Blockcipher-Based Key Derivation without PRP/PRF Switching

Fabrice Benhamouda[1], Shai Halevi[1], Panos Kampanakis[1], and Hugo Krawczyk[1]

[1]AWS, USA

May 16, 2025

Abstract

We examine the use of blockcipher-based key derivation beyond the birthday bound, arguing that the analysis step of PRP/PRF switching can be eliminated in many cases. To support this, we consider a modified "ideal model" for keying cryptographic applications in the multi-instance setting, where keys are chosen to be random *but distinct*, rather than completely independent).

Our analysis shows that typical cryptographic applications remain secure in this model. One consequence is that it is typically safe to derive close to 2^n keys using an n-bit blockcipher in counter mode. In particular, considering the practice of nonce-derived keys for authenticated encryption, our results imply that modes such as XAES-256-GCM that use CMAC-based key derivation are safe to use with more than 2^{64} distinct nonces.

1 Introduction

Key derivation is the process of computing many cryptographic keys from one underlying secret.[1] We consider the practice of using an n-bit blockcipher with one main key to derive multiple cryptographic keys, that can then be used in other cryptographic constructions such as blockciphers or keyed hash functions. In particular, we are concerned with this practice when the number of derived keys is very large, beyond the so-called birthday-bound of $2^{n/2}$. Throughout this manuscript we refer to the scheme that would use the derived keys as the *target application*.

One complication that arises in the analysis of some blockcipher-based key derivation schemes beyond the birthday bound, is that the derived keys may fail to exhibit collisions in places where truly random keys would have them. Consider using an n-bit blockcipher with non-repeating nonces to derive multiple n-bit keys for the target application: We want to argue that if the target application itself is secure, and so is the blockcipher used for key derivation, then the resulting composed system is also secure. The standard method of reducing security of the composed scheme to that of the target application involves considering an "ideal model" where the keys for different instances of the target application are chosen uniformly at random and independently. In the real construction, on the other hand, the keys are not quite independent even if the underlying blockcipher is perfectly secure, since it is a permutation and therefore no two keys can ever be the same.

Below the birthday bound, such analysis would begin with a PRP/PRF switching step, considering a hybrid game in which the blockcipher is replaced by a random function (rather than a permutation). Once this step is done, the keys to the different instances of the target application become random and independent, and the rest of the argument follows. But PRP/PRF switching induces significant statistical difference once the number of keys approach the birthday bound of $2^{n/2}$. An analysis that includes this step therefore cannot attest to the security of the composed system beyond the birthday bound.

[1]Key derivation covers two independent aspects: *Extraction* refers to deriving a uniform cryptographic key from a high-entropy (but not uniform) secret, and *expansion* refers to converting a single uniform cryptographic key to multiple ones. In this note we are only concerned with expansion.

Taking a step back, our intuition tells us that the failure to show a reduction is an artifact of the analysis, rather than a real security issue with the composed scheme. If anything, the composed system seems *safer* than the "ideal" model with random and independent keys, exactly because in that model keys could collide by accident, whereas in the composed system they cannot. The PRP/PRF switching step seems only needed to introduce a defect that exists in the "ideal" model but not in the actual construction. In this work we therefore consider an alternative "ideal" model, in which keys were chosen uniformly at random *subject to being unique*. It follows by definition that key derivation using a secure blockcipher and non-repeating nonces emulates well this multi-instance, unique-key model, regardless of the number of derived keys: If the target application is secure in the multi-instance unique-key model, then we know that the composed scheme is secure also beyond the birthday bound.

A somewhat more complicated case involves target applications that require keys longer than the block size of the key-deriving cipher. In this scenario, the keys for the different instances are even more correlated, they are chosen (pseudo)randomly subject to the constraints that no two n-bit blocks in any of the keys are equal. It follows by definition that key derivation using a secure blockcipher and non-repeating nonces emulates well this multi-instance, unique-block model for any number of derived keys. Hence a secure target application in the multi-instance unique-block model is also secure when keys are derived as above, even beyond the birthday bound. This discussion motivates the following question, which is the focus of the current work:

What target applications are secure in the multi-instance unique-key/block models?

Jumping ahead, our results imply that nearly all target applications remain secure in these models.

1.1 A Motivating Application: XAES-256-GCM vs. DNDK

Our main motivation for the current study was our attempt to analyze the recent XAES-256-GCM mode of Valsorda [16]. This mode takes as input an AES-256 main key K and two 96-bit nonces U, V, uses CMAC-AES-256 with key K to derive from U another 256-bit data key K_U, then encrypts the message with AES-256-GCM keyed by K_U with nonce V.

One stated design goals of XAES-256-GCM is to be able to use a single main key K to encrypt up to 2^{80} different messages. But as far as we know, no formal analysis appears in the literature to show that it indeed meets that design goal. Perhaps one reason is that this scheme cannot be proven secure beyond the birthday bound of 2^{64} in the setting with independent keys, exactly because of the missing-collisions issue from above: If the nonces U are all distinct, then so are all the derived AES-256 keys K_U.

One way to resolve that issue is to re-design the scheme itself, replacing the PRP-based key derivation by an efficient PRF construction. This approach was taken for example in Gueron's DNDK construction [9], that has the same design goals as XAES but uses techniques from Bellare et al. and Iwata [2, 12] to get a PRF-based key derivation, hence enabling analysis in the setting with independent keys. But building a PRF just for the purpose of key derivation adds some complexity and cost, is this added complexity really needed? [2]

We remark that essentially the same question was raised by Bose et al. [6] in the context of AES-GCM-SIV: They pointed out that key-derivation in AES-GCM-SIV has some added complexity in order to make it a good PRF, and that their ideal-cipher analysis does not need this property and can therefore be used to prove security of this construction with cheaper key-derivation. The current work expands on that point, moving it to the standard model and considering general target applications.

1.2 Related Work

Multi-instance security. Starting with Bellare, Boldyreva, and Micali [1], there has been a lot of work concerning security of cryptographic primitives in the multi instance setting, studying how the security

[2] Very recently, Gueron and Ristenpart proposed a variant of DNDK with a more efficient key-derivation [10], and explicitly mentioned that analyzing XAES "would require a different formal analysis", since its key derivation is not a PRF. Our work provides such an analysis.

guarantees degrade as the number of instances grow. In the standard model where the keys to the different instances are independent, Bellare et al. described in [1] a simple reduction that proves linear degradation in the number of instances: An ϵ-successful attack in the u-instance setting implies an (ϵ/u)-successful attack in the single instance setting with comparable complexity.

Most later works used idealized models such as the random-permutation or ideal-cipher models, in order to get better bounds or analyze different aspects. Bellare and Tackmann [3] analyzed authenticated-encryption schemes such as GCM and variants thereof in the multi-instance setting, using the ideal-cipher model to argue about offline computation that the attacker may be able to perform. Luykx, Mennink, and Paterson [14] proved that for certain type of schemes (including GCM), the optimal strategy for an ideal-cipher-model adversary is to spend all their resources attacking just a single instance. The work of Luykx et al. assumed that different instances of the target application use independent permutations, hence they were already implicitly using the unique-key model.[3] Bose, Hoang, and Tessaro [6] used the ideal-cipher model to analyze multi-instance authenticated encryption, focusing on the effect of key collisions. They proved that some schemes (such as AES-GCM-SIV) remain secure even when key-collisions occur, as long as not too many messages are encrypted under the same key. They proved that even in the presence of accidental key-collisions, an attacker cannot do much better in the multi-instance setting than in the single-instance setting, essentially because the attacker cannot identify collisions seven if they occur.

