Kolmogorov Comes to Cryptomania: On Interactive Kolmogorov Complexity and Key-Agreement

Marshall Ball* Yanyi Liu† Noam Mazor ‡ Rafael Pass §

March 11, 2024

Abstract

Only a handful candidates for computational assumptions that imply secure key-agreement protocols (KA) are known, and even fewer are believed to be quantum safe. In this paper, we present a new hardness assumption—the *worst-case hardness* of a promise problem related to an *interactive* version of Kolmogorov Complexity. Roughly speaking, the promise problem requires telling apart tuples of strings (π, x, y) with relatively (w.r.t. $\mathsf{K}(\pi)$) low time-bounded *Interactive Kolmogorov Complexity* (IK^t), and those with relatively high Kolmogorov complexity, given the promise that $\mathsf{K}^t(x|y) < s, \mathsf{K}^t(y|x) < s$ and $s = \log n$, and where $\mathsf{IK}^t(\pi; x; y)$ is defined as the length of the shortest pair of t-bounded TMs (A, B) such that the interaction of (A, B) lead to the transcript π and the respective outputs x, y.

We demonstrate that when t is some polynomial, then not only does this hardness assumption imply the existence of KA, but it is also *necessary* for the existence of secure KA. As such, it yields the first natural hardness assumption characterizing the existence of key-agreement protocols.

We additionally show that when the threshold s is bigger (e.g., $s = 55 \log n$), then the (worst-case) hardness of this problem instead characterizes the existence of one-way functions (OWFs). As such, our work also clarifies exactly what it would take to base KA on the existence of OWFs, and demonstrates that this question boils down to demonstrating a worst-case reduction between two closely related promise problems.

*New York University. Email marshall@cs.nyu.edu. Part of this work was done while vising the Simons Institute and supported in part by the Simons Foundation.

†Cornell Tech. E-mail: yl2866@cornell.edu. Part of this work was done while visiting the Simons Institute.

‡Cornell Tech. E-mail: noammaz@gmail.com. Part of this work was done while at Tel Aviv University and while visiting the Simons Institute. Research partly supported by Israel Science Foundation grant 666/19, NSF CNS-2149305 and NSF CNS-2128519.

§Tel-Aviv University and Cornell Tech. E-mail: rafaelp@tau.ac.il. Part of this work was done while visiting the Simons Institute. Supported in part by NSF Award CNS 2149305, AFOSR Award FA9550-18-1-0267, AFOSR Award FA9550-23-1-0387 and a JP Morgan Faculty Award. This material is based upon work supported by DARPA under Agreement No. HR00110C0086. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government, DARPA or the AFOSR.

1 Introduction

The notion of a *key-agreement* (a.k.a. *key-exchange*) protocol, introduced by Diffie and Hellman [DH76] in their seminar paper "New Directions in Cryptography" from the 1976 ushered in a new era for Cryptography. Key Agreement (KA) protocols enable two parties—Alice and Bob—that have never previously met to use communication to establish a secret key (which later can be used to securely communicate), with the guarantee that an eavesdropper (referred to as Eve) who observes the transcript of communication between Alice and Bob cannot learn the secret key.

Key agreement protocols are perhaps the most important primitive enabling secure communication on the Internet—it is safe to say that a majority of electronic commerce applications would not be possible without secure key agreement protocols. However, despite the importance of key-exchange protocols, and so-called "public-key cryptography" in general (or "Cryptomania" in language of Impagliazzo [Imp95]), we only know of a handful of candidate hard problems from which KA protocols can be constructed. More specifically, these include (a) *number-theory problems* based on either factoring [RSA78; Rab79] or discrete logarithms [DH76; ElG84], (b) coding-theory based problems [McE78], (c) lattice problems as finding shortest/longest vectors in lattices [AD97; Reg09; BCNHR22], and (d) noisy linear-algebra based problems [Ale03; ABW10]. Out of these, the number-theory based problems can be efficiently solved by quantum algorithms [Sho99], and the coding-theory, lattice and noisy linear algebra problems are all very related. Indeed, all popular candidates have significant algebraic structure, as well other structure (e.g., being contained in SZK or $\mathrm{AM} \cap \mathrm{coAM}$). While this structure is useful for constructing cryptographic protocols/primitives, there is always the fear that this structure may eventually make the problem tractable—which indeed is what happened with the number-theory based candidates with respect to quantum algorithms.

This is in great contrast with so-called "private-key primitives" (a.k.a. "Minicrypt" in the language of [Imp95]) and its central primitive of a *one-way function (OWF)* for which lots of candidates are known. Furthermore, recently natural average-case problems *characterizing* the existence of OWFs were demonstrated [LP20; LP21; LP22], and even more recently even *worst-case hard problems* (a.k.a. OWF-complete problems) characterizing the existence of OWFs were demonstrated [LP23], and independently by [HN23] for the case of OWFs with uniform security.

As beautifully expressed by Boaz Barak in 2013 [Bar14; Bar13]:

> *The bottom line is that based on the currently well studied schemes, structure is strongly associated with (and perhaps even implied by) public key cryptography. This is troubling news, since it makes public key crypto somewhat of an "endangered species" that could be wiped out by a surprising algorithmic advance.*

While this text was written 10 years ago, the situation has not changed since then. In particular, the following problem has remained wide open:

> *Can Key Agreement be based on some "unstructured" hardness assumption?*

We may further ask whether, similar to recent characterizations of OWFs, there exists some problem that *characterizes* KA—this would enable more clearly understanding whether structure is inherent for primitives in Cryptomania:

> *Can we identify some problem whose worst-case hardness is equivalent to the existence of key agreement?*

In this work, we simultaneously provide positive answers to both the above questions. We demonstrate the existence of a (seemingly) unstructured problem, whose worst-case hardness is *equivalent* to the existence of KA.

1.1 Our Results

We will demonstrate the existence of a problem whose worst-case hardness characterizes the existence of KA. This problem is motivated by a recent trends of works demonstrating Kolmogorov-complexity style problems whose average-case hardness (with respect to the uniform distribution) characterize the existence of OWFs [LP20; LP21; LP22], and most notably, the very recent result of [LP23] that develops *non-black-box techniques* to demonstrate a promise problem whose *worst-case hardness* characterizes OWFs ([HN23] independently develop a similar technique in the uniform setting).

We highlight, however, that whereas our work is inspired by these works characterizing OWFs, the actual details are quite different. Most notably, whereas these earlier works can work with the (standard) notion of time-bounded Kolmogorov complexity, we will need to introduce a new *interactive* variant of time-bounded Kolmogorov complexity, that we believe is of independent interest.

Interactive Kolmogorov Complexity Roughly speaking, the *Time-bounded Interactive Kolmogorov Complexity* (IK^t) of a tuples (π, x, y) is the minimal combined length $|P_\mathsf{A}| + |P_\mathsf{B}|$ of time-t programs, P_A and P_B such that the interaction of $(P_\mathsf{A}, P_\mathsf{B})$ lead to the transcript π and respective outputs x, y. We should think of IK^t as the natural generalization of time-bounded Kolmogorov complexity, K^t to interactive algorithms.

In more detail, let U be a universal TM. Given two programs P_A and P_B, and a bound t on the number of steps, let $(\mathsf{U}(P_\mathsf{A}, 1^t), \mathsf{U}(P_\mathsf{B}, 1^t))$ denote the interaction between $\mathsf{U}(P_\mathsf{A}, 1^t)$ and $\mathsf{U}(P_\mathsf{B}, 1^t)$, when simulating each program for t steps. (For concreteness, and following the literature on communication complexity [Yao79], we will focus on a specific model of interaction, where the players take turns to send single bits to one another.)

Definition 1.1 (IK^t)**.** *For a function $t\colon \mathbb{N} \to \mathbb{N}$, and $(\pi, x, y) \in \{0,1\}^*$, the t-*bounded interactive Kolmogorov complexity *of π, x, y, denoted by $\mathsf{IK}^t(\pi; x; y)$, is the minimal number $\ell \in \mathbb{N}$ for which there exists (deterministic) programs P_A and P_B such that (a) $|P_\mathsf{A}| + |P_\mathsf{B}| = \ell$, and (b) The interaction $(\mathsf{U}(P_\mathsf{A}, 1^t), \mathsf{U}(P_\mathsf{B}, 1^t))$ yields the transcript π and the respective outputs x, y.*

The Relative IK^t problem: $\mathsf{RIK}^t\mathsf{P}[\alpha, \beta]$ We are now ready to formalize the promise problem, $\mathsf{RIK}^t\mathsf{P}$, that we will be considering. Roughly speaking, the promise problem requires telling apart (a) tuples of strings (π, x, y) with *relatively* low t-bounded Interactive Kolmogorov Complexity, and those with (b) *relatively* high Kolmogorov complexity.[1] When we say relatively high/low, we mean relative to the Kolmogorov complexity of π.

More formally,

Definition 1.2 ($\mathsf{RIK}^t\mathsf{P}$)**.** *(Relative IK^t problem) For functions $\sigma_Y < \sigma_N$ and t, let $\mathsf{RIK}^t\mathsf{P}[\sigma_Y, \sigma_N]$ denote the following promise problem:*

[1] Recall that the Kolmogorov complexity, $\mathsf{K}(w)$, of a string w is simply the length of the shortest program, w.r.t. some fixed UTM, that outputs the string w.

- $\mathcal{Y} = \{(\pi, x, y) \in (\{0,1\}^n)^3 \colon \mathsf{IK}^t(\pi; x; y) \leq \mathsf{K}(\pi) + \sigma_Y\}$,
- $\mathcal{N} = \{(\pi, x, y) \in (\{0,1\}^n)^3 \colon \mathsf{K}(\pi, x, y) \geq \mathsf{K}(\pi) + \sigma_N\}$.

In essence, the problem considers the cost (in terms of description length) of generating (π, x, y) relative to the cost of generating just π:

- **YES**-instances correspond to tuples (π, x, y) where there IS NO significant "cost" in terms of description length to generate (π, x, y) through (a) interaction and (b) efficiently, as opposed to just generating π (without interaction and perhaps in efficiently).
- **NO**-instances, on the other hand, correspond to tuples (π, x, y) where there IS a significant cost even just to generate (π, x, y) as opposed to generating just π.

Let us highlight that this problem is closely related to the notion of *computational depth* introduced by Antunes et al [AFVMV06], and defined as $CD^t(x) = \mathsf{K}^t(x) - \mathsf{K}(x)$. The **YES**-instances of our $\mathsf{RIK}^t\mathsf{P}$ problem can be thought of as those with small "interactive" computational depth; as we shall discuss shortly, this will be instrumental to us to enables a worst-case to average-case reduction.

Characterizing KA For our purposes, simply the $\mathsf{RIK}^t\mathsf{P}$ problem will not suffice; we will need to condition the problem on the promise that x and y are "close" to each other in "Kolmogorov distance". For a function $\Delta \colon \mathbb{N} \to \mathbb{N}$, let[2]

$$Q_\Delta = \{(\pi, x, y) \in (\{0,1\}^n)^3 \colon \mathsf{K}^t(x \mid y) \leq \Delta(n), \mathsf{K}^t(y \mid x) \leq \Delta(n)\}.$$

Observe that when $\Delta = O(\log n)$ and $t \in \text{poly}$, Q_Δ can be decided in polynomial time.

In the following, we let $\mathsf{RIK}^t\mathsf{P}[\sigma_Y, \sigma_N]|_{Q_\Delta}$ denote the promised problem $(\mathcal{Y} \cap Q_\Delta, \mathcal{N} \cap Q_\Delta)$, for $(\mathcal{Y}, \mathcal{N}) = \mathsf{RIK}^t\mathsf{P}[\sigma_Y, \sigma_N]$ We are now ready to state our characterization of KA.

Theorem 1.3 (KA characterization)**.** *The following are equivalent for every polynomial $t(n) > n^{1.1}$, and every $\Delta(n) \leq \log n$*

1. *Key-agreement protocols exist.*
2. $\mathsf{RIK}^t\mathsf{P}[10 \log n, 50 \log n]|_{Q_\Delta} \notin \mathsf{ioBPP}$.

As far as we know, Theorem 1.3 yields the first plausible unstructured assumption that implies the existence of KA, let alone the fact that this assumption also is *necessary* for the existence of KA. Additionally, as far as we know, it is also only plausible non-lattice-based *worst-case* hardness assumption that implies the existence of KA.

An Alternative Worst-case Characterization of OWFs As mentioned above, the recent work of [LP23] provides a characterization of OWF through the worst-case hardness of a natural promise problem. We here note that the same $\mathsf{RIK}^t\mathsf{P}[10 \log n, 50 \log n]|_{Q_\Delta}$ problem which characterizes KA when Δ is small, characterizes OWF when Δ is just slightly larger.

Theorem 1.4 (OWF characterization)**.** *The following are equivalent for every polynomial $t(n) > n^{1.1}$, and every $\Delta(n) \geq 55 \log n$*

[2] $\mathsf{K}^t(w)$ is defined just as $\mathsf{K}(w)$ except that we restrict to programs whose running is bounded by $t(|w|)$.

1. *One-way functions exist.*

2. $\mathsf{RIK}^t\mathsf{P}[10\log n, 50\log n]|_{Q_\Delta} \notin \mathsf{ioBPP}$.

We mention that there is nothing special about the constants in the above Theorems; in fact, the results holds as long as the constant, and their difference, is sufficiently large—we refer the reader to the formal statements in Section 3 (see Theorems 3.4 and 3.5).

KA v.s. OWFs Note that the *only* difference between the characterization of KA and that of OWFs is the size of Δ—for KA it needs to be less than $\log n$ whereas for OWFs it needs to be at least $55\log n$. As such, as a direct consequence, we get that the question of basing KA on OWF is equivalent to demonstrating a *worst-case* reduction from $\mathsf{RIK}^t\mathsf{P}_{Q_\Delta}$ with "large" Δ to a "small" Δ. We note, however, that while decreasing the threshold may seem like just a quantitative question, there is a qualitative difference between the "small" threshold and the "large" threshold cases: in the large threshold case (i.e., OWF), s is required to be bigger than the $\mathsf{RIK}^t\mathsf{P}$ threshold for **NO**-instances, whereas for the small threshold case (i.e., KA) it is required to be smaller than the $\mathsf{RIK}^t\mathsf{P}$ threshold for **NO**-instances.

A Note on Worst-case to Average-case Reductions using Computational Depth Antunes and Fortnow [AF09] elegantly used (standard) computational depth to connect worst-case hardness of a problem when restricting attention to elements with small computational depth and average-case hardness on sampleable distributions; this connection, however, only gave so-called "errorless average-case hardness" (i.e., average-case hardness w.r.t. algorithms that never make mistake—they either give the right answer or output $\perp$) which is not useful for cryptography. Liu and Pass [LP23] recently showed that such a worst-case to average-case reduction can be performed also in the "two-sided error" case (relevant for cryptography) when considering a particular computational problem related to time-bounded Kolmogorov complexity. Since as mentioned above, our **YES**-instances can be thought of those with small "interactive" computational depth, we will be able to leverage those techniques to use worst-case hardness.

1.2 Proof Overview

We here provide a detailed proof overview for the proof of Theorem 1.3. The proof of Theorem Theorem 1.4 follows using similar techniques, but leveraging machinery already developed for OWFs (e.g., [LP20; LP23]).

1.2.1 KA from Worst-case Hardness of $\mathsf{RIK}^t\mathsf{P}$

Weak KA First, we observe that by the Key-agreement Amplification Theorem of Holenstein [Hol06] and an application of the Goldreich-Levin theorem [GL89], to obtain (full-fledged) KA, it suffices to obtain a weak form of KA, which we simply refer to as *Weak KA* defined as follows: There exist some $\epsilon = 1/\text{poly}$ such that *agreement* between A and B happens with probability $1-\epsilon$. *Security* requires that Eve cannot guess the key (output by Alice) with probability better than $1 - 20\epsilon$.

The Weak KA protocol We will next show how to build a Weak KA assuming the worst-case hardness of $\mathsf{RIK}^t\mathsf{P}$. Our protocols proceeds as follows. Alice and Bob on input n, perform the following steps:

- *Sample random programs:* Alice and Bob respectively sample random lengths $\ell_A, \ell_B \in [n]$, and random length ℓ-program, P_A and P_B respectively.
- *Run random programs:* They next runs their respective programs for at most $t(n)$ steps (i.e., letting P_A communicate with P_B, each of them running for at most $t(n)$ steps, and let x, y denote the respective outputs.
- *Correcting Bob's outputs:* Bob samples a random length $\ell' \in [\log n]$, and a random length ℓ' program P' and lets y' denote the output of $P'(y)$ after at most $t(n)$ steps. (Intuitively, this steps enables Bob to recover Alice's output x from his own output y given the condition that $\mathsf{K}^t(x|y) \leq \log n$.)
- *Equality check:* Finally, they run a equality check to determine whether the outputs x and y' are the same; in more detail, Alice picks a universal hash $h : \{0,1\}^n \to \{0,1\}^{20 \log n}$ and sends over $h, z = h(x)$. Bob verifies if $z = h(y')$ and if so sends back the message $\mathsf{success}$, and otherwise abort.
- *Outputs:* If Bob sent the message $\mathsf{success}$, the parties respectively output x, y'; otherwise, they respectively output $0, 0$.

Agreement We claim that with probability $1 - 1/n^{20}$, Alice and Bob will agree (i.e., the final outputs are the same). Note that the only time Alice and Bob will not agree is in case, $h(x) = h(y')$, yet $y' \neq x$. This can only happen when x and y' lead to a collision in the universal hash function, but this only happens with probability $1/n^{20}$.

