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 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. Our bounds also improve existing analyses in the single-user setting, in particular when messages of variable lengths are encrypted. We also validate security against a general class of key-derivation methods, including one that *halves* the complexity of the final proposal.
As an intermediate step, we consider 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 viewpoint generalizes the currently adopted one in mu security, and can be used to analyze re-keying practices.

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, 21, 22, 26, 29, 35], and its relevance stems from the *en masse* deployment of symmetric cryptography, 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 [18] scheme by Gueron, Langley, and Lindell, an AES-based scheme for *authenticated encryption with associated data* (AEAD) which

is meant to resist nonce misuse. 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 the 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 reuse 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 across multiple users *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 the key length or a worse security guarantee.

On the way, we analyze the building blocks of AES-GCM-SIV in a refined model of mu security where the amount of data processed by each user is bounded, and where keys come from arbitrary distributions. These results could be of independent interest.

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

MULTI-USER SECURITY. *Multi-user* (mu) security was introduced by Bellare, Boldyreva and Micali [3] in the public-key setting as an explicit security target, although in the symmetric setting the notion had already been targeted in attacks [10,11], and was used implicitly as a technical tool in [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's encryption queries $\textsc{Enc}(i, M)$ 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 the real from the ideal-world.

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. The advantage of any *single-user* adversary encrypting, in total, L blocks of data and making p queries to the cipher (which we model as ideal) is upper bounded by $\epsilon_{su}(L, p) \leq \frac{L^2}{2^n} + \frac{p}{2^k}$ (cf. e.g. [5]). If the attacker now adaptively distributes its queries across u users, a hybrid argument shows that the bound is $\epsilon_{mu}(L, p, u) \leq u \cdot \epsilon_{su}(L, p + L) \leq \frac{2uL^2}{2^n} + \frac{u(p+L)}{2^k}$.

Usually, we do not want to fix u, and allow the adversary to encrypt its budget of L blocks *adaptively* across as many users as it sees fit. In particular, the adversary could (1) query one message only with length L, or (2) query L messages with length 1, each to a different user. 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 [2,21,22,29,35] have shown that this is overly pessimistic, and the security loss can be much smaller; in fact, often $\epsilon_{mu}(L, p) \approx \epsilon_{su}(L, p)$ holds.

BOUNDING THE PER-USER DATA COMPLEXITY. Note that even if $\epsilon_{mu}(L,p) \approx \epsilon_{su}(L,p)$ above, the matching attack could be 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 would require a single honest user to willingly encrypt multiple exabytes of data, and there are many scenarios where we can easily enforce this not to happen. If we enforce a per-user upper bound B on the number of encrypted blocks, an L-block adversary would be forced to spread its effort across at least L/B users, and the advantage could become even *smaller*. Indeed, tightening existing bounds, we show below that for CTR, the advantage of such an attacker is at most

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

for some constant a. This bound shows that the fewer blocks we encrypt per user, the higher the security: Beyond-birthday security is possible, e.g., for $k = n = 128$ and $B = 2^{32}$, the bound is of the order $L/2^{96} + p/2^{128}$. Also, the bound is independent of the number of users, and in particular the role of off-line computation – captured here by p – is also independent of L. Note that most previous results on mu security target deterministic security games, such as PRFs/PRPs [2, 21, 22, 29, 35] or deterministic AE [9, 26], 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.

AES-GCM-SIV: OVERVIEW AND BOUNDS. The above 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. The fewer data we encrypt per key, the higher the security.

AES-GCM-SIV adapts the re-keying idea to the AEAD setting, making it in particular *nonce based* – i.e., to encrypt a message M with a nonce N, we use a key-derivation function (KDF) KD to derive a key $K_N \leftarrow \mathsf{KD}(K,N)$ from the master secret key K and the nonce N, and then encrypt the message M with the nonce N under the key K_N using a base AE scheme AE. Now, the keys K_N can be thought as belonging to different (virtual) users. Existing analyses [20,24] show indeed that, assuming KD is a good PRF, a mu security bound for AE can be lifted to 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. If nonces are not re-used, B is the maximum block length of an encrypted message.

