most B-block queries per user, and p ideal-cipher queries,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_1] \le \epsilon_1 \ .$$

Next, for the second constraint of badness, let $\mathsf{KeyGen}[k]$ be the key-generation algorithm such that, on input $(\mathsf{st}, \mathsf{aux})$, runs $(K, \mathsf{st}') \leftarrow \mathsf{KeyGen}(\mathsf{st}, \mathsf{aux})$, and then outputs (J, st'), where J is the k-bit suffix of K. Then $\mathsf{KeyGen}[k]$ is $\frac{\beta}{2^k}$-smooth. Consider the following adversary $\mathcal{A}^*$ attacking the mu-ind security of SE, with respect to the key-generation algorithm $\mathsf{KeyGen}[k]$. It runs $\mathcal{A}$ and uses its oracle New and Enc to respond to the latter's queries of the same type. For each verification query of $\mathcal{A}$, adversary $\mathcal{A}^*$ simply returns false. When $\mathcal{A}$ finishes querying and asks for the keys, then $\mathcal{A}^*$ also finishes querying and gives $\mathcal{A}$ the keys that it receives. Then the transcript of $\mathcal{A}^*$ in the ideal world has the same distribution as $\mathscr{R}_2(\mathcal{X}_1)$. Since adversary $\mathcal{A}^*$ uses at most q encryption queries whose total block length is at most L, at most B-block queries per user, and p ideal-cipher queries,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_2] \le \epsilon_2 \ .$$

For the third constraint of badness, consider a sequence of encryption queries $(i_1, M_1, N_1, A_1), \ldots, (i_q, M_q, N_q, A_q)$ with answers $C_1, \ldots, C_q$ respectively. Fix $1 \le r, s \le q$. Let $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ and $K'_{\mathsf{in}} \parallel K'_{\mathsf{out}}$ be the keys of users i_r and i_s respectively. Consider the table generated by the r-th encryption query, and the eval entry generated by the s-th encryption query. Recall that this table is generated by (1) padding C_r with random bits to have full block length, and padding M_r to have full block length, (2) parsing $\mathsf{IV} \parallel C_{r,1} \parallel \cdots \parallel C_{r,m} \leftarrow C_r$, and $M_{r,1} \parallel \cdots \parallel M_{r,m} \leftarrow M_r$, with $|C_{r,\ell}| = |M_{r,\ell}| = n$, and (3) producing $(K_{\mathsf{out}}, X_1, C_{r,1} \oplus M_{r,1}), \ldots, (K_{\mathsf{out}}, X_m, C_{r,m} \oplus M_{r,m})$, with $X_\ell \leftarrow \mathsf{add}(\mathsf{IV}, \ell - 1)$. We now compute the probability that $K_{\mathsf{out}} = K'_{\mathsf{out}}$ and $C_{r,\ell} \oplus M_{r,\ell} = T$. Consider the following cases.

Case 1: $r \ge s$. Hence $C_{r,\ell}$ is picked at random, independent of $M_{r,\ell}$ and T. Since KeyGen is β-pairwise AU, the chance that $K_{\mathsf{out}} = K'_{\mathsf{out}}$ and $C_{r,\ell} \oplus M_{r,\ell} = T$ is at most $\beta/2^{k+n}$ if $i_r \ne i_s$, and at most 2^{-n} if $i_r = i_s$.

Case 2: $r < s$. Then T is picked at random, independent of $M_{r,\ell}$ and $C_{r,\ell}$. Since KeyGen is β-pairwise AU, the chance that $K_{\mathsf{out}} = K'_{\mathsf{out}}$ and $C_{r,\ell} \oplus M_{r,\ell} = T$ is at most $\beta/2^{k+n}$ if $i_r \ne i_s$, and at most 2^{-n} if $i_r = i_s$.

Thus in any case, the chance that $K_{\mathsf{out}} = K'_{\mathsf{out}}$ and $C_{r,\ell} \oplus M_{r,\ell} = T$ is at most $\beta/2^{k+n}$ if $i_r \ne i_s$, and at most 2^{-n} if $i_r = i_s$. Since the total number of triples in all tables is at most L, and there are at most B eval entries created due to encryption queries for user i_s, and note that $q \le L/2$ (as each encryption/verification query consists of at least two blocks, one due to the associated data, and another due to the message/ciphertext),

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_3] \le \sum_{1 \le s \le q} \frac{\beta L}{2^{k+n}} + \frac{\beta B}{2^n} = \frac{\beta L q}{2^{k+n}} + \frac{\beta q B}{2^n} \le \frac{\beta L^2}{2^{k+n}} + \frac{0.5 \beta L B}{2^n} \ .$$

For the fourth constraint of badness, fix an entry (dec, K, X, Y) created by decrypting a verification query of user j. Consider an entry $(\mathsf{eval}, i, N, M, A, T)$.

Let K_{out} be the k-bit suffix of the key K_i of user i, and note that K is the k-bit suffix of the key K_j of user j. There are two cases:

Case 1: The verification query above is made before the the encryption query corresponding to the eval entry. Then T is a random string, independent of Y. If $i = j$ then $K = K_{\mathsf{out}}$, and the chance that $Y = T$ is 2^{-n}. If $i \neq j$ then since KeyGen is β-pairwise AU, the chance that $K = K_{\mathsf{out}}$ and $Y = T$ is at most $\beta/2^{k+n}$.

Case 2: The verification query above is made after the the encryption query corresponding to the eval entry. Since there are at most L dec entries and at most L triples in the tables, given T, there are still at least $2^n - 2L - p \geq 2^{n-1}$ equally likely choices of Y. Hence if $i = j$ then $K = K_{\mathsf{out}}$, and the chance that $Y = T$ is at most $2/2^n$. On the other hand, since KeyGen is β-pairwise AU, if $i \neq j$ then the chance that $K = K_{\mathsf{out}}$ and $Y = T$ is at most $2\beta/2^{k+n}$.

Thus in both cases, the chance that $K = K_{\mathsf{out}}$ and $Y = T$ is at most $2\beta/2^{k+n}$ if $i \neq j$, and at most $2/2^n$ if $i = j$. Summing this over at most q eval entries and at most L dec entries, and note that there are at most B eval entries per user,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_4] \leq \frac{2LB}{2^n} + \frac{2\beta Lq}{2^{k+n}} \leq \frac{2LB}{2^n} + \frac{2\beta L^2}{2^{k+n}} \enspace .$$

For the fifth constraint of badness, consider an entry $(\mathsf{vf}, i, N, C, A, \mathsf{false})$ in $\mathcal{X}_1$, and let T be the IV of C and M be the associated decrypted message. Note that if $\mathcal{X}_1 \in \mathcal{B}_5$ then $\mathscr{R}_1(\mathcal{X}_1)$ is GMAC^+-good and $\mathscr{R}_2(\mathcal{X}_1)$ is CTR-good. Fix $j \leq q$. There is at most one entry $(\mathsf{eval}, j, N', M', A', T)$ in $\mathscr{R}_1(\tau)$; otherwise $\mathscr{R}_1(\mathcal{X}_1)$ is not good, and thus $\mathcal{X}_1 \notin \mathcal{B}_5$. Let $K_i = K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ and $K_j = K'_{\mathsf{in}} \parallel K'_{\mathsf{out}}$. If $j \neq i$ then the probability that $K'_{\mathsf{out}} = K_{\mathsf{out}}$ and $\mathsf{xor}(H(K_{\mathsf{in}}, M, A), N) = \mathsf{xor}(H(K'_{\mathsf{in}}, M', A'), N')$ is at most $\frac{2c\beta \cdot \mathbf{E}[|M|_n + |A|_n]}{2^{n+k}}$, because H is c-regular, xor is 2-regular, and KeyGen is β-pairwise AU. If $i = j$ then $K_{\mathsf{in}} \parallel K_{\mathsf{out}} = K'_{\mathsf{in}} \parallel K'_{\mathsf{out}}$, and we consider three following cases.

Case 1: $(M, N, A) = (M', N', A')$. Let C' be the answer of $\textsc{Enc}(j, N', M', A')$ as indicated in $\mathcal{X}_1$. For the blockcipher $\tilde{E}$ above, since $C' = \mathsf{CTR}[E].\mathsf{E}(K_{\mathsf{out}}, M'; T)$, we also have $C' = \mathsf{CTR}[\tilde{E}].\mathsf{E}(K_{\mathsf{out}}, M'; T)$, due to the consistency between E and $\tilde{E}$. On the other hand, recall that M is generated by running $\mathsf{SE.D}^{\tilde{E}}(K_{\mathsf{out}}, C)$. Since $M = M'$ and C and C' share the same IV, we must have $C = C'$. This means that the adversary queries $\textsc{Vf}(i, N, C, A)$ first, and then later queries $\textsc{Enc}(i, N', M', A')$ and accidentally gets the same answer C. This case happens with probability at most $2^{-|C|} \leq 2^{-n}$.

Case 2: $(M, A) = (M', A')$, but $N \neq N'$. Due to the injectivity of xor, this case cannot happen.

Case 3: $(M, A) \neq (M', A')$. Since KeyGen is β-pairwise AU, H is c-AXU and xor is 2-regular and linear and injective,

$$\begin{aligned} &\Pr[\mathsf{xor}(H(K_{\mathsf{in}}, M, A), N) = \mathsf{xor}(H(K_{\mathsf{in}}, M', A'), N')] \\ &\leq \frac{2c\beta \cdot \mathbf{E}\big[|M|_n + |A|_n + |M'|_n + |A'|_n\big]}{2^n} \leq \frac{2c\beta \cdot \big(B + \mathbf{E}\big[|M|_n + |A|_n\big]\big)}{2^n} \enspace . \end{aligned}$$

As the three cases above are mutually exclusive, if $i = j$ then the chance that $\mathsf{xor}(H(K_{\mathsf{in}}, M, A), N) = \mathsf{xor}(H(K_{\mathsf{in}}, M', A'), N')$ is at most $\frac{2c\beta\cdot\left(B+\mathbf{E}\left[|M|_n+|A|_n\right]\right)}{2^n}$. Sum over all $j \le q$, and then over all q vf entries, and note that $B \ge 2$ and $q \le L/2$ (as each encryption/verification query consists of at least two blocks, one due to the associated data, and another due to the message/ciphertext),

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_5] \le \frac{2c\beta qL}{2^{n+k}} + \frac{2c\beta(L+qB)}{2^n} \le \frac{c\beta L^2}{2^{n+k}} + \frac{2c\beta LB}{2^n} \enspace .$$

Finally, for the last constraint, consider an entry $(\mathsf{vf}, i, N, C, A, \mathsf{false})$ and let M be the decrypted message associated with this entry. Let $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ be the key of user i, and let T be the IV of C. Consider one entry $(\mathtt{prim}, K, X, T, \cdot)$. Since KeyGen is β-pairwise AU, H is c-regular, and xor is 2-regular, the chance that $K = K_{\mathsf{out}}$ and $X = \mathsf{xor}(H(K_{\mathsf{in}}, M, A), N)$ is at most $\frac{2\beta c\cdot\mathbf{E}[|M|_n+|A|_n]}{2^{n+k}}$. Sum that over all vf entries and p $\mathtt{prim}$ entries,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_6] \le \frac{2c\beta Lp}{2^{n+k}} \enspace .$$

Summing up,

$$\begin{aligned}
\Pr[\mathcal{X}_1 \text{ is bad}] &\le \sum_{j=1}^{6} \Pr[\mathcal{X}_1 \in \mathcal{B}_j] \\
&\le \epsilon_1 + \epsilon_2 + \frac{(2c\beta + 0.5\beta + 2)LB}{2^n} + \frac{\beta(c+3)L^2 + 2c\beta Lp}{2^{n+k}} \\
&\le \frac{1}{2^{n/2}} + \frac{\beta ap}{2^k} + \frac{(3c\beta + 7\beta)L^2 + 4\beta cLp}{2^{n+k}} + \frac{(4c\beta + 0.5\beta + 6)LB}{2^n} \enspace .
\end{aligned}$$

Bounding transcript ratio. Fix a good transcript τ such that $\mathsf{p}_{\mathsf{S}_1}(\tau) > 0$. In particular, this means that there is no vf of answer true. Create the multisets $S_1, \ldots, S_5$ as follows.