Security Clearly, when the message abort is sent, then Eve can always guess the key. Intuitively, we need to show that (1) "non-trivial agreement" (i.e., $x = y'$) happens with large probability, and (2) when this happens, Eve cannot guess x with too large probability.

In more detail, consider some Eve that succeeds in guessing the key with probability, $1 - 1/n^{19}$. We will show how to use Eve to decide $\mathsf{RIK}^t\mathsf{P}[10 \log n, 50 \log n]|_{Q_\Delta}$ (in probabilistic polynomial time). Given some instance (π, x, y), need to decide whether IK^t is relatively low. The idea is to see if Eve is able to guess x given the transcript π, and if so output YES (since x can be predicted by Eve who is a small algorithm). In more detail, the decider given (π, x, y) needs to embed π into a transcript to feed to Eve. We think of π as the transcript of (P_A, P_B), extend it to a full transcript by picking a random hash $h : \{0,1\}^n \to \{0,1\}^{20 \log n}$ and let $\pi' = (\pi, h, h(x), \mathsf{success})$, and run Eve on π'. If Eve guesses x, output YES, and otherwise NO.

We will show that if Eve succeeds with probability $1 - 1/n^{19}$ when running the protocol on security parameter n (for some sufficiently large n), then we can decide $\mathsf{RIK}^t\mathsf{P}$ on *all* instances of length $3n$. To show this, we will consider **YES**- and **NO**-instances separately, and for simplicity of exposition, we here assume that Eve is deterministic.

- Consider some **YES** instance $(\pi, x, y) \in (\{0,1\}^n)^3$. Let $\ell = \mathsf{IK}^t(\pi, x, y)$. Since it is a **YES**-instance, we have that $\mathsf{IK}^t(\pi, x, y) - \mathsf{K}(\pi, x, y) \leq 10 \log n$; that is, $\mathsf{K}(\pi, x, y) \geq \ell - 10 \log n$. Due

to the promise Q_Δ, we further have that $\mathsf{K}^t(x|y) \leq \log n$ (and that $\mathsf{K}^t(y|x) \leq \log n$, but that will not be relevant in the analysis of **YES** instances).

If decider fails on (π, x, y) with probability $\geq 1/3$ (recall that the decider is randomized) then with probability $1/3 \cdot 2^{-\ell - \log n} \cdot n^{-3}$, Eve fails and the transcript and outputs are *non-aborting* and equal to $(\pi, x, y' = x)$. (We have probability $1/n$ of picking the right machine length ℓ_A for Alice, probability $1/n$ of picking the right machine length ℓ_B for Bob, and $1/\log n$ probability of picking the right length ℓ' for Bob "correcting machine" —we here rely on the fact that $\mathsf{K}^t(x|y) \leq \log n$, and finally we have that the probability of actually picking all the right programs (given that we picked the right lengths for them) is $2^{-\ell - \log n}$.)

Recall that Eve succeeds with probability $1 - 1/n^{19}$ and thus fails with probability bounded by $1/n^{19}$; it follows that the probability that she fails *and* $\mathsf{IK}^t(\pi, x, y) = \ell$ also is bounded by $1/n^{19}$. Thus, the number of tuples (π, x, y) on which Eve fails and $\mathsf{IK}^t(\pi, x, y) = \ell$ is upper bounded by

$$\frac{n^{-19}}{1/3 \cdot 2^{-\ell - \log n} \cdot n^{-3}} \leq 2^{\ell - 14 \log n}.$$

since if we denote by k, the number of such tuples, we have that $k \cdot 1/3 \cdot 2^{-\ell - \log n} \cdot n^{-3} \leq n^{-19}$

It follows that the Kolmogorov complexity of the tuple must be smaller than $\ell - 14 \log n$ (to describe the index in the list of all tuples on which Eve fails with proability $1/3$)+ $\log n$ (to describe n) plus $\log n$ (to describe ℓ) plus $O(1)$ (to describe Eve) plus an additional $\log n$ terms to deal with "self delimiting" (i.e., to be able decode); in total $\mathsf{K}(\pi, x, y) < \ell - 10 \log n$, which is a contradiction. (We note that this last part of the argument is inspired by argument in the recent paper [LP23], relying on computational depth—as mentioned, our **YES**-instances can be thought of those with small "interactive computational depth) Let us also highlight that this part of the argument is *non-black-box*—it requires using the code of Eve.

- Consider some **NO** instance $(\pi, x, y) \in (\{0,1\}^n)^3$ that is in the promise (i.e., $\mathsf{K}^t(y|x) \leq \log n$, and $\mathsf{K}^t(y|x) \leq \log n$ but the latter statement will not be relevant in the analysis of **NO** instances).

  We will show that the decider will output NO with probability $2/3$ in this case. Or equivalently, that if the decider outputs YES with probability $> 1/3$, then $\mathsf{K}(\pi, x, y) < \mathsf{K}(\pi) + 50 \log n$, so (π, x, y) cannot be a **NO**-instance. Recall that the decider only outputs YES when Eve manages to guess the key x. Intuitively, when this happens, we can compress x given π and $h(x)$ by using the code of Eve (which is constant, plus the description of n, which can specified in $2 \log n$ bits). Furthermore, since due the promise (in particular, that $\mathsf{K}^t(y|x) < \log n$), we can also generate y (given x) using $2 \log n$ bits. Since $|h(x)| = 20 \log n$, in total, $25 \log n$ suffice to specify x, y given π. There is just one issue: even though we assumed that Eve is deterministic, the decider is randomized, so to perform the above compression, we naively would also need to include the random tape of the decider, and more specifically, the description of the hashfunction h (which may be long).

  To resolve this issue, we rely on a classic results in the literature on Kolmogorov complexity: *symmetry of information (SoI)* [Zvo] which states that for all strings a, b,

$$\mathsf{K}(a) - \mathsf{K}(a|b) \leq \mathsf{K}(b) - \mathsf{K}(b|a) + 10 \log(|a| + |b|)$$

  We will let $a = (\pi, x, y)$ and $b = h$; we may assume without loss of generality that $|h| = \ell'$ for some $\ell' = \ell'(n)$ and that the hashfunction is selected as a uniform string of length ℓ'. Given a,

fix some b (i.e., the hashfunction h) such that (a) the decider outputs YES given this choice of h, and (b) $\mathsf{K}(b|a)$ is as high as possible. By a standard counting argument, it follows that we can find a string satisfying (a) such that $\mathsf{K}(b|a) \geq \ell' - O(1) \geq \mathsf{K}(b) - O(1)$

Above, we have argued that $\mathsf{K}(a|b) \leq \mathsf{K}(\pi) + 25 \log n$ for a choice of b when the decider outputs YES. It follows by SoI that $\mathsf{K}(a) \leq \mathsf{K}(a|b) + 10 \log n^2 \leq \mathsf{K}(\pi) + 45 \log n$ (if we use any standard universal hashfunction). (In the actual formal proof, we present a direct proof using a similar structure as the proof of the SoI Theorem; this enables obtaining better constants).

Dealing with Non-uniform Attackers The above analysis only deals with uniform attacker—in particular, we relied on the fact that the description size of Eve is $O(1)$. As we now discuss, using a trick from [LP23], we can extend the analysis to also work in the non-uniform setting. In more details, assuming that $\mathsf{RIK}^t\mathsf{P}[5 \log n, 50 \log n]|_{Q_\Delta} \notin \mathsf{ioP}/\mathrm{poly}$, we can prove that the above protocol is secure against non-uniform adversaries. To prove this, we assume towards a contradiction that Eve is a non-uniform algorithm that breaks the key-agreement protocol, and construct a non-uniform algorithm that decides $\mathsf{RIK}^t\mathsf{P}[5 \log n, 50 \log n]|_{Q_\Delta}$. The issue with the above proof is that we cannot simply use Eve to bound the Kolmogorov complexity of (π, x, y) as done in the above proof, as the description length of Eve now is large. However, if such Eve exists, we can simulate a good attacker using a fixed-size (inefficient) Turing machine: Let M be the Turing machine that, given a constant c such that n^c is a bound on the size of Eve, and input π', first finds the circuit E'_n of size at most n^c that maximize the advantage in predicting the output of Alice in the protocol, and then execute $E'(\pi')$. In this case, M has prediction advantage at least as the advantage of Eve, and we can use M to compress (π, x, y).

A Note on Universal KA and the Need for Equality Checking By our results, the above protocol (after amplification) is a so-called universal KA protocol-namely, a protocol having the property that if KA exists, then the protocol is a KA. A universal KA protocol was previously presented by [HKNRR05], following ideas similar to those used by Levin to construct a universal OWF [Lev85]. We also highlight that our protocol resembles the universal OWF construction of [Lev85] in the sense that we are letting the players pick a random program. The problem with such an approach is that we are not guaranteed that the programs picked by Alice and Bob actually lead to agreement: [HKNRR05] thus lets Alice pick two program (one for her and one for Bob) and first checks if those programs lead to agreement (with high probability). To reduce security to $\mathsf{RIK}^t\mathsf{P}$, we cannot afford to do so but rather must have Alice and Bob individually pick random programs, and this is why we require performing equality checking to determine agreement, which increases the round complexity. This is also the reason why our approach does not apply to public key encryption.

1.3 Worst-case Hardness of $\mathsf{RIK}^t\mathsf{P}$ from KA

We next show how the existence of KA implies hardness of $\mathsf{RIK}^t\mathsf{P}$.

Hardness of $\mathsf{RIK}^t\mathsf{P}$ from DH-Style KA For starters, let us show this statement for the special case when the KA protocol has the special property that (a) the transcript of the protocol is uniform, and (b) the length of the transcript is the sum of the length of the random tapes of Alice

and Bob. For instance, the original KA protocol of Diffie and Hellman [DH76] has this property; for concreteness, let us refer to those as **DH-style protocols**.

Note that any KA protocol provides a way to generate a transcript π and outputs x, y such that (a) $x = y$ (i.e., we have agreement), and (b) $\mathsf{IK}^t(\pi, x, y)$ is upper bounded by the length of the randomness of the respective parties plus $2 \times 2 \log n$ (to describe n for each of the players, and the constant size descriptions of Alice and Bob), which for the special case of DH-style protocols is upperbounded by $\mathsf{K}(\pi)$ plus $4 \log n$ with high probability over π, x, y. That is,

$$\mathsf{IK}^t(\pi, x, x) \leq \mathsf{K}(\pi) + 4 \log n$$

On the other hand, by the security of the KA protocol, we have that (π, x, y) (as sampled from the protocol) is indistinguishable from (π, z, z) where π is sampled from the protocol and $z \leftarrow U_n$. And with high probability over such strings, we have

$$\mathsf{K}(\pi, z, z) \geq \mathsf{K}(\pi) + n - O(1)$$

Now, consider any polynomial-time decider for $\mathsf{RIK}^t\mathsf{P}$; we shall argue that such a decider must make some mistake. First, note that if the decider succeeds with probability 0.99 of *all* **YES** instances, then it must output YES with probability at least 0.98 on instances sampled from the KA. By indistinguishability, it must then still output YES with probability 0.97 on instances of the form (π, z, z) where π sampled from the protocol and $z \leftarrow U_n$; furthermore, by an averaging argument, we have that for 0.9 fraction of instances so sampled, the decider outputs YES with probability 0.9 over its own randomness. But at least a fraction 0.99 of these instances are **NO** instances, so there are (lots of) instances on which the decider makes mistakes.

DH-Style KA with Small Keys We note that the above argument actually works for any DH-style protocol where the length of the key is $\geq 51 \log n$. In this case, we can pad the key to length n (by adding extra 0s), and observe that (a), the analysis of **YES**-instances remains unchanged, and (b) for **NO**-instances, we get that $\mathsf{K}(\pi, z, z) \geq \mathsf{K}(\pi) + 51 \log n - O(1) \geq 50 \log n$. This observation will be useful to us later on.

Hardness of $\mathsf{RIK}^t\mathsf{P}$ from Cond EP-KA To extend the above approach to work for any KA (not necessarily a DH-style one), we will following the blue print from [LP20] used in order to show that OWFs imply average-case hardness of time-bounded Kolmogorov complexity. In particular, in analogy with the notion of an *conditionally entropy-preserving PRG* from [LP20], we define notion of an *conditionally entropy-preserving KA (cond-EP-KA)*. Roughly speaking, an *entropy-preserving KA* (EP-KA) is a KA protocol where the min-entropy of the transcript is close to the length of the randomness of both parties. A cond-EP KA is one where there exists some event E such that conditional on E, the protocol is entropy preserving, and furthermore both agreement and security only need to hold conditioned on E.[3]

Roughly speaking, the cond-EP KA property is exactly what is needed to make the above proof (for DH-style protocols) go through. This follows from the fact that with high probability,

[3]This is essentially a straight-forward extension of the notion of an cond EP-PRG from [LP20] which roughly speaking required entropy-preserving and pseudorandomness conditioned on some event E. For technical reasons, we here defined entropy-preserving with respect to min-entropy as opposed to entropy as was done in [LP20]. Additionally, to simplify the proof, we will also allow the event E to be *randomized*.

strings sampled from a source with high min-entropy must have high Kolmogorov complexity (by a standard counting argument). We just need to also verify $\mathsf{IK}^t(\pi, x, y)$ is small when π, x, y is sampled from the protocol, which holds independently of what the event is.

Cond EP-KA from KA It remains to show how to get an cond EP-KA from any KA. We will proceed in two steps. First, we will show how to get a "weak" cond-EP KA where the key is simply unpredictable (as opposed to indistinguishable from random), and next we will simply use the Goldreich-Levin Theorem to strengthen it into a (full-fledged) cond-EP KA with key of length $51 \log n$, which as observed above suffices to conclude the argument.

It just remain to show how to turn any KA into a (weak) cond-EP KA. To do this, our starting point will again be the approach from [LP20] (of constructing a cond EP-PRG from OWFs) extended to the interactive setting; our analysis, however, is significantly different and more complicated: Alice picks a pairwise independent hashfunction h (which acts as a good extractor by the Left-over-hash Lemma (LHL) [HILL99]), sends h to Bob; next Alice and Bob run the original KA protocols, and finally, they both pick random indexes $i_A, i_B \in [r(n)]$ where $r(n)$ is an upperbound on the length of their randomness, and finally respectively send each other $h(r_A)$ and $h(r_B)$ *truncated* to $i_A - 2\log n$ and $i_B - 2\log n$ bits, where r_A, r_B denote their respective random strings. Note that this is no longer a secure KA, since with high probability, we are leaking a large part of the randomness of the parties. But, if i_A, i_B happen to be picked as the min-entropies of (respectively) $r_A|\pi$, $r_B|\pi$—let us refer to this event as E—then by the LHL, the output of the hashfunctions will be $1/n$ close (in statistical distance) to uniform. We may next rely on the characterization that two distribution are δ-close (in statistical distance) if and only if there is a coupled way of choosing elements from the two distributions such that the two samples are equal with probability $1-\delta$—let us refer to this event as W; thus, we can define a randomized event E' (that E happens and that the coupled samples are equal) conditioned on which the outputs of the hashfunction is the uniform distribution conditioned on some large event (with probability $1 - 1/n$). In particular, this directly yields that conditioned on E', the probability of every full transcript (including the hashes) is at most

$$\begin{aligned}\Pr[\pi, h] \cdot 2^{-\mathrm{H}_\infty(r_A|\pi) - \mathrm{H}_\infty(r_B|\pi) + 4\log n} &= \Pr[\pi] \cdot 2^{-\mathrm{H}_\infty(r_A, r_B|\pi) + 4\log n - |h|} \\ &= \Pr[\pi] \cdot 2^{-|r_A| - |r_B| + \log(1/\Pr[\pi]) + 4\log n - |h|} = 2^{-|r_A| - |r_B| + 4\log n - |h|}\end{aligned}$$

and thus the min-entropy is at least $|r_A| + |r_B| + |h| - 4\log n$.

Additionally, we would like to deduce that conditioned E', security also still holds. Intuitively, this should trivially follows since we have argued that the output of the hashfunction is uniform conditioned on some large event (we can clearly simulate the uniform distribution, and if an attacker can guess the key when we condition on W, then we can also guess the key given the uniform distribution with a factor 2 loss). There is, however, a final obstacles with the above. To argue security, we require the min-entropy of r_A (and r_B respectively) to be high not just conditioned on π but also conditioned on x (and y respectively). Luckily, by the agreement property of the KA protocols, one can show that with high probability over the execution of the KA, x and y are *deterministically* determined as a function of π (to see this, we may consider Alice and Bob as inefficient but stateless players: if they agree, then the secret key must be fixed as a function of the transcript). As such, conditioning also on x (or y) does not change the min-entropy of r_A (or r_B).

2 Preliminaries

2.1 Notations

All logarithms are taken in base 2. We use calligraphic letters to denote sets and distributions, uppercase for random variables, and lowercase for values and functions. Let poly stand for the set of all polynomials. Let PPT stand for probabilistic poly-time, and n.u.-poly-time stand for non-uniform poly-time. An n.u.-poly-time algorithm A is equipped with a (fixed) poly-size advice string set $\{z_n\}_{n\in\mathbb{N}}$ (that we typically omit from the notation), and we let A_n stand for A equipped with the advice z_n (used for inputs of length n). Let neg stand for a negligible function. Given a vector $v \in \Sigma^n$, let v_i denote its i^{th} entry, let $v_{<i} = (v_1, \ldots, v_{i-1})$ and $v_{\leq i} = (v_1, \ldots, v_i)$. Similarly, for a set $\mathcal{I} \subseteq [n]$, let $v_{\mathcal{I}}$ be the ordered sequence $(v_i)_{i\in\mathcal{I}}$. For $x, y \in \{0,1\}^*$, we use $x||y$ to denote the concatenation of x and y. For a set $\mathcal{S} \subseteq \{0,1\}^*$, we use $\mathcal{S}||y$ to denote the set $\{x||y\colon x \in \mathcal{S}\}$.