PRF vs. PRP. Bellare, Goldreich, and Krawczyk [2] and Hall, Wagner, Kelsey, and Schneier [11] initiated the study of constructing PRFs from PRPs, showing that things like XORing two PRPs or truncating the output of a PRP yield functions that are close to PRFs. Some later studies of blockcipher-based constructions of PRF-like functions from block ciphers include Mennink-Nenes [15] and Chen, Lambooij, and Mennink [7].

Iwata [12] used the technique from [2] for a counter-mode-like encryption, showing how to use $q(N+1)$ PRP applications to encrypt q messages of N blocks each, with security bound $\approx qN^2/2^n$ (vs. $(qN)^2/2^n$ for standard counter-mode-based encryption with non-repeating nonces).

Bernstein [4] proved a tighter bound on PRP/PRF switching below the birthday bound, obtaining meaningful bounds all the way up to $2^{n/2}$ uses (but not much more). Specifically, when using PRP rather than PRF in unpredictability-type applications, Bernstein proved a multiplicative bound of $1/(1-\frac{q-1}{2^n})^{q/2}$ when making q queries, which is sometimes much better than the naïve additive bound of $\binom{q}{2}/2^n$. Combining Bernstein's observation with the standard single-to-multi reduction in the independent-key model (that has a factor of q) yields a single-to-multi reduction in the unique-block model with security degradation bound of $q/(1-\frac{q-1}{2^n})^{q/2}$. See some more discussion in Theorem 1.

Lee, Luykx, Mennink, and Minematsu [13] investigated using key-derivation in settings when the target application is itself a blockcipher, and in some cases (such as iterated Even-Mansour) had shown that the key-derivation need not be a PRF.

1.3 Our Contribution

Our focus in this work is on standard-model analysis of the use of PRPs beyond the birthday bound. Our (simple) observation is that the standard single-to-multi-instance security reduction can be applied also in the unique-key/block models whenever the target application is either (a) an unpredictability-type application, or (b) an indistinguishability-type application in any setting where it is possible to check for key/block collisions.

For unpredictability-type applications, this means that we have a single-to-multi reduction in the unique-block model with security degradation bound very close to q, nearly all the way to $q = 2^n$. The same holds also for indistinguishability-type applications whenever is possible to check for key/block collisions, which includes the following settings:

- When the target application itself enables checking whether a candidate key/block is indeed used in a specific application instance;[4]

[3] Ideal-cipher analysis is not sensitive to the distinction between unique keys and unique blocks.

[4] This is common for full keys, but not for blocks.

- When the single-instance application remains secure in the presence of a key/block-collision oracle (see Section 2.3 for details).

Morally, these results say that most target applications are as secure in the unique-key/block model as in the model with independent keys. In some cases, this conclusion relies on a stronger hardness assumption on the target application, but even in those cases the strengthening of the assumption seems very mild.

We note that security of indistinguishability-type applications in the presence of a collision oracle is *sufficient* for them to remain secure in the unique key/block models, but we do not believe it to be a *necessary* condition, see some discussion in Section 3.2.2. On the other hand, not *every* indistinguishability application remains secure in the multi-instance unique-key/block models. For example it is easy to see that using the derived keys for one-time-pad encryption fails beyond the birthday bound, for example it becomes possible to check whether or not the encrypted blocks are all the same. Indeed, one-time-pad is an indistinguishability-type application that inherently does not allow checking for key collisions, and it is insecure in the presence of a key/block-collision oracle.

1.4 Organization

In Section 2 we present our notations and basic definitions, formalizing what it means to be an application that uses cryptographic keys and how the security of such applications is defined in the multi-instance setting. Then in Section 3 we describe our main security reductions for both unpredictability and indistinguishability applications. In Section 4 we offer some conclusions and revisit XAES-256-GCM. In Appendix A we describe some unsuccessful attempts to extend the reductions from Section 3.

2 Preliminaries

In this manuscript we consider game-based security definitions, which are probabilistic games between the application and an adversary A. We use the term target-application to refer both to the game itself and to the scheme that implements the interfaces available to the adversary. When the distinction is not clear from the context, we will explicitly talk about a game or a scheme to make it clear.

Throughout the writeup we denote by n the block-size used in key derivation (typically $n = 128$), and k is the key-size of the target application, which we assume is a multiple of n (typically either $k = n$ or $k = 2n$).

2.1 Target Applications and their Security Games

A single instance of the target application takes as input a random k-bit key denoted K, and provides some interfaces that depend on K (such as encrypting/decrypting using that key). The game begins with a uniform choice of a key $K \in \{0,1\}^k$. Then A is run on input 1^k and is given access to oracles that implement the target-application interfaces, computing key-dependent (and possibly randomized and/or stateful) functions. A runs for a while, accessing these interfaces as it wishes, and eventually halts with some output X. The *transcript* of this run (denoted Tr) consists of A's inputs, the queries that it makes to the oracles, and the answers to these queries.

The security game comes with a notion of what is means for the adversary to *win*, and the notion of security relates the win probability of an adversary to some resources that they have (such as the number of queries and their length). For presentation convenience, we often suppress below the mention of those resources, instead just talking about adversaries "from some class."

Unpredictability vs. Indistinguishability. Security games typically come in two varieties, depending on how the adversarial advantage is defined. Unpredictability refers to a game where the adversary is expected to almost never win (such as message authentication), whereas indistinguishability refers to games where it should be hard for the adversary to distinguish between two options (such as CPA security).

Definition 1 (Unpredictability). *An unpredictability application comes with a* win *predicate, applied to the key, transcript, and the adversary's output, where* $\mathsf{win}(K, \mathsf{Tr}, X) \in \{0,1\}$ *determines whether the adversary won that specific run. The advantage of the adversary in a single-instance unpredictability game is defined as its win probability,* $\mathsf{adv}^{unp}_A = \Pr[\mathsf{win}(K, \mathsf{Tr}, X) = 1]$*, where the probability is taken over all the randomness in the game, including the initial uniform choice of the key* K*.*

A scheme enjoys security up to some bound ϵ *in the single-instance unpredictability setting against some class of adversaries, if for every adversary* A *in that class it holds that* $\mathsf{adv}^{unp}_A \leq \epsilon$*.*

Definition 2 (Indistinguishability). *In an indistinguishability game, a random bit* $B \leftarrow \{0,1\}$ *is chosen at the onset of the game together with the key* K *and made available to the target-application interfaces. The adversary* A *outputs a single bit* $X \in \{0,1\}$*, and the advantage is defined as* $\mathsf{adv}^{ind}_A = \Pr[X = 1|B = 1] - \Pr[X = 1|B = 0]$*. The probabilities are taken over all the randomness in the game, including the choice of* K *(and conditioned on the choice of* B*).*

A scheme enjoys security up to some bound ϵ *in the single-instance indistinguishability setting against some class of adversaries, if for every adversary* A *in that class it holds that* $\mathsf{adv}^{ind}_A \leq \epsilon$*.*

Within some real-world context, a scheme is considered "secure" if it enjoys security against a relevant class of adversaries up to a small enough ϵ. For example, the GCM specification in NIST SP800-38D [8] spells out $\epsilon = 2^{-32}$ (at least against certain classes of attacks). It also mandates certain bounds to limit the resources that are available to the (single-instance) adversary, such as not encrypting more than 2^{32} messages per key or not encrypting messages longer than $2^{36} - 32$ bytes.

2.2 Multi-Instance Security

Multi-instance security means that an adversary that interacts with many instances of the scheme, each with its own key, is not able to "break" any of these instances except with a small enough probability.