Concretely, in AES-GCM-SIV, the underlying AE is GCM-SIV$^+$, a slight modification of GCM-SIV [19]. This relies in turn on SIV ("synthetic IV") [34], 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

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

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 [38] similar to, but different from, the one used in GCM [28]. It composes an xor-universal hash function with n-bit key, with a block cipher of 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 an appropriate domain separation is used.

For *nonce-misuse resistance* (so-called mrae security), the best published bound for AES-GCM-SIV with key length 128 bits is of order

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

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, $\epsilon(Q)$ is the PRF advantage of KD against Q queries, and it is $Q/2^{96}$ for the considered instantiation.

Our bounds in the mu setting. The analysis of AES-GCM-SIV *uses* mu security as a tool, but still only gives su security bounds. A valid question is whether its security substantially degrades in the mu setting or not.

We answer this question, and show that for a large class of suitable instantiations of KD, *multi*-user mrae security of AES-GCM-SIV is of order

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

where L, B, and d are upper bounds, respectively, of the overall number of encrypted/verified blocks, of the number of blocks encrypted per user-nonce pair, and of the number of users that re-use a particular nonce value.

This shows a number of 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 the previous works. (Note in particular that the PRF advantage term $\epsilon(Q)$ disappears from the bound.) 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 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 length 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.

The rest of the introduction will explain some ideas behind the bound and the techniques, which we believe to be more broadly applicable.

Challenges. On the way to our end result, we give a number of results of independent interest. Interestingly, while we will recycle ideas on the way, the approach is less modular than one expects. First off, we analyze CTR and GMAC$^+$ in a regime where the amount of data processed by each user is bounded. We will then obtain an analysis of the mu security GCM-SIV$^+$. Here, due to the key re-use, the technique for generic composition used in the original SIV scheme fails, but we will be able to recycle many low-level parts of the proofs for CTR and GMAC$^+$.

At this point, however, it is unclear 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 user-nonce 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 [20, 24] do, all security will vanish at this point. Indeed, the existing mu analysis of GCM succumbs to this problem [9], and the problem seems unavoidable here too, since we are considering a deterministic security game.

Bounded nonce re-use across users. The way out from this problem is to assume every 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 a nonce N, and re-use them over and over in encryption queries for different users, and if the same ciphertext appears twice after roughly $2^{k/2}$ queries, we are likely to be in the real world, as ciphertexts are random and independent in the ideal world. This however requires us to *re-use* the same nonce across $2^{k/2}$ users. A first interesting point we observe is that the security of KD as PRF degrades gracefully with the number of users d that can re-use the same input/nonce.

Unfortunately, this is not enough. The catch is that a bound d on the number of users re-using a nonce is only meaningful for encryption queries, e.g., if nonces are chosen randomly. For authenticity, an attacker would attempt to issue verification queries for as many users as it wishes, and we cannot restrict the choice of nonces. In particular, we cannot prevent that $2^{k/2}$ verification queries for different users with the same nonce may end up using colliding user keys. The question is how far this is an issue.

To get some intuition, consider the security of KD as a MAC, i.e., the adversary issues, in a first stage, queries (i, N), producing output $\mathsf{KD}(K_i, N)$ (where K_i is the key of the i-th user), 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. The caveat is that 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 do so for a specific class of natural KD constructions outlined below, and point out that the setting of AE is 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, while producing a bound which does not depend on key collisions. 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 [18] 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 [24] suggests using the so-called XOR construction to achieve higher security, as multiple analyses [7, 14, 25, 31, 33] confirm better bounds than for truncation [16]. Still, the resulting KD would need 4 resp. 6 calls. They also consider a faster construction, based on CENC [23], which would require 3 resp. 4 calls. All of these constructions are required to be good PRFs in existing analyses.