2.2 Distributions and Random Variables

When unambiguous, we will naturally view a random variable as its marginal distribution. The support of a finite distribution $\mathcal{P}$ is defined by $\mathrm{Supp}(\mathcal{P}) := \{x\colon \Pr_{\mathcal{P}}[x] > 0\}$. For a (discrete) distribution $\mathcal{P}$, let $x \leftarrow \mathcal{P}$ denote that x was sampled according to $\mathcal{P}$. Similarly, for a set $\mathcal{S}$, let $x \leftarrow \mathcal{S}$ denote that x is drawn uniformly from $\mathcal{S}$. For $m \in \mathbb{N}$, we use $\mathcal{U}_m$ to denote a uniform random variable over $\{0,1\}^m$ (that is independent from other random variables in consideration). The statistical distance (also known as, variation distance) of two distributions $\mathcal{P}$ and $\mathcal{Q}$ over a discrete domain $\mathcal{X}$ is defined by $\mathrm{SD}(\mathcal{P},\mathcal{Q}) := \max_{\mathcal{S}\subseteq\mathcal{X}} |\mathcal{P}(\mathcal{S}) - \mathcal{Q}(\mathcal{S})| = \frac{1}{2}\sum_{x\in\mathcal{S}} |\mathcal{P}(x) - \mathcal{Q}(x)|$. We use the following standard definitions:

Definition 2.1 (Indistinguishability)**.** *Distribution ensembles $\mathcal{P} = \{\mathcal{P}_n\}_{n\in\mathbb{N}}$ and $\mathcal{Q} = \{\mathcal{Q}_n\}_{n\in\mathbb{N}}$ are n.u.-poly-time-indistinguishable, if*

$$\left| \Pr_{x\leftarrow\mathcal{P}_n}[\mathsf{D}(x) = 1] - \Pr_{x\leftarrow\mathcal{Q}_n}[\mathsf{D}(x) = 1] \right| \leq \mathrm{neg}(n)$$

for any PPT *algorithm* D*.*

Definition 2.2 (Computable distribution)**.** *A distribution ensemble $\mathcal{P} = \{\mathcal{P}_n\}$ is* computable*, if there exists (potentially inefficient) algorithm S and a computable function $m \in \mathrm{poly}$, such that for every $n \in \mathbb{N}$, $\mathsf{S}(1^n; \mathcal{U}_{m(n)})$ is distributed according to $\mathcal{P}_n$.*

We will also use the following lemma, proved in Appendix B.

Lemma 2.3 (Coupling)**.** *Let X_1 and X_2 be distributions over a set Ω, such that $\mathrm{SD}(X_1, X_2) = \epsilon$. Then there exist random variables W_1 and W_2, jointly distributed with X_1 and X_2 respectively, such that $\Pr[W_1 = 1] = \Pr[W_2 = 1] = 1 - \epsilon$, and $X_1|_{W_1=1} \equiv X_2|_{W_2=1}$.*

2.3 Entropy

For a random variable X, let $\mathrm{H}(X) = \mathrm{E}[\log \frac{1}{\Pr[X=x]}]$ denote the (Shannon) entropy of X, and let $\mathrm{H}_\infty(X) = \min_{x\in Supp(X)} \log \frac{1}{\Pr[X=x]}$ denote the *min-entropy* of X.

$$\min_{x\in\mathrm{Supp}(X)} \log \frac{1}{\Pr[X = x]}.$$

For a random variable X and an event E, we use $\mathrm{H}_\infty(X \mid E)$ to denote the min-entropy of the distribution $X|_E$. The max-entropy of a distribution X, denoted by $\mathrm{H}_0(X)$, is defined by

$$\mathrm{H}_0(X) = \log|\mathrm{Supp}(X)|.$$

We will use the following facts.

Lemma 2.4 (Implicit in [LP20; IRS22], explicit in [LP23])**.** *Let X be a random variable distributed over $S \subseteq \{0,1\}^n$, E be an set $\subseteq S$. It holds that*

$$\Pr[x \leftarrow X : x \in E] \leq \frac{\log|S| + 1 - H(X)}{\log|S| - \log|E|}$$

Fact 2.5. *Let X and Y be independent random variables. Then $\mathrm{H}_\infty(X,Y) = \mathrm{H}_\infty(X) + \mathrm{H}_\infty(Y)$.*

Fact 2.6. *Let X be a random variable and E an event. Then $\mathrm{H}_\infty(X \mid E) \geq \mathrm{H}_\infty(X) - \log\frac{1}{\Pr[E]}$.*

2.4 Promise problems

A promise problem $\mathcal{L} = (\mathcal{Y}, \mathcal{N})$ is a pair of disjoint subsets of $\{0,1\}^*$.

Definition 2.7 (Infinitely-often BPP (ioBPP))**.** *A promise problem $(\mathcal{Y}, \mathcal{N})$ is in* ioBPP *if there exists a* PPT *algorithm* A *such that the following holds for infinitely many n's with $(\mathcal{Y} \cup \mathcal{N}) \cap \{0,1\}^n \neq \emptyset$:*

- *For every $x \in \{0,1\}^n \cap \mathcal{Y}$: $\Pr[\mathsf{A}(x) = 1] \geq 2/3$.*
- *For every $x \in \{0,1\}^n \cap \mathcal{N}$: $\Pr[\mathsf{A}(x) = 1] \leq 1/3$.*

A promise problem is in ioP/poly *if the above holds with respect to $n.u. - poly - time$ algorithm* A*.*

For a set $\mathcal{Q} \subseteq \{0,1\}^*$, we denote by $\mathcal{L}|_\mathcal{Q}$ the promise problem $(\mathcal{Y} \cap \mathcal{Q}, \mathcal{N} \cap \mathcal{Q})$.

2.5 One-Way Functions

We now formally define basic cryptographic primitives. We start with the definition of one-way functions.

Definition 2.8 (One-way function)**.** *A polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ is called a* one-way function *if for every polynomial-time algorithm* A*,*

$$\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{A}(1^n, f(x)) \in f^{-1}(f(x))\right] = \mathrm{neg}(n)$$

Definition 2.9 (Weak one-way function)**.** *Let $m \in$ poly be a polynomial-time computable function. A polynomial-time computable function $f : \{0,1\}^{m(n)} \to \{0,1\}^*$ is called α-*weak one-way function *if for every polynomial-time algorithm* A*, for every large enough n,*

$$\Pr_{x \leftarrow \{0,1\}^{m(n)}}\left[\mathsf{A}(1^n, f(x)) \in f^{-1}(f(x))\right] \leq 1 - \alpha(n)$$

f is a weak one-way function *if it is $1/p$-weak one way function, for some $p \in$ poly.*

Theorem 2.10 (Weak to strong OWFs, [Yao82])**.** *One-way functions exist if and only if weak-one way functions exist.*

2.6 Two-party protocols and Key-Agreement Protocols

For a two-party protocol (A, B), we denote by $(\pi, x, y) \leftarrow (\mathsf{A}(a), \mathsf{B}(b))(z)$ the transcript π, the output of A, x, and the output of B, y, sampled from a random interaction of $\mathsf{A}(a, z)$ and $\mathsf{B}(b, z)$. Such a protocol is t-time, if both A and B run in time at most $t(n)$ on input (a, b, z) of length n. A protocol is PPT if it is t-time for $t \in$ poly. A protocol is deterministic if both A and B are. Two-party protocol is an ℓ-bit protocol if the length of the outputs of the parties in $(\mathsf{A}, \mathsf{B})(1^n)$ is $\ell(n)$.

Definition 2.11 (Key-Agreement)**.** *A poly-time two-party protocol* (A, B) *is a* (α, δ)-key-agreement *protocol if the following holds:*

- α-Agreement: $\Pr_{(\pi,x,y)\leftarrow(\mathsf{A},\mathsf{B})(1^n)}[x = y] \geq \alpha(n)$.
- δ-leakage: *For every poly-time algorithm* Eve, $\Pr_{(\pi,x,y)\leftarrow(\mathsf{A},\mathsf{B})(1^n)}[\mathsf{Eve}(\pi) = x] \leq \delta(n)$.

Such a protocol is a key-agreement *protocol if it is a* $(1 - \mathrm{neg}(n), \mathrm{neg}(n))$*-key-agreement.*

The following lemma shows that it is possible to amplify an 1-bit weak key-agreement protocol into a key-agreement. This lemma is a simple case of the more general result of Holenstein [Hol06].

Lemma 2.12 (Key-agreement amplification, [Hol06])**.** *The following holds for every constants* $\alpha > \beta$*. Assume there exists a* 1*-bit,* $(1 - n^{-\alpha}, 1 - n^{-\beta})$*-key agreement protocol. Then, there exists a key-agreement protocol.*

We use Goldrich-Levin to generalize the above lemma to n-bit protocols. The proof of the following lemma is in Appendix B.

Lemma 2.13. *The following holds for every constants* $\alpha > \beta$*. Assume there exists an* n*-bit,* $(1 - n^{-\alpha}, 1 - n^{-\beta})$*-key agreement protocol. Then, there exists a key-agreement protocol.*

The following well-known fact states that in every two-party protocol with independent inputs, given the transcript there is no dependency between the views of the parties.

Fact 2.14. *Let* (A, B) *be a two-party protocol, and let* $R_{\mathsf{A}}, R_{\mathsf{B}}$ *be independent random variables. Let* $(\Pi, X, Y) \leftarrow (\mathsf{A}(R_{\mathsf{A}}), \mathsf{B}(R_{\mathsf{B}}))$ *be the transcript and outputs of the parties in an execution of* (A, B) *on random inputs. Then, for every* $\pi.r_{\mathsf{A}}$ *and* r_{B}*,*

$$R_{\mathsf{A}}|_{\Pi=\pi,R_{\mathsf{B}}=r_{\mathsf{B}}} \equiv R_{\mathsf{A}}|_{\Pi=\pi} \quad \textit{and} \quad R_{\mathsf{B}}|_{\Pi=\pi,R_{\mathsf{A}}=r_{\mathsf{A}}} \equiv R_{\mathsf{B}}|_{\Pi=\pi}.$$

2.7 Hashing and Extraction

We will use 2-universal families in the proofs.

Definition 2.15 (2-universal family)**.** *A family of function* $\mathcal{F} = \left\{ f \colon \{0,1\}^n \to \{0,1\}^\ell \right\}$ *is* 2-universal *if for every* $x \neq x' \in \{0,1\}^n$ *it holds that* $\Pr_{f\leftarrow\mathcal{F}}[f(x) = f(x')] = 2^{-\ell}$*.*

A universal a family is explicit *if given a description of a function* $f \in \mathcal{F}$ *and* $x \in \{0,1\}^n$*,* $f(x)$ *can be computed in polynomial time (in* n, ℓ*).*

For example, the family of all binary matrices of size $n \times \ell$ is a 2-universal hash family, when thinking on $M(x) = x \cdot M$, for $M \in \{0,1\}^{n \times \ell}$ and $x \in \{0,1\}^n$. An important property of 2-universal families is that they can be used to construct a strong extractor. This is stated in the leftover hash lemma:

Lemma 2.16 (Leftover hash lemma [ILL89])**.** *Let $n \in \mathbb{N}$, $\varepsilon \in [0,1]$, and let X be a random variable over $\{0,1\}^n$. Let $\mathcal{H} = \left\{h \colon \{0,1\}^n \to \{0,1\}^\ell\right\}$ be a 2-universal hash family with $\ell \leq \mathrm{H}_\infty(X) - 2\log 1/\varepsilon$. Then,*

$$\mathrm{SD}((H, H(X)), (H, \mathcal{U}_\ell)) \leq \varepsilon$$

for $\mathcal{U}_\ell$ being the uniform distribution over $\{0,1\}^\ell$ and H being the uniform distribution over $\mathcal{H}$.

The Goldreich-Levin theorem is useful to extract pseudorandomness. We will use the following version of it.

Lemma 2.17 (Goldreich-Levin [GL89; Yao82])**.** *There exists an oracle-aided* PPT A *such that the following holds. Let $n, \ell \in N$ be numbers, and $\mathcal{Q}$ a distribution over $\{0,1\}^n \times \{0,1\}^*$, and let D be an algorithm such that*

$$\left| \Pr_{\substack{(x,z)\leftarrow\mathcal{Q},\\ r=(r_1,\ldots,r_\ell)\leftarrow(\{0,1\}^n)^\ell}} [\mathsf{D}(z, r, \mathrm{GL}(x, r_1), \ldots, \mathrm{GL}(x, r_\ell)) = 1] - \Pr_{\substack{(x,z)\leftarrow\mathcal{Q},\\ r=(r_1,\ldots,r_\ell)\leftarrow(\{0,1\}^n)^\ell}} [\mathsf{D}(z, r, \mathcal{U}_\ell) = 1] \right| \geq \alpha$$

for some α, where $\mathrm{GL}(x, r) := \langle x, r\rangle$ is the Goldreich-Levin predicate. Then

$$\Pr_{(x,z)\leftarrow\mathcal{Q}}\left[\mathsf{A}^{\mathsf{D}}(1^n, 1^\ell, 1^{\lceil 1/\alpha\rceil}, z) = x\right] \geq \mathrm{poly}(\alpha, 2^{-\ell}, 1/n).$$

2.8 Kolmogorov Complexity

We introduce the notion of (time-bounded) Kolmogorov complexity. Roughly speaking, the *t-time-bounded Kolmogorov complexity*, $\mathrm{K}^t(x \mid z)$, of a string $x \in \{0,1\}^*$ conditioned on a string $z \in \{0,1\}^*$ is the length of the shortest program $P = (M, y)$ such that, when simulated by an universal Turing machine, $P(z)$ outputs x in $t(|x|)$ steps. Here, a program P is simply a pair of a Turing Machine M and an input y, where $P(z)$ is defined as $M(y, z)$. When there is no running time bound (i.e., the program can run in an arbitrary number of steps), we obtain the notion of (time-unbounded) Kolmogorov complexity.

Let U be some fixed Universal Turing machine that can emulate any program P with polynomial overhead. Let $\mathsf{U}(P(z), 1^t)$ denote the output of $P(z)$ when emulated on U for t steps. In this paper we fix U to be universal Turing machine with poly log time overhead (That is, when simulating a t-time TM M on input $|y| = n$, $\mathsf{U}((M, y))$ runs in time $O(t(n)\log^c(n))$ for some constant c). We now define the notion of Kolmogorov complexity.

Definition 2.18. *Let t be a polynomial. For all $x \in \{0,1\}^*$ and $z \in \{0,1\}^*$, define*

$$\mathrm{K}^t(x \mid z) = \min_{P\in\{0,1\}^*} \{|P| : \mathsf{U}(P(z), 1^{t(|x|)}) = x\}$$

where $|P|$ is referred to as the description length *of P. When there is no time bound, we define*

$$\mathrm{K}(x \mid z) = \min_{P \in \{0,1\}^*} \{|P| : \mathsf{U}(P(z), 1^{t'}) = x \text{ for some finite } t'\}$$

When there is no condition (i.e., z is an empty string), we simply omit " $\mid z$" and let $\mathrm{K}^t(x)$ (resp $\mathrm{K}(x)$) denote the "plain" t-time-bounded (resp time-unbounded) Kolmogorov complexity of x.

We use $\mathsf{K}(x, y)$ to denote the Kolmogorov complexity of some generic self-delimiting encoding of the pair x, y. Recall that we use $\mathsf{K}(x||y)$ to denote the complexity of the concatenation of x and y. We will use the following well-known facts:

Fact 2.19. *For every $x, y \in \{0,1\}^*$,*

$$\mathsf{K}(x, y) \leq \mathsf{K}(x) + \mathsf{K}(y) + \log(\mathsf{K}(x)) + 2\log\log(\mathsf{K}(x)) + O(1).$$

And more generally,

Fact 2.20 (Chain rule)**.** *For every $x, y \in \{0,1\}^*$,*

$$\mathsf{K}(x, y) \leq \mathsf{K}(x) + \mathsf{K}(y \mid x) + \log(\mathsf{K}(y \mid x)) + 2\log\log(\mathsf{K}(y \mid x)) + O(1).$$

The following lemma bounds the Kolmogorov complexity of an output of randomized algorithm. The proof is by following the proof of the well-known Symmetry of Information theorem to get more tight constants in our settings.

Lemma 2.21. *Let $x, y \in \{0,1\}^*$, and assume that there is an algorithm D (that halts on every input), a polynomial $t(n) = n^c$, and a function $f_y \colon \{0,1\}^{t(|x|)} \to \{0,1\}^\ell$ such that*

$$\Pr_{r \leftarrow \{0,1\}^{t(|x|)}}[D(x, r, f_y(r)) = y] > 1/4.$$

Then,

$$\mathsf{K}(x, y) \leq \mathsf{K}(x) + \ell + \log(\ell) + 2\log\log(\ell) + O(|D|) + O(c).$$

Proof. Consider the following algorithm D'.

Algorithm 2.22 (D')**.**

Input: Algorithm D, $c \in \mathbb{N}$, $\ell \in \mathbb{N}$, $i \in [2^{\ell+2}]$ and $x \in \{0,1\}^*$.