For a parameter q, we consider an adversary that can make up to q queries to a special *new-instance* oracle to initiate a new instance of the scheme. The i'th new-instance query chooses a new random key $K_i \in \{0,1\}^k$ according to some distribution (see below). Thereafter, the adversary can access a new set of oracles, implementing all the interfaces of the single-instance game with respect to this new instance with key K_i. A multi-instance adversary can interact with all those oracles, and we denote the sequence of q keys in a multi-instance run by $\vec{K}$ and the multi-instance trace by $\vec{\mathsf{Tr}}$. For $i \leq q$, Tr_i is the portion of the transcript of the multi-instance adversary consisting of all the queries and answers from the i'th-instance oracles.

Independent keys, unique keys, or unique blocks. The focus of the current work is on cases where keys to the different instances are not chosen independently, but rather when keys (or blocks within them) are guaranteed to be unique. Fixing the block-size n and key-size k and denoting $b = k/n$ (which is assumed to be a small integer), we let K_i denote the i'th k-bit key in $\vec{K}$ and $K_i[j]$ the j'th n-bit block in K_i ($1 \leq i \leq q$ and $1 \leq j \leq b$). We consider the following distributions over the choice of keys:

IK. Independent keys, the uniform distribution over $\{0,1\}^{k \cdot q}$;

UK. Unique keys, the uniform distribution over $\vec{K} \in \{0,1\}^{k \cdot q}$, subject to $K_i \neq K_{i'}$ for all $i \neq i'$;

UB. Unique blocks, the uniform distribution over $\vec{K} \in \{0,1\}^{k \cdot q}$, subject to $K_i[j] \neq K_{i'}[j']$ for all $(i,j) \neq (i',j')$.

Note that with our formulation, unique keys are a special case of unique blocks for $k = n$ (i.e. $b = 1$). Hence we only formulate the definitions below for unique blocks, not unique keys. We are interested in the advantage of a multi-instance adversary when keys are chosen from these distributions. The advantage is again defined differently for unpredictability and indistinguishability applications.

Definition 3 (Multi-instance unpredictability)**.** *In a multi-instance unpredictability game, the adversary outputs* (i, X)*, and it wins if* X *is a winning output relative to the* i*'th instance alone,* $\mathsf{win}^q\big(\vec{K}, \vec{\mathsf{Tr}}, (i, X)\big) := \mathsf{win}(K_i, \mathsf{Tr}_i, X)$*. As before, the advantage is just the win probability, relative to the appropriate key distribution:* $\mathsf{adv}_A^{unp,\mathcal{Y}} = \Pr_{\mathcal{Y}}[\mathsf{win}^q(\vec{K}, \vec{\mathsf{Tr}}, X) = 1]$ *for* $\mathcal{Y} \in \{\mathbf{IK}, \mathbf{UB}\}$*.*

Definition 4 (Multi-instance indistinguishability)**.** *In a multi-instance indistinguishability game, a single "global" bit* $B \leftarrow \{0,1\}$ *is chosen at the onset and all the instances use the same bit.*[5] *As in Definition 2, the adversary outputs a bit* $X \in \{0,1\}$*, and it wins if* $X = B$*. The advantage relative to the appropriate key distribution is* $\mathsf{adv}_A^{ind,\mathcal{Y}} = \Pr_{\mathcal{Y}}[X = 1|B = 1] - \Pr_{\mathcal{Y}}[X = 1|B = 0]$ *for* $\mathcal{Y} \in \{\mathbf{IK}, \mathbf{UB}\}$*.*

We note that the probability distributions in the definitions above assume that different instances are independent of each other, except for the keys that are not quite independent and the global bit for indistinguishability applications.

A scheme is said to enjoy multi-instance security up to some bound ϵ in the relevant setting against some class of adversaries, if for every adversary A in that class it holds that $\mathsf{adv}_A^{*,*} \leq \epsilon$.

2.2.1 Models for Adversarial Resources

There are two common models in the literature for accounting for the resources of the adversary in multi-instance games: Either the adversary is assumed to have a total budget of queries/bits/etc. that it can spread across the different instances however it wants, or it gets a per-instance budget. Both cost models can be justified, depending on what we assume on the resource allocation among legitimate users of the target application:

- The total-budget setting is a worst-case model, that should be used when the scheme is deployed in unrestricted environments, where even the resource allocation among legitimate users could potentially be adversarial. For example if an encryption scheme is deployed in a system without any key-rotation or usage restrictions, then an attacker may have a free choice between focusing all their resources on just one instance or spreading them across many instances, whichever works best for the attack.

- The per-instance-budget reflects settings where the scheme it to be deployed in a system that can be relied upon to limit the resources of each instance. For example, an encryption standard may include a per-key usage restriction such as no more than N encrypted bits per key. An attacker that interacts with such a system may be able to encrypt many more than N bits, but it will not be able to use just one key to encrypt all of them, since standard-compliant systems ensure that keys are rotated out before encrypting N bits.

Results in the standard model (including this work) typically do not depend on the cost model, they just show that a multi-instance adversary with some resources can be converted to a single-instance adversary with similar resources. But the implications of these results for practical deployment depend on the cost-model for those deployments. While it may be possible to enforce per-instance usage restrictions, it seems very hard (or impossible) to enforce an overall upper bound on the resources available to the attackers.

2.3 Collision Oracles

As we discussed in the introduction, we can show that single-instance security of indistinguishability-type applications implies multi-instance security in the unique-key/block setting in settings that allow to identify key/block collisions. Namely, if it is possible to test if some key candidate is used by an instance of the application. We formalize this by considering a stronger notion of security for the single-instance game, giving the adversary extra power by letting them access an oracle that answers *a single query* about key (or block) collisions.

[5] One could also consider a per-instance bit and ask that the adversary outputs (i, X) where X is a guess for the bit of the i'th instance. We use the global-bit definition since it is more popular in the literature and generally a stronger requirement.

Specifically, we consider a single-instance target application with key $K^* \in \{0,1\}^k$, and add to it an oracle which is willing to answer a single query, consisting of a set of n-bit blocks $\vec{L}$. The oracle answers whether or not any block in K^* appears in $\vec{L}$. We denote that oracle by $O^{\mathsf{BC}}_{K^*}(\cdot)$ (BC for Block-collisions) (and stress again that this is only a one-time oracle, returning a single bit).

We denote the adversarial advantage in a *single-instance indistinguishability game* in the presence of the oracle $O^{\mathsf{BC}}_{K^*}$ by $\mathsf{adv}^{\mathsf{BC},ind}_A$. The security notion that we consider asserts that the adversary still has small enough advantage in the single-instance game, even given access to a collision oracle. We informally refer to this as "enhanced" single-instance security.

Is "Enhanced" Security a Stronger Assumption? There are certain cases in which it is easy to see that "enhanced" security is not any stronger than just assuming security of the target application in the single-instance setting. First, the oracle $O^{\mathbf{UB}}_{K^*}$ returns just a single bit, so it is clearly useless in recovering the underlying key K^*, or winning any unpredictability game.

Second, most applications enable checking whether a key-candidate is indeed the key used by the construction, rendering the key-collision oracle redundant for the unique-key setting $k = n$. For example, in an encryption application, one can use the application interfaces to obtain an encryption C^* of some (long enough) message M^* under the key K^*. To determine if a candidate key K is equal to K^*, one simply tries to decrypt C^* under K and checks if they get the same M^*. On the other hand, in the unique-block setting where $n < k$, the target applications do not usually enable this check.