- For each entry $(\mathtt{prim}, K, X, Y, \cdot)$ in τ, add a triple (K, X, Y) to S_1.
- For each triple (K, X, Y) in tables of τ, add it to S_2.
- For each entry (dec, K, X, Y), if $(K, X, Y) \notin S_3$ then add (K, X, Y) to S_3.
- For each entry $(\mathsf{eval}, i, N, M, A, T)$ in $\mathscr{R}_1(\tau)$, add (K_{out}, X, T) to S_4, where $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ is the key of user i in τ, and $X = \mathsf{xor}(H(K_{\mathsf{in}}, M, A), N)$.
- For each entry $(\mathsf{vf}, i, N, C, A, \mathsf{false})$ in τ, if $(K_{\mathsf{out}}, X, T) \notin S_5$ then add this triple to S_5, where T is the IV of C, $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ is the key of user i in τ, M is the decrypted message associated with this entry indicated by τ, and $X = \mathsf{xor}(H(K_{\mathsf{in}}, M, A), N)$.

Due to (1) the goodness of τ, (2) the fact that add can only produce outputs starting with 1 but xor produces output starting with 0, and (3) the way we generate dec entries,

- For each $j \le 5$, the multiset S_j contains no item twice, meaning that it is actually a set.

– The sets $S_1, \cdots, S_5$ are pairwise disjoint.
– There are no triples (K, X, Y) and (K, X', Y') in $S_1 \cup S_2 \cup S_3 \cup S_4$ such that $X = X'$ or $Y = Y'$.

Now, the probability $\mathsf{p_{S_0}}(\tau)$ is the chance that all the following events happen:

– Samp: If we query NEW using the queries as indicated in τ, the generated keys will be the values indicated by τ.
– Real_j, for $1 \le j \le 4$: For each $(K, X, Y) \in S_j$, querying $E_K(X)$ returns Y.
– Real_5: For each $(K, X, Y) \in S_5$, querying $E_K(X)$ does *not* return Y.

On the other hand, the probability $\mathsf{p_{S_1}}(\tau)$ is the chance Samp and Real_1 and the following events happen:

– Ideal_1: For the padding version of CTR, let $C_1, \ldots, C_q$ be the ciphertexts indicated by τ. For the padding-free version of CTR, let $C_1, \ldots, C_q$ be the *pre-truncated* ciphertexts indicated by τ.[10] Then, if we sample q random strings of length $|C_1|, \cdots, |C_q|$ respectively, then we get $C_1, \ldots, C_q$ respectively. Note that $|C_1| + \cdots + |C_q| = n(|S_2| + |S_4|)$.
– Ideal_2: Create a blockcipher $\tilde{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ as follows: for every $K \in \{0,1\}^k$, sample $\tilde{E}(K, \cdot) \leftarrow_{\$} \mathrm{Perm}(n)$, subject to the constraint that for every $(K, X, Y) \in S_1 \cup S_2$, we have $\tilde{E}(K) = Y$. Now, for every $(K', X', Y') \in S_3$, if we query $E(K', X')$ then we get Y'.

For each $2 \le j \le 5$, let P_j denote $\Pr[\mathsf{Real}_j \mid \mathsf{Real}_1 \cap \cdots \mathsf{Real}_{j-1}]$. As KeyGen does not use E, event Samp is independent of other events, and thus

$$\frac{\mathsf{p_{S_0}}(\tau)}{\mathsf{p_{S_1}}(\tau)} = \frac{\Pr[\mathsf{Real}_5 \cap \cdots \cap \mathsf{Real}_1]}{\Pr[\mathsf{Ideal}_1 \cap \mathsf{Ideal}_2 \cap \mathsf{Real}_1]} = \frac{P_2 \cdot P_3 \cdot P_4 \cdot P_5}{\Pr[\mathsf{Ideal}_1 \cap \mathsf{Ideal}_2 \mid \mathsf{Real}_1]} \ .$$

In the last ratio, since Ideal_1 is independent of other events, the denominator can be factored to $\Pr[\mathsf{Ideal}_1] \cdot \Pr[\mathsf{Ideal}_2 \mid \mathsf{Real}_1]$. Moreover, note that $\Pr[\mathsf{Ideal}_2 \mid \mathsf{Real}_1] = \Pr[\mathsf{Real}_3 \mid \mathsf{Real}_1 \cap \mathsf{Real}_2] = P_3$. Hence

$$\frac{\mathsf{p_{S_0}}(\tau)}{\mathsf{p_{S_1}}(\tau)} = \frac{P_2 \cdot P_4 \cdot P_5}{\Pr[\mathsf{Ideal}_1]} \ .$$

For each $K \in \{0,1\}^k$, let $Z_1(K), Z_2(K), Z_3(K), Z_4(K)$ denote the number of triples (K, X, Y) in $S_1, S_2, S_1 \cup S_2 \cup S_3, S_4$ respectively. Then

$$\begin{aligned} P_2 \cdot P_4 &= \prod_{K \in \{0,1\}^k} \prod_{i = Z_1(K)}^{Z_1(K)+Z_2(K)-1} \frac{1}{2^n - i} \prod_{j = Z_3(K)}^{j = Z_3(K)+Z_4(K)-1} \frac{1}{2^n - j} \\ &\ge \prod_{K \in \{0,1\}^k} 2^{-n \cdot (Z_2(K) + Z_4(K))} = 2^{-n(|S_2| + |S_4|)} = \Pr[\mathsf{Ideal}_1] \ . \end{aligned}$$

[10] Given a table $\mathcal{T}$ and a message M, the pre-truncated ciphertext can be obtained as follows. Suppose that $\mathcal{T}$ contains $(K, X_1, Y_1), \ldots, (K, X_m, Y_m)$. Then the pre-truncated ciphertext is $(Y_1 \parallel \cdots \parallel Y_m) \oplus M'$, where M' is obtained by padding 0's to M to have full block length.

Thus

$$\frac{\mathsf{ps}_0(\tau)}{\mathsf{ps}_1(\tau)} \geq P_5 \enspace .$$

We now give a lower bound for P_5. Note that $|S_1 \cup \cdots \cup S_4| \leq p + L + q \leq 2^{n-1}$, because (i) there are p ideal-cipher queries in τ, contributing p triples in S_1, (ii) each encryption query (i, N, M, A) contributes one triple in S_4, and at most $(|M|_n + |A|_n)$ triples in S_2, and (iii) each verification query (i, N, C, A) contributes at most $(|C|_n + |A|_n)$ triples in S_3. Now, for each $(K, X, Y) \in S_5$, there are only two cases.

Case 1: There is a triple $(K, X', Y') \in S_1 \cup \cdots \cup S_4$ such that either (i) $X' = X$ but $Y' \neq Y$, or (ii) $Y' = Y$ but $X' \neq X$. In this case, given that E is consistent $S_1 \cup \cdots \cup S_4$, if we query $E_K(X)$ then the answer will not be Y.

Case 2: There is no triple $(K, X', Y') \in S_1 \cup \cdots \cup S_4$ such that either $X = X'$ or $Y = Y'$. Hence, conditioning that E is consistent with $S_1 \cup \cdots \cup S_4$, since there are at least $2^n - |S_1 \cup \cdots \cup S_4| \geq 2^{n-1}$ equally likely choices for $E_K(X)$, the conditional probability that $E(K, X) = Y$ is at most $2/2^n$.

Hence in both case, conditioning that E is consistent with $S_1 \cup \cdots \cup S_4$, if we query $E_K(X)$ then the conditional probability that we get Y is at most $2/2^n$. By union bound, $P_5 \geq 1 - |S_5| \cdot 2/2^n \geq 1 - 2q/2^n$. Hence

$$\frac{\mathsf{ps}_0(\tau)}{\mathsf{ps}_1(\tau)} \geq 1 - \frac{2q}{2^n} \geq 1 - \frac{0.5LB}{2^n} \enspace .$$

F.1 Proof of Theorem 5

We now discuss how to adapt the proof of Theorem 4 to deal with a weakly regular hash H. The definition of bad transcripts is exactly the same, and so is the bound on the transcript ratio; the changes are the probabilities that bad transcripts occur, specifically for events $\mathcal{B}_1, \mathcal{B}_5$, and $\mathcal{B}_6$. Note that we assume an upper bound d on the number of users re-using a particular nonce N, and this is going to be used below. Let $\mathcal{X}_1$ is the random variable for the transcript in the ideal system.

Analysis of $\mathcal{B}_1$. Let ϵ_1 be the value that the GMAC^+ proof uses to upper-bound the probability of bad transcripts, for any adversary $\overline{\mathcal{A}}$ that makes at most q evaluation queries whose total block length is at most L, at most B-block queries per user, and p ideal-cipher queries, and for any β-pairwise AU key-generation algorithm, assuming that each nonce is reused across at most d users. As in the proof of Theorem 4,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_1] \leq \epsilon_1 \enspace .$$

The only change here is that now we need to use Theorem 3 (instead of Theorem 2) to obtain ϵ_1. In particular, applying Theorem 3 with $\lambda = 2$ and note that $q \leq L/2$,

$$\epsilon_1 \leq \frac{(1 + 2\beta c)LB}{2^n} + \frac{2\beta c L p + (2\beta c + \beta)L^2}{2^{n+k}} + \frac{d(p + L)}{2^k} \enspace .$$

Analysis of $\mathcal{B}_5$. First, consider the case that $\mathcal{X}_1$ falls into $\mathcal{B}_5$ due to some entries $(\mathsf{vf}, i, N, C, A, \mathsf{false})$ and $(\mathsf{eval}, j, N', M', A', T)$ such that either (1) $(M, A) \neq (\varepsilon, \varepsilon)$ or (2) $(M', A') \neq (\varepsilon, \varepsilon)$ or (3) $(M, A) = (M', A')$ and $i = j$, where M is the decrypted message of the verification entry. As in the proof of Theorem 4, this case happens with probability at most $\frac{c\beta L^2}{2^{n+k}} + \frac{2c\beta LB}{2^n}$.

Next consider an entry $(\mathsf{vf}, i, N, C, A, \mathsf{false})$ such that both decrypted message M and associated data A are empty. Consider an entry $(\mathsf{eval}, j, N', M', A', T)$ such that $(M', A') = (\varepsilon, \varepsilon)$, $j \neq i$, and T is the IV of C. Let $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ and $K'_{\mathsf{in}} \parallel K'_{\mathsf{out}}$ be the keys of users i and j respectively. Since H is weakly regular, $H(K_{\mathsf{in}}, M, A) = H(K'_{\mathsf{in}}, M', A') = 0^n$. For these pair of entries to cause $\mathcal{X}_1$ to fall into $\mathcal{B}_5$, we must have $\mathsf{xor}(0^n, N) = \mathsf{xor}(0^n, N')$, meaning that $N = N'$, due to the injectivity of xor. Since the nonce N is used across at most d users, there are at most d choices for the index j. On the other hand, the chance that $K_{\mathsf{out}} = K'_{\mathsf{out}}$ is at most 2^{-k}. Summing this over d choices of j, and over q verification queries, we obtain a bound $qd/2^k \leq Ld/2^k$. Hence

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_5] \leq \frac{c\beta L^2}{2^{n+k}} + \frac{2c\beta LB}{2^n} + \frac{Ld}{2^k} \ .$$

Analysis of $\mathcal{B}_6$. First consider the case that some verification entry, in which either the decrypted message or the associated data is non-empty, causes $\mathcal{X}_1$ to fall into $\mathcal{B}_6$. As in the proof of Theorem 4, one can bound the chance that this case happens by $\frac{2c\beta Lp}{2^{n+k}}$. Next, consider an entry $(\mathsf{vf}, i, N, C, A, \mathsf{false})$, in which both the decrypted message M and the associated data A are the empty string. For each entry $(\mathtt{prim}, K, X, Y, +)$, view it as throwing a ball into bin Y. Likewise, for each entry $(\mathtt{prim}, K, X, Y, -)$, view it as throwing a ball into bin X. Thus there are at most $p \leq 2^{(1-\epsilon)n-1}$ throws. For each j-th throw, given the result of the prior throws, the conditional probability that the j-th ball lands into any particular bin is at most 2^{1-n}. From Lemma 10, with probability at least $1 - 2^{-n/2}$, each bin contains at most a balls.