Operation:

1. *Compute $t = t(|x|)$, and for every $r \in \{0,1\}^t$ and $z \in \{0,1\}^\ell$, execute $D(x, r, z)$.*
2. *Let $\mathcal{S}$ be the list of every outputs y' such that $\left|\left\{r \in \{0,1\}^t \colon \exists z \text{ s.t. } D(x, r, z) = y'\right\}\right| > 1/4 \cdot 2^t$, ordered according to the lexicographic order.*
3. *output the i-th element in $\mathcal{S}$.*

Note that the input to D' can be encoded using $O(D)+O(c)+\log\ell+2\log\log\ell+\ell+O(1)+\mathsf{K}(x)$ bits. To see that the lemma holds, it is enough to see that y is among the $4\cdot 2^\ell$ first elements in $\mathcal{S}$. By assumption, y is in the set $\mathcal{S}$. Moreover, since for every fixed r there are at most 2^ℓ different values of z, and thus at most 2^ℓ outputs of D, we get that the size of $\mathcal{S}$ is at most

$$|\mathcal{S}| \leq \frac{2^t \cdot 2^\ell}{2^{t-2}} = 4 \cdot 2^\ell.$$

□

We will also use the following bound on the Kolmogorov complexity of strings sampled from distributions with high min-entropy.

Lemma 2.23. *For every $n \in \mathbb{N}$, and every distribution $\mathcal{D}$, it holds that*

$$\Pr_{x\leftarrow\mathcal{D}}[\mathsf{K}(x) \geq \mathrm{H}_\infty(\mathcal{D}) - \log n] \geq 1 - 1/n.$$

3 Interactive Kolmogorov Complexity

To define interactive Kolmogorov complexity, we consider an interactive universal TM U. Given two programs $P_\mathsf{A} = (M_\mathsf{A}, w_\mathsf{A})$ and $P_\mathsf{B} = (M_\mathsf{B}, w_\mathsf{B})$, and a bound t on the number of steps, let $(\mathsf{U}(P_\mathsf{A}, 1^t), \mathsf{U}(P_\mathsf{B}, 1^t))$ denote the interaction between $\mathsf{U}(P_\mathsf{A}, 1^t)$ and $\mathsf{U}(P_\mathsf{B}, 1^t)$: The interaction is carried out in rounds. In odd numbered rounds, A is active and B is idle, and vice versa in even numbered round. In each round, the active program can read a received bit from the last round, perform some computation and then output a bit, at which point we move on to the next round.

The emulation of each program stops when the program either halts or reaches t steps; when the first program reach the bound t on the number of steps, a special symbol is sent to the other program, that can then run (without sending additional messages) until it halts or also reaches t steps. The transcript of the interaction is the concatenation of all bits sent during the emulation (excluding the special halt symbol). The output of A (B resp.) is the output of $\mathsf{U}(P_\mathsf{A}, 1^t)$ ($\mathsf{U}(P_\mathsf{B}, 1^t)$ resp.) at the end of the emulation.

Definition 3.1 (IK^t)**.** *For a function $t\colon \mathbb{N} \to \mathbb{N}$, and $(\pi, x, y) \in \{0,1\}^*$, the t-bounded interactive Kolmogorov complexity of π, x, y, denoted by $\mathsf{IK}^t(\pi; x; y)$, is the minimal number $\ell \in \mathbb{N}$, such that the following holds for programs P_A and P_B with $|P_\mathsf{A}| + |P_\mathsf{B}| = \ell$. In the interaction $(\mathsf{U}(P_\mathsf{A}, 1^t), \mathsf{U}(P_\mathsf{B}, 1^t))$, the transcript is π and the outputs of P_A and P_B are x and y resp.*

Consider the following promise problem, $\mathsf{RIK}^t\mathsf{P}$:

Definition 3.2 ($\mathsf{RIK}^t\mathsf{P}$)**.** *(Relative IK^t problem) For functions $\sigma_Y < \sigma_N$ and t, let $\mathsf{RIK}^t\mathsf{P}[\sigma_Y, \sigma_N]$ denote the following promise problem:*

- $\mathcal{Y} = \left\{(\pi, x, y) \in (\{0,1\}^n)^3 \colon \mathsf{IK}^t(\pi; x; y) \leq \mathsf{K}(\pi) + \sigma_Y\right\}$,
- $\mathcal{N} = \left\{(\pi, x, y) \in (\{0,1\}^n)^3 \colon \mathsf{K}(\pi, x, y) \geq \mathsf{K}(\pi) + \sigma_N\right\}$.

The following lemma shows that for the right choice of parameters, the above is a well-defined promise problem (that is, that $\mathcal{Y} \cap \mathcal{N} = \emptyset$).

Lemma 3.3. *For every t, large enough n and $(\pi, x, y) \in (\{0,1\}^n)^3$,*

$$\mathsf{K}(\pi, x, y) \leq \mathsf{IK}^t(\pi; x; y) + 2\log n.$$

Proof. Let P_{A} and P_{B} be the programs with $|P_{\mathsf{A}}| + |P_{\mathsf{B}}| = \mathsf{IK}^t(\pi; x; y)$ that, when interacting, produce π as the transcript and x, y as the outputs. Then, by Fact 2.19,

$$\begin{aligned}
&\mathsf{K}(\pi, x, y) \\
&\leq |P_{\mathsf{A}}| + |P_{\mathsf{B}}| + \log|P_{\mathsf{A}}| + 2\log\log|P_{\mathsf{A}}| + O(1) \\
&\leq |P_{\mathsf{A}}| + |P_{\mathsf{B}}| + \log 2n + 2\log\log 2n + O(1) \\
&\leq |P_{\mathsf{A}}| + |P_{\mathsf{B}}| + 2\log n,
\end{aligned}$$

where the second inequality holds since by assumption P_{A} is the minimal program that, when interacting with P_{B}, the transcript and output of P_{A} are π, x. Thus, $|P_{\mathsf{A}}| \leq 2n + O(1)$. □

We also define the following condition on the input, in which x and y are close to each other. For a function $\Delta\colon \mathbb{N} \to \mathbb{N}$, let

$$Q_\Delta = \big\{(\pi, x, y) \in (\{0,1\}^n)^3 \colon \mathsf{K}^t(x \mid y) \leq \Delta(n), \mathsf{K}^t(y \mid x) \leq \Delta(n)\big\}.$$

Observe that when $\Delta = O(\log n)$ and $t \in \text{poly}$, Q_Δ can be decided in polynomial time. By defining $\mathsf{K}^t(x \mid x) = 0$ for every $x \in \{0,1\}^*$, we get that

$$Q_0 = \big\{(\pi, x, y) \in (\{0,1\}^n)^3 \colon x = y\big\}.$$

3.1 Characterization of Key-Agreement and One-Way Functions

We now state our main theorems. The first theorem characterize the existence of key-agreement protocols.

Theorem 3.4 (KA characterization)**.** *The following are equivalent for every constants $\epsilon > 0$, $d \geq 0$, $c_1 > 3$ and c_2 with $c_2 - c_1 > 9 + 2d$, for every polynomial $t(n) > n^{1+\epsilon}$, and for $\Delta(n) = d\log n$:*

1. *Key-agreement protocols exist.*
2. $\mathsf{RIK}^t\mathsf{P}[c_1\log n, c_2\log n]|_{Q_\Delta} \notin \mathsf{ioBPP}$.

The proof that Item 2 implies Item 1 is given in Section 4. The proof that Item 1 implies Item 2 is given in Section 5.

Theorem 3.5 (OWFs characterization)**.** *The following are equivalent for every constants $\epsilon > 0$, $c_1 \geq 0$, c_2 and d with $c_2 - c_1 > 11$ and $d > c_2 + 4$, for every polynomial $t(n) \geq n^{1+\epsilon}$, and for every function $\Delta(n) \geq d\log n$:*

1. *One-way functions exist.*
2. $\mathsf{RIK}^t\mathsf{P}[c_1\log n, c_2\log n]|_{Q_\Delta} \notin \mathsf{ioBPP}$.

The proof that Item 2 implies Item 1 is given in Section 6. The proof that Item 1 implies Item 2 is given in Section 7.

4 Worst-Case hardness of $\mathsf{RIK}^t\mathsf{P} \implies \mathbf{KA}$

In this section we prove the following theorem, that states that the worst-case hardness of $\mathsf{RIK}^t\mathsf{P}|_{Q_\Delta}$ implies the existence of key-agreements protocols.

Theorem 4.1. *Let c_1, c_2 and $d \geq 0$ be constants such that $0 \leq c_1 < c_2 - 9 - 2d$, let $\Delta(n) = d \log n$, and let $t(n)$ be a polynomial. Then the following holds: If $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]|_{Q_\Delta} \notin \mathsf{ioBPP}$, key-agreement protocols exist.*

To prove the theorem, fix $t = t(n)$, c_1, c_2, d and Δ as in Theorem 4.1, and let $c = c_2 - 2 - d$. For simplicity, in the following, for a string $x \in \{0,1\}^*$, let $(x)_n$ be the first n bits of $x0^n$ (that is, an n bits string containing the first n bits of x, or appropriate padding if necessary). Let $\mathcal{H} = \left\{h \colon \{0,1\}^n \to \{0,1\}^{c \log n}\right\}$ be a 2-universal family, and consider the following protocol:

Protocol 4.2 ((A, B))**.**

Parameter: function $t \colon \mathbb{N} \to \mathbb{N}$, $c \in \mathbb{N}$.

Input: 1^n.

Operation:

1. A *samples* $\ell_\mathsf{A} \leftarrow [n]$ *and* $P_\mathsf{A} \leftarrow \{0,1\}^{\ell_\mathsf{A}}$, B *samples* $\ell_\mathsf{B} \leftarrow [n]$ *and* $P_\mathsf{B} \leftarrow \{0,1\}^{\ell_\mathsf{B}}$. B *samples* $\ell_C \in [\Delta(n)]$, *and* $P_C \in \{0,1\}^{\ell_C}$.
2. A *and* B *interact according to* $(\mathsf{U}(P_\mathsf{A}, 1^{t(n)}), \mathsf{U}(P_\mathsf{B}, 1^{t(n)}))$. *Let π be the transcript of the interaction, and let x, y be the outputs of* A *and* B, *respectively.*
3. B *executes* $\mathsf{U}(P_C(y), 1^{t(n)})$ *to get the output y'. Let $\widehat{x} = (x)_n$ and $\widehat{y} = (y')_n$.*
4. A *samples a hash function $h \leftarrow \mathcal{H}$, and sends $h, h(\widehat{x})$. If $h(\widehat{y}) \neq h(\widehat{x})$,* B *sends* 0 *and the parties output 0^n. Otherwise,* B *sends* 1, *and the parties output $\widehat{x}$ and $\widehat{y}$ respectively.*

Let o_A and o_B denote the output of A and B in the above protocol, respectively. The transcript of the above protocol is of the form $\widetilde{\pi} = (\pi, h, h(\widehat{x}), b)$ for $b \in \{0,1\}$. Below we bound the agreement probability and the leakage of the above protocol. In the following, let $\widetilde{\Pi}$ be the random variable that distributed according to the transcript of Protocol 4.2, and let $\Pi, X, \widehat{X}, Y, \widehat{Y}, H, O_\mathsf{A}$ and O_B be the random variables distributed according to $\pi, x, \widehat{x}, y, \widehat{y}, o_\mathsf{A}$ and o_B, respectively.

4.1 Agreement

We start by analyzing the agreement probability of Protocol 4.2.

Lemma 4.3. *For every $n \in \mathbb{N}$, it holds that $\Pr[O_\mathsf{A} = O_\mathsf{B}] \geq 1 - n^{-c}$.*

Proof. The only events in which A and B do not agree, is when $\widehat{x} \neq \widehat{y}$ but $h(\widehat{x}) = h(\widehat{y})$. by the 2-universal property of $\mathcal{H}$, for every such $\widehat{x} \neq \widehat{y}$, $\Pr_h[h(\widehat{x}) = h(\widehat{y})] = n^{-c}$. Thus,

$$\begin{aligned}
\Pr[O_\mathsf{A} = O_\mathsf{B}] &= 1 - \Pr\left[\widehat{X} \neq \widehat{Y}, H(\widehat{X}) = H(\widehat{Y})\right] \\
&= 1 - \Pr\left[\widehat{X} \neq \widehat{Y}\right] \Pr\left[H(\widehat{X}) = H(\widehat{Y}) \mid \widehat{X} \neq \widehat{Y}\right] \\
&\geq 1 - n^{-c}.
\end{aligned}$$

□

4.2 Security

We next bound the leakage of Protocol 4.2.

Lemma 4.4. *Assume there exists an algorithm* Eve *such that* $\Pr\left[\mathsf{Eve}(1^n, \widetilde{\Pi}) = O_{\mathsf{A}}\right] \geq 1 - n^{-c+1}$ *for infinitely many* n*'s. Then* $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]|_{Q_\Delta} \in \mathsf{ioBPP}$.

Consider the following algorithm, that given an attacker Eve that guesses the output of Protocol 4.2 with too good probability, decides $\mathsf{RIK}^t\mathsf{P}|_{Q_\Delta}$. In the following we only focusing on the executions of Protocol 4.2 in which $|x| = |y| = |\pi| = n$ (and show this is enough). In this case, $\widehat{x} = x$.

Algorithm 4.5.

Input: $\pi \in \{0,1\}^n, x \in \{0,1\}^n, y \in \{0,1\}^n$.

Operation:

1. *Sample* h *and let* $\pi' = (\pi, h, h(x), 1)$.
2. *Execute* $\mathsf{Eve}(1^n, \pi')$ *to get* x'.
3. *If* $x' = x$ *answer "Yes". Otherwise answer "No".*

That is, given π, x, y (and assuming that $(\pi, x, y) \in Q_\Delta$), Algorithm 4.5 simply checks if Eve outputs x on the input $\pi' = (\pi, h, h(x), 1)$. In the following, Let $\mathcal{Y}$ and $\mathcal{N}$ be the Yes and No instances of the problem $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]$, respectively. Similarly, let $\mathcal{Y}|_{Q_\Delta}$ and $\mathcal{N}|_{Q_\Delta}$ be the Yes and No instances of the problem $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]|_{Q_\Delta}$.

Proof of Lemma 4.4. Assume there exists an algorithm Eve such that $\Pr\left[\mathsf{Eve}(1^n, \widetilde{\Pi}) = O_{\mathsf{A}}\right] \geq 1 - n^{-c+1}$ for infinitely many n's. We will show that Algorithm 4.5 decides $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]|_{Q_\Delta}$ for every such n. In the following, fix $n \in \mathbb{N}$ with $\Pr\left[\mathsf{Eve}(1^n, \widetilde{\Pi}) = O_{\mathsf{A}}\right] \geq 1 - n^{-c+1}$.

Soundness: We now show that for every No instance, Algorithm 4.5 answer "Yes" with probability at most 1/3. To do so, first notice that, by definition, for every $(\pi, x, y) \in \{0,1\}^{3n}$ for which Algorithm 4.5 answer "Yes" with probability at least 1/3, the following holds: With probability at least 1/3 over the choice of h and randomness r_E for Eve, $\mathsf{Eve}(1^n, (\pi, h, h(x), 1); r_E) = x$. We can now finish the proof using the Symmetry of Information theorem [Zvo]. However, to get better parameters, we instead use Lemma 2.21. Let $D(n, \pi, h(x), h, r_E) = \mathsf{Eve}(1^n, (\pi, h, h(x), 1); r_E)$, and define $f_x(h, r_E) = h(x)$. By Lemma 2.21, for every large enough n and assuming that $(\pi, x, y) \in Q_\Delta$,

$$\begin{aligned}
\mathsf{K}(\pi, x, y) &\leq \mathsf{K}(n, \pi, x, y) \\
&\leq \mathsf{K}(n, \pi, x) + \mathsf{K}(y \mid x) + 2\log \mathsf{K}(y \mid x) \\
&\leq \mathsf{K}(n, \pi) + |h(x)| + \log|h(x)| + 2\log\log|h(x)| + O(|D|) + \Delta(n) + 2\log(\Delta(n)) \\
&\leq \mathsf{K}(\pi) + \log n + 2\log\log n + |h(x)| + 2\log|h(x)| + O(1) + \Delta(n) + 2\log(\Delta(n)) \\
&< \mathsf{K}(\pi) + c\log n + 2\log n + \Delta(n) \\
&= \mathsf{K}(\pi) + c_2 \log n,
\end{aligned}$$

where the second equality holds by Fact 2.20, and the third inequality holds by Fact 2.19 and the fact that $\log n \geq 2\log\log n$ for large enough n. Namely, by the above (π, x, y) is not in $\mathcal{N}$. We get that for every No instance $(\pi, x, y) \in \mathcal{N}|_{Q_\Delta}$, Algorithm 4.5 answer "no" with probability smaller than 1/3.