Even when not implied by standard security, for most applications it seems hard to imagine a constructive use of collision oracles in any real attack. As one concrete example, consider the case of target applications based on AES-256, with $k = 256$ and $n = 128$. It may be the case that guessing 128 bits of the 256-bit AES key would make some attacks on AES-256 much more efficient, thereby making it possible to implement the block-collision oracle somewhat efficiently (hence making that oracle somewhat redundant). For example, some non-trivial related-key attacks are known against AES-256 [5], and it may be the case that these attacks become more efficient given a guess of one of the two key blocks. If indeed those attacks are simpler given half the key, then in applications where related-key attacks are relevant this would give an attacker an efficient method of checking if a candidate block is part of the key: Simply try to run the related-key attack and check if it succeeds.

We believe that assuming "enhanced" security of the target application in the single-instance setting is a very mild assumption to make, and can be safely used whenever there is any advantage to using it.

3 The Main Reductions

We now present and prove our main technical results, reducing the multi-instance security of applications within the unique-key/block models to their single-instance security. Recall that we denote by n the block-size, by k the key-size of the target application (which in most applications is either n or $2n$), and by $b = k/n$ the number of blocks in a key of the target application.

3.1 Unpredictability Target Applications

Theorem 1. *Fix any n, b, $k = b \cdot n$, and $q < 2^n/b^2$, and consider an unpredictability target applications with k-bit keys, that enjoys security up to some bound ϵ in the single-instance setting against some class of attackers. Namely, for any adversary A in that class, it holds that $\mathsf{adv}^{unp}_A \leq \epsilon$.*

Then for any q-instance adversary A' from the same class, the advantage of A' in the unique-block model is bounded by $\mathsf{adv}^{unp,\mathbf{UB}}_{A'} \leq \left(\epsilon + \frac{b(b-1)}{2^n}\right) \cdot q/(1 - \frac{b^2(q-1)}{2^n})$.

Before proving this theorem we note a few points about it:

- The multi-instance A' being "from the same class" in the theorem above means that every q-instance A' can be transformed to a single-instance A that never makes more queries to its oracles than A' does.

Moreover, A never makes more queries to its oracles than A' does *to any sof the instances separately*, and the expected number of oracle queries made by A is a $\frac{1}{q}$-fraction of the total queries of A'. (The single-instance A also does little local computation, just implementing for A' some of its oracles.)

- Combining the standard multi-to-single reduction in the independent-key model with Bernstein's bound from [4, Thm 2.3] yields a similar statement to above, but with the factor $1/(1-\frac{b^2(q-1)}{2^n})$ replaced by $1/(1-\frac{bq-1}{2^n})^{bq/2}$. While the factor in Theorem 1 remains close to one nearly all the way to $q = 2^n/b^2$, the factor from the combination of the standard reduction with [4] is only meaningful below the birthday bound.

  For example, in the parameter setting that we care about for XAES, $n = 128, q = 2^{80}, b = 2$, our factor is $1/(1-\frac{4(2^{80}-1)}{2^{128}}) \approx 1 + 2^{-46}$ while the factor from that combination is $1/(1-\frac{2^{81}-1}{2^{128}})^{2^{80}} \approx e^{2^{33}}$.

Proof. We describe a simple reduction that turns a q-instance adversary A' with advantage $\mathsf{adv}^{unp,\mathbf{UB}}_{A'} = \alpha$ into a single-instance adversary A with advantage $\mathsf{adv}^{unp}_{A} \geq (1-\frac{b^2 q}{2^n}) \cdot \frac{\alpha}{q} - \frac{b(b-1)}{2^n}$.

The reduction A is essentially just the standard hybrid argument. It begins by choosing at random an index $j \in [1, q]$, then choosing the secret keys for all the instances except the j'th one, uniformly at random *subject to having all their blocks unique*. The reduction A runs the q-instance adversary A', sending queries to instance j to its (single-instance) oracle, and answering all other queries by itself using the keys that it knows. If the q-instance oracle halts with output (j, X) (for the index j chosen by the reduction), then A outputs the same X. Otherwise A outputs $\perp$.

If there are no collisions, then the view of the q-instance adversary in the reduction is nearly identical to its view in the unique-block model, except the oracle key that was chosen at random rather than with unique blocks. The probability that the oracle key does not have unique blocks is at most $b(b-1)/2^n$ by a classical birthday bound.

Let red be the view of the q-instance adversary A' in the reduction, and red' be its view in the modified game where the oracle's key is chosen with unique blocks.

Denote the keys that A chose by $\vec{K} = (K_1, \ldots, K_{j-1}, K_{j+1}, \ldots, K_q)$, denote the oracle's key by K_j, and let col denote the block-collision event, where K_j and $\vec{K}$ have some blocks in common.

In red', each block of K_j is chosen uniformly at random (even if the distributions of each block are not independent). Thus, the probability that at least one of these b blocks collides with the $b(q-1)$ blocks of the other $q-1$ keys is $\Pr_{\mathsf{red}'}[\mathsf{col}] \leq b^2(q-1)/2^n$, by union bound over the b blocks.

Finally, we remark that in the game red' conditioned on $\overline{\mathsf{col}}$ (i.e. the event where there are no collisions), the index j that the reduction chose is independent of the view of the adversary A'. Finally, we remark that in the game red' conditioned on $\overline{\mathsf{col}}$ (i.e. the event where there are no collisions), the index j that the reduction chose is independent of the view of the adversary A'.

Hence we can bound the reduction's win probability by:

$$
\begin{aligned}
\mathsf{adv}^{unp}_{A} &= \Pr_{\mathsf{red}}[A \text{ outputs } (i,X) \text{ s.t. } i = j \;\&\; \mathsf{win}(K_i, \mathsf{Tr}_i, X)] \\
&\geq \Pr_{\mathsf{red}'}[A \text{ outputs } (i,X) \text{ s.t. } i = j \;\&\; \mathsf{win}(K_i, \mathsf{Tr}_i, X)] - \frac{b(b-1)}{2^n} \\
&\geq \Pr_{\mathsf{red}'}\left[A \text{ outputs } (i,X) \text{ s.t. } i = j \;\&\; \mathsf{win}(K_i, \mathsf{Tr}_i, X) \;\&\; \overline{\mathsf{col}}\;\right] - \frac{b(b-1)}{2^n} \\
&= \left(1 - \frac{b^2(q-1)}{2^n}\right) \cdot \Pr_{\mathsf{red}'}\left[A \text{ outputs } (i,X) \text{ s.t. } \mathsf{win}(K_i, \mathsf{Tr}_i, X) \mid \overline{\mathsf{col}}\;\right] \cdot \frac{1}{q} - \frac{b(b-1)}{2^n} \\
&\geq \left(1 - \frac{b^2(q-1)}{2^n}\right) \cdot \frac{\alpha}{q} - \frac{b(b-1)}{2^n}.
\end{aligned}
$$

We have $(1-\frac{b^2(q-1)}{2^b}) \cdot \frac{\alpha}{q} - \frac{b(b-1)}{2^n} \leq \mathsf{adv}^{unp}_{A} \leq \epsilon$ due to the assumed security of the target application in the single-instance setting, and therefore $\alpha \leq \left(\epsilon + \frac{b(b-1)}{2^n}\right) \cdot q/(1-\frac{b^2(q-1)}{2^n})$. □

For the unique-key setting we have $b = 1$ so we get a bound of $\epsilon \cdot q/(1 - \frac{q-1}{2^n})$.

These expressions imply that the advantage of the q-instance adversary is essentially q times that of the single-instance adversary, almost all the way up to $bq = 2^n$. Consider for example the unique-block case with $n = 128$-bit blocks and $k = 256$-bit keys (so $b = 2$), with $q = 2^{125}$ keys, way above the birthday bound for 128-bit blocks. Plugging in the numbers we get $1/(1 - \frac{b^2(q-1)}{2^n}) \approx 1/(1 - \frac{2^2 \cdot 2^{125}}{2^{128}}) = 2$, so the q-instance adversary has at most $2q\times$ larger advantage than the single-instance adversary, still essentially linear in q.