Rather than studying 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 $\mathsf{AES\text{-}GCM\text{-}SIV}$ proposal, but was discarded due to security concerns. 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 $\mathsf{AES\text{-}GCM\text{-}SIV}$, because the derived keys are not truly random. However, 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.[3]

In platforms that support AES hardware acceleration, the difference in performance between the KDF in Equation (1) and the current one in $\mathsf{AES\text{-}GCM\text{-}SIV}$

[3] This key-derivation scheme is also used to derive sub-keys from tweaks in the setting of FPE within the DFF construction [37]. DFF is a replacement for FF2 [36], a scheme proposed to NIST for standardization but eventually rejected due to a birthday-bound key-recovery attack [15]. The security of DFF is formalized and studied in [6], but their analysis is still in the su setting, namely there is only one master key for KD.

is not important, as demonstrated via the experiments in [18]. Still, we believe it is important for schemes to be minimal, and thus to understand the security of the simplest possible instantiations of the KDF.

Sub-optimality of POLYVAL. We also observe that the universal hash POLYVAL within GMAC$^+$ is somewhat suboptimal. 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, but substantially weakens the mu 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 ensured 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, this change becomes necessary if GCM-SIV$^+$ or GMAC$^+$ are used as standalone schemes.

Relation to existing works. We elaborate further on our improvements in the su setting over recent analyses [20, 24]. As mentioned 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 [20,24] 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 [20, 24], 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 is non-trivial: The adversary can adopt an arbitrarily adaptive attack pattern. Handling such adversaries was the object of recent works in the mu regime [2, 21, 22, 26, 29, 35].

Standard vs ideal-model. We also note that the bound of [24] 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 [24] 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 in local computation than obtaining 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 [26] on classical GCM also seemingly focuses on the standard model and thus seems to fail to capture such hidden terms. In contrast, [20] 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 the 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 $x \xleftarrow{\$} S$ 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. 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 [21] (which was in turn inspired by Maurer's framework [27]), 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 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 [13, 32]. 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. [13, 32] *Supposed we can partition transcripts into good and bad transcripts. Further, suppose that there exists* $\epsilon \geq 0$ *such that* $1 - \frac{\mathsf{p}_{\mathsf{S}_0}(\tau)}{\mathsf{p}_{\mathsf{S}_1}(\tau)} \leq \epsilon$ *for every good transcript* τ *such that* $\mathsf{p}_{\mathsf{S}_1}(\tau) > 0$*. Then,*

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

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 [30]. 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.[4] 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.

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

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

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

AUTHENTICATED ENCRYPTION SCHEME. An authenticated encryption scheme AE with associated data (also referred to as an AEAD scheme), the algorithms AE.E and AE.D are both deterministic. In particular, 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 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

[4] 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.

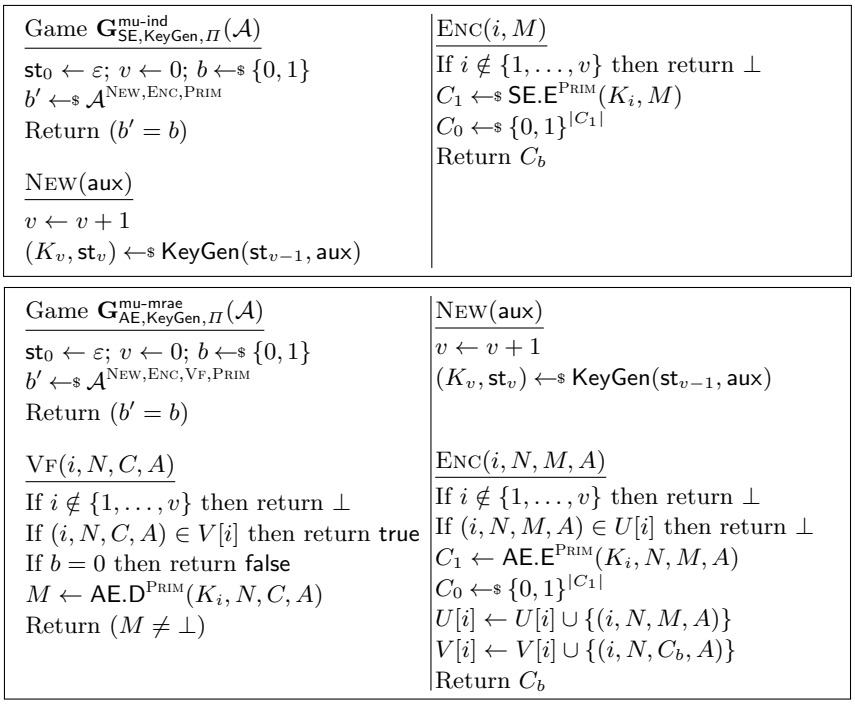

Fig. 1: **Security definitions for chosen-plaintext security of IV-based encryption (top), as well as nonce-misuse resistance for authenticated encryption (bottom).** We assume (without making this explicit) that Prim implements the ideal-primitive Π.