Completeness: For the other direction, let $\beta = n^{-c+1}$, and assume toward a contradiction that Algorithm 4.5 does not output "Yes" with probability at least 2/3 on every input $(\pi, x, y) \in \{0,1\}^{3n}$ from $\mathcal{Y}|_{Q_\Delta}$. Let $\mathcal{S}$ be the set of $(\pi, x, y) \in Q_\Delta \cap \{0,1\}^{3n}$ on which Algorithm 4.5 answers "No" with probability at least 1/3. By definition of Algorithm 4.5, for every $(\pi, x, y) \in \mathcal{S}$ it holds that that

$$\Pr_{h, r_E}[\mathsf{Eve}(1^n, (\pi, h, h(x), 1); r_E) \neq x] \geq 1/3. \tag{1}$$

Moreover, for every $(\pi, x, y) \in Q_\Delta$ with $\mathsf{IK}^t(\pi; x; y) = \ell$, it holds that in a random run of the protocol,

$$\Pr[(\Pi, X, Y) = (\pi, x, y), O_{\mathsf{A}} = x] \geq 1/n^3 \cdot 2^{-\ell - \Delta(n)}. \tag{2}$$

Indeed, let $P_{\mathsf{A}}^{\pi,x,y}$ and $P_{\mathsf{B}}^{\pi,x,y}$ be the programs of length $\left|P_{\mathsf{A}}^{\pi,x,y}\right| + \left|P_{\mathsf{B}}^{\pi,x,y}\right| = \ell$ that realize the t-bounded interactive Kolmogorv complexity of π, x, y (that is, the program that when interact produce π as transcript and x, y as outputs), and let $P_C^{\pi,x,y}$ be the program of length $\mathsf{K}^t(x \mid y) \leq \Delta(n)$ that realize the conditional t-bounded Kolmogorov complexity of x given y. Then, in Step 2 of Protocol 4.2, with probability at least $1/n^3$ it holds that $\ell_{\mathsf{A}} = |P_{\mathsf{A}}|$, $\ell_{\mathsf{B}} = |P_{\mathsf{B}}|$, and $\ell_C = |P_C|$. Given this event, with probability $2^{-\ell_{\mathsf{A}} - \ell_{\mathsf{B}} - \ell_C}$ it holds that $P_{\mathsf{A}} = P_{\mathsf{A}}^{\pi,x,y}$, $P_{\mathsf{B}} = P_{\mathsf{B}}^{\pi,x,y}$, and $P_C = P_C^{\pi,x,y}$. Finally, since $P_C(y) = x$, with the above probability A and B agree, and $h(\widehat{x}) = h(\widehat{y})$ (that is, A's output is x).

For every $\ell \in \mathbb{N}$, let $\mathcal{S}_\ell = \left\{(\pi, x, y) \in \mathcal{S} \colon (\pi, x, y) \in \{0,1\}^{3n}, \mathsf{IK}^t(\pi; x; y) = \ell\right\}$, and fix ℓ^* such that the algorithm fails on $\mathcal{S}_{\ell^*} \cap \mathcal{Y}|_{Q_\Delta}$. Combining Equations (1) and (2) get that,

$$\begin{aligned}
\beta &\geq \Pr\left[\mathsf{Eve}(1^n, \widetilde{\Pi}) \neq O_{\mathsf{A}}, O_{\mathsf{A}} = X\right] \\
&\geq 1/3 \cdot \Pr[(\Pi, X, Y) \in \mathcal{S}, O_{\mathsf{A}} = X] \\
&\geq \Omega(1/n^3 \cdot 2^{-\ell^* - \Delta(n)} \cdot |\mathcal{S}_{\ell^*}|),
\end{aligned}$$

and thus,

$$|\mathcal{S}_{\ell^*}| \leq \Omega(n^3 \cdot \beta \cdot 2^{\ell^* + \Delta(n)})$$

On the other hand, there exists an algorithm that given n, ℓ and $i \in [|\mathcal{S}_\ell|]$, output the i-th element in $\mathcal{S}_\ell$ according to the lexicographic order (by enumerating over all the possible values of $(\pi, x, y) \in Q_\Delta \cap (\{0,1\}^n)^3$ with $\mathsf{IK}^t(\pi; x; y) = \ell$ and computing the failure probability of Algorithm 4.5). Thus, by Fact 2.19, for every $(\pi, x, y) \in \mathcal{S}_{\ell^*}$ it holds that

$$\begin{aligned}
\mathsf{K}(\pi) &\leq \mathsf{K}(n, \pi, x, y) \\
&\leq \log n + 2\log\log n + \log \ell^* + 2\log\log \ell^* + \log|\mathcal{S}_{\ell^*}| + O(1) \\
&\leq 3\log n + (3\log n + \log\beta + \ell^* + \Delta(n)) \\
&= 6\log n + \ell^* + (1-c)\log n + \Delta(n) \\
&< \ell^* - c_1 \log n
\end{aligned}$$

where the last inequality holds since $c > c_1 + d + 7$, and $\Delta(n) = d \log n$. The above implies that $\mathcal{S}_{\ell^*} \cap \mathcal{Y} = \emptyset$, and thus $\mathcal{S}_{\ell^*} \cap \mathcal{Y}|_{Q_\Delta} = \emptyset$. Namely, Algorithm 4.5 has no error. □

4.3 Proving Theorem 4.1.

We are now ready to use Lemma 2.12 in order to prove Theorem 4.1.

Proof of Theorem 4.1. By Lemmas 4.3 and 4.4, Protocol 4.2 has agreement $1 - n^{-c}$ and leakage $1 - n^{-c+1}$. Thus, by Lemma 2.13, Protocol 4.2 can be amplified into a key-agreement protocol. □

Remark 4.6 (The non-uniform setting)**.** *A similar theorem can be proven when assuming that* $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]|_{Q_\Delta} \notin \mathsf{ioP/poly}$*, and when the key-agreement is secure against non-uniform adversaries. In this case, we assume that* Eve *is a non-uniform algorithm that breaks the key-agreement protocol, and want to construct a non-uniform (randomized) algorithm that decides* $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]|_{Q_\Delta}$*.*

The issue with the above proof is that we cannot simply use Eve *to bound the Kolmogorov complexity of* π, x, y *as done in the proof of Lemma 4.4, as* Eve *does not have constant size. However, we can find* Eve *using a small Turing machine: Let* M *be the (inefficient) Turing machine that, given a constant* c *such that* n^c *is a bound on the size of* Eve*, and input* $(1^n, \widetilde{\pi})$*, first find the circuit* E'_n *of size at most* n^c *that maximize the advantage in predicting the output of* A *in Protocol 4.2, and then execute* $E'(1^n, \widetilde{\pi})$*. Observe that* M *has prediction advantage at least as the advantage of* Eve*. The theorem now follows using the same proof, by replacing* Eve *in the proof of Lemma 4.4 with* M*, and replacing* Eve *in Algorithm 4.5 with* $E' = \{E'_n\}_{n \in \mathbb{N}}$*.*

5 KA $\implies$ Hardness of $\mathsf{RIK}^t\mathsf{P}$

In this part, it is shown that if a key-agreement protocol exists, $\mathsf{RIK}^t\mathsf{P}$ is hard. We prove the following theorem.

Theorem 5.1. *Assume there exists a key-agreement protocol. Then for any* $\epsilon > 0$*, any* $t(n) \geq n^{1+\epsilon}$*, any* $\Delta(n) \geq 0$ *and for every constants* $c_2 > c_1 > 3$*,* $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]|_{Q_\Delta} \notin \mathsf{ioBPP}$*.*

To prove Theorem 5.1, we will need the following definition of conditional entropy-preserving key-agreement protocol, in which the min-entropy of the transcript is almost equal to the amount of randomness used by the parties. In the following, randomized event is an event that can be dependent in additional random variables, jointly distributed with the inputs of the parties of the protocol (see Lemma 2.3 for an example).

Definition 5.2 (Conditional entropy-preserving key-agreement (Cond-EP KA))**.** *Let* $s^{\mathsf{A}} = s^{\mathsf{A}}(n), s^{\mathsf{B}} = s^{\mathsf{B}}(n)$ *be efficiently computable functions. A deterministic two-party protocol* (A, B) *is* $(d, s^{\mathsf{A}}, s^{\mathsf{B}})$-cond-EP KA *if the following holds. For every* $n \in \mathbb{N}$*, let* (Π_n, X_n, Y_n) *be the distribution of the transcript and the outputs in the interaction* $(\mathsf{A}(\mathcal{U}_{s^{\mathsf{A}}(n)}), \mathsf{B}(\mathcal{U}_{s^{\mathsf{B}}(n)}))(1^n)$*. Then there exists a sequence of randomized events* $\{E_n\}_{n \in \mathbb{N}}$ *such that:*

1. Agreement: $\Pr[X_n = Y_n \mid E_n] = 1$
2. Secrecy: *For every poly-time algorithm* D*,*

$$\left|\Pr[\mathsf{D}(1^n, \Pi_n, X_n) = 1 \mid E_n] - \Pr\left[\mathsf{D}(1^n, \Pi_n, \mathcal{U}_{|X_n|}) = 1 \mid E_n\right]\right| = \mathrm{neg}(n).$$

3. Entropy: $\mathrm{H}_\infty(\Pi_n \mid E_n) \geq s^{\mathsf{A}}(n) + s^{\mathsf{B}}(n) - d\log n$.

A protocol is d-cond-EP KA *if it is* $(d, s^{\mathsf{A}}, s^{\mathsf{B}})$*-cond-EP KA for some functions* $s^{\mathsf{A}}, s^{\mathsf{B}}$.

We say that such a protocol has transcript of length n if $\Pr[|\Pi_n| = n] = 1$. The main lemma in this part states that the conclusion of Theorem 5.1 holds assuming that conditional entropy-presrving key-agreement protocol exists.

Lemma 5.3. *Let* $c_2 > c_1 > 2.1$ *and* $\epsilon > 0$ *be constant. Assume there is a* 0.1*-Cond-EP KA protocol with key of length* $(c_2+2)\log n$*, transcript of length* n*, and running time* $n^{1+\epsilon}$*. Then* $\mathsf{RIK}^t\mathsf{P}[c_1\log n, c_2\log n]|_{Q_\Delta} \notin \mathsf{ioBPP}$ *for every* $t(n) \geq 2n^{1+\epsilon}$*,* $\Delta(n) \geq 0$*, and for every constants* $c_2 > c_1 > 2$.

Proof of Lemma 5.3. Let c_1, c_2, ϵ and Δ be as in Lemma 5.3, and let (A, B) be a $(0.1, s^{\mathsf{A}}, s^{\mathsf{B}})$-cond EP KA protocol with key-length, transcript-length and running time as stated in Lemma 5.3. In the following, assume toward a contradiction that $\mathsf{RIK}^t\mathsf{P}[c_1\log n, c_2\log n]|_{Q_\Delta} \in \mathsf{ioBPP}$, and let M be the algorithm that, for infinitely many n's, decides $\mathsf{RIK}^t\mathsf{P}[c_1\log n, c_2\log n]|_{Q_\Delta}$ with error probability at most 0.01 on every input of length $3n$. We will show that M contradicts the secrecy assumption of (A, B).

To do so, let $s(n) = s^{\mathsf{A}}(n) + s^{\mathsf{B}}(n)$, and for every n, let Π_n, X_n, Y_n be the distribution of the transcript and outputs in a random execution of the protocol (A, B), and let E_n be the event promised by Definition 5.2. Let $\Pi_n|_{E_n}, X_n|_{E_n}$ be the joint distribution of Π_n, X_n conditioned on E_n. Define $\Pi'_n = \Pi_n|_{E_n}$, and let X'_n be equal to $(X_n|_{E_n}||0^{n-|X|})$. Let $Z'_n \leftarrow (\{0,1\}^{|X|}||0^{n-|X|})$. By the secrecy assumption, no algorithm can distinguish between (Π'_n, X'_n) and (Π'_n, Z_n) noticeable advantage. Fix n such that M succeed. We start by bounding the Kolmogorov complexity of Π'.

By definition, $\mathrm{H}_\infty(\Pi') = \mathrm{H}_\infty(\Pi \mid E_n) \geq s^{\mathsf{A}}(n) + s^{\mathsf{B}}(n) - 0.1\log n$. Thus, by Lemma 2.23, with probability at least 0.99 over $\pi \leftarrow \Pi'_n$, it holds that

$$\mathsf{K}(\pi) \geq s^{\mathsf{A}}(n) + s^{B}(n) - 0.1\log n - O(1). \tag{3}$$

Moreover, since π is an transcript of the interaction of (A, B), it can be compressed by encoding n, and the randomness used in the interaction. Thus, by Fact 2.19,

$$\mathsf{K}(\pi) \leq s^{\mathsf{A}}(n) + s^{\mathsf{B}}(n) + \log n + 2\log\log n + O(1).$$

We now use M to construct an algorithm D that breaks the secrecy assumption of (A, B). Let D be the algorithm that given π, z outputs $M(\pi, z0^{|\pi|-|z|}, z0^{|\pi|-|z|})$. We want to show that,

$$\left|\Pr[\mathsf{D}(\Pi_n, X_n) = 1 \mid E_n] - \Pr\left[\mathsf{D}(\Pi_n, \mathcal{U}_{|X_n|}) = 1 \mid E_n\right]\right| \geq 1/2. \tag{4}$$

Observe that the distribution of Π'_n, X'_n is exactly equal to the distribution of $(\Pi_n, X_n||0^{n-|X_n|})|_{E_n}$. Moreover, by Equation (3) with probability at least 0.99 over $(\pi, x) \leftarrow (\Pi'_n, X'_n)$ it holds that,

$$\mathsf{IK}^t(\pi; x0^{n-|x|}; x0^{n-|x|}) \leq 2(\log n + 2\log\log n) + s^{\mathsf{A}}(n) + s^{\mathsf{B}}(n) + O(1) \leq \mathsf{K}(\pi) + c_1\log n,$$

where the above holds by our choice of c_1 and by Fact 2.19, since given n and the right randomness, A and B output π, x, x (and since padding x with $0^{n-|x|}$ can be done in time at most $t/2$). Thus,

it holds that a sample from (Π'_n, X'_n, X'_n) is an Yes instances with probability at least 0.99. By the assumption that the error probability of M is at most 0.01, it holds that,

$$\begin{aligned}\Pr[\mathsf{D}(\Pi_n, X_n) = 1 \mid E_n] &= \Pr\big[M(\Pi'_n, X'_n, X'_n) = 1\big] \\ &\geq \Pr\big[M(\Pi'_n, X'_n, X'_n) = 1 \mid (\Pi'_n, X'_n, X'_n) \in \mathcal{Y}|_{Q_\Delta}\big] - 0.01 \geq 0.98.\end{aligned} \tag{5}$$

On the other hand, the distribution of Π'_n, Z'_n is equal to the distribution $(\Pi_n|_{E_n}, \mathcal{U}_{|X_n|}||0^{n-|X_n|})$. Since $\mathrm{H}_\infty(\Pi_n|_{E_n}, \mathcal{U}_{|X_n|}) \geq s(n) - 0.1\log n + (c_2+2)\log n$ (recall that $|X_n| = (c_2+2)\log n$), it holds by Lemma 2.23 that, with probability at least 0.99 over the choice of $\pi \leftarrow \Pi|_{E_n}$ and $z \leftarrow \{0,1\}^{|X_n|}$,

$$\mathsf{K}(\pi, z0^{n-|z|}, z0^{n-|z|}) \geq s(n) - 0.01\log n + (c_2+2)\log n - O(1) \geq \mathsf{K}(\pi) + c_2 \log n,$$

and thus a sample from (Π'_n, Z'_n, Z'_n) is a No instance with probability at least 0.99. By the assumption that the error probability of M is at most 0.01, it holds that,

$$\Pr\big[\mathsf{D}(\Pi_n, \mathcal{U}_{|X_n|}) = 1 \mid E_n\big] \leq \Pr\big[M(\Pi'_n, Z'_n, Z'_n) = 1 \mid (\Pi'_n, Z'_n, Z'_n \in \mathcal{N}|_{Q_\Delta}\big] + 0.01 \leq 0.02. \tag{6}$$

Combining Equations (5) and (6) yields Equation (4). □

5.1 Key-Agreement to weak-cond-EP Key-agreement

In the rest of this section we show how to constract conditional entropy-preserving key-agreement from a key-agreement protocol. We start by constructing a weaker form of cond-EP KA, defined below, in which instead of requeiring that the secret key is indistinguishable from uniform, we only require it to be unpredictable.

Definition 5.4 (Weak-Cond-EP KA)**.** *Let $s^{\mathsf{A}} = s^{\mathsf{A}}(n), s^{\mathsf{B}} = s^{\mathsf{B}}(n)$ be efficiently computable functions. A detrministic two-party protocol (A, B) is $(d, s^{\mathsf{A}}, s^{\mathsf{B}})$-*weak-cond-EP KA *if the following holds. For every $n \in \mathbb{N}$, let (Π_n, X_n, Y_n) be the distribution of the transcript and the outputs in the interaction $(\mathsf{A}(\mathcal{U}_{s^{\mathsf{A}}(n)}), \mathsf{B}(\mathcal{U}_{s^{\mathsf{B}}(n)}))(1^n)$. Then there exists a sequence of randomized events $\{E_n\}_{n\in\mathbb{N}}$ such that:*

1. Agreement: $\Pr[X_n = Y_n \mid E_n] = 1$
2. Secrecy: *For every poly-time algorithm* Pred, $\Pr[\mathsf{Pred}(1^n, \Pi_n) = X_n \mid E_n] = neg(n)$.
3. Entropy: $\mathrm{H}_\infty(\Pi_n \mid E_n) \geq s^{\mathsf{A}}(n) + s^{\mathsf{B}}(n) - d\log n$.

A protocol is weak-cond-EP KA *if it is $(d, s^{\mathsf{A}}, s^{\mathsf{B}})$-weak-cond-EP KA for some $d \in \mathbb{N}$, and some functions $s^{\mathsf{A}}, s^{\mathsf{B}}$.*

We now prove the next lemma, that states that weak-cond EP key-agreement can be constructed from a key-agreement protocol.