3.2 Indistinguishability Target Applications

The reduction of Theorem 1 does not quite work for indistinguishability target applications, since the low-probability event where collisions occurs may introduce a large enough disadvantage to offset the advantage of α/q that the reduction gets when there are no collisions.

To see the issue, consider for example a q-instance adversary A' that has advantage α in the unique-key setting, but such that if there are collisions then A' always targets one of the colliding key instances and always outputs the wrong guess. Applying the reduction from above to this q-instance adversary A', the advantage of the resulting single-instance adversary A would be roughly $(1 - \frac{q-1}{2^n}) \cdot \frac{\alpha}{q} - \frac{q-1}{2^n}$, which could even be negative for q beyond the birthday bound.

This difficulty seems an artifact of our analysis method rather than a real issue with the construction, as such a problematic q-instance adversary A' seems quite unlikely: Either an adversary can tell whether or not there are collisions between the keys that they have access to (in which case the reduction should be able to do the same and eliminate these collisions), or they cannot tell (in which case they would have the same advantage whether or not there are collisions and the issue above does not arise). Below we discuss possible approaches to bypass this technical difficulty.

3.2.1 Relying on "enhanced" security

To overcome the technical issue from above, we would like the reduction to be able to identify the rare collision event, so it can output some default value if a collision occurs and only use the output of A' when there are no collisions. The most straightforward way of identifying collisions is using the collision oracles from Section 2.3. Namely, we can rely on enhanced single-instance security that holds even in the presence of these oracles and use $\mathsf{adv}^{\mathsf{BC},ind}_A$ rather than just adv^{ind}_A. As we discussed in Section 2.3, such enhanced security is sometimes equivalent to the standard notion of security, and it seems a rather safe assumption to make even if not.

Theorem 2. *Fix any n, b, $k = bn$, and $q < 2^n/b$, and consider an indistinguishability target applications with k-bit keys. Assume that the application enjoys enhanced security up to some bound ϵ in the single-instance setting against some class of attackers, even in the presence of a block-collision oracle $O^{\mathsf{BC}}_{K^*}(\cdot)$. Namely, for any adversary A in that class it holds that $\mathsf{adv}^{\mathsf{BC},ind}_A \leq \epsilon$.*

Then for any q-instance adversary A' from the same class, it holds that $\mathsf{adv}^{unp,\mathbf{UB}}_{A'} \leq \big(\epsilon + \frac{2b(b-1)}{2^n}\big) \cdot q/(1 - \frac{b^2(q-1)}{2^n})$ in the unique-block model.

Proof. Let A' be a q-instance adversary with advantage $\mathsf{adv}^{ind,\mathbf{UB}}_{A'} = \alpha$ in the unique-block model. We describe a reduction that turns A' into a single-instance adversary A with advantage $\mathsf{adv}^{\mathsf{BC},ind}_A \geq (1 - \frac{b^2(q-1)}{2^n}) \cdot \frac{\alpha}{q} - \frac{2b(b-1)}{2^n}$.

The reduction is similar to above, except that it needs to detect collisions and it needs to handle the global bit B. A begins by choosing at random an index $j \in [1, q]$, then choosing the secret keys for all the instances except the j'th one, uniformly at random *subject to having unique blocks*. As before, we denote the keys that A chooses by $\vec{K} = (K_1, \ldots, K_{j-1}, K_{j+1}, \ldots, K_q)$, the oracle key by K_j, and the block-collision event by col. Then A calls to its key-collision oracle $O^{\mathsf{BC}}_{K_j}(\vec{K})$ to detect the col event, and if the oracle indicates a collision then A outputs 0 and halts.

Otherwise, A runs the q-instance adversary A', sending queries to instance j to its (single-instance) oracle, and answering other queries by itself. To answer these queries, A uses the keys $\vec{K}$ that it knows, and it uses the bit $B = 0$ in all the queries to instances 1 through $j-1$ and the bit $B = 1$ in all the instances $j+1$ through q. (Of course, A does not know if the oracles that implement the j'th instance uses $B_j = 0$ or $B_j = 1$, this is what A needs to find out.) When the q-instance adversary A' halts with some output $X \in \{0,1\}$, then A outputs the same X.

As before, let $\mathsf{red}, \mathsf{red}'$ be the view of the A' when the oracle key K_j (consisting of b blocks) is chosen uniformly at random or with unique blocks, respectively. Namely, red' is just red conditioned on no collision inside K_j, an event with probability at least $1 - b(b-1)/2^n$. For the distribution red' where the oracle's key is chosen with unique blocks, we have $\Pr_{\mathsf{red}'}[\overline{\mathsf{col}}] \geq 1 - \frac{b^2(q-1)}{2^n}$, in which case the keys in the reduction are chosen from the exact same distribution as in the unique-block model.

We now consider the hybrid games $H_0, H_1, \ldots, H_q$, all involving the q-instance adversary A' with unique-block keys. In the j'th hybrid H_j all the instances upto and including K_j use the bit $B = 0$ and all the instances from $j+1$ though q use the bit $B = 1$. Then $\Pr_{H_0}[A' \text{ outputs } 1] = \Pr_{\mathbf{UB}}[X = 1 | B = 1]$ and $\Pr_{H_q}[A' \text{ outputs } 1] = \Pr_{\mathbf{UB}}[X = 1 | B = 0]$, and therefore $\Pr_{H_0}[A' \text{ outputs } 1] - \Pr_{H_q}[A' \text{ outputs } 1] = \alpha$.

At the same time, conditioned on $\overline{\mathsf{col}}$ and some value $j \in \{1, \ldots, q\}$ in the reduction, if the oracle of A (which is what is used for the j'th instance) has $B = 1$ then the view of A' is exactly that of H_{j-1} and if the oracle of A has $B = 0$ then the view of A' is exactly that of H_j. Finally, since A only outputs 1 when no collision occurs, we get:

$$
\begin{aligned}
\mathsf{adv}_A^{\mathsf{BC},ind} &= \Pr_{\mathsf{red}}[A \text{ outputs } 1 | B = 1] - \Pr_{\mathsf{red}}[A \text{ outputs } 1 | B = 0] \\
&\geq \Pr_{\mathsf{red}'}[A \text{ outputs } 1 | B = 1] - \Pr_{\mathsf{red}'}[A \text{ outputs } 1 | B = 0] - \frac{2b(b-1)}{2^n} \\
&\geq \Pr_{\mathsf{red}'}[\overline{\mathsf{col}}] \cdot \left(\Pr_{\mathsf{red}'}[A \text{ outputs } 1 | B = 1 \ \&\ \overline{\mathsf{col}}] - \Pr_{\mathsf{red}'}[A \text{ outputs } 1 | B = 0 \ \&\ \overline{\mathsf{col}}] \right) - \frac{2b(b-1)}{2^n} \\
&\geq \big(1 - \frac{b^2(q-1)}{2^n}\big) \cdot \frac{1}{q} \cdot \sum_{i=1}^{q} \big(\Pr_{H_{j-1}}[A' \text{ outputs } 1] - \Pr_{H_j}[A' \text{ outputs } 1] \big) - \frac{2b(b-1)}{2^n} \\
&= \big(1 - \frac{b^2(q-1)}{2^n}\big) \cdot \frac{1}{q} \cdot \big(\Pr_{H_0}[A' \text{ outputs } 1] - \Pr_{H_q}[A' \text{ outputs } 1] \big) - \frac{2b(b-1)}{2^n} \\
&= \big(1 - \frac{b^2(q-1)}{2^n}\big) \cdot \frac{\alpha}{q} - \frac{2b(b-1)}{2^n}.
\end{aligned}
$$

Hence we have $\big(1 - \frac{b^2(q-1)}{2^n}\big) \cdot \frac{\alpha}{q} - \frac{2b(b-1)}{2^n} \leq \mathsf{adv}_A^{\mathsf{BC},ind} \leq \epsilon$ due to the assumed security of the target application in the single-instance setting, and therefore $\alpha \leq \big(\epsilon + \frac{2b(b-1)}{2^{n-1}}\big) \cdot q/(1 - \frac{b^2(q-1)}{2^n})$. □

For the unique-key setting we have $b = 1$ so we get a bound of $\epsilon \cdot q/(1 - \frac{q-1}{2^n})$.