Let T be the IV of C and let $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ be the key of user i. Since H is weakly regular, $H(K_{\mathsf{in}}, M, A) = 0^n$. From the balls-into-bins result above, there are at most a balls in bin T, and also at most a balls in bin $\mathsf{xor}(0^n, N)$. Thus there are at most $2a$ entries $(\mathtt{prim}, K, \mathsf{xor}(0^n, N), T, \cdot)$. For each such entry, the chance that $K = K_{\mathsf{out}}$ is at most 2^{-k}. Hence the chance that the verification entry above causes $\mathcal{X}_1$ to fall into $\mathcal{B}_6$ is at most $2a/2^k$. Summing this across at most q verification queries, we obtain a bound $2aq/2^k \leq aL/2^k$. Hence

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_6] \leq \frac{2c\beta Lp}{2^{n+k}} + \frac{aL}{2^k} \ .$$

G Proof of Lemma 3

Let $r = k/n \in \{1, 2\}$. Suppose that $R_0, \ldots, R_5$ are sampled uniformly without replacement from a set S of size at least $\frac{15}{16} \cdot 2^n$. Pick an arbitrary string $K \in$

$\{0,1\}^{n+k}$. Since $\mathsf{KD}_1.\mathsf{Map}$ outputs $(R_0 \parallel R_1 \parallel R_2)[1:n+k]$, the chance that $\mathsf{KD}_1.\mathsf{Map}(R_0,\ldots,R_5) = K$ is at most

$$\frac{1}{(\frac{15}{16}\cdot 2^n - 2)^{r+1}} \le \frac{1}{(\frac{7}{8}\cdot 2^n)^{r+1}} \le \frac{1}{(7/8)^3 \cdot 2^{n(r+1)}} \le \frac{2}{2^{k+n}} \enspace .$$

On the other hand, the chance that $\mathsf{KD}_0.\mathsf{Map}(R_0,\ldots,R_5) = K$ is at most

$$\frac{1}{(\lfloor(\frac{15}{16}\cdot 2^n - 5)/2^{n/2}\rfloor)^{2(r+1)}} \le \frac{1}{(\frac{29}{32}\cdot 2^{n/2})^{2(r+1)}} \le \frac{1}{(29/32)^6 \cdot 2^{n(r+1)}} \le \frac{2}{2^{k+n}} \enspace .$$

This concludes the proof.

H Proof of Proposition 2

We will first construct adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ such that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\mathsf{AE},E}(\overline{\mathcal{A}}_1) &\le \mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A}_1), \text{ and}\\ \mathsf{Adv}^{\mathsf{mu\text{-}auth}}_{\mathsf{AE},E}(\overline{\mathcal{A}}_2) &\le 2\,\mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A}_2) \enspace .\end{aligned}$$

If we can do that, one can construct $\mathcal{A}$ as follows. It picks a number $a \leftarrow\!\!{\scriptstyle\$}\; \{0,1,2\}$. If $a=0$ then it runs $\mathcal{A}_1$, uses its oracles to answer the latter's queries accordingly, and outputs the same bit that $\mathcal{A}_1$ outputs. If $a \in \{1,2\}$ then it runs $\mathcal{A}_2$, uses its oracles to answer the the latter's queries accordingly, and outputs the same bit that $\mathcal{A}_2$ outputs. Then

$$\begin{aligned}\mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A}) &= \frac{1}{3}\,\mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A}_1) + \frac{2}{3}\,\mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A}_2)\\ &\ge \frac{1}{3}\,\mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\mathsf{AE},E}(\overline{\mathcal{A}}_1) + \frac{1}{3}\,\mathsf{Adv}^{\mathsf{mu\text{-}auth}}_{\mathsf{AE},E}(\overline{\mathcal{A}}_2) \enspace .\end{aligned}$$

We now construct $\mathcal{A}_1$. Without loss of generality, assume that $\overline{\mathcal{A}}_1$ does not repeat a prior query, and assumes that for each encryption query (i,N,M,A), it must call $\textsc{New}(\cdot)$ at least i times before, so that user i was initialized. Adversary $\mathcal{A}_1$ initializes a counter $v \leftarrow 0$ and a map $V = \bot$, and then runs $\overline{\mathcal{A}}_1$. For each encryption query (i,N,M,A) of $\overline{\mathcal{A}}_1$, if $V[i,N] = \bot$ then $\mathcal{A}_1$ calls $\textsc{New}(\mathsf{aux})$ with $\mathsf{aux} = (i,N)$, updates $V[i,N] \leftarrow v+1$, and increments v. It returns $\textsc{Enc}(j,N,M,A)$ to $\overline{\mathcal{A}}_1$, with $j \leftarrow V[i,N]$. Finally, when $\overline{\mathcal{A}}_1$ outputs a bit then $\mathcal{A}_1$ outputs the same bit. Then

$$\mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\mathsf{AE},E}(\overline{\mathcal{A}}_1) \le \mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A}_1) \enspace .$$

Next, we construct $\mathcal{A}_2$ as follows. Without loss of generality, assume that $\overline{\mathcal{A}}_2$ does not repeat a prior query, and assumes that for each encryption/verification query $(i,N,\cdot,A)$, it must call $\textsc{New}(\cdot)$ at least i times before, so that user i was initialized. Adversary $\mathcal{A}_2$ initializes a counter $v \leftarrow 0$ and a map $V = \bot$, and then runs $\overline{\mathcal{A}}_2$. For each encryption/verification query (i, N, X, A) of $\overline{\mathcal{A}}_2$, if $V[i, N] = \bot$ then $\mathcal{A}_2$ calls $\textsc{New}(\mathsf{aux})$ with $\mathsf{aux} = (i, N)$, updates $V[i, N] \leftarrow v+1$, and increments v. If this is an encryption query then it returns $\textsc{Enc}(j, N, X, A)$ to $\overline{\mathcal{A}}_2$ with $j \leftarrow V[i, N]$. Otherwise it calls $\textsc{Vf}(j, N, X, A)$, with $j \leftarrow V[i, N]$. Finally, $\mathcal{A}_2$ will output 1 if and only if some verification query returns true. Let c be the challenge bit of game $\mathbf{G}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_2)$. Then

$$\Pr[\mathbf{G}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A}_2) \mid c = 1] = \Pr[\mathbf{G}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\overline{\mathcal{A}}_2)] \enspace .$$

On the other hand, if $c = 0$ then $\overline{\mathcal{A}}_2$ always receives false for any verification query. Thus

$$\Pr[\mathbf{G}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A}_2) \mid c = 0] = \frac{1}{2} \enspace .$$

Summing up,

$$\mathsf{Adv}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\mathcal{A}_2) = \frac{1}{2}\,\mathsf{Adv}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\overline{\mathcal{A}}_2)$$

as claimed.

I Proof of Lemma 4

For two outputs K and K' generated by KeyGen, by symmetry, there are only four cases.

Case 1: K and K' are independent, random strings. For any two strings $(J, J') \in (\{0,1\}^{k+n})^2$, the chance that $(K, K') = (J, J')$ is $1/2^{2(k+n)}$.

Case 2: $K = \mathsf{KD}[k](\pi_i, N)$ for some $\pi_i \leftarrow\!\!\$\ \mathrm{Perm}(n)$, and $K' \leftarrow\!\!\$\ \{0,1\}^{k+n}$. For any two strings $(J, J') \in (\{0,1\}^{k+n})^2$, since $\mathsf{KD}[E]$ is 2-unpredictable, the chance that $(K, K') = (J, J')$ is at most

$$\frac{2}{2^{k+n}} \cdot \frac{1}{2^{k+n}} = \frac{2}{2^{2(k+n)}} \enspace .$$

Case 3: $K = \mathsf{KD}[k](\pi_i, N)$ for some $\pi_i \leftarrow\!\!\$\ \mathrm{Perm}(n)$, and $K' \leftarrow\!\!\$\ \mathsf{KD}[k](\pi_i, N')$, with $N \neq N'$. For any two strings $(J, J') \in (\{0,1\}^{k+n})^2$, since $\mathsf{KD}[E]$ is 2-unpredictable, the chance that $K = J$ is at most $2/2^{n+k}$. For $s \in \{0, \ldots, 5\}$, let $R_s \leftarrow \mathsf{pad}(N, s)$ and $R'_s \leftarrow \mathsf{pad}(N', s)$. Given $(R_0, \pi_i(R_0)), \ldots, (R_5, \pi_i(R_5))$, the values of $\pi_i(R'_0), \ldots, \pi_i(R'_5)$ are sampled uniformly without replacement from a set of at least $2^n - 6 \geq \frac{15}{16} \cdot 2^n$. Since $\mathsf{KD}[E]$ is 2-unpredictable, given that $K = J$, the conditional probability that $K' = J'$ is at most $2/2^{k+n}$. Hence the chance that $K = J$ and $K' = J'$ is at most

$$\frac{2}{2^{k+n}} \cdot \frac{2}{2^{k+n}} = \frac{4}{2^{2(k+n)}} \enspace .$$

Case 4: $K = \mathsf{KD}[k](\pi_i, N)$ and $K' \leftarrow\!\!\$\ \mathsf{KD}[k](\pi_j, N')$, for $\pi_i, \pi_j \leftarrow\!\!\$\ \mathrm{Perm}(n)$. For any two strings $(J, J') \in (\{0,1\}^{k+n})^2$, since $\mathsf{KD}[E]$ is 2-unpredictable, the chance that $(K, K') = (J, J')$ is at most

$$\frac{2}{2^{k+n}} \cdot \frac{2}{2^{k+n}} = \frac{4}{2^{2(k+n)}} \enspace .$$

Combining all cases, KeyGen is indeed 4-pairwise AU.

J Proof of Lemma 5

We shall use the H-coefficient technique. Let the real system implement game $\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}$ for challenge bit $b = 1$ (meaning EVAL is always implemented via $\mathsf{KD}[E]$), and let the ideal system implement game $\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}$ for challenge bit $b = 0$, (meaning EVAL is always implemented via $\mathsf{KD}[k]$). Without loss of generality, suppose that $\mathcal{A}$ never repeats a prior query. Since we consider computationally unbounded adversaries, without loss of generality, assume that the adversary is deterministic. Assume that $\mathcal{A}$ makes no redundant queries, meaning that if $\mathcal{A}$ queries (K, x) to E to get y, then it will not query (K, y) to E^{-1} to get x, and vice versa. Assume that for each evaluation query (i, N), the adversary called NEW() at least i times before, so that the key K_i was initialized.

When the adversary finishes querying, in the real world, we grant it the keys $K_1, K_2, \cdots$. In the ideal world, we instead grant it strings $K_1, K_2, \cdots \leftarrow_{\$} \{0,1\}^k$ independent of anything else. This can only help the adversary. Besides the revealed keys, a transcript consists of the following information:

- **Evaluation queries:** For each query (i, N) to EVAL, we will store six entries $(\mathsf{eval}, i, x_0, y_0), \ldots, (\mathsf{eval}, i, x_5, y_5)$, where each $x_s = \mathsf{pad}(N, s)$. For each $s \in \{0, \ldots, 5\}$, in the real world, $y_s = E_{K_i}(x_s)$, and in the ideal world, $y_s = \pi_i(x_s)$, where π_i is the secret permutation of user i. Clearly, the answer for this query in both worlds is $\mathsf{KD.Map}(y_0, \ldots, y_5)$. Thus we may grant the adversary more information that what it is supposed to receive, but this only helps the adversary. There are $6q$ eval entries.
- **Ideal-cipher queries:** For each query (K, x) to E for answer y, we store a corresponding entry $(\texttt{prim}, K, x, y, +)$. Likewise, for a query (K, y) to E^{-1} for answer x, we store a corresponding entry $(\texttt{prim}, K, x, y, -)$.

A transcript does not explicitly record the NEW() queries of $\mathcal{A}$, because the adversary is deterministic, and NEW returns no output.