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 $\text{Enc}(i, N, M, A)$ 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, [17, 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 $\frac{q(q-1)}{4N}$.*

Game $\mathbf{G}^{\mathsf{mu\text{-}prf}}_{\mathsf{F},\mathsf{KeyGen},\Pi}(\mathcal{A})$	$\textsc{New}(\mathsf{aux})$	$\textsc{Eval}(i, M)$
$v \leftarrow 0$; $\mathsf{st}_0 \leftarrow \emptyset$	$v \leftarrow v + 1$	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 $(b' = b)$		Return Y_b

Fig. 2: **Definition of multi-user PRF security.** Again, PRIM implements the ideal primitive Π.

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 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. There, F's access to Π is modeled by having oracle access to PRIM. For any adversary $\mathcal{A}$, and key-generation algorithm KeyGen, we define

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

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

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

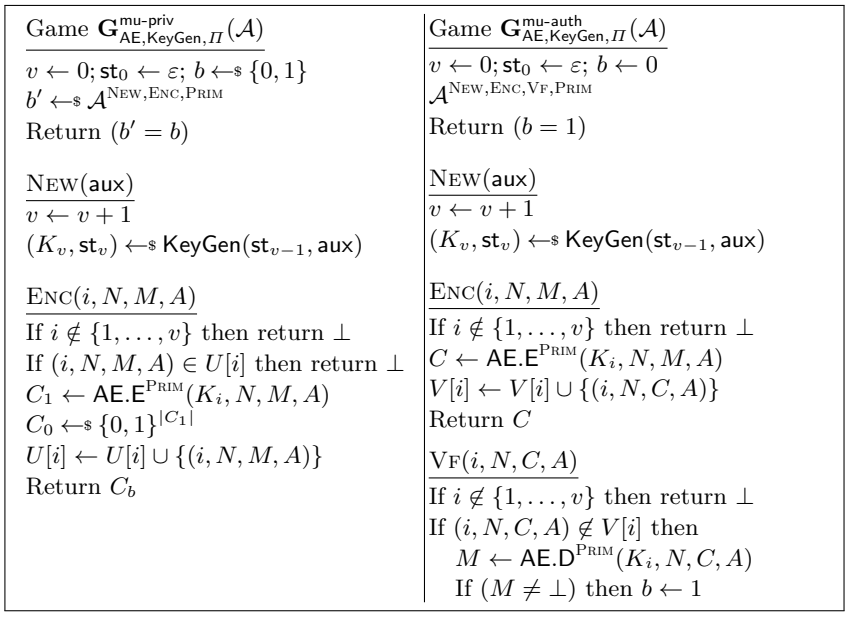

Fig. 3: **Games to define privacy(left), and authenticity (right) of an AE scheme AE with respect to a key-generation algorithm** $\mathsf{KeyGen} : \mathcal{K} \times \mathcal{N} \rightarrow \{0,1\}^{\mathsf{AE.kl}}$**.** The oracle Prim implements the ideal primitive Π. In the authenticity notion,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

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 oracle. We require that the verification queries be made *after* all the 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. The proof is in the full version of this paper [12].

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*

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

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 $\mathsf{AES\text{-}GCM\text{-}SIV}$. 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, $\mathsf{AES\text{-}GCM\text{-}SIV}$ 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 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 the full version of this paper [12] 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 [20,24] 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 the 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 [38], 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 the full version of this paper [12] 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

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

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

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

Mu-prf security of GMAC^+. We upper bound the mu-prf advantage for GMAC^+. 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 every 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 $\epsilon > 0$.

The proof of the following theorem is in the full version of this paper [12].

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 $k = n = 128$, $C = \beta = 1$, $B = 2^{32}$, $L = qB$, $q \leq 2^{95}$, 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 the full version of this paper [12].