Lemma 5.5 (KA to weak-Cond-EP KA)**.** *Assume there exists a key-agreement protocol. Then, there exists a weak-cond-EP KA.*

In the following, let (A,B) be a t-time key agreement protocol, for $t \in \text{poly}$. Assume that A and B get $t(n)$ random bits as input (and are detrministic algorithms). We claim that the next protocol $(\widehat{\mathsf{A}},\widehat{\mathsf{B}})$ is a weak-cond-EP protocol. Let $\mathcal{H}_{t(n)} = \left\{h\colon \{0,1\}^{t(n)} \to \{0,1\}^{t(n)}\right\}$ be an explicit 2-universal family, such that, for every $i \leq t(n)$, the family $\mathcal{H}_i = \left\{h_{\leq i}\colon h \in \mathcal{H}_{t(n)}\right\}$ is a 2-universal family, for $h_{\leq i}(x) = h(x)_{\leq i}$ (for example, the family of all matrices of size $t(n) \times t(n)$). We further assume without loss of generality that the description size of $h \in \mathcal{H}_{t(n)}$ is $\log(\left|\mathcal{H}_{t(n)}\right|)$ (that is, sampling $h \in \mathcal{H}$ is equivalent to sample $|h|$ uniform bits).

Protocol 5.6 ($(\widehat{\mathsf{A}},\widehat{\mathsf{B}})$)**.**

Common input: 1^n.

$\widehat{\mathsf{A}}$*'s input:* $r^{\mathsf{A}} \in \{0,1\}^{t(n)}$, $z^{\mathsf{A}} \in [t(n)]$, $h \in \mathcal{H}_{t(n)}$.

$\widehat{\mathsf{B}}$*'s input:* $r^{\mathsf{B}} \in \{0,1\}^{t(n)}$, $z^{\mathsf{B}} \in [t(n)]$.

Operation:

1. $\widehat{\mathsf{A}}$ *and* $\widehat{\mathsf{B}}$ *interact according to* $(\mathsf{A}(r^{\mathsf{A}}),\mathsf{B}(r^{\mathsf{B}}))(1^n)$.
2. $\widehat{\mathsf{A}}$ *sends* h *to* $\widehat{\mathsf{B}}$ *(in* $2|h|$ *rounds, in which* B *answers with* 0*).* $\widehat{\mathsf{A}}$ *computes* $h(r^{\mathsf{A}})$ *and* $\widehat{\mathsf{B}}$ *computes* $h(r^{\mathsf{B}})$.
3. $\widehat{\mathsf{A}}$ *and* $\widehat{\mathsf{B}}$ *interact in additional* $2t(n)$ *rounds to send* $h(r^{\mathsf{A}})_{\leq z^{\mathsf{A}}-\log n}$ *and* $h(r^{\mathsf{B}})_{\leq z^{\mathsf{B}}-\log n}$*: In the* $(2i)$*th round,* $\widehat{\mathsf{A}}$ *sends* $h(r^{\mathsf{A}})_i$ *if* $i \leq z^{\mathsf{A}} - \log n$ *or* 0 *otherwise. In the* $(2i+1)$*th round,* $\widehat{\mathsf{B}}$ *sends* $h(r^{\mathsf{B}})_i$ *if* $i \leq z^{\mathsf{B}} - \log n$ *or* 0 *otherwise.*
4. $\widehat{\mathsf{A}}$ *and* $\widehat{\mathsf{B}}$ *output the outputs of* A *and* B*, respectively.*

Fix $n \in N$. Let $R^{\mathsf{A}}, R^{\mathsf{B}}, \Pi, X$ and Y be the random variables distributed according to the randomness $r^{\mathsf{A}}, r^{\mathsf{B}}$ of the parties, the transcript π and the outputs of the parties x, y, in a random execution of $(\mathsf{A},\mathsf{B})(1^n)$. For every transcript π of $(\mathsf{A},\mathsf{B})(1^n)$, let

$$\begin{aligned} z^{\mathsf{A}}(\pi) &= \left\lceil \mathrm{H}_\infty(R^{\mathsf{A}} \mid \Pi = \pi) \right\rceil \\ &= \left\lceil \log\left|\left\{r^{\mathsf{A}} \in \{0,1\}^{t(n)}\colon \exists r^{\mathsf{B}} \in \{0,1\}^{t(n)} \text{ s.t. } (\mathsf{A}(r^{\mathsf{A}}),\mathsf{B}(r^{\mathsf{B}}))(1^n) = (\pi,\cdot,\cdot)\right\}\right| \right\rceil, \end{aligned}$$

and similarly,

$$\begin{aligned} z^{\mathsf{B}}(\pi) &= \left\lceil \mathrm{H}_\infty(R^{\mathsf{B}} \mid \Pi = \pi) \right\rceil \\ &= \left\lceil \log\left|\left\{r^{\mathsf{B}} \in \{0,1\}^{t(n)}\colon \exists r^{\mathsf{A}} \in \{0,1\}^{t(n)} \text{ s.t. } (\mathsf{A}(r^{\mathsf{A}}),\mathsf{B}(r^{\mathsf{B}}))(1^n) = (\pi,\cdot,\cdot)\right\}\right| \right\rceil. \end{aligned}$$

Let $z^{\mathsf{A}}, z^{\mathsf{B}} \in [t(n)]$ be numbers such that $\Pr\left[z^{\mathsf{A}}(\Pi) = z^{\mathsf{A}}, z^{\mathsf{B}}(\Pi) = z^{\mathsf{B}}\right] \geq 1/t(n)^2$, and let E_1 be the event over $R^{\mathsf{A}}, R^{\mathsf{B}}$ that $z^{\mathsf{A}}(\Pi) = z^{\mathsf{A}}$ and $z^{\mathsf{B}}(\Pi) = z^{\mathsf{B}}$. Then,

$$\Pr_{R^{\mathsf{A}},R^{\mathsf{B}}}[E_1] \geq 1/t(n)^2. \tag{7}$$

For $H \leftarrow \mathcal{H}_{t(n)}$, let $\widehat{\Pi}$ be the distribution of the transcript of the protocol $(\widehat{\mathsf{A}}(R^{\mathsf{A}}, z^{\mathsf{A}}, H), \widehat{\mathsf{B}}(R^{\mathsf{B}}, z^{\mathsf{B}}))(1^n)$. Let $M^{\mathsf{A}} = H(R^{\mathsf{A}})_{\leq z^{\mathsf{A}}-\log n} 0^{t(n)-z^{\mathsf{A}}+\log n}$ be the messages sent by $\widehat{\mathsf{A}}$ in Step 3 of the protocol,

and similarly, let $M^{\mathsf{B}} = H(R^{\mathsf{B}})_{\leq z^{\mathsf{B}}-\log n}0^{t(n)-z^{\mathsf{B}}+\log n}$ be the messages sent by $\widehat{\mathsf{B}}$. Let $\widehat{M}^{\mathsf{A}} \leftarrow (\{0,1\}^{z^{\mathsf{A}}-\log n}||0^{t(n)-z^{\mathsf{A}}+\log n})$ and $\widehat{M}^{\mathsf{B}} \leftarrow (\{0,1\}^{z^{\mathsf{B}}-\log n}||0^{t(n)-z^{\mathsf{B}}+\log n})$ be the random variables obtains by replacing the hashed randomness with uniformly chosen bits in the messages of $\widehat{\mathsf{A}}$ and $\widehat{\mathsf{B}}$. We will use the following two claims.

Claim 5.7.

$$\mathrm{SD}((\Pi, H, M^{\mathsf{A}}, M^{\mathsf{B}})|_{E_1}, (\Pi|_{E_1}, H, \widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}})) \leq 2/\sqrt{n}.$$

Proof. Immediate from Lemma 2.16 and the definition of the event E_1. □

Claim 5.8. *There exist an event $E_2 \subseteq E_1$ and an efficient oracle-aided algorithm Red such that $\Pr[E_2 \mid E_1] \geq 1/2$, and the following holds. Assume there exists an algorithm Pred such that $\Pr\left[\mathsf{Pred}(\widehat{\Pi}) = X \mid E_2\right] = \delta$. Then, $\Pr\left[\mathsf{Red}^{\mathsf{Pred}}(\Pi, z^{\mathsf{A}}, z^{\mathsf{B}}) = X\right] \geq \delta/2t(n)^2$.*

Moreover $\mathrm{H}_\infty(\widehat{\Pi}|_{E_2}) \geq \left|R^{\mathsf{A}}\right| + \left|R^{\mathsf{B}}\right| + |H| - 2\log(t(n)) - 2\log n - 1$.

Proof. Let Red be the algorithm that, given π, z^{A} and z^{B}, samples $h \leftarrow \mathcal{H}_{t(n)}$, $m^{\mathsf{A}} \leftarrow \widehat{M}^{\mathsf{A}}$ and $m^{\mathsf{B}} \leftarrow \widehat{M}^{\mathsf{B}}$ and executes $\mathsf{Pred}(\pi, h, m^{\mathsf{A}}, m^{\mathsf{B}})$. In the following we show that, for some event E_2, if $\Pr\left[\mathsf{Pred}(\widehat{\Pi}) = X \mid E_2\right] = \delta$, then

$$\Pr\left[\mathsf{Pred}(\Pi, H, \widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}}) = X \mid E_1\right] \geq \delta/2, \tag{8}$$

which concludes the claim by Equation (7). We start by showing that

$$\mathrm{SD}((\Pi, X, H, M^{\mathsf{A}}, M^{\mathsf{B}})|_{E_1}, (\Pi, X, H, \widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}})|_{E_1}) \leq 1/10. \tag{9}$$

That is, Claim 5.7 holds also when we add the output X. To see the above, for every π, let $x(\pi)$ be the value of x that maximise the probability $\Pr[X = x \mid \Pi = \pi]$. By Claim 5.7 and data processing, it holds that,

$$\mathrm{SD}((\Pi, x(\Pi), H, M^{\mathsf{A}}, M^{\mathsf{B}})|_{E_1}, (\Pi, x(\Pi), H, \widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}})|_{E_1}) \leq 2/\sqrt{n} \leq 1/30.$$

It thus enough to show that

$$\mathrm{SD}((\Pi, X, H, \widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}})|_{E_1}, (\Pi, x(\Pi), H, \widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}})|_{E_1}) \leq 1/30,$$

and,

$$\mathrm{SD}((\Pi, X, H, M^{\mathsf{A}}, M^{\mathsf{B}})|_{E_1}, (\Pi, x(\Pi), H, M^{\mathsf{A}}, M^{\mathsf{B}})|_{E_1}) \leq 1/30.$$

By data processing, it is enough to bound the later, as $\widehat{M}^{\mathsf{A}}$ and $\widehat{M}^{\mathsf{B}}$ are independent from Π and X. Observer that, since $x(\Pi)$ is fixed given Π, it holds that

$$\mathrm{SD}((\Pi, X, M^{\mathsf{A}}, M^{\mathsf{B}})|_{E_1}, (\Pi, x(\Pi), M^{\mathsf{A}}, M^{\mathsf{B}})|_{E_1}) = \Pr[X \neq x(\Pi) \mid E_1].$$

Assume toward a contradiction that $\Pr[X \neq x(\Pi) \mid E_1] > 1/20$. Since $\Pr[E_1] \geq 1/t(n)^2$, we get that $\Pr[X \neq x(\Pi)] > 1/(20t(n)^2)$. We now show that, by definition of $x(\Pi)$, the agreement probability of (A, B) given E_1 in this case is at most $1 - 1/(20t(n)^2)$, which is a contradiction to the agreement property of key-agreements protocols, as $t(n)$ is polynomial. Indeed, by Fact 2.14, the inputs $R^{\mathsf{A}}, R^{\mathsf{B}}$ are in a product distribution given the transcript π. By data processing, also the outputs

X and Y are in product distribution given the transcript. We get that for every fixed π and an output y of $\widehat{\mathsf{B}}$, the probability that $X = y$ is at most $\Pr[X = x(\pi)]$. Taking expectation over Y and Π, we get that $\Pr[X = Y] \leq \Pr[X = x(\Pi)] \leq 1 - 1/(20t(n)^2)$, as we wanted to show.

Overall, we got that Equation (9) holds.

Next, we use Lemma 2.3 and Equation (9) to show Equation (8). By Lemma 2.3 and Equation (9), there are some random variables W (jointly distributed with $(\Pi, X, M^{\mathsf{A}}, M^{\mathsf{B}})$) and $\widehat{W}$ (jointly distributed with $(\Pi, X, \widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}})$), such that $\Pr[W = 1] = \Pr\Big[\widehat{W} = 1\Big] \geq 9/10$, and,

$$(\Pi, X, M^{\mathsf{A}}, M^{\mathsf{B}})|_{E_1, W=1} \equiv (\Pi, X, \widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}})|_{E_1, \widehat{W}=1}. \tag{10}$$

Let E_2 be the event that $W = 1$ and E_1 occurs, and assume that $\Pr\Big[\mathsf{Pred}(\widehat{\Pi}) = X \mid E_1, W = 1\Big] = \Pr\Big[\mathsf{Pred}(\widehat{\Pi}) = X \mid E_2\Big] = \delta$. By definition of W, we get that $\Pr\Big[\mathsf{Pred}(\Pi, H, \widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}}) = X \mid E_1, \widehat{W} = 1\Big] = \delta$, and thus Equation (8) holds.

Lastly, to see the moreover part of the claim, fix $\pi \in \{0,1\}^*, h \in \mathcal{H}, m^{\mathsf{A}} \in \{0,1\}^{t(n)}$ and $m^{\mathsf{B}} \in \{0,1\}^{t(n)}$. We want to upper bound the probability

$$\Pr\Big[(\Pi, H, M^{\mathsf{A}}, M^{\mathsf{B}}) = (\pi, h, m^{\mathsf{A}}, m^{\mathsf{B}}) \mid E_2\Big] = \Pr\Big[(\Pi, H, \widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}}) = (\pi, h, m^{\mathsf{A}}, m^{\mathsf{B}}) \mid E_1, \widehat{W} = 1\Big],$$

where the equality holds by Equation (10). Since $\Pr\Big[\widehat{W} = 1\Big] \geq 9/10$, and by the definition of $\widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}}$ it holds that

$$\begin{aligned}\Pr\Big[(\Pi, H, \widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}}) = (\pi, h, m^{\mathsf{A}}, m^{\mathsf{B}}) \mid E_1, \widehat{W} = 1\Big] &\leq 10/9 \cdot \Pr\Big[(\Pi, H, \widehat{M}^{\mathsf{A}}, \widehat{M}^{\mathsf{B}}) = (\pi, h, m^{\mathsf{A}}, m^{\mathsf{B}}) \mid E_1\Big] \\ &\leq \Pr[\Pi = \pi \mid E_1] \cdot 2^{-|H| - z^{\mathsf{A}} - z^{\mathsf{B}} + 2\log n + 1}.\end{aligned}$$

Thus, to conclude the claim it is enough to show that for every π, $\Pr[\Pi = \pi \mid E_1] \leq 2^{-|R^{\mathsf{A}}| - |R^{\mathsf{B}}| + z^{\mathsf{A}} + z^{\mathsf{B}} + 2\log t(n)}$. By Equation (7), Fact 2.6 and the definition of E_1, it is enough to show that for every π with $z^{\mathsf{A}}(\pi) = z^{\mathsf{A}}$ and $z^{\mathsf{B}}(\pi) = z^{\mathsf{B}}$, it holds that

$$\Pr[\Pi = \pi] \leq 2^{-|R^{\mathsf{A}}| - |R^{\mathsf{B}}| + z^{\mathsf{A}} + z^{\mathsf{B}}}. \tag{11}$$

To see Equation (11), let $\mathcal{R}^{\mathsf{A}}(\pi) = \Big\{r^{\mathsf{A}} \in \{0,1\}^{t(n)} \colon \exists r^{\mathsf{B}} \in \{0,1\}^{t(n)} \ s.t.\ (\mathsf{A}(r^{\mathsf{A}}), \mathsf{B}(r^{\mathsf{B}}))(1^n) = (\pi, \cdot, \cdot)\Big\}$ and $\mathcal{R}^{\mathsf{B}}(\pi) = \Big\{r^{\mathsf{B}} \in \{0,1\}^{t(n)} \colon \exists r^{\mathsf{A}} \in \{0,1\}^{t(n)} \ s.t.\ (\mathsf{A}(r^{\mathsf{A}}), \mathsf{B}(r^{\mathsf{B}}))(1^n) = (\pi, \cdot, \cdot)\Big\}$ be sets such that $z^{\mathsf{A}}(\pi) = \lceil \log|\mathcal{R}^{\mathsf{A}}(\pi)| \rceil$ and $z^{\mathsf{B}}(\pi) = \lceil \log|\mathcal{R}^{\mathsf{A}}(\pi)| \rceil$. It holds that,

$$\Pr[\Pi = \pi] = \Pr\Big[R^{\mathsf{A}} \in \mathcal{R}^{\mathsf{A}}(\pi), R^{\mathsf{B}} \in \mathcal{R}^{\mathsf{B}}(\pi)\Big] = \frac{|\mathcal{R}^{\mathsf{A}}(\pi)| \cdot |\mathcal{R}^{\mathsf{B}}(\pi)|}{2^{|R^{\mathsf{A}}| + |R^{\mathsf{B}}|}} \leq 2^{-|R^{\mathsf{A}}| - |R^{\mathsf{B}}| + z^{\mathsf{A}} + z^{\mathsf{B}}}$$

as we wanted to show. □

Proving Lemma 5.5.