DEFINING BAD TRANSCRIPTS. We say that a transcript is *bad* if one of the following happens:

1. There are entries $(\mathsf{eval}, i, x', y')$ and $(\texttt{prim}, K, x, y, +)$ such that K is the key of user i as indicated by the transcript, and $x = x'$.
2. There are entries $(\mathsf{eval}, i, x', y')$ and $(\texttt{prim}, K, x, y, -)$ such that K is the key of user i as indicated by the transcript, and $x = x'$.
3. There are entries $(\mathsf{eval}, i, x', y')$ and $(\texttt{prim}, K, x, y, +)$ such that K is the key of user i as indicated by the transcript, and $y = y'$.
4. There are entries $(\mathsf{eval}, i, x', y')$ and $(\texttt{prim}, K, x, y, -)$ such that K is the key of user i as indicated by the transcript, and $y = y'$.
5. There are entries (eval, i, x, y) and $(\mathsf{eval}, j, x, y')$ of the same input x, with $i \neq j$, such that according to the transcript, users i and j have the same key.
6. There are entries (eval, i, x, y) and $(\mathsf{eval}, j, x', y)$ of the same output y, with $i \neq j$, such that according to the transcript, users i and j have the same key.

If a transcript is not bad then we say that it is *good*.

Probability of bad transcripts. Let $\mathcal{X}_1$ be the random variable for the transcript in the ideal system. We now bound the probability that $\mathcal{X}_1$ is bad. Let $\mathcal{B}_j$ be the set of transcripts that violates the jth constraint of badness. We first bound the probability that $\mathcal{X}_1$ meets the first constraint of badness. Consider an entry $(\texttt{prim}, K, x, y, +)$ in $\mathcal{X}_1$. Since the adversary is d-repeating, there are at most d entries $(\mathsf{eval}, i, x, y')$ of the same input x, and for such an entry, the chance that $K_i = K$ in the ideal system, is exactly 2^{-k}. Summing this over at most p ideal-cipher queries, we have

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_1] \le \frac{dp}{2^k} \enspace .$$

Next, we bound the probability that $\mathcal{X}_1$ meets the second constraint of badness. View each entry (eval, i, x, y) as throwing a ball into bin y. Note that our $6q \le 2^{(1-\epsilon)n-1}$ throws are inter-dependent, but for each j-th throw, conditioning on the result of the prior throws, the chance that the j-th ball falls into any particular bin is at most $1/(2^n - 6q) \le 2^{1-n}$. Suppose that each bin contains at most a balls, which happens with probability at least $1 - 2^{-n/2}$, according to Lemma 10. Consider an entry $(\texttt{prim}, K, x, y, -)$ in $\mathcal{X}_1$. There are at most a entries $(\mathsf{eval}, i, x', y)$ of the same output y, and for such an entry, the chance that $K_i = K$ in the ideal system, is exactly 2^{-k}. Summing this over at most p ideal-cipher queries, we have

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_2] \le \frac{1}{2^{n/2}} + \frac{ap}{2^k} \enspace .$$

Next, for the third constraint of badness, consider entries $(\texttt{prim}, K, x, y, +)$ and $(\mathsf{eval}, i, x', y')$ in $\mathcal{X}_1$. If the ideal-cipher query is made after the evaluation query then given y', the random variable y can take at least $2^n - 6q - p \ge 2^{n-1}$ equally likely values. If the ideal-cipher query is made before the evaluation query then given y, the random variable y' can take at least $2^n - 6q - p \ge 2^{n-1}$ equally likely values. In either case, the chance that $y = y'$ and $K = K_i$ in the ideal system is at most $2/2^{k+n}$. Summing this over at most p ideal-cipher queries and $6q$ evaluation entries,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_3] \le \frac{12pq}{2^{n+k}} \enspace .$$

For the fourth constraint of badness, consider entries $(\texttt{prim}, K, x, y, -)$ and $(\mathsf{eval}, i, x', y')$ in $\mathcal{X}_1$. If the ideal-cipher query is made after the evaluation query then given x', the random variable x can take at least $2^n - 6q - p \ge 2^{n-1}$ equally likely values. If the ideal-cipher query is made before the evaluation query then given x, the random variable x' can take at least $2^n - 6q - p \ge 2^{n-1}$ equally likely values. In either case, the chance that $x = x'$ and $K = K_i$ in the ideal system is at most $2/2^{k+n}$. Summing this over at most p ideal-cipher queries and $6q$ evaluation entries,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_4] \le \frac{12pq}{2^{n+k}} \enspace .$$

For the fifth constraint of badness, consider a nonce N. For each $N \in \{0,1\}^r$, let $Q_N \le d$ be the random variable for the number of evaluation queries that use nonce N. For each $x \in \{\mathsf{pad}(N,0), \ldots, \mathsf{pad}(N,5)\}$, there are at most

$$\binom{Q_N}{2} \le \frac{(Q_N)^2}{2} \le \frac{dQ_N}{2}$$

pairs of entries (eval, i, x, y) and $(\mathsf{eval}, j, x, y')$, with $i \ne j$. For each such pair, the chance that $K_i = K_j$ in the ideal system is 2^{-k}. Hence

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_5] \le \sum_N \frac{6d \cdot \mathbf{E}[Q_N]}{2} \cdot \frac{1}{2^k} = \frac{3d}{2^k} \cdot \mathbf{E}\Big[\sum_N Q_N\Big] \le \frac{3dq}{2^k},$$

where the last inequality is due to the fact that $\sum_N Q_N$ is exactly the total number of evaluation queries. For the last constraint of badness, note that for any entries (eval, i, x, y) and $(\mathsf{eval}, j, x', y')$, if $i \ne j$ then y and y' are independent, uniformly distributed over $\{0,1\}^n$. For each such pair, the chance that $K_i = K_j$ and $y = y'$ is $2^{-(k+n)}$. Summing over at most

$$\binom{6q}{2} \le 18q^2$$

pairs of eval entries,

$$\Pr[\mathcal{X}_1 \in \mathcal{B}_6] \le \frac{18q^2}{2^{k+n}}$$

Summing up,

$$\Pr[\mathcal{X}_1 \text{ is bad}] \le \sum_{j=1}^{6} \Pr[\mathcal{X}_1 \in \mathcal{B}_j] \le \frac{1}{2^{n/2}} + \frac{24pq + 18q^2}{2^{k+n}} + \frac{ap + d(p+3q)}{2^k} \quad .$$

Transcript ratio. Let $\mathbf{S}_0$ be the real system, and $\mathbf{S}_1$ be the ideal system. We now show that

$$\mathsf{p}_{\mathbf{S}_0}(\tau) \ge \mathsf{p}_{\mathbf{S}_1}(\tau)$$

for any good transcript τ such that $\mathsf{p}_{\mathbf{S}_1}(\tau) > 0$. Fix such a transcript τ. Note that in computing $\mathsf{p}_{\mathbf{S}}(\tau)$, for $\mathbf{S} \in \{\mathbf{S}_0, \mathbf{S}_1\}$, we can ignore the sign of the ideal-cipher entries, and treat that as a forward query. For a key $K \in \{0,1\}^k$, let $S_1(K) = \{(\mathsf{eval}, i, x, y) \mid K_i = K\}$, and $S_2(K) = \{(\mathtt{prim}, K, x, y, \cdot)\}$. Since τ is good, $|S_1(K)|$ is divisible by 6 for every $K \in \{0,1\}^k$. Suppose that τ contains exactly u users, q evaluation queries, and p ideal-cipher queries. Then

$$\mathsf{p}_{\mathbf{S}_1}(\tau) = 2^{-ku} \cdot \frac{1}{\Big(2^n \cdots (2^n - 5)\Big)^q} \cdot \prod_{K \in \{0,1\}^k} \prod_{i=0}^{|S_2(K)|-1} \frac{1}{2^n - i} \quad .$$

In the real world, because the transcript is good, for each key K, the sets $S_1(K)$ and $S_2(K)$ call E_K on different inputs, and thus

$$\begin{aligned}\mathsf{p_{S_0}}(\tau) &= 2^{-ku} \prod_{K\in\{0,1\}^k} \prod_{i=0}^{|S_1(K)|+|S_2(K)|-1} \frac{1}{2^n - i}\\ &\geq 2^{-ku} \prod_{K\in\{0,1\}^k} \frac{1}{\Big(2^n\cdots(2^n-5)\Big)^{|S_1(K)|/6}} \prod_{i=0}^{|S_2(K)|-1} \frac{1}{2^n-i} \enspace .\end{aligned}$$

Since

$$\sum_{K\in\{0,1\}^k} |S_1(K)| = 6q,$$

it follows that $\mathsf{p_{S_0}}(\tau) \geq \mathsf{p_{S_1}}(\tau)$.

K Proof of Lemma 7

Without loss of generality, assume that if $\mathcal{A}_0$ queries $\textsc{Enc}(i, N, M, A)$ for an answer C, then later it will not query $\textsc{Vf}(i, N, C, A)$. We now construct $\mathcal{A}_1$. Adversary $\mathcal{A}_1$ runs $\mathcal{A}_0$, and uses its $\textsc{Enc}$ and $\textsc{New}$ oracles to respond to the latter's queries accordingly. For each verification query (i, N, C, A) of $\mathcal{A}_0$, if there is no prior encryption query (i, N, M, A') then $\mathcal{A}_1$ simply ignores it. Otherwise, $\mathcal{A}_1$ uses its $\textsc{Vf}$ oracle to respond. Finally, when $\mathcal{A}_0$ outputs a bit b', $\mathcal{A}_1$ outputs the same bit. Next, we construct $\mathcal{A}_2$. Adversary $\mathcal{A}_2$ runs $\mathcal{A}_0$, and uses its $\textsc{Enc}$ and $\textsc{New}$ oracles to respond to the latter's queries accordingly. For each verification query (i, N, C, A) of $\mathcal{A}_0$, if there is some prior encryption query (i, N, M, A') then $\mathcal{A}_2$ simply ignores it. Otherwise, $\mathcal{A}_2$ uses its $\textsc{Vf}$ oracle to respond. When $\mathcal{A}_0$ outputs a bit, $\mathcal{A}_2$ outputs the same bit. Then

$$\mathsf{Adv}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_0) \leq \mathsf{Adv}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_1) + \mathsf{Adv}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_2) \enspace .$$

We now bound $\mathsf{Adv}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_1)$ by building an adversary $\overline{\mathcal{A}}$ for distinguishing $\mathsf{KD}[E]$ and $\mathsf{KD}[k]$. Adversary $\overline{\mathcal{A}}$ simulates game $\mathbf{G}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_1)$, but each time it needs to generate a session key, it uses its $\textsc{Eval}$ oracle instead of $\mathsf{KD}[E]$. However, if $\overline{\mathcal{A}}$ previously queried $\textsc{Eval}(i, N)$ for an answer K, next time it simply uses K without querying. When $\mathcal{A}_1$ outputs a bit, adversary $\overline{\mathcal{A}}$ outputs the same answer. Let c be the challenge bit in game $\mathbf{G}^{\text{dist}}_{\mathsf{KD}[E]}(\overline{\mathcal{A}})$. Then

$$\begin{aligned}\Pr[\mathbf{G}^{\text{dist}}_{\mathsf{KD}[E]}(\overline{\mathcal{A}}) \Rightarrow \mathsf{true} \mid c = 1] &= \Pr[\mathbf{G}^{\text{mu-auth}}_{\overline{\mathsf{AE}}}(\mathcal{A}_1)], \text{ and}\\ \Pr[\mathbf{G}^{\text{dist}}_{\mathsf{KD}[E]}(\overline{\mathcal{A}}) \Rightarrow \mathsf{false} \mid c = 0] &= \Pr[\mathbf{G}^{\text{mu-auth}}_{\mathsf{KtE}[\mathsf{KD}[k],\mathsf{AE}],E}(\mathcal{A}_1)] \enspace .\end{aligned}$$

Subtracting, we get

$$\mathsf{Adv}^{\text{dist}}_{\mathsf{KD}[E]}(\overline{\mathcal{A}}) = \frac{1}{2}\big(\mathsf{Adv}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_1) - \mathsf{Adv}^{\text{mu-auth}}_{\mathsf{KtE}[\mathsf{KD}[k],\mathsf{AE}],E}(\mathcal{A}_1)\big) \enspace .$$

Note that $\overline{\mathcal{A}}$ makes at most $p + L \le 2^{n-4}$ ideal-cipher queries, and q Eval queries. Moreover, $\overline{\mathcal{A}}$ is d-repeating. Hence using Lemma 5,

$$\mathsf{Adv}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\overline{\mathcal{A}}) \le \frac{1}{2^{n/2}} + \frac{24(L+p)q + 18q^2}{2^{k+n}} + \frac{a(L+p) + d(L+p+3q)}{2^k} \ .$$

Putting this all together,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{mu\text{-}auth}}_{\mathsf{AE},E}(\mathcal{A}_0) &\le \mathsf{Adv}^{\mathsf{mu\text{-}auth}}_{\mathsf{KtE}[\mathsf{KD}[k],\mathsf{AE}],E}(\mathcal{A}_1) + \mathsf{Adv}^{\mathsf{mu\text{-}auth}}_{\mathsf{AE},E}(\mathcal{A}_2) \\ &\quad + \frac{2}{2^{n/2}} + \frac{48(L+p)q + 36q^2}{2^{k+n}} + \frac{2(a+d)L + 2(a+d)p + 6dq}{2^k} \ .\end{aligned}$$

This concludes the proof.