3.2.2 Eliminating Enhanced Security?

Below we discuss briefly two plausible approaches for removing the reliance on "enhanced" security. We were not able to make these approaches work, and provide more details in the appendix.

Extracting the collision event from the q-instance adversary. One way to eliminate the need for collision oracles is to have the reduction implement these oracles by itself. The hope is that if the q-instance adversary A' behaves differently when there are collisions than when there are none, then the reduction can observe these differences and ignore A''s output if there are collisions.

Unfortunately, we could not find a signal that the reduction can use to identify collisions while maintaining the advantage when there are no collisions. We sketch in the appendix a few things that we tried, and explain

why they fail. Finding a method that works, or a convincing reason why no such method exists, remains a theoretical open problem that would be interesting to resolve.

Making a knowledge-type assumption. Failing to constructively extract the knowledge of collisions from the q-instance adversary, we could resort to a knowledge-type assumption to do the extraction for us. Such assumptions are a way of formalizing the intuition that an adversary can only behave differently when there are collisions if they know that a collision occurred. The assumption essentially states that for any adversary A that behaves differently when there are collisions, there is another adversary B that behaves exactly like A but also outputs a bit indicating if a collision occurred. We then could use B in the reduction from above rather than A, replacing the collision-oracles from Theorem 2 by this extra bit that B outputs.

As we explain in the appendix, a more-or-less straightforward formulation of such an assumption turns out to be insufficient in our case. While it is possible to formulate a knowledge assumption that would make the reduction go through, it is a rather contrived one. Assumptions of this type were proven false (or shown to conflict with obfuscation-type assumptions) in the past. We feel that it does not really add to our understanding of what target applications remain secure in the unique-key/block setting, see more details in the appendix.

3.3 Other Reductions

One could also describe other reductions to support the security of target applications in the unique-key/block models. In particular, we can show that security of any q-instance target application in the unique-block model can be reduced to a form of "enhanced" security of the same application in the independent-key model with $q' \approx q/(1-\frac{\lambda b^2 q}{2^n})$ instances. (This applies to both indistinguishability and unpredictability applications.)

Specifically, we need to assume that the target application is secure in the q'-instance independent-key model even in the presence of an oracle that tells the adversary which of the q' keys have blocks that are involved in collisions. Then, given an adversary that expects unique blocks and access to $q' \leq \frac{2^n}{\lambda b^2}$ instances with independent keys, the reduction would ask the collision oracle which of these keys are involved in collisions and simply discard those keys, running the unique-block adversary only on the remaining keys that have unique blocks.

The probability that any one key is involved in a block collision is at most $b^2(q'-1)/2^n$, so the expected number of keys that are involved in collisions is at most $b^2q'(q'-1)/2^n$. Moreover, the events associated with different keys being involved in collisions are negatively correlated, so we can use the Chrenoff bound to conclude that the probability of having more than $\lambda b^2 q'(q'-1)/2^n$ is exponentially small in λ. Hence for $q = q'(1-\frac{\lambda b^2(q-1)}{2^n})$ there will be at least q remaining instances for the reduction to work with, except with a probability exponentially small in λ.

We note, however, that assuming this "enhanced security" is often a much stronger assumption than what we have for Theorem 2 above. Indeed this oracle not only indicates whether or not there are collisions, but actually tells the adversary where they are, which gives the adversary a lot of power. Knowing that two specific keys collide would lead to devastating attacks in many applications, for example the attacker would ask one user to encrypt and use the ciphertext with another user that has the same key. (Indeed the analysis of Bose et al. [6] established security even in the presence of key-collisions exactly because the attacker cannot identify those collisions.)

4 Conclusion

Motivated by the analysis of the XAES-256-GCM mode of Valsorda [16], in this work we considered the multi-instance unique-key/block models where keys from difference isntances of a target-application are chosen not completely at random, but such that all keys/blocks are distinct.

We exhibited sufficient conditions under which we can prove that a secure application in the single-instance model remains secure also in these multi-instance unique-key/block models. In particular we have

shown it for unpredictability-type applications and for indistinguishability-type applications in settings where key/block collisions can be detected.

4.1 Is XAES-256-GCM Secure Above the Birthday Bound?

Returning to our motivated application, what do our results say about the security of XAES-256-GCM?

The XAES-256-GCM encryption scheme takes as input an AES-256 key $K \in \{0,1\}^{256}$ and two 96-bit nonces[6] $U, V \in \{0,1\}^{96}$ and use them to encrypt a plaintext messages P. It uses AES-256 in CMAC mode to derive from U another 256-bit key K_U, then encrypts the message P using AES-256-GCM with key K_U and nonce V. Specifically, it sets

Input: Key K, Nonces U, V, Plaintext P

1. $L = X \times \text{AES-256}_K(0^{128})$ // Multiplication by $X \in GF(2^{128})$
2. $M_1 = (\text{const}_1 \| U) \oplus L$, $M_2 = (\text{const}_2 \| U) \oplus L$
3. $K_U = \text{AES-256}_K(M_1) \parallel \text{AES-256}_K(M_2)$ // CMAC-based key derivation
4. Output $C = \text{AES-256-}GCM_{K_U}(V; P)$ // GCM encryption

where const_1 =`0x00015800` and const_2 =`0x00025800` are two different 32-bit constants.

One of the design goals of XAES-256-GCM is to be able to use one main key to encrypt up to 2^{80} messages with distinct nonces. This means that the key-derivation will be used to derive $q \leq 2^{80}$ AES-256 keys (from different U values), and each one of those may be used to encrypt multiple messages (for different V values).

Assuming that AES-256 itself is a secure PRP, the question becomes whether or not the target application of AES-256-GCM is secure in the unique-block model with $b = 2$ for $q \leq 2^{80}$. AES-256-GCM is an authenticated encryption mode, which means that it offers both CCA-security (which is an indistinguishability-type application) and unforgeability of ciphertexts (which is an unpredictability-type application). Our results imply that this combination provides unforgeability of ciphertexts (assuming that the single-instance AES-256-GCM provides it), and it provides CCA-security if the underlying single-instance AES-256-GCM enjoys enhanced security.

It seems entirely reasonable to assume that a single-instance AES-256-GCM enjoys enhanced security, indeed it is hard to imagine a constructive way of using a block-collision oracle to attack the CCA-security of the scheme. Moreover, as we explained above this is only a sufficient condition, we don't believe it to be a necessary one. Our results therefore lend strong support to XAES achieving its design goals.

References

[1] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 259–274. Springer, 2000. doi:10.1007/3-540-45539-6_18.

[2] Mihir Bellare, Oded Goldreich, and Hugo Krawczyk. Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 270–287. Springer, 1999. doi:10.1007/3-540-48405-1_17.

[3] Mihir Bellare and Björn Tackmann. The multi-user security of authenticated encryption: AES-GCM in TLS 1.3. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 247–276. Springer, 2016. doi:10.1007/978-3-662-53018-4_10.