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,*

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

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 $\mathsf{F.kl} \geq k$. 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 $K_{\mathsf{in}} \,\|\, K_{\mathsf{out}}$ 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 $\mathsf{GCM\text{-}SIV}^+$ 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 $\mathsf{xor}(X, N) \in 0\{0,1\}^{n-1}$

$\underline{\mathsf{AE.E}(K_{\mathsf{in}} \,\|\, K_{\mathsf{out}}, N, M, A)}$	$\underline{\mathsf{AE.D}(K_{\mathsf{in}} \,\|\, K_{\mathsf{out}}, N, C, A)}$
$\mathsf{IV} \leftarrow \mathsf{F}(K_{\mathsf{in}} \,\|\, K_{\mathsf{out}}, N, M, A)$	$\mathsf{IV} \,\|\, C' \leftarrow C;\ M \leftarrow \mathsf{SE.D}^{E}(K_{\mathsf{out}}, C)$
$C \leftarrow \mathsf{SE.E}^{E}(K_{\mathsf{out}}, M; \mathsf{IV})$	$T \leftarrow \mathsf{F}^{E}(K_{\mathsf{in}} \,\|\, K_{\mathsf{out}}, N, M, A)$
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.**

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 notion of pairwise AU for key-generation algorithms can be found in Section 4.2. See the full version of this paper [12] 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,*

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

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 $(\mathtt{prim}, K, X, Y, +)$. Likewise, for each query $E^{-1}(K, Y)$ for answer X, the transcript stores an entry $(\mathtt{prim}, K, X, Y, -)$. 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 $\mathsf{CTR}.\mathsf{E}[E](K_i, M; T)$ 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 the full version of this paper [12].

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* $L \leq 2^{(1-\epsilon)n-4}$, *and encryption queries of at most* B

blocks per user, and $p \le 2^{(1-\epsilon)n-4}$ *ideal-cipher queries,*

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

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 [20]. 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 $n \ge 1$ 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

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

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

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 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 [24] propose to use either the XOR construction [8,14] or the CENC construction [23]. Both the 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 $\Pr[\mathsf{KD.Map}(R_0, \ldots, R_5) = s] \leq \gamma / 2^{n+k}$. Lemma 3 below shows that both $\mathsf{KD}_0[E]$ and $\mathsf{KD}_1[E]$ are 2-unpredictable; see the full version of this paper [12] 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 $n \geq 128$ 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 $\mathrm{Perm}(n)$.

$\underline{\mathsf{KeyGen}(\mathsf{st}, \mathsf{aux})}$
$(N, i) \leftarrow \mathsf{aux}$; $(\pi_1, S_1, \ldots, \pi_m, S_m) \leftarrow \mathsf{st}$
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
  $K \xleftarrow{\$} \{0,1\}^{k+n}$; 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 $\pi_{m+1} \xleftarrow{\$} \mathrm{Perm}(n)$; $S_{m+1} \leftarrow \{N\}$; $\mathsf{st} \leftarrow (\pi_1, S_1, \ldots, \pi_{m+1}, S_{m+1})$
Return $(\mathsf{KD}[k](\pi_i, N), \mathsf{st})$

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

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 the full version of this paper [12]. 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 $\mathsf{KtE}[\mathsf{KD}[k], \mathsf{AE}]$.

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* 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 the full version of this paper [12].

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

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

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

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

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.

Lemma 5 below bounds the indistinguishability advantage between $\mathsf{KD}[E]$ and $\mathsf{KD}[k]$. The proof is in the full version of this paper [12].