L Proof of Lemma 8

Without loss of generality, assume that if the adversary makes an encryption query (i, N, M, A) for an answer C, then later it will not query $\textsc{Vf}(i, N, C, A)$. Assume that it always calls all q $\textsc{New}()$ queries at the beginning. Assume that the adversary does not repeat prior queries, and does not make redudant ideal-cipher queries.

Transition to game G_0. We will construct from $\mathcal{A}$ another adversary $\overline{\mathcal{A}}$ that plays game G_0 as shown in Fig. 10. The adversary is given oracle access to E and its inverse, and $\textsc{New}()$ as usual; the latter initializes a user v with master key $K_v \leftarrow_{\$} \{0,1\}^n$ upon each call. It is also given another evaluation oracle $\textsc{Eval}(i, N)$ that implements $\mathsf{KD}[E](K_i, N)$. The adversary has to output a tuple $(\boldsymbol{I}, \boldsymbol{N}, \boldsymbol{C}, \boldsymbol{A})$ of vectors. We require that $\boldsymbol{I}[j] \in \{1, \ldots, v\}$ for every j, and if the adversary previously queried $\textsc{Eval}(i, N)$ and then $(\boldsymbol{I}[j], \boldsymbol{N}[j]) \ne (i, N)$ for all $j \le |\boldsymbol{I}|$. The game then iterates through every verification query $(\boldsymbol{I}[j], \boldsymbol{N}[j], \boldsymbol{C}[j], \boldsymbol{A}[j])$, and try to decrypt $\overline{\mathsf{AE}}.\mathsf{D}^E(K_{\boldsymbol{I}[j]}, \boldsymbol{N}[j], \boldsymbol{C}[j], \boldsymbol{A}[j])$. If some verification query results in a non-$\bot$ answer then the game returns true, meaning the adversary wins the game. Otherwise the game returns false.

We now construct the adversary $\overline{\mathcal{A}}$ as shown in Fig. 11. It runs $\mathcal{A}$ and uses its $\textsc{New}$ oracle to respond to the latter's queries of the same type, and maintains a counter v starting at 0. For each encryption query (i, N, M, A) of $\mathcal{A}$, adversary $\overline{\mathcal{A}}$ will first query $\textsc{Eval}(i, N)$ to get the session key K, and then computes $C \leftarrow \mathsf{AE}.\mathsf{E}^E(K, N, M, A)$ and returns the answer C to $\mathcal{A}$. (However, if it previously queried $\textsc{Eval}(i, N)$ before then it simply reuses the prior answer as K.) For each decryption query (i, N, C, A) of $\mathcal{A}$, adversary $\overline{\mathcal{A}}$ increments v, and updates $(\boldsymbol{I}[v], \boldsymbol{N}[v], \boldsymbol{C}[v], \boldsymbol{A}[v]) \leftarrow (i, N, C, A)$. Finally, when $\mathcal{A}$ terminates, $\overline{\mathcal{A}}$ outputs $(\boldsymbol{I}, \boldsymbol{N}, \boldsymbol{C}, \boldsymbol{A})$ that it has maintained. Since $\mathcal{A}$ is simple, $\overline{\mathcal{A}}$ does not violate the requirements of game G_0. Moreover, $\overline{\mathcal{A}}$ makes at most q evaluation queries, $L+p$ ideal-cipher queries (but only p of them are backward queries), and q verification queries. For this constructed adversary $\overline{\mathcal{A}}$,

$$\Pr[G_0] = \mathsf{Adv}^{\mathsf{mu\text{-}auth}}_{\mathsf{AE},E}(\mathcal{A}) \ .$$

Game G_0	$\textsc{New}()$
$v \leftarrow 0$; $S \leftarrow\!\!\$\ \emptyset$; $(\boldsymbol{I}, \boldsymbol{N}, \boldsymbol{C}, \boldsymbol{A}) \leftarrow\!\!\$\ \overline{\mathcal{A}}^{\textsc{New},\textsc{Eval},E,E^{-1}}$	$v \leftarrow v + 1$; $K_v \leftarrow\!\!\$\ \{0,1\}^k$
For $j = 1$ to $\lvert \boldsymbol{I} \rvert$ do	$\textsc{Eval}(i, N)$
If $\boldsymbol{I}[j] \notin \{1, \ldots, v\}$ then return false	//Implement $\mathsf{KD}[E](K_i, N)$
If $(\boldsymbol{I}[j], \boldsymbol{N}[j]) \in S$ then return false	If $i \notin \{1, \ldots, v\}$ then return $\perp$
For $j = 1$ to $\lvert \boldsymbol{I} \rvert$ do	$K \leftarrow \mathsf{KD}[E](K_i, N)$
$M \leftarrow \overline{\mathsf{AE}}^E.\mathsf{D}(K_{\boldsymbol{I}[j]}, \boldsymbol{N}[j], \boldsymbol{C}[j], \boldsymbol{A}[j])$	$S \leftarrow S \cup \{(i, N)\}$
If $M \neq \perp$ then return true	Return K
Return false	

Fig. 10: **Game G_0 in the proof of Lemma 8.**

Adversary $\overline{\mathcal{A}}^{\textsc{New},\textsc{Eval},E,E^{-1}}$	$\textsc{Enc}(i, N, M, A)$
$v \leftarrow 0$; $\mathcal{A}^{\textsc{New},\textsc{Enc},\textsc{Vf},E,E^{-1}}$	$K \leftarrow \textsc{Eval}(i, N)$
Return $(\boldsymbol{I}, \boldsymbol{N}, \boldsymbol{C}, \boldsymbol{A})$	$C \leftarrow \mathsf{AE.E}^E(K, N, M, A)$
$\textsc{Vf}(i, N, C, A)$	Return C
$v \leftarrow v + 1$; $\boldsymbol{I}[v] \leftarrow i$	
$\boldsymbol{N}[v] \leftarrow N$; $\boldsymbol{C}[v] \leftarrow C$; $\boldsymbol{A}[v] \leftarrow A$	

Fig. 11: **Constructed adversary $\overline{\mathcal{A}}$ in the proof of Lemma 8.**

Bounding $\Pr[G_0]$. Each time $\overline{\mathcal{A}}$ makes a forward query $E_K(X)$, if (1) there is some $N \in \mathcal{N}$ such that $X \in \Omega = \{\mathsf{pad}(N, 0), \ldots, \mathsf{pad}(N, 5)\}$, (2) there is no prior backward query $E^{-1}(K, Y)$ for answer $X' \in \Omega \backslash \{X\}$, then we immediately grant the adversary the free queries $E_K(X^*)$, for all $X' \in \Omega \backslash \{X\}$. Thus the adversary makes at most $6(L + p)$ ideal-cipher queries, but at most p of them are backward ones. Assume that $\overline{\mathcal{A}}$ does not repeat prior queries, and it does not make redundant ideal-cipher queries: if it gets $E(K, X)$ for answer Y then it will not later query $E^{-1}(Y)$ to get answer X again, and vice versa. Recall that this adversary makes all q $\textsc{New}()$ queries at the beginning.

Let G_1 be the following variant of game G_0. In G_1, the evaluation oracle implements $\mathsf{KD}[k]$. Specifically, when $\textsc{New}()$ initializes user v, it also samples a secret permutation $\pi_v \leftarrow\!\!\$\ \mathrm{Perm}(n)$ along with the key K_v. On query $\textsc{Eval}(i, N)$, the oracle lets $X_s \leftarrow \mathsf{pad}(N, s)$ for every $s \in \{0, \ldots, 5\}$, computes $Y_s \leftarrow \pi_i(X_s)$, creates internal entries $(\mathtt{prim}, K_i, X_s, Y_s, +)$, and returns $\mathsf{KD.Map}(Y_0, \ldots, Y_5)$. For each query $E(K, X)$ for answer Y, the game creates an entry $(\mathtt{prim}, K, X, Y, +)$. Likewise, for each query $E^{-1}(K, Y)$ for answer X, the game creates an entry $(\mathtt{prim}, K, X, Y, -)$. So there are at most $6Q$ $\mathtt{prim}$ entries, where $Q = L + p + q$. We say that the entries are *incompatible* if there are different entries $(\mathtt{prim}, K, X, Y, \cdot)$ and $(\mathtt{prim}, K, X', Y', \cdot)$ of the same key K such that either $X = X'$ or $Y = Y'$. If incompatibility happens then game G_1 returns false, meaning that the adversary *loses* the game. Otherwise, the game programs E to be consistent with the $\mathtt{prim}$

entries, and then processes the verification queries as in game G_0. In particular, in decrypting $\overline{\mathsf{AE}}^E.\mathsf{D}(K_{\boldsymbol{I}[j]}, \boldsymbol{N}[j], \boldsymbol{C}[j], \boldsymbol{A}[j])$, the KDF of $\overline{\mathsf{AE}}$ is still $\mathsf{KD}[E]$.

To bound the gap between games G_0 and G_1, we construct an adversary $\mathcal{A}^*$ that tries to distinguish $\mathsf{KD}[E]$ and $\mathsf{KD}[k]$, as defined in game $\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\mathcal{A}^*)$, but it will be additionally granted the master keys $K_1, \ldots, K_q$ when it finishes querying. Note that Lemma 5 still applies to this key-revealing setting. The goal of $\mathcal{A}^*$ is to simulate G_0 if it interacts with $\mathsf{KD}[E]$, and simulate G_1 if it interacts with $\mathsf{KD}[k]$. To achieve that, it runs $\overline{\mathcal{A}}$, uses its oracles to answer the queries of the latter of the same type, and maintains `prim` entries for the ideal-cipher queries. When $\overline{\mathcal{A}}$ finishes querying, $\mathcal{A}^*$ asks to be granted master keys $K_1, \ldots, K_q$; at this point it is not allowed to make further queries. Adversary $\mathcal{A}^*$ now creates `prim` entries corresponding to the past evaluation queries. If there are entries $(\mathtt{prim}, K, X, Y, \cdot)$ and $(\mathtt{prim}, K, X', Y', \cdot)$ of the same key K, but either (1) $X = X'$ but $Y \neq Y'$, or (2) $X \neq X'$ but $Y = Y'$, then $\mathcal{A}^*$ terminates and outputs 0, indicating that it has been interacting with $\mathsf{KD}[k]$. Otherwise, it will process the verification queries of $\overline{\mathcal{A}}$ as in game G_0. However, recall that at this point it cannot make further queries to E/E^{-1}. So during the handling of the verification queries, when it is supposed to query $E(K, X)$, if there is an entry $(\mathtt{prim}, K, X, Y, \cdot)$ then it simply uses the answer Y without querying E at all. If there is no such entry then it picks an answer $Y \leftarrow_{\$} \{0,1\}^n \backslash S$, where S is the set of all strings Y^* such that there is an entry $(\mathtt{prim}, K, X^*, Y^*, \cdot)$, and creates a new entry $(\mathtt{prim}, K, X, Y, +)$. If there is a verification query that results in a non-$\perp$ answer then $\mathcal{A}^*$ outputs 1, indicating that it has been interacting with $\mathsf{KD}[E]$. Otherwise it outputs 0.