Proof of Lemma 5.5. Let (A,B) be a t-time key agreement protocol, and fix $n \in N$. Let $(\widehat{\mathsf{A}},\widehat{\mathsf{B}})$ be as defined in Protocol 5.6, and let E_2 be the event promised by Claim 5.8. Let $R^{\mathsf{A}}, R^{\mathsf{B}}, Z^{\mathsf{A}}, Z^{\mathsf{B}}, H, X, Y, \Pi$ and $\widehat{\Pi}$ be as defined above, and let $\widehat{R}^{\mathsf{A}} = (R^{\mathsf{A}}, Z^{\mathsf{A}}, H)$ and $\widehat{R}^{\mathsf{B}} = (R^{\mathsf{B}}, Z^{\mathsf{B}})$.

By Claim 5.8, condition on the event E_2, the distribution of the transcript of $(\widehat{\mathsf{A}}(\widehat{R}^{\mathsf{A}}), \widehat{\mathsf{B}}(\widehat{R}^{\mathsf{B}}))(1^n)$ has min-entropy at least

$$\left|R^{\mathsf{A}}\right| + \left|R^{\mathsf{B}}\right| + |H| - 2\log(t(n)) - 2\log n - 1 = \left|\widehat{R}^{\mathsf{A}}\right| + \left|\widehat{R}^{\mathsf{A}}\right| - 4\log(t(n)) - 2\log n - 1.$$

Let E be the event that E_2 happened, $Z^{\mathsf{A}} = z^{\mathsf{A}}$, $Z^{\mathsf{B}} = z^{\mathsf{B}}$, and $X = Y$. By definition, given E the protocol has agreement 1. To see the secrecy, observe that since

$$\Pr\left[X = Y \mid E_2, Z^{\mathsf{A}} = z^{\mathsf{A}}, Z^{\mathsf{B}} = z^{\mathsf{B}}\right]$$

is noticeable, it is enough to prove the secrecy given $E_3 = \left\{E_2, Z^{\mathsf{A}} = z^{\mathsf{A}}, Z^{\mathsf{B}} = z^{\mathsf{B}}\right\}$. However, since the distribution of $(\widehat{\Pi}, X)|_{E_2, Z^{\mathsf{A}}=z^{\mathsf{A}}, Z^{\mathsf{B}}=z^{\mathsf{B}}}$ is exactly the distribution considered in Claim 5.8, we get that the existence of an algorithm Pred that break the secrecy of $(\widehat{\mathsf{A}},\widehat{\mathsf{B}})|_{E_2, Z^{\mathsf{A}}=z^{\mathsf{A}}, Z^{\mathsf{B}}=z^{\mathsf{B}}}$ implies the existence of a protocol that breaks the secrecy of (A,B) (by guessing the values of z^{A} and z^{B}). □

5.2 Weak-cond-EP Key-agreement to cond-EP Key-agreement

We now show how to construct a conditional entropy-preserving key-agreement protocol from a weak conditional entropy-preserving key-agreement. We use Goldreich-Levin to prove the following lemma.

Lemma 5.9 (Weak-Cond-EP KA to Cond-EP KA)**.** *Assume there exists a weak-cond-EP KA protocol* (A,B)*. Then, for every constants* $c \in \mathbb{N}$ *and* $\epsilon > 0$*, there exists a* 0.1*-cond-EP KA with key of length* $c\log n$*, transcript of length* n *and running time* $t(n) = n^{1+\epsilon}$*.*

Let (A,B) be a m-bits $(d, s^{\mathsf{A}}, s^{\mathsf{B}})$-weak-Cond-EP KA, and let $\delta > 0$ be a constant to be chosen later. Let $(\mathsf{A}',\mathsf{B}')$ be the following protocol:

Protocol 5.10 $((\mathsf{A}',\mathsf{B}'))$**.**

A'*'s input:* $n, r^{\mathsf{A}} \in \{0,1\}^{s^{\mathsf{A}}(\lfloor n^{\delta} \rfloor)}$*,* $r_1, \ldots, r_{c\log n} \in (\{0,1\}^m)^{c\log n}$*.*

B'*'s input:* $n, r^{\mathsf{B}} \in \{0,1\}^{s^{\mathsf{B}}(\lfloor n^{\delta} \rfloor)}$*.*

Operation:

1. A' *and* B' *interact according to* $(\mathsf{A}(r^{\mathsf{A}}), \mathsf{B}(r^{\mathsf{B}}))(1^{\lfloor n^{\delta} \rfloor})$*. Let* x *and* y *be the outputs of* A *and* B*, respectively.*
2. A' *sends* $r_1, \ldots, r_{c\log n}$ *(bit by bit, where* B *answers with* 0*'s).*
3. *Let* k *be the number of bits sent so far. If* $k < n$*,* A' *and* B' *continue sending* 0*'s until the length of the transcript is* n*.*
4. A' *outputs* $\mathrm{GL}(x, r_1), \ldots, \mathrm{GL}(x, r_{c\log n})$*, and* B' *outputs* $\mathrm{GL}(y, r_1), \ldots, \mathrm{GL}(y, r_{c\log n})$*.*

Proof of Lemma 5.9. Let (A,B) be an m-bit $(d, s^{\mathsf{A}}, s^{\mathsf{B}})$-weak-cond EP KA, and let $\{E_n\}_{n\in\mathbb{N}}$ be the events and constant promised by Definition 5.4. Let $e \in \mathbb{N}$ be a number, such that n^e is an upper bound on the running time of (A,B), and on the time taking to compute $s^{\mathsf{A}}(n), s^{\mathsf{B}}(n)$. Let $\delta = \epsilon/10(e+d)$. Finally, let $(\mathsf{A}',\mathsf{B}')$ be as defined in Protocol 5.10. Then, the running time of $(\mathsf{A}(n,\cdot),\mathsf{B}(n,\cdot))$ is at most $n^{1+\epsilon}$. Moreover, the length of the transcript of $(\mathsf{A}'(n,\cdot),\mathsf{B}'(n,\cdot))$ is exactly n.

Let $\left\{E_{\lfloor n^\delta \rfloor}\right\}_{n\in\mathbb{N}}$ be the sequence of events required in Definition 5.2. Observe that, for every $n \in \mathbb{N}$, given the event $E_{\lfloor n^\delta \rfloor}$, A',B' always agree, and by Fact 2.5, the min-entropy of the transcript is at least $s^{\mathsf{A}}(\lfloor n^\delta \rfloor) + s^{\mathsf{B}}(\lfloor n^\delta \rfloor) - d\log\lfloor n^\delta\rfloor + c\log n \cdot m \geq s^{\mathsf{A}}(\lfloor n^\delta \rfloor) + s^{\mathsf{B}}(\lfloor n^\delta \rfloor) + c\log n \cdot m - 0.1\log n$, as required.

For the secrecy property, let Π_n, X_n, Y_n be as defined in Definition 5.4. By Lemma 2.17, it is enough to show that there is no PPT algorithm Pred such that $\Pr\left[\mathsf{Pred}(1^n, \Pi_{\lfloor n^\delta\rfloor}) = X_{\lfloor n^\delta\rfloor} \mid E_{\lfloor n^\delta\rfloor}\right] \geq 1/\mathrm{poly}(n)$ for infinitely many n's. Assume towards a contradiction that such Pred exists. Then, the algorithm Pred', that given $1^{n'}, \Pi_{n'}$ samples $n \leftarrow [(n'+1)^{1/\delta}]$ and executes $\mathsf{Pred}(1^n, \Pi_{n'})$ has noticeable probability to predict $X_{n'}$ given the event $E_{n'}$, for infinitely many n's. This is a contradiction to the secrecy property of (A, B).

□

5.3 Proving Theorem 5.1

Proof. Fix c_1, c_2 as in Theorem 5.1, and assume there exists a KA protocol. By Lemma 5.5, there exists a weak-cond-EP KA protocol. By Lemma 5.9 there exists a 0.1-cond EP KA with key of length $(c_2+2)\log n$ and running time $t'(n) = n^{1+\epsilon}/2$. Finally, the theorem follows by Lemma 5.3.
□

6 Worst-Case hardness of $\mathsf{RIK}^t\mathsf{P} \implies$ OWF

In this section we prove the following theorem, that states that the worst-case hardness of $\mathsf{RIK}^t\mathsf{P}$ implies the existence of one-way functions.

Theorem 6.1. *Let c_1, c_2 be constants such that $0 \leq c_1 < c_2 - 11$, and let $t\colon \mathbb{N} \to \mathbb{N}$ be an efficiently computable function. Then the following holds: If $\mathsf{RIK}^t\mathsf{P}[c_1\log n, c_2 \log n] \notin \mathsf{ioBPP}$, one-way functions exist.*

The above is stronger than stated in Theorem 3.5, as if $\mathsf{RIK}^t\mathsf{P}[c_1\log n, c_2\log n]|_{Q_\Delta} \notin \mathsf{ioBPP}$ for some Δ, then $\mathsf{RIK}^t\mathsf{P}[c_1\log n, c_2\log n] \notin \mathsf{ioBPP}$

To prove the theorem, fix $t = t(n)$, c_1 and c_2 as in Theorem 4.1, and let $c = c_2 - 3$. For simplicity, in the following, for a string $x \in \{0,1\}^*$, let $(x)_n$ be the first n bits of $x0^n$ (that is, an n bits string containing the first n bits of x, or appropriate padding if necessary). Let $\mathcal{H} = \left\{h\colon \{0,1\}^{2n} \to \{0,1\}^{c\log n}\right\}$ be a 2-universal family, and consider the following function f:

Algorithm 6.2 (f)**.**

Parameter: function $t\colon \mathbb{N}\to\mathbb{N}$, $c \in \mathbb{N}$.

Input: $\ell_{\mathsf{A}} \in [n]$, $\ell_{\mathsf{B}} \in [n]$, $a \in \{0,1\}^n$, $b \in \{0,1\}^n$, $h \in \mathcal{H}$.

Operation:

1. *Let* $P_{\mathsf{A}} = a_{\leq \ell_{\mathsf{A}}}$ *and* $P_{\mathsf{B}} = b_{\leq \ell_{\mathsf{B}}}$
2. *f simulates the interaction of* $(\mathsf{U}(P_{\mathsf{A}}, 1^{t(n)}), \mathsf{U}(P_{\mathsf{B}}, 1^{t(n)}))$. *Let π be the transcript of the interaction, and let x, y be the outputs of* A *and* B*, respectively.*
3. *Let* $(\ell, \pi, h, h((x)_n || (y)_n))$ *be the output of f.*

Below we assume there is an algorithm Eve that invert the above function with high probability, and show how to use such Eve in order to decide $\mathsf{RIK}^t\mathsf{P}$. In the following, let $L_A \leftarrow [n], L_B \leftarrow [n], A \leftarrow \{0,1\}^n, B \leftarrow \{0,1\}^n$ and $H \leftarrow \mathcal{H}$. Let Π, X and Y be the random variables that gets the values of π, x and y in the execution of $f(L_A, L_B, A, B, H)$.

Lemma 6.3. *Assume there exists an algorithm* Eve *such that*

$$\Pr\left[\mathsf{Eve}(1^n, f(L_A, L_B, A, B, H)) \in f^{-1}(f(L_A, L_B, A, B, H))\right] \geq 1 - n^{-c_1 - 6}$$

for infinitely many n's. Then $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n] \in \mathsf{ioBPP}$.

We prove Lemma 6.3, but first let us use it in order to prove Theorem 6.1.

Proof of Theorem 6.1. Immediate from Lemma 6.3 and Theorem 2.10. □

To prove Lemma 6.3 we will use the following lemma, on interactive Kolmogorov complexity.

Lemma 6.4. *Let n be a large enough number, $r < n$, and $t = t(n)$ be a function. For every $\pi \in \{0,1\}^n$ such that $\mathsf{K}(\pi) = k$ the following holds.*

$$\left|\left\{x, y \in (\{0,1\}^n)^2 \colon \mathsf{IK}^t(\pi; x; y) \leq k + r\right\}\right| \leq 2^{r + 7\log n}$$

Proof of Lemma 6.4. Assume towards a contradiction that

$$\left|\left\{x, y \in (\{0,1\}^n)^2 \colon \mathsf{IK}^t(\pi; x; y) \leq r + s\right\}\right| > 2^{r + 7\log n}.$$

We will show that $\mathsf{K}(\pi) < k$. For every $\pi' \in \{0,1\}^n$, let $S_{\pi'} = \left|\left\{x, y \in (\{0,1\}^n)^2 \colon \mathsf{IK}^t(\pi; x; y) \leq k + r\right\}\right|$. Observe that there are at most $d = \frac{(k+r)^2 2^{k+r}}{2^{r+7\log n}}$ strings $\pi' \in \{0,1\}^n$ with $S_{\pi'} > 2^{r+7\log n}$, since there are at most $(k+r)^2 2^{k+r}$ pairs of programs $P_{\mathsf{A}}, P_{\mathsf{B}}$ such that $|P_{\mathsf{A}}| + |P_{\mathsf{B}}| \leq (k+r)$.

Thus, to encode π, it is enough to describe the set of all $\pi' \in \{0,1\}^n$ with $S_{\pi'} > 2^{r+7\log n}$, and then describe the index of π inside this set. To describe the above set it is enough to describe n, k, and s. Thus,

$$\begin{aligned}
\mathsf{K}(\pi) &\leq \log n + 2\log\log n + \log k + 2\log\log k + \log r + 2\log\log r + \log d \\
&\leq 4\log n + 2\log(k+r) + k - 7\log n \\
&\leq 7\log n + k - 8\log n \\
&< k
\end{aligned}$$

□

6.1 Proving Lemma 6.3

Next, we prove Lemma 6.3. Consider the following algorithm, that given an attacker Eve that inverts the function f with too good probability, decides $\mathsf{RIK}^t\mathsf{P}$. In the following we only focusing on the executions of f in which $|x| = |y| = |\pi| = n$.

Algorithm 6.5.

Input: $\pi \in \{0,1\}^n, x \in \{0,1\}^n, y \in \{0,1\}^n$.

Operation:

1. *For every* $\ell \in [2n]$*:*
   (a) *Sample* $h \leftarrow \mathcal{H}$ *and compute* $h(x||y)$.
   (b) *Execute* $\mathsf{Eve}(\ell, \pi, h, h(x||y))$ *to get* L_A, L_B, A, B, h. *Execute* $(A_{\leq L_A}, B_{\leq L_A})$ *for* $t(n)$ *steps to get* π', x', y'.
   (c) *If* $L_A + L_B \leq \ell$, $\pi' = \pi$, $x' = x$ *and* $y' = y$, *answer "Yes".*
2. *Answer "No".*

That is, given π, x, y (and assuming that $(\pi, x, y) \in Q_\Delta$), Algorithm 4.5 tries to use Eve to get programs $P_\mathsf{A}, P_\mathsf{B}$ that produce (π, x, y). In the following, let $\mathcal{Y}$ and $\mathcal{N}$ be the Yes and No instances of the problem $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]$, respectively.

Proof of Lemma 4.4. Assume there exists an algorithm Eve such that

$$\Pr\left[\mathsf{Eve}(1^n, f(L_A, L_B, A, B, H)) \in f^{-1}(f(L_A, L_B, A, B, H))\right] \geq 1 - n^{-c_1 - 6}$$

for infinitely many n's. We will show that Algorithm 6.5 decides $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]$ for every such n. In the following, fix such $n \in \mathbb{N}$.

Soundness: First, notice that for every $(\pi, x, y) \in \{0,1\}^{3n}$ for which Algorithm 6.5 answer "yes" with probability larger than $1/3$, the following holds with the same probability: For some $\ell \in [2n]$, Eve outputs P_A and P_B that outputs x and y. Thus, there exists a PPT algorithm D, such that with probability at least $1/3$ over the choice of h and the randomness r_E of Eve the following holds. Given $h(x||y)$ and the right choice of ℓ, $D(n, \pi, \ell, h(x||y), h, r_E)$ outputs x, y. By Lemma 2.21, we get that for every large enough n,

$$\begin{aligned}
\mathsf{K}(\pi, x, y) &\leq \mathsf{K}(n, \pi, x, y) \\
&\leq \mathsf{K}(n, \pi) + |h(x||y)| + 2\log|h(x||y)| + \log \ell + 2\log\log \ell + O(1) \\
&\leq \mathsf{K}(\pi) + \log n + 2\log\log n + |h(x||y)| + 2\log|h(x||y)| + \log \ell + 2\log\log \ell + O(1) \\
&< \mathsf{K}(\pi) + c\log n + 3\log n \\
&= \mathsf{K}(\pi) + c_2 \log n,
\end{aligned}$$

Namely, by the above (π, x, y) is not in $\mathcal{N}$. We get that for every No instance $(\pi, x, y) \in \mathcal{N}$, Algorithm 4.5 answer "no" with probability at least $2/3$.

Completeness: We start by showing that the following holds for every $(\pi, x, y) \in (\{0,1\}^n)^3$ with $\mathsf{K}(\pi) = k$. If, for some $\ell \leq k + c_1 \log n$, Eve finds some pre-image of $(\ell, \pi, H, H(x||y))$ with probability at least $3/4$, then with probability at least $2/3$ Algorithm 6.5 outputs "yes" on (π, x, y). Indeed, by Lemma 6.4, there are at most $2^{(c_1+7)\log n}$ different values that the outputs (x', y') can get for every pair of programs of total length $k + c_1 \log n$ that produce π as transcript. Since $\mathcal{H}$ is a 2-universal family, and since $c \geq c_1 + 8$, with probability at most $2^{-c \log n} \cdot 2^{c_1+7 \log n} \leq 1/n$ over the choice of H there is such a pair that with $H(x'||y') = H(x||y)$. Thus with probability at least $3/4 - 1/n > 2/3$, Eve must output programs that output x and y. In this case, Algorithm 6.5 outputs "yes".