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

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

Putting this all together,

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

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 the full version of this paper [12], 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*

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

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 $\mathsf{KD}[E]$. See the full version [12] for the proof. 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*

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

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 $K^j_{\mathsf{in}} \| K^j_{\mathsf{out}} \leftarrow \mathsf{KD.Map}(R_{0,j}, \ldots, R_{5,j})$. Now, if some K_j is the master key of some user i then $K^j_{\mathsf{in}} \,\|\, K^j_{\mathsf{out}}$ 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 $V_j \leftarrow E^{-1}(K^j_{\mathsf{out}}, T)$ 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 $\mathsf{xor}(H(K^j_{\mathsf{in}}, M_j, A^*), N) = V_j$. 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 $E^{-1}_K(Y)$ 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 $K_{\mathsf{in}} \,\|\, K_{\mathsf{out}}$ 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 $K_{\mathsf{in}} \,\|\, K_{\mathsf{out}}$ still has at least $k + n - 1$ bits of (conditional) min-entropy. In this case, the chance that $\mathsf{AE.D}(K_{\mathsf{in}} \,\|\, K_{\mathsf{out}}, N, C, M)$ 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,[5] 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 $M \leftarrow \mathsf{CTR}[E].\mathsf{D}(K_{\mathsf{out}}, C)$. This is highly non-trivial because somehow the adversary already sees the keys

[5] We note that this is not the classic balls-into-bins setting, because the balls are thrown in an inter-dependent way. In the full version [12], we give an analysis of this biased balls-into-bins setting.

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 $(\mathtt{prim}, K, \mathsf{pad}(N^*, 0), R_0^*, +), \ldots, (\mathtt{prim}, K, \mathsf{pad}(N^*, 5), R_5^*, +)$ and $(\mathtt{prim}, J, U, T^*, -)$, with $U \in 0\{0,1\}^{n-1}$ and $J' \parallel J \leftarrow \mathsf{KD}.\mathsf{Map}(R_0^*, \ldots, R_5^*)$. 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

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

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 Eval oracle, it is tempting to consider the variant G_1 of game G_0 where 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 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 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 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 the 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*

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

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 the full version of this paper [12]. 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* $p \leq 2^{(1-\epsilon)n-8}$ *ideal-cipher queries, and* $q \leq 2^{(1-\epsilon)n-8}$ *encryption/verification queries whose total block length is at most* $L \leq 2^{(1-\epsilon)n-8}$ *and encryption queries of at most* B *blocks per (user, nonce) pair. Then,*

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

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 the key length to 256 bits 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 Mihir Bellare, Shay Gueron, Yehuda Lindell, and anonymous CRYPTO reviewers 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*, pages 1515–1532, 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. E. Biham. How to forge DES-encrypted messages in 2^{28} steps. Technical Report CS0884, Technion - Israel Institute of Technology, 1996.
11. E. Biham. How to decrypt or even substitute DES-encrypted messages in 2^{28} steps. *Inf. Process. Lett.*, pages 117–124, 2002.
12. P. Bose, V. T. Hoang, and S. Tessaro. Revisiting AES-GCM-SIV: Multi-user security, faster key derivation, and better bounds. Cryptology ePrint Archive, Report 2018/136, 2018. https://eprint.iacr.org/2018/136.
13. 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.
14. 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.
15. M. Dworkin and R. Perlner. Analysis of VAES3 (FF2). Cryptology ePrint Archive, Report 2015/306, 2015. http://eprint.iacr.org/2015/306.
16. 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.
17. S. Goldwasser and M. Bellare. Lecture notes on cryptography. Summer Course "Cryptography and Computer Security" at MIT, 1999.
18. 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.
19. 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.
20. S. Gueron and Y. Lindell. Better bounds for block cipher modes of operation via nonce-based key derivation. In *CCS 2017*, pages 1019–1036, 2017.
21. 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.
22. 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.

23. 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.
24. T. Iwata and Y. Seurin. Reconsidering the security bound of AES-GCM-SIV. *IACR Trans. Symm. Cryptol.*, 2017(4):240–267, 2017.
25. 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.
26. A. Luykx, B. Mennink, and K. G. Paterson. Analyzing multi-key security degradation. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 575–605. Springer, Heidelberg, Dec. 2017.
27. 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.
28. 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.
29. 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.
30. 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.
31. J. Patarin. A proof of security in $O(2^n)$ 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.
32. 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.
33. 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.
34. 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.
35. 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.
36. J. Vance. VAES3 scheme for FFX: An addendum to "The FFX mode of operation for Format Preserving Encryption. Submission to NIST, May 2011.
37. J. Vance and M. Bellare. Delegatable Feistel-based Format Preserving Encryption mode. Submission to NIST, Nov 2015.
38. 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.