We now analyze the advantage of $\mathcal{A}^*$. Let b be the challenge bit in the game $\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\mathcal{A}^*)$. Then

$$\Pr[\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\mathcal{A}^*) \Rightarrow \mathsf{true} \mid b = 1] = \Pr[G_0] \ .$$

On the other hand, we claim that

$$\Pr[\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\mathcal{A}^*) \Rightarrow \mathsf{false} \mid b = 0] \leq \Pr[G_1] + \frac{q(18q + 144L + 144p)}{2^{k+n}} \ . \tag{10}$$

Subtracting, we obtain

$$\mathsf{Adv}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\mathcal{A}^*) \geq \Pr[G_0] - \Pr[G_1] - \frac{q(18q + 144L + 144p)}{2^{k+n}} \ .$$

We now justify Equation (10). Note that the difference between $\Pr[G_1]$ and $\Pr[\mathbf{G}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\mathcal{A}^*) \Rightarrow \mathsf{false} \mid b = 0]$ is bounded by the chance that there are distinct entries $(\mathtt{prim}, K, X, Y, \cdot)$ and $(\mathtt{prim}, K', X', Y', \cdot)$ such that $K = K'$, $X = X'$, and $Y = Y'$, because in that case, game G_1 will terminate prematurely due to incompatibility, but $\mathcal{A}^*$ still proceeds into handling the verification queries. Due to symmetry, there are only three cases.

Case 1: The first entry is created by some $\textsc{Eval}(i, N)$ and the second entry by $\textsc{Eval}(j, N)$, with $i \neq j$. Since π_i and π_j are independent, Y and Y' are independent, uniformly random strings. Since K_i and K_j are independent with π_i and π_j, the chance that $K_i = K_j$ and $Y = Y'$ is at most $2^{-(k+n)}$. Summing over at most

$$\binom{6q}{2} \le 18q^2$$

pairs of entries created by the q evaluation queries, the chance this case happens is at most $18q^2/2^{k+n}$.

Case 2: The first entry is created by some $\textsc{Eval}(i, N)$ and the second entry by querying $E(K', X')$. If the evaluation query is made first then given Y, there are still at least $2^n - 6Q \ge 2^{n-1}$ equally likely choices for Y', and thus the chance that $K_i = K'$ and $Y' = Y$ is at most $2/2^{k+n}$. Likewise, if the ideal-cipher query is made first then given Y', there are still at least 2^{n-1} equally likely choices for Y, and thus the chance that $K_i = K'$ and $Y = Y'$ is also at most $2/2^{k+n}$. Summing this over $36(L+p)q$ possible pairs, the chance that this case happens is at most $72(L+p)q/2^{k+n}$.

Case 3: The first entry is created by some $\textsc{Eval}(i, N)$ and the second entry by querying $E^{-1}(K', Y')$. Similar to case 2, the chance that this case happens is at most $72(L+p)q/2^{k+n}$.

Combining all cases leads us to Equation (10). On the other hand, $\mathcal{A}^*$ is d-repeating and makes at most q evaluation queries, and $6(L+p) \le 2^{n-4}$ ideal-cipher queries. From Lemma 5,

$$\mathsf{Adv}^{\mathsf{dist}}_{\mathsf{KD}[E]}(\mathcal{A}^*) \le \frac{1}{2^{n/2}} + \frac{144(L+p)q + 18q^2}{2^{k+n}} + \frac{6a(L+p) + d(6L+6p+3q)}{2^k} \ .$$

Hence

$$\Pr[G_0] - \Pr[G_1] \le \frac{1}{2^{n/2}} + \frac{288(L+p)q + 36q^2}{2^{n+k}} + \frac{6a(L+p) + d(6L+6p+3q)}{2^k}.$$

What remains is to bound $\Pr[G_1]$.

Bounding $\Pr[G_1]$. Consider the following balls-into-bins game. For each entry $(\mathtt{prim}, K, X, Y, +)$, view this as throwing a ball to bin Y. Likewise, for each entry $(\mathtt{prim}, K, X, Y, -)$, view this as throwing a ball to bin X. Thus there are at most $6Q \le 2^{(1-\epsilon)n-1}$ throws. For each j-th throw, given the result of the prior throws, the conditional probability that the j-th ball lands into any particular bin is at most 2^{1-n}. From Lemma 10, with probability at least $1 - 2^{-n/2}$, each bin contains at most a balls.

Consider the following balls-into-bins game. For each query $\textsc{Eval}(i, N)$ for answer Z, we view this as throwing a ball into bin $Z[n+1 : n+k]$. Likewise, for each 6-tuple $(\mathtt{prim}, K, \mathsf{pad}(N, 0), R_0, +), \ldots, (\mathtt{prim}, K, \mathsf{pad}(N, 1), R_5, +)$ created by querying E, we view this as throwing a ball into bin $Z[n+1 : n+k]$, where $Z \leftarrow \mathsf{KD.Map}(R_0, \ldots, R_5)$. Thus there are at most $Q \le \min\{2^{(1-\epsilon)k-1}, \frac{1}{16} \cdot 2^n\}$ throws. For each j-th throw, given the result of the prior throws, since KD is 2-unpredictable, the conditional probability that the j-th ball lands into any particular bin is at most 2^{1-k}. From Lemma 10, with probability at least $1 - 2^{-k/2} \geq 1 - 2^{-n/2}$, each bin contains at most a balls.

We claim that for each $j \leq q$, the probability that the verification query $(\boldsymbol{I}[j], \boldsymbol{N}[j], \boldsymbol{C}[j], \boldsymbol{A}[j])$ makes game G_1 return true is at most

$$\frac{11}{2^n} + \frac{8a + 7a^2}{2^k} + \mathbf{E}\big[|\boldsymbol{C}[j]|_n + |\boldsymbol{A}[j]|_n\big]\Big(\frac{na}{2^k} + \frac{48c(L+p+q)}{2^{n+k}}\Big) \ . \tag{11}$$

This claim will be justified later. Summing over q verification queries, and accounting for the $2/2^{n/2}$ probability due to the two balls-into-bins games,

$$\Pr[G_1] \leq \frac{2}{2^{n/2}} + \frac{11q}{2^n} + \frac{(8a+7a^2)q}{2^k} + \frac{naL}{2^k} + \frac{48c(L+p+q)L}{2^{n+k}} \ .$$

Hence

$$\begin{aligned}\mathsf{Adv}^{\mathsf{mu\text{-}auth}}_{\mathsf{AE},E}(\mathcal{A}) \leq \frac{3}{2^{n/2}} + \frac{11q}{2^n} + \frac{288(L+p)q + 36q^2 + 48c(L+p+q)L}{2^{n+k}} \\ + \frac{(8a+7a^2+3d)q + (na+6a+6d)L + 6(a+d)p}{2^k} \ .\end{aligned}$$

Below, we will prove Equation (11).

Handling each verification query. Fix $j^* \leq q$, and let (i, N, C, A) be the j^*-th verification query. Let T be the IV in C and let M be the random variable for the decrypted message $\mathsf{CTR}[E].\mathsf{D}(K_{\mathsf{out}}, N, C, A)$, where $K_{\mathsf{in}} \parallel K_{\mathsf{out}}$ is the random variable for the session key of user i for nonce N. The message M is determined from C if we know all pairs $(X, E(K_{\mathsf{out}}, X))$ for every string $X \in 1\{0,1\}^{n-1}$. Consider the following cases.

Case 1: There is no entry $(\texttt{prim}, K_i, X, \cdot, \cdot)$ with $X \in \{\mathsf{pad}(N,0), \ldots, \mathsf{pad}(N,5)\}$. Assume that all prim entries are compatible; otherwise the adversary simply loses the game. Let $V \leftarrow \mathsf{xor}(H(K_{\mathsf{in}}, M, A), N)$. Processing this verification query will return true only if $E(K_{\mathsf{out}}, V) = T$. After programming E, given all prim entries and K_i, since KD is 2-unpredictable, there are at least 2^{k-1} equally choices for the K_{out}. Assume that $K_{\mathsf{out}} \neq K_i$, which happens with probability at least $1 - 2/2^k \geq 1 - 2/2^n$.

Suppose that there is no entry $(\texttt{prim}, K_{\mathsf{out}}, V, \cdot, \cdot)$. Note that V starts with 0. Given all prim entries, $(N, C, A, K_i, K_{\mathsf{in}}, K_{\mathsf{out}})$ and all pairs $(X, E(K_{\mathsf{out}}, X))$ for every $X \in 1\{0,1\}^{n-1}$, (i) one can determine V, but (ii) there are at least $2^{n-1} - 6Q \geq 2^{n-2}$ equally likely choices for $E(K_{\mathsf{out}}, V)$, and thus the chance that $T = E(K_{\mathsf{out}}, V)$ is at most $4/2^n$.

Suppose that there is an entry $(\texttt{prim}, K_{\mathsf{out}}, V, T', \cdot)$, with $T' \neq T$. Then after programming E, the chance that $T = E(K_{\mathsf{out}}, V)$ is 0.

We now bound the chance that there is an entry $(\texttt{prim}, K_{\mathsf{out}}, V, T, \cdot)$. Clearly this entry, if exists, is not created by $\textsc{Eval}(i, \cdot)$ queries since $K_{\mathsf{out}} \neq K_i$. First consider the case that either (1) $A \neq \varepsilon$, or (2) $M \neq \varepsilon$, or (3) H is c-regular.

After programming E, given all $\texttt{prim}$ entries, $(N, C, A, K_i, K_{\mathsf{out}})$ and all pairs $(X, E(K_{\mathsf{out}}, X))$ for every $X \in 1\{0,1\}^{n-1}$, (i) one can uniquely determine M, but (ii) since KD is 2-unpredictable, for any string K, the conditional probability that $K_{\mathsf{in}} = K$ is at most 2^{1-n}. Hence, for any entry $(\texttt{prim}, K, X, \cdot, \cdot)$ that is not created by $\textsc{Eval}(i, \cdot)$ queries, due to the (possibly weak) c-regularity of H and the 2-regularity of xor, the chance that $K_{\mathsf{out}} = K$ and $V = X$ is at most

$$\frac{2}{2^k} \cdot \frac{2 \cdot c \cdot \mathbf{E}\big[|C|_n + |A|_n\big]}{2^{n-1}} = \frac{8c \cdot \mathbf{E}\big[|C|_n + |A|_n\big]}{2^{k+n}} \enspace .$$

Summing over $6Q$ entries $(\texttt{prim}, K, X, \cdot, \cdot)$, we obtain a bound $\frac{48Qc \cdot \mathbf{E}\left[|C|_n + |A|_n\right]}{2^{n+k}}$. Next we consider the case that both A and M are the empty string, and H is weakly c-regular. Note that one can check if $M = \varepsilon$ without knowing the key K_{out}, by comparing $|C|$ with n. Now since H is weakly regular, $H(K_{\mathsf{in}}, M, A) = 0^n$, and thus $V = \mathsf{xor}(0^n, N)$. From the balls-into-bins assumption above, there are at most $2a$ entries $(\texttt{prim}, K, V, T, \cdot)$, and the chance that one such K is K_{out} is at most $2a/2^k$.

Summing up, for this case, the chance that the verification query (i, N, C, A) makes game G_1 return true is at most

$$\frac{6}{2^n} + \frac{48Qc \cdot \mathbf{E}\big[|C|_n + |A|_n\big]}{2^{n+k}} + \frac{2a}{2^k} \enspace .$$

Case 2: There is an entry $(\texttt{prim}, K_i, Z, \cdot, -)$ for $Z \in \Omega = \{\mathsf{pad}(N, 0), \ldots \mathsf{pad}(N, 5)\}$. Regardless of how the adversary chooses (i, N), there are at most $6a$ entries $(\texttt{prim}, \cdot, V, \cdot, -)$ with $V \in \Omega$. Since the key K_i is sampled independent of the $(\texttt{prim}, \cdot, \cdot, \cdot, -)$ entries, this case happens with probability at most $\frac{6a}{2^k}$.