[6] The description in [16] includes a single 192-bit nonce N, but the two halves serve different purposes so we separate them here, $N = U|V$.

[4] Daniel J. Bernstein. Stronger security bounds for permutations. https://cr.yp.to/antiforgery/permutations-20050323.pdf, 2005.

[5] Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009. doi:10.1007/978-3-642-10366-7_1.

[6] Priyanka Bose, Viet Tung Hoang, and Stefano Tessaro. Revisiting AES-GCM-SIV: multi-user security, faster key derivation, and better bounds. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 468–499. Springer, 2018. doi:10.1007/978-3-319-78381-9_18.

[7] Yu Long Chen, Eran Lambooij, and Bart Mennink. How to build pseudorandom functions from public random permutations. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 266–293. Springer, 2019. doi:10.1007/978-3-030-26948-7_10.

[8] Morris Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC. Technical Report NIST Special Publication (SP) 800-38D (Final), November 28, 2007, U.S. Department of Commerce, Washington, D.C., 2007. doi:10.6028/NIST.SP.800-38D.

[9] Shay Gueron. Double-nonce-derive-key-gcm (DNDK-GCM) general design paradigms and application. Internet Draft, draft-gueron-cfrg-dndkgcm-00, April 2024. Accessed August 2024. URL: https://datatracker.ietf.org/doc/draft-gueron-cfrg-dndkgcm/.

[10] Shay Gueron and Thomas Ristenpart. DNDK: Combining nonce and key derivation for fast and scalable AEAD. Cryptology ePrint Archive, Paper 2025/785, 2025. URL: https://eprint.iacr.org/2025/785.

[11] Chris Hall, David A. Wagner, John Kelsey, and Bruce Schneier. Building prfs from prps. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO '98*, volume 1462, pages 370–389. Springer, 1998. URL: https://doi.org/10.1007/BFb0055742, doi:10.1007/BFB0055742.

[12] Tetsu Iwata. New blockcipher modes of operation with beyond the birthday bound security. In Matthew J. B. Robshaw, editor, *Fast Software Encryption, FSE 2006*, volume 4047 of *Lecture Notes in Computer Science*, pages 310–327. Springer, 2006. doi:10.1007/11799313_20.

[13] Jooyoung Lee, Atul Luykx, Bart Mennink, and Kazuhiko Minematsu. Connecting tweakable and multi-key blockcipher security. *Des. Codes Cryptogr.*, 86(3):623–640, 2018. URL: https://doi.org/10.1007/s10623-017-0347-3, doi:10.1007/S10623-017-0347-3.

[14] Atul Luykx, Bart Mennink, and Kenneth G. Paterson. Analyzing multi-key security degradation. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 575–605. Springer, 2017. doi:10.1007/978-3-319-70697-9_20.

[15] Bart Mennink and Samuel Neves. Optimal prfs from blockcipher designs. *IACR Trans. Symmetric Cryptol.*, 2017(3):228–252, 2017. URL: https://doi.org/10.13154/tosc.v2017.i3.228-252, doi:10.13154/TOSC.V2017.I3.228-252.

[16] Filippo Valsorda. XAES-256-GCM. https://words.filippo.io/dispatches/xaes-256-gcm/, June 2024. Accessed August 2024.

A Attempts at a Reduction without Collision Oracles

As we said in Section 3.2.2, we could not find a way for the reduction to extract from the q-instance adversary the "knowledge" of whether or not there are block collisions among its keys. Below we describe a few attempts and explain why they fail. In these attempts we tried to prove the weaker notion of indistinguishability with a per-instance bit B_j, where the attacker needs to output (i, X) indicating which instance it is trying to break, but even this weaker notion we were not able to prove without a collision oracle.

For this setting, the problematic scenario is a q-instance adversary A' that has a significant advantage of outputting (i, X) where X is the correct guess for the i'th oracle when there are no collisions, but such that in the case of collisions it (a) outputs the index of one of the colliding keys with high probability, and (b) has a significant disadvantage in guessing the bit b in that case.

First try: one set of keys. Consider a reduction that begins the same as the one from Theorem 1, choosing an index j and a set of keys $\vec{K^*} = \{K_1^*, \ldots, K_{j-1}^*, K_{j+1}^*, \ldots, K_q^*\}$, and having access to an oracle with key K_j^*. The reduction then tries to gather statistics about the behavior of the adversary A' on this set of q keys. The hope of that if A' is indeed a problematic adversary as above, then there will be an observable difference between the case where there are block collisions and the case where there are none. Specifically, that the oracle's key will be chosen a lot more often when there are collisions than when there are none.

The reduction could run A' many times on this set of keys in different random orders, to estimate the probability that it outputs the index of the oracle's key. Them, it will only use the output of A' if that probability is below some threshold τ. Let us index by $k = 1, 2, \ldots$ the runs of A', and let j_k be the index of the oracle's key and (i_k, X_k) be the output of A' in the k'th run. The reduction computes the empirical probability of the event $i_k = j_k$, and denote it by μ. If $\mu > \tau$ then the reduction halts with a random output bit. Otherwise it runs the reduction from before, running A' one more time with the oracle at a random index j^*, getting the output (i^*, X^*), and outputting X^* if $i^* = j^*$ and a random bit otherwise.

Unfortunately, this does not work. Consider the q-instance adversary A' that first recovers all the keys (by exhaustive search) and then does the following:

- If there are keys with colliding blocks then always output one of them (always the same one) with a wrong guess for b;
- Otherwise use some hash function H, applied to the set of keys in sorted order, to get an index $i \in [q]$, let K_i be the i'th key in the sorted order, and output the index of K_i (in the original order) with a correct guess for B.

With this adversary, for every set of keys there is exactly one key whose index would be output, regardless of the ordering of keys. When run inside the reduction, either that is the oracle's key (in which case the probability would be above τ) or it is not (in which case it never happens that $i^* = j^*$). Either way, the reduction would always output a random bit and therefore will not have any advantage.

Second try: a few sets of keys. Perhaps, instead of fixing just one set of keys, the reduction can try a few of them, replacing just a handful of keys in each try. Replacing just a handful of keys is unlikely to change whether or not there are block collisions, but will perhaps force A' to vary its output, enabling a better estimate of the relevant probabilities. This does not work either, however. Consider a similar adversary A' to above, except in the no-collision case it looks for the keys whose blocks are closest to each other in Hamming distance and always outputs one of them (always the same one). Changing just a few keys is unlikely to effect the two closest ones, so by the same argument from above the reduction will almost never use the output of A', and therefore will not have almost any advantage.

Third try: many sets of keys. Perhaps the solution would be to use a great many sets of keys when trying to evaluate statistics of A'. For example, the reduction would start by choosing a "main set" of keys, on which it hopes to use the adversary's output. Denote it by $\vec{K^*} = \{K_1^*, \ldots, K_{j-1}^*, K_{j+1}^*, \ldots, K_q^*\}$, and denote the oracle's key by K_j^*.

For every chosen key $i \neq j$, the reduction then tries to estimate whether or not K_i^* has a block collision with the oracle's key K_j^*. This is done by selecting many auxiliary sets of $q-2$ keys, each time running A' with K_i^*, K_j^* and the auxiliary keys (in random order) and measuring the frequency of A' outputting the index of K_i^* and the index of K_j^* in these runs (denoted μ_i and $\mu_{i,j}$, respectively). Also when A' outputs the index of K_i^* the reduction can measure the frequency of correct guesses for b, since it knows the key K_i^*, and denote that frequency by ρ_i. (Obviously the reduction cannot do the same when A' outputs the index of K_j^*.)