Next, let $\beta = n^{-c_1-6}$, and assume toward a contradiction that Algorithm 4.5 does not output "yes" with probability at least $2/3$ on every input $(\pi, x, y) \in \{0,1\}^{3n}$ from $\mathcal{Y}$. That is, there exists $(\pi, x, y) \in \mathcal{Y}$ such that with probability at least $1/4$, for every $\ell \leq \mathsf{K}(\pi) + c_1 \log n$, Eve does not invert successfully $(\ell, \pi, H, H(x||y))$. In particular, Eve fails on $(\mathsf{IK}^t(\pi; x; y), \pi, H, H(x||y))$ with probability at least $1/4$. Let $\mathcal{S}$ be the set of $(\pi, x, y) \in \{0,1\}^{3n}$ on which Eve fails to invert $(\mathsf{IK}^t(\pi; x; y), \pi, H, H(x||y))$ with probability at least $1/4$. We next bound the size of $\mathcal{S}$.

Observe that for every (π, x, y) with $\mathsf{IK}^t(\pi; x; y) = \ell$, it holds that in a random execution of f,

$$\Pr[(\Pi, X, Y) = (\pi, x, y), L_A + L_B = \ell] \geq 1/n^2 \cdot 2^{-\ell}. \tag{12}$$

For every $\ell \in \mathbb{N}$, let $\mathcal{S}_\ell = \left\{(\pi, x, y) \in \mathcal{S} \colon (\pi, x, y) \in \{0,1\}^{3n}, \mathsf{IK}^t(\pi; x; y) = \ell\right\}$, and fix ℓ^* such that the algorithm fails on $\mathcal{S}_{\ell^*} \cap \mathcal{Y}|_{Q_\Delta}$. Combining Equations (1) and (2) get that,

$$\begin{aligned}
\beta &\geq \Pr\left[\mathsf{Eve}(1^n, f(L_A, L_B, A, B, H)) \notin f^{-1}(f(L_A, L_B, A, B, H))\right] \\
&\geq 1/4 \cdot \Pr[(\Pi, X, Y) \in \mathcal{S}] \\
&\geq \Omega(1/n^2 \cdot 2^{-\ell^*} \cdot |\mathcal{S}_{\ell^*}|),
\end{aligned}$$

and thus,

$$|\mathcal{S}_{\ell^*}| \leq \Omega(n^2 \cdot \beta \cdot 2^{\ell^*})$$

On the other hand, there exists an algorithm that given n, ℓ and $i \in [|\mathcal{S}_\ell|]$, output the i-th element in $\mathcal{S}_\ell$ according to the lexicographic order (by enumerating over all the possible values of $(\pi, x, y) \in \{0,1\}^{3n}$ with $\mathsf{IK}^t(\pi; x; y) = \ell$ and computing the failure probability of Eve). Thus, by Fact 2.19, for every $(\pi, x, y) \in \mathcal{S}_{\ell^*}$ it holds that

$$\begin{aligned}
\mathsf{K}(\pi) &\leq \mathsf{K}(n, \pi, x, y) \\
&\leq \log n + 2 \log\log n + \log \ell^* + 2 \log\log \ell^* + \log|\mathcal{S}_{\ell^*}| + O(1) \\
&\leq 3 \log n + (2 \log n + \log \beta + \ell^*) \\
&= 5 \log n + \ell^* - (c_1 + 6) \log n \\
&< \ell^* - c_1 \log n
\end{aligned}$$

The above implies that $\mathcal{S}_{\ell^*} \cap \mathcal{Y} = \emptyset$. Namely, Algorithm 4.5 has no error. □

7 OWF $\implies$ Worst-Case Hardness of $\mathsf{RIK}^t\mathsf{P}$

In this section we prove that one-way functions imply the hardness of $\mathsf{RIK}^t\mathsf{P}$.

Theorem 7.1. *For any constant c_1, c_2, d, $d-4 > c_2 > c_1 \geq 0$, if one-way functions exist, then for any $\varepsilon > 0$, any polynomial $t(n) \geq n^{1+\epsilon}$, and for $\Delta(n) = d \log n$, $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]|_{Q_\Delta} \notin \mathsf{ioBPP}$.*

To prove the above theorem, we will use the following definition and theorem from [LP20].

Definition 7.2. *An efficiently computable function $g : \{0,1\}^n \times \{0,1\}^{n+\gamma \log n} \to \{0,1\}$ is a ε-*conditionally-secure α-entropy-preserving pseudorandom generator (ε-cond α-EP-PRG) *if there exist a sequence of events $= \{E_n\}_{n\in\mathbb{N}}$ such that the following conditions hold:*

- **(pseudorandomness):** *For every PPT attacker $\mathcal{A}$ and sufficiently large $n \in \mathbb{N}$,*

$$|\Pr[s \leftarrow \{0,1\}^n : \mathcal{A}(1^n, g(s)) = 1 \mid E_n] - \Pr[r \leftarrow \{0,1\}^{n+\gamma \log n}; \mathcal{A}(1^n, r) = 1]| < \varepsilon(n), \quad (13)$$

- **(entropy-preserving):** *For all sufficiently large $n \in \mathbb{N}$, $H([g(\mathcal{U}_n \mid E_n)]_n) \geq n - \alpha \log n$, where $[\cdot]_n$ denote the function that truncates any string to its n-bit prefix.*

We refer to the constant α as the entropy-loss constant*.*

We say that g has *rate-1 efficiency* if its running time on inputs of length n is bounded by $n + O(n^\varepsilon)$ for some constant $\varepsilon < 1$. The proof of Theorem 7.1 is immediate from the following theorem and lemma.

Theorem 7.3 ([LP20])**.** *Assume that one-way functions exist. Then, for any $\gamma > 0, \alpha > 0$, there exists a rate-1 efficient 0.1-cond α-EP-PRG $g : \{0,1\}^n \to \{0,1\}^{n+\gamma \log n}$.*

Lemma 7.4. *For any constant c_1, c_2, d, $d-4 > c_2 > c_1 \geq 0$, if there exists a rate-1 efficient 0.1-cond $(0.1c_1)$-EP-PRG $g : \{0,1\}^n \to \{0,1\}^{n+(c_2+2)\log n}$, then for any $\varepsilon > 0$, any polynomial $t(n) \geq n^{1+\epsilon}$, and for $\Delta(n) = d \log n$, $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]|_{Q_\Delta} \notin \mathsf{ioBPP}$.*

Proof of Theorem 7.1. Immediate from Theorem 7.3 and Lemma 7.4. □

Proof of Lemma 7.4. We first aim at a weaker goal: showing the hardness of $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]$ (without the condition Q_Δ). And we will show that our proof also shows the hardness of the problem even when conditioned on Q_Δ.

For any constant c_1, c_2, let g be the cond EP-PRG $g : \{0,1\}^n \to \{0,1\}^{n+(c_2+2)\log n}$. We will use an algorithm that decides $\mathsf{RIK}^t\mathsf{P}$ to break the cond EP-PRG g. Let m denote $m = m(n) = n + (c_2+2)\log n$. We will consider an "embedding" h that maps a string $\in \{0,1\}^m$ to a transcript (π, x, y) where $|\pi| = |x| = |y| = 2n$, and h is defined as follows. For any $s \in \{0,1\}^m$, h will create a transcript where Alice will produce the whole string s and Bob will keep "silent". In more detail, $h(s)$ is defined to be

$$\pi = s_1 0 s_2 0 \ldots s_n 0, x = s_{n+1,\ldots,m} 0^{2n-(m-n)}, y = 0^{2n}$$

Note that $h(s) = (\pi, x, y)$ will define a transcript where Alice sends s_1, Bob sends 0, Alice sends s_2, and so on until $2n$ rounds. And after the interaction Alice will output the remaining part of s (padded with zeros until it is of length $2n$), and Bob will simply output $2n$ zeros.

Consider any polynomial $t(n) \geq n^{1+\epsilon}, \varepsilon > 0$. Observe that for a pseudorandom string $s \in \{0,1\}^m$, $h(s)$ will have small IK^t-complexity. Namely, for any $v \in \{0,1\}^n, s = g(v)$,

$$\mathsf{IK}^t(h(s)) \leq n + \gamma \quad (14)$$

where γ is a sufficiently large constant. Intuitively, this follows from our choice of f and the fact that s is pseudorandom. In more detail, we argue that the transcript $h(s)$ can be produced as follows: Alice will hardwire the string v and the code of g, and she will run g on input v to obtain $s = g(v)$. Alice will then send each of the first $|v|$ bits in s to Bob, and outputs what remains padded to length $2|v|$. Bob will simply answer each Alice's message by 0, and count the number of rounds. Bob will then output as many zeros as rounds. Notice that both Alice and Bob run in time $t(n)$ (since g is rate-1 efficient), Alice can be described using $|v| + O(1)$ bits, and Bob is just a machine of constant length. On the other hand, for a random string $r \in \{0,1\}^m$, $h(r)$ will have K-complexity at least

$$\mathrm{K}(h(r)) \geq m - 1 - \log n > n + c_2 \log n + 1 \tag{15}$$

with probability at least $1 - \frac{1}{n}$, which follows from a standard counting argument for K-complexity.

Notice that we can also define another mapping h', which embeds a string $\in \{0,1\}^m$ to a transcript $\in (\{0,1\}^{2n+1})^3$, in an analogous way. The above two observations will also hold if we consider h'.

We move on to proving that $\mathsf{RIK}^t\mathsf{P}$ is hard. Towards this, we assume for contradiction that there exists a polynomial time algorithm that decides $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n] = (\mathcal{Y}, \mathcal{N})$ on infinitely many input lengths. By a Chernoff-type argument, we can show that there exists a *ppt* algorithm M such that M succeeds with probability at least 0.99 on each YES/NO instance. Note that M will succeed on every $(\pi, x, y) \in \mathcal{Y} \cup \mathcal{N}$ such that $|\pi| = |x| = |y| = l$ for infinitely many $l \in \mathbb{N}$. It follows either M succeeds for infinitely many l of the form $l = 2n$, or for infinitely many l of the form $l = 2n + 1$. We assume that the former is the case and present the rest of the proof. If the later is the case, the lemma will essentially follow from the same proof with minor changes (where we replace the mapping h by h').

Given the algorithm M that decides $\mathsf{RIK}^t\mathsf{P}$, we will use it to construct an attacker that breaks the cond EP-PRG g. Our attacker $\mathcal{A}$, on input 1^n and a string $s \in \{0,1\}^{m(n)}$, and $\mathcal{A}$ needs to decide whether s is pseudorandom or random. $\mathcal{A}$ will use the embedding h, and compute $(\pi, x, y) = h(s)$. Then $\mathcal{A}$ simply outputs $M(\pi, x, y)$.

We turn to analyzing our attacker $\mathcal{A}$. Recall that M decides $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]$ on every (π, x, y), $|\pi| = |x| = |y| = l$, for infinitely many l of the form $l = 2n$. Fix some sufficiently large n such that M succeeds on $l = 2n$, and let E_n be the event associated with the cond EP-PRG g on seed length n. The following two claims will show that $\mathcal{A}$ distinguishes g from random on input length n.

Claim 7.5. *$\mathcal{A}(1^n, r)$ will output 0 with probability at least $\frac{1}{2} - \frac{2}{n} - 0.01$ where $r \leftarrow \mathcal{U}_m$.*

Proof. Given $r \leftarrow \mathcal{U}_m$, let $(\pi, x, y) = h(r)$. Recall that π is half random and half all-zero, and by a standard counting argument for K, we have that with probability at least $1/2$, $\mathrm{K}(\pi) \geq n - 1$. In addition, it follows from Equation 15 that with probability at least $1 - \frac{1}{n}$, $\mathrm{K}(\pi, x, y) > n + c_2 \log n + 1$. By a Union bound, $\mathrm{K}(\pi, x, y) - \mathrm{K}(\pi) \leq c_2 \log n$ with probability $1/2 - 1/n$. Finally, notice that if this is the case, (π, x, y) will be a NO instance, and M will output 0 with probability at least 0.99, and this claim follows from again a union bound. □

Claim 7.6. *$\mathcal{A}(1^n, s)$ will output 0 with probability at most 0.3 where $s \leftarrow \mathcal{U}_n \mid E_n$.*

Proof. Let S denote the random variable distributed as $g(\mathcal{U}_n \mid E_n)$. Let Π, X, Y be the random variable such that $(\Pi, X, Y) = h(S)$. It follows from Equation 14 that the IK^t-complexity of Π, X, Y

is at most

$$\mathsf{IK}^t(\Pi; X; Y) \leq n + \gamma$$

with probability 1. We turn to proving that Π also has high K-complexity with high probability. By the definition of Π and h, it holds that

$$H(\Pi) \geq H([S]_n)$$

which is at least

$$n - 0.1c_1 \log n$$

since g is entropy-preserving conditioned on E_n. Let $W = \mathsf{supp}(\Pi)$ and $Z = \{z \in \{0,1\}^{2n} : \mathrm{K}(z) < n - (c_1/2)\log n\} \cap W$. Notice that $2^{H(\Pi)} \leq |W| \leq 2^n$, and by a standard counting argument $|Z| \leq 2^{n-(c_1/2)\log n+1}$. By Lemma 2.4, we have that the probability that $\Pi \in Z$ is at most

$$\Pr[\Pi \in Z] \leq \frac{\log |W| + 1 - H(\Pi)}{\log |W| - \log |Z|} \leq 0.2$$

In other words, the probability that Π has K-complexity $\geq n - (c_1/2)\log n$ is at least 0.8. Conditioned on this event, and recall that Π, X, Y always has small IK^t-complexity, we conclude that

$$\mathsf{IK}^t(\Pi; X; Y) - \mathrm{K}(\Pi) \leq (c_1/2)\log n + \gamma \leq c_1 \log n$$

which implies that (Π, X, Y) is a YES instance. Recall that M will output 1 with probability at least 0.99 on YES instances, it follows by a union bound that $M(\Pi, X, Y)$ will output 1 with probability $0.8 - 0.01$, which concludes the proof of this claim. □

Finally, note that our proof only concerns with x, y where y is an all-zero string, and x is an almost all-zero string except for its first $(c_2 + 2)\log n$ bits. It follows that

$$\mathrm{K}^t(y \mid x) \leq \mathrm{K}^t(x \mid y) \leq 2\log n + (c_2 + 2)\log n \leq \Delta$$

Thus, for any $s \in \{0,1\}^m$, $(\pi, x, y) = h(s)$, it holds that $(\pi, x, y) \in Q_\Delta$, and our proof also shows the hardness of $\mathsf{RIK}^t\mathsf{P}[c_1 \log n, c_2 \log n]|_{Q_\Delta}$. □

References

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. "Public-key cryptography from different assumptions". In: *Proceedings of the forty-second ACM symposium on Theory of computing*. 2010, pp. 171–180 (cit. on p. 2).

[AD97] Miklós Ajtai and Cynthia Dwork. "A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence". In: *stoc29*. See also ECCC TR96-065. 1997, pp. 284–293 (cit. on p. 2).

[AF09] Luis Antunes and Lance Fortnow. "Worst-case running times for average-case algorithms". In: *2009 24th Annual IEEE Conference on Computational Complexity*. IEEE. 2009, pp. 298–303 (cit. on p. 5).

[AFVMV06] Luis Antunes, Lance Fortnow, Dieter Van Melkebeek, and N Variyam Vinodchandran. "Computational depth: concept and applications". In: *Theoretical Computer Science* 354.3 (2006), pp. 391–404 (cit. on p. 4).

[Ale03] Michael Alekhnovich. "More on average case vs approximation complexity". In: *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* IEEE. 2003, pp. 298–307 (cit. on p. 2).

[Bar13] Boaz Barak. "Structure vs Combinatorics in Computational Complexity". In: (2013) (cit. on p. 2).

[Bar14] Boaz Barak. "Structure vs combinatorics in computational complexity". In: *Bulletin of EATCS* 1.112 (2014) (cit. on p. 2).

[BCNHR22] Andrej Bogdanov, Miguel Cueto Noval, Charlotte Hoffmann, and Alon Rosen. "Public-Key Encryption from Homogeneous CLWE". In: *Theory of Cryptography: 20th International Conference, TCC 2022, Chicago, IL, USA, November 7–10, 2022, Proceedings, Part II*. Springer. 2022, pp. 565–592 (cit. on p. 2).

[DH76] Whitfield Diffie and Martin E. Hellman. "New Directions in Cryptography". In: *IEEE Transactions on Information Theory* (1976), pp. 644–654 (cit. on pp. 2, 9).

[ElG84] Taher ElGamal. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms". In: *Annual International Cryptology Conference (CRYPTO)*. 1984, pp. 10–18 (cit. on p. 2).

[GL89] Oded Goldreich and Leonid A. Levin. "A Hard-Core Predicate for all One-Way Functions". In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing (STOC)*. 1989, pp. 25–32 (cit. on pp. 5, 14).

[HILL99] Johan Hastad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. "A pseudorandom generator from any one-way function". In: *SIAM Journal on Computing* (1999), pp. 1364–1396 (cit. on p. 10).

[HKNRR05] Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. "On robust combiners for oblivious transfer and other primitives". In: *Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings 24*. Springer. 2005, pp. 