Case 3: Entries $(\texttt{prim}, K_i, \mathsf{pad}(N, 0), \cdot, +), \ldots, (\texttt{prim}, K_i, \mathsf{pad}(N, 5), \cdot, +)$ exist. Note that those entries are not created by queries $\textsc{Eval}(i, \cdot)$, because the adversary is not allowed to query $\textsc{Eval}(i, N)$ and then output a verification query (i, N, C, A). Except for entries created by $\textsc{Eval}(i, \cdot)$, the key K_i is independent of the remaining entries. Since KD is 2-unpredictable, assume that $K_{\mathsf{out}} \neq K_i$, which happens with probability at least $1 - \frac{2Q}{2^k} \cdot \frac{1}{2^k} \geq 1 - \frac{2Q}{2^{k+n}}$. We consider the following sub-cases.

Case 3.1: There is no entry $(\texttt{prim}, K_{\mathsf{out}}, Z, T, \cdot)$, where Z is a string starting with 0. Assume that the entries are compatible; otherwise the adversary simply loses the game. After programming E, given all entries, $(K_i, K_{\mathsf{in}}, K_{\mathsf{out}})$, and all pairs $(X, E(K_{\mathsf{out}}, X))$ for every $X \in 1\{0,1\}^{n-1}$, (i) one can determine M from C, and then compute $V \leftarrow \mathsf{xor}(H(K_{\mathsf{in}}, M, A), N)$, but (ii) either there is an entry $(\texttt{prim}, K_{\mathsf{out}}, V, T', \cdot)$ with $T' \neq T$, meaning $E(K_{\mathsf{out}}, V) = T' \neq T$, or there is no entry $(\texttt{prim}, K_{\mathsf{out}}, V, \cdot, \cdot)$, meaning that there are still at least $2^{n-1} - 6Q \geq 2^{n-2}$ equally likely choices for $E(K_{\mathsf{out}}, V)$, and therefore the chance that $E(K_{\mathsf{out}}, V) = T$ is at most $4/2^n$. Hence in this case the chance that the verification query above can make G_1 answer true is at most $4/2^n$.

Case 3.2: There is an entry $(\texttt{prim}, K_{\mathsf{out}}, Z, T, +)$ where Z is a string starting with 0. Now, the entry $(\texttt{prim}, K_{\mathsf{out}}, Z, T, +)$, if exists, does not come from

$\textsc{Eval}(i, \cdot)$ queries, since $K_i \neq K_{\mathsf{out}}$ as mentioned above. Note that due to the balls-into-bins assumption above, there are at most a^2 septets $(\mathtt{prim}, J, \cdot, T, +)$, $(\mathtt{prim}, K, \mathsf{pad}(N', 0), R_0, +), \ldots, (\mathtt{prim}, \mathsf{pad}(N', 5), R_5, +)$ such that (i) J is the k-bit suffix of $\mathsf{KD.Map}(R_0, \ldots, R_5)$, and (ii) those entries are not from the evaluation queries of user i. For each such septet, the chance that $K = K_i$ is 2^{-k}. Hence this case happens with probability at most $a^2/2^k$.

Case 3.3: There is an entry $(\mathtt{prim}, K_{\mathsf{out}}, Z, T, -)$, where $Z \in 0\{0,1\}^{n-1}$. Assume that $Z \notin \Omega = \{\mathsf{pad}(N, 0), \ldots, \mathsf{pad}(N, 5)\}$. This happens with probability at least $1 - 6a^2/2^k$, since due to the balls-into-bins assumption above, there are at most $6a^2$ septets of entries $(\mathtt{prim}, J, Z', T, -), (\mathtt{prim}, K, \mathsf{pad}(N, 0), R_0, +), \ldots,$ $(\mathtt{prim}, K, \mathsf{pad}(N, 5), R_5, +)$ such that J is the k-bit suffix of $\mathsf{KD.Map}(R_0, \ldots, R_5)$ and $Z' \in \Omega$.

If incompatibility does not happen then either $(\mathtt{prim}, K_i, \mathsf{pad}(N, 0), \cdot, +), \ldots,$ $(\mathtt{prim}, K_i, \mathsf{pad}(N, 5), \cdot, +)$ all belong to the ideal-cipher queries, or they all are created by queries $\textsc{Eval}(j, \cdot)$ for some $j \neq i$. Consider all septets $\mathcal{T}$ of entries $(\mathtt{prim}, K, \mathsf{pad}(N, 0), R_0, +), \ldots, (\mathtt{prim}, K, \mathsf{pad}(N, 5), R_5, +)$, $(\mathtt{prim}, J, U, T, -)$, in which $J' \parallel J \leftarrow \mathsf{KD.Map}(R_0, \ldots, R_5)$, such that (1) either the first six entries belong to the ideal-cipher queries, or they are created by queries $\textsc{Eval}(j, \cdot)$, with $j \neq i$, and (2) $J \neq K$ and (3) $U \in 0\{0,1\}^{n-1} \backslash \{\mathsf{pad}(N, 0), \ldots, \mathsf{pad}(N, 5)\}$. For such a septet $\mathcal{T}$, denote $K = \mathsf{MKey}(\mathcal{T})$, $J = \mathsf{OKey}(\mathcal{T})$, $J' = \mathsf{IKey}(\mathcal{T})$ and $U = \mathsf{Input}(\mathcal{T})$, and let $\mathsf{Msg}(\mathcal{T}, C)$ be the message obtained by decrypting $\mathsf{CTR}[E].\mathsf{D}(\mathsf{OKey}(\mathcal{T}), C)$. This query (i, N, C, A) makes game G_1 return true only if incompatibility does not happen, and there is a septet $\mathcal{T}$ such that $\mathsf{xor}(H(\mathsf{IKey}(\mathcal{T}), \mathsf{Msg}(\mathcal{T}, C), A), N) = \mathsf{Input}(\mathcal{T})$ and $K_i = \mathsf{MKey}(\mathcal{T})$.

Now consider the following game G_2 that is equivalent of G_1, but adds some extra bookkeeping. In the first phase, we let $\mathsf{bad} \leftarrow \mathsf{false}$, and then run $\overline{\mathcal{A}}^{\textsc{New}, \textsc{Eval}, E, E^{-1}}$ as usual. After the adversary finishes querying and outputs its verification queries, let (i, N, C, A) be the j^*-th verification query. Excluding the $\mathtt{prim}$ entries created by $\textsc{Eval}(i, \cdot)$ queries, we check for compatibility of the remaining entries. If incompatibility happens we terminate the game and return false. Otherwise, among septets $\mathcal{T}$ such that $\mathsf{MKey}(\mathcal{T}) = K_i$, we check if $\mathsf{xor}(H(\mathsf{IKey}(\mathcal{T}), \mathsf{Msg}(\mathcal{T}, C), A), N) = \mathsf{Input}(\mathcal{T})$, and if this happens, we set bad to true. Note that here $\mathsf{OKey}(\mathcal{T}) \neq \mathsf{MKey}(\mathcal{T}) = K_i$, and thus the additional calls to $E(\mathsf{OKey}(\mathcal{T}), \cdot)$ to compute $\mathsf{Msg}(\mathcal{T}, C)$ will not create further incompatibility with the $\mathtt{prim}$ entries created by $\textsc{Eval}(i, \cdot)$ queries. In the second phase, we check the compatibility of all entries, program E, and proceed to handle the verification queries.

Note that in this case, the verification query (i, N, C, A) makes game G_1 return 1 only if game G_2 sets bad. Thus it suffices to prove that

$$\Pr[G_2 \text{ sets } \mathsf{bad}] \leq \frac{1}{2^n} + \frac{na \cdot \mathbf{E}\big[|C|_n + |A|_n\big]}{2^k} \quad .$$

Analyzing game G_2. Let game G_3 be identical to game G_2, but in G_3, we drop the second phase. Then

$$\Pr[G_3 \text{ sets } \mathsf{bad}] = \Pr[G_2 \text{ sets } \mathsf{bad}]$$

since the part we drop does not modify bad. Consider the following game G_4 that is identical to game G_3, except the following. In game G_3, recall that we check $\mathsf{xor}(H(\mathsf{IKey}(\mathcal{T}), \mathsf{Msg}(\mathcal{T}, C), A), N) = \mathsf{Input}(\mathcal{T})$ for only septets $\mathcal{T}$ such that $\mathsf{MKey}(\mathcal{T}) = K_i$. In G_4, we instead check $\mathsf{xor}(H(\mathsf{IKey}(\mathcal{T}), \mathsf{Msg}(\mathcal{T}, C), A), N) = \mathsf{Input}(\mathcal{T})$ for all septets, and then among septets $\mathcal{T}$ of affirmative answers, we set bad if there is some $\mathcal{T}$ such that $\mathsf{MKey}(\mathcal{T}) = K_i$. The two games are equivalent, and thus

$$\Pr[G_4 \text{ sets } \mathsf{bad}] = \Pr[G_3 \text{ sets } \mathsf{bad}]$$

Let Bad be the event that in game G_4, the adversary can find $na(|C|_n + |A|_n)$ or more septets of affirmative answers. Note that the septets and their checking are independent of the key K_i. Hence it suffices to prove that $\Pr[\mathsf{Bad}] \le 1/2^n$. The difficulty here is that the adversary can *adaptively* make (i, N, C, A) after seeing the queries and answers. This creates an issue in using the regularity of H in checking $\mathsf{xor}(H(\mathsf{IKey}(\mathcal{T}), \mathsf{Msg}(\mathcal{T}, C), A), N) = \mathsf{Input}(\mathcal{T})$, since (C, A, N) might depend on $\mathsf{IKey}(\mathcal{T})$. However, we claim that for any *fixed* choice (i^*, N^*, C^*, A^*),

$$\Pr\big[\mathsf{Bad} \cap \big((i, N, C, A) = (i^*, N^*, C^*, A^*)\big)\big] \le 2^{1-(3n\ell+2n)} \tag{12}$$

where $\ell = |C^*|_n + |A^*|_n \ge 2$. By union bound, the chance that Bad happens is at most

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

To justify Equation (12), fix such a value (i^*, N^*, C^*, A^*). Consider the following game G_5. Initially, we let $\mathsf{bad} \leftarrow \mathsf{false}$, and run $\overline{\mathcal{A}}^{\textsc{New}, \textsc{Eval}, E, E^{-1}}$ as usual. After the adversary finishes querying, we ignore its verification queries. Excluding $\mathtt{prim}$ entries created by $\textsc{Eval}(i, \cdot)$ queries, we check for compatibility of the remaining entries. If incompatibility happens, we terminate the game and return false. Otherwise, consider all septets $\mathcal{T}$ of $(\mathtt{prim}, K, \mathsf{pad}(N^*, 0), R_0, +), \ldots,$ $(\mathtt{prim}, K, \mathsf{pad}(N^*, 5), R_2, +), (\mathtt{prim}, J, U, T^*, -)$, with $J' \| J \leftarrow \mathsf{KD.Map}(R_0, \ldots, R_5)$ such that (1) either the first six entries belong to the ideal-cipher queries, or they are created by queries $\textsc{Eval}(j, \cdot)$, with $j \ne i$, and (2) $J \ne K$ and (3) $U \in 0\{0,1\}^{n-1} \backslash \{\mathsf{pad}(N^*, 0), \ldots, \mathsf{pad}(N^*, 5)\}$. Check

$$\mathsf{xor}(H(\mathsf{IKey}(\mathcal{T}), \mathsf{Msg}(\mathcal{T}, C^*), A^*), N^*) = \mathsf{Input}(\mathcal{T})$$

for all such septets $\mathcal{T}$, and if there are $na\ell$ or more septets of affirmative answers then we set bad to true. Then

$$\Pr\big[\mathsf{Bad} \cap \big((i, N, C, A) = (i^*, N^*, C^*, A^*)\big)\big] \le \Pr[G_5 \text{ sets } \mathsf{bad}] \ .$$

Consider the following game G_6. It is identical to game G_5, but we grant the adversary the keys K_j, for every $j \neq i^*$, at the beginning. This should only help the adversary. Then

$$\Pr[G_5 \text{ sets } \mathsf{bad}] \le \Pr[G_6 \text{ sets } \mathsf{bad}] \enspace .$$