Finally, if there exists $i \neq j$ such that $\mu_i + \mu_{i,j} \geq \tau$ and $\rho_i \leq 1/2$ then the reduction concludes that the keys K_i^*, K_j^* have a block collision and aborts with a random bit. Else it runs A' once more with the keys $\vec{K}^*$, and if A' outputs (j, b) then the reduction outputs b (and otherwise it outputs a random bit).

But even this does not work. Let ϵ be some not-too-small constant, say $\epsilon = 1/4$, and consider the following adversary A':

- For the non-collision case, A' designates a set of "special" keys of density $\approx \epsilon/q$, such that a set of q random keys contains at least one special key with probability ϵ. Given access to a set $\vec{K}$ of q keys, A' does the following:
  - If none of them is "special" then A' outputs an arbitrary index with a random bit.
  - Otherwise let K_1 be the special key and K_2 be the key that's closest to it in Hamming distance (breaking ties in some fixed way, say according to lexicographic order). With probability $1/2$ A' outputs the index of K_1 with the right guess for b, and otherwise it outputs the index of K_2 with a random bit.
- If there is a block collision, then let (K_1, K_2) be the colliding pair ordered in some fixed way (say, lexicographically). A' outputs with probability $1/2$ the index of K_1 with a wrong guess for b, with probability $1/4$ the index of K_2 with a random bit, and otherwise they outputs a random index with a random bit.

The advantage of this A' is $\epsilon/2$, but note that the advantage is concentrated only on the special keys. However, the reduction above will almost never use the guess of A' when the oracle's key is special.

To see why, first notice that when there are collisions then A outputs the index of one of the colliding pair only with probability $3/4$. To reject colliding pairs, the threshold τ must therefore be set to $3/4$ (or less). Next, consider the case when K_j^* is special and there are no collisions in $\vec{K}^*$, and let K_i^* be is closest to K_j^* in Hamming distance. When considering K_i^*, the reduction will embed K_i^*, K_j^* in a set with $q-2$ other random keys, and the probability of K_i^* remaining the closest to K_j^* in such a random set is about $1/2$. Hence the reduction will see the index of K_j^* with probability $1/2$, the index of K_i^* with probability $1/4$, and a random index with probability $1/4$, and moreover, the probability of the right guess when outputting the index of K_i^* will be $1/2$. Hence the reduction has $\mu_i + \mu_{i,j} \geq \tau$ and $\rho_i \leq 1/2$, so the output of A' will be ignored.

We note that in addition to not working, this reduction is also quite expensive: Not only it goes over all $q-1$ chosen keys, but for each one it must run a lengthy probability-estimation process, to make the estimation error small enough to be useful. (Above we simply assumed that it was able to compute the various quantities exactly.)

A.1 Formulating a Knowledge Assumption

In light of the failure to extract the collision event from the q-instance adversary, we could try to use a knowledge-type assumption to do the extraction for us. Such an assumption essentially states that for any q-instance adversary A' that behaves differently when there are collisions, there is another adversary A'' of similar complexity that behaves exactly like A but also outputs a bit indicating if a collision occurred. Then the reduction from the proof of Theorem 2 can use that bit in lieu of a collision-oracle.

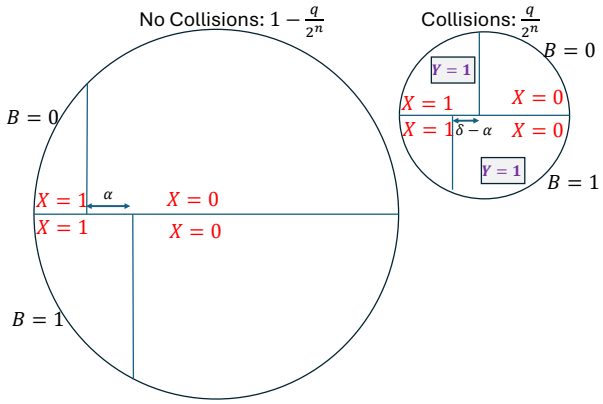

Figure 1: A bad adversary for the reduction

The requirement that A'' behaves exactly like A' is formulated by postulating deterministic adversaries and demanding that on every input, the transcript of A'' is exactly the same as that of A', except that A'' has another output Y in addition to the output X of A'.

The requirement that A'' "knows when there are collisions" is more problematic. Clearly, we want the output Y of A'' to indicate collisions, but what can we require? A natural approach is to demand that if the event $X = B$ (in which A' correctly guesses the global bit B) happens with higher probability in the no-collision case than in the collision case, then A'' has a similar advantage in determining if a collision occurs.

Namely, for any deterministic adversary A' we denote

$$\delta_{A'} := \Pr_{\vec{K}\leftarrow\mathcal{D}_{\mathsf{noCol}},B\leftarrow\{0,1\}}\left[A'^{\vec{K},B}\Rightarrow B\right] - \Pr_{\vec{K}\leftarrow\mathcal{D}_{\mathsf{col}},B\leftarrow\{0,1\}}\left[A'^{\vec{K},B}\Rightarrow B\right].$$

where $\mathcal{D}_{\mathsf{col}}, \mathcal{D}_{\mathsf{noCol}}$ are the distribution of keys with and without collisions, respectively, and $A'^{\vec{K},B}\Rightarrow B$ is the event where A', given access to oracles with keys $\vec{K}$ and global bit B, outputs the bit B. Then we want to require that for every A' there exists A'' that behaves exactly like A' but also outputs a bit Y such that $\Pr_{\mathcal{D}_{\mathsf{col}}}[Y=1] \approx \Pr_{\mathcal{D}_{\mathsf{noCol}}}[Y=1] + \delta_{A'}$.

Unfortunately, this requirement is insufficient for the reduction from Theorem 2 to go through. An illustration of this failure is depicted in Fig. 1: Consider an adversary A' with a small constant advantage (say $\alpha_{A'} = 0.1$), and with $\delta_{A'}$ slightly larger than $\alpha_{A'}$ (say $\delta_{A'} \approx 0.11$). That is, A' has advantage of 0.1 in guessing the bit B when there are no collisions, but a slight disadvantage of 0.01 in guessing B when there are collision. The reduction from the proof of Theorem 2 does not work with this adversary, its "advantage" could be as small as $(1-\frac{q}{2^n})\cdot\frac{0.1}{q} - \frac{q}{2^n}\cdot 0.01$ which is negative for $q > 4\cdot 2^{n/2}$.

But the extra bit Y may not help here, consider the adversary A'' depicted in Fig. 1: It always outputs $Y = 0$ when there are no collisions and it outputs $Y = 1$ with probability $\delta_{A'}$ when there are collisions, so we would like to think of it as "knowing" when collisions occur. But this A'' only outputs $Y = 1$ in situations were (there are collisions and yet) A' outputs $X = B$. Using the event $Y = 1$ as an indicator to ignore the output of A'' will be counter-productive, it will only increase the disadvantage of A'' when there are collisions, thereby decreasing further the advantage of the reduction.

To get a knowledge assumption that can be used for the reduction, we have to demand not only that A'' outputs $Y = 1$ more often when there are collisions than where are none, but also that this $Y = 1$ output is positively correlated with the event $X \neq B$ when A' guesses wrong. But this makes the entire assumption rather meaningless, if A' can do that it means that it can do better than A' in guessing the bit B. This assumption essentially says that adversaries have roughly the same advantages when collisions occur as when there are no collisions. While this could be true, it seems too much like requiring that adversaries play nice

with our reduction rather than trying to handle arbitrary adversaries. As we said, we feel that making such an assumption does not really add to our understanding of what target applications remain secure in the unique-key/block setting.