Consider the following game G_7. It is identical to game G_6 but here we lazily implement π_j for every $j \neq i^*$, and also lazily implement E. That is, initially π_j is undefined on all inputs. Only when we need to compute $\pi_j(X)$ do we sample the output from a proper distribution. Likely, initially $E(\cdot,\cdot)$ is undefined on all inputs. Only when we need to compute $E(K,X)$ or $E^{-1}(K,Y)$ do we sample the outputs from proper distributions. Moreover, we eagerly do the compatibility checking and programming E. That is, each time the adversary makes an ideal-cipher query we immediately do the compatibility checking and terminate the game if incompatibility is detected. Likewise, each time the adversary makes a query to $\textsc{Eval}(j,)$ with $j \neq i^*$, we immediately do the compatibility checking, terminate the game if incompatibility is detected, and program E otherwise. Since G_7 is simply a different implementation of G_6,

$$\Pr[G_7 \text{ sets } \mathsf{bad}] = \Pr[G_6 \text{ sets } \mathsf{bad}] \enspace .$$

Next, in game G_7, for any $j \neq i^*$, querying $\textsc{Eval}(j, N')$ either causes premature termination without setting bad, or effectively calls $R_s \leftarrow E(K_j, \mathsf{pad}(N', s))$ for every $s \in \{0,\dots,5\}$, and returns $\mathsf{KD.Map}(R_0,\dots,R_5)$. Since the adversary knows the keys K_j, without loss of generality, assume that the adversary makes no evaluation query $\textsc{Eval}(j,\cdot)$ for any $j \neq i^*$, and makes at most $6Q$ ideal-cipher queries: whenever it is supposed to query $\textsc{Eval}(j, N')$, it will query $E(K_j, \mathsf{pad}(N', s))$ for every $s \in \{0,\dots,5\}$ instead. (If those queries are redundant then the adversary can reuse the past answers without querying E.) Now, we say that an entry $(\mathtt{prim}, J, U, T^*, -)$ is *bad* if it belongs to an affirmative septet. Consider the following game G_8. It is essentially the same as game G_7, but we set bad if there are at least $n\ell$ bad entries. Note that each entry $(\mathtt{prim}, J, U, T^*, -)$ can belong to at most a septets, due to the balls-into-bins assumption above. Hence

$$\Pr[G_7 \text{ sets } \mathsf{bad}] \le \Pr[G_8 \text{ sets } \mathsf{bad}] \enspace .$$

Game G_9 is essentially the same as game G_8, but at the beginning, we grant the adversary free queries $E(K, \mathsf{pad}(N^*, 0)), \dots, E(K, \mathsf{pad}(N^*, 5))$, for every $K \in \{0,1\}^k$, and then grant it free queries $E(K,X)$ for every $K \in \{0,1\}^k$ and $X \in 1\{0,1\}^{n-1}$. (However, if some query $E(K,X)$ repeats a prior query then we will not grant this query.) Note that our free queries may prohibit the adversary from making some backward queries as those now become redundant. However, the entries $(\mathtt{prim}, K, U, \cdot, -)$ corresponding to those backward queries are not bad, because those U's will belong to $1\{0,1\}^{n-1} \cup \{\mathsf{pad}(N^*, 0), \dots, \mathsf{pad}(N^*, 5)\}$. Hence

$$\Pr[G_8 \text{ sets } \mathsf{bad}] \le \Pr[G_9 \text{ sets } \mathsf{bad}] \enspace .$$

What is left is to analyze the chance that game G_9 sets bad.

Analyzing game G_9. Consider the following balls-into-bins game. For each 6-tuple of queries $E(K, \mathsf{pad}(N^*, 0)), \ldots, E(K, \mathsf{pad}(N^*, 5))$ with answer $R_0, \ldots, R_5$ respectively, we view them as throwing a ball into bin $Z[n+1:n+k]$, where $Z \leftarrow \mathsf{KD.Map}(R_0, \ldots, R_5)$. So totally we throw 2^k balls. Since $\mathsf{KD}[E]$ is 2-unpredictable, for the j-th throw, given the result of prior throws, the conditional probability that the j-th ball lands in any particular bin is at most 2^{1-k}. Assume that each bin contains at most $\lceil k\ell/2 \rceil \le n\ell$ balls, which happens with probability at least $1 - 2^{-(3\ell+2)k} \ge 1 - 2^{-(3\ell+2)n}$, according to Lemma 11. Now, partition the septets according to their backward queries. Each partition contains at most $n\ell$ septets, due to the balls-into-bins assumption above. Since there are at most p backward queries, there are at most p partitions.

Now, for each entry $(\mathtt{prim}, J, U, T^*, -)$, for it to be bad, at least one of the $n\ell$ septets in its partition must be an affirmative one. Given all prior $\mathtt{prim}$ entries, the random variable U has at least $2^{n-1} - 6Q - 6 \ge 2^{n-2}$ equally likely values, and thus the conditional probability that this entry is bad is at most $n\ell/2^{n-2}$. Hence the chance that there are $n\ell$ bad entries is at most

$$\binom{p}{n\ell}\Big(\frac{n\ell}{2^{n-2}}\Big)^{n\ell} \le \frac{p^{n\ell}}{(n\ell)!}\Big(\frac{n\ell}{2^{n-2}}\Big)^{n\ell} \le \frac{(n\ell/64)^{n\ell}}{(n\ell)!} \le \frac{(n\ell/64)^{n\ell}}{(n\ell/e)^{n\ell}} \le \frac{1}{16^{n\ell}} \le \frac{1}{2^{(3\ell+2)n}} \ ,$$

where the second equality is based on the hypothesis that $p \le 2^{n-8}$, the third inequality is due to the fact that $m! \ge (m/e)^m$ for any integer $m \ge 1$, and the last inequality is due to the fact that $\ell \ge 2$. Summing up,

$$\Pr[G_9 \text{ sets } \mathsf{bad}] \le 2^{1-(3\ell+2)n} \ .$$

This concludes the proof.

M Proof of Theorem 6

From Proposition 1, we can construct d-repeating adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ such that

$$\mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\overline{\mathsf{AE}},E}(\mathcal{A}) \le \mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_1) + \mathsf{Adv}^{\mathsf{mu\text{-}auth}}_{\overline{\mathsf{AE}},\Pi}(\mathcal{A}_2) + \frac{2q}{2^n} \ .$$

Moreover, any query of $\mathcal{A}_1$ or $\mathcal{A}_2$ is also a query of $\mathcal{A}$. In particular, $\mathcal{A}_1$ makes at most q encryption queries of total L blocks, and at most B blocks per (user, nonce) pair, and p ideal-cipher queries. Likewise, $\mathcal{A}_2$ makes at most q encryption/verification queries of total L blocks, and encryption queries of B blocks per (user, nonce) pair, and p-ideal cipher queries. Let $\mathsf{AE}^* = \mathsf{KtE}[\mathsf{KD}[k], \mathsf{AE}]$.

Privacy analysis. We first bound the privacy advantage of $\mathcal{A}_1$. From Lemma 6,

$$\mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_1) \le \mathsf{Adv}^{\mathsf{mu\text{-}priv}}_{\mathsf{AE}^*,E}(\mathcal{A}_1) + \frac{2}{2^{n/2}} + \frac{48(L+p)q + 36q^2}{2^{k+n}} + \frac{2a(L+p) + 2d(L+p+3q)}{2^k} \ .$$

Now, since $q \le L/2$ (as each query consist of at least two blocks, one from associated data, another from plaintext/ciphertext), the bound above can be simplified as

$$\mathsf{Adv}^{\text{mu-priv}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_1) \le \mathsf{Adv}^{\text{mu-priv}}_{\mathsf{AE}^*,E}(\mathcal{A}_1) + \frac{2}{2^{n/2}} + \frac{33L^2 + 24Lp}{2^{k+n}} + \frac{(2a+5d)L + (2a+2d)p}{2^k} \enspace .$$

Authenticity analysis. We next bound the authenticity advantage of $\mathcal{A}_2$. From Lemma 9, we can construct an adversary $\mathcal{A}_3$ such that

$$\begin{aligned}\mathsf{Adv}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_2) &\le \mathsf{Adv}^{\text{mu-auth}}_{\mathsf{AE}^*,E}(\mathcal{A}_3) + \frac{5}{2^{n/2}} + \frac{11q}{2^n} + \frac{336(L+p)q + 72q^2}{2^{n+k}} \\ &\quad + \frac{48c(L+p+q)L}{2^{n+k}} + \frac{(8a+7a^2+9d)q + (na+8a+8d)L + 8(a+d)p}{2^k} \enspace .\end{aligned}$$

Moreover, any query of $\mathcal{A}_3$ is also a query of $\mathcal{A}_2$. In particular, $\mathcal{A}_3$ makes at most q encryption/verification queries of total L blocks, and encryption queries of B blocks per (user, nonce) pair, and p-ideal cipher queries. Again, since $q \le L/2$, the bound above can be simplified as

$$\begin{aligned}\mathsf{Adv}^{\text{mu-auth}}_{\overline{\mathsf{AE}},E}(\mathcal{A}_2) &\le \mathsf{Adv}^{\text{mu-auth}}_{\mathsf{AE}^*,E}(\mathcal{A}_3) + \frac{5}{2^{n/2}} + \frac{(186+72c)L^2 + (168+48c)Lp}{2^{n+k}} \\ &\quad + \frac{11q}{2^n} + \frac{(12a+4a^2+13d+na)L + 8(a+d)p}{2^k} \enspace .\end{aligned}$$

Combining privacy and authenticity. From the analysis above, on the one hand,

$$\begin{aligned}\mathsf{Adv}^{\text{mu-mrae}}_{\overline{\mathsf{AE}},E}(\mathcal{A}) &\le \mathsf{Adv}^{\text{mu-priv}}_{\mathsf{AE}^*,E}(\mathcal{A}_1) + \mathsf{Adv}^{\text{mu-auth}}_{\mathsf{AE}^*,E}(\mathcal{A}_3) + \frac{7}{2^{n/2}} + \frac{13q}{2^n} \\ &\quad + \frac{(14a+4a^2+18d+na)L + 10(a+d)p}{2^k} \\ &\quad + \frac{(219+72c)L^2 + (192+48c)Lp}{2^{n+k}} \enspace .\end{aligned}$$

On the other hand, using Proposition 2, we can construct an adversary $\overline{\mathcal{A}}$ such that

$$\mathsf{Adv}^{\text{mu-priv}}_{\mathsf{AE}^*,E}(\mathcal{A}_1) + \mathsf{Adv}^{\text{mu-auth}}_{\mathsf{AE}^*,E}(\mathcal{A}_3) \le 3\,\mathsf{Adv}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\overline{\mathcal{A}}),$$

where the key-generation algorithm KeyGen is given in Fig. 7. Adversary $\overline{\mathcal{A}}$ makes at most q encryption/verification queries of total L blocks, and encryption queries of B blocks per *user*, and p-ideal cipher queries. Hence

$$\begin{aligned}\mathsf{Adv}^{\text{mu-mrae}}_{\overline{\mathsf{AE}},E}(\mathcal{A}) &\le 3\,\mathsf{Adv}^{\text{mu-mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\overline{\mathcal{A}}) + \frac{7}{2^{n/2}} + \frac{13q}{2^n} \\ &\quad + \frac{(14a+4a^2+18d+na)L + 10(a+d)p}{2^k} \\ &\quad + \frac{(219+72c)L^2 + (192+48c)Lp}{2^{n+k}} \enspace .\end{aligned}$$

Finally, using Theorem 4 with $\beta = 4$ if H is c-regular, or Theorem 5 with $\beta = 4$ if H is weakly c-regular

$$\mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\mathsf{AE},\mathsf{KeyGen},E}(\overline{\mathcal{A}}) \leq \frac{1}{2^{n/2}} + \frac{(4a+d)p + (2d+a)L}{2^k} + \frac{(12c+28)L^2 + 16cLp}{2^{n+k}} + \frac{(16c+8.5)LB}{2^n}$$

and note that $q \leq LB/4$,

$$\mathsf{Adv}^{\mathsf{mu\text{-}mrae}}_{\overline{\mathsf{AE}},E}(\mathcal{A}) \leq \frac{10}{2^{n/2}} + \frac{(17a + 4a^2 + 24d + na)L + (22a+13d)p}{2^k} + \frac{(48c+30)LB}{2^n} + \frac{(303+108c)L^2 + (192+96c)Lp}{2^{n+k}} \ .$$

This concludes the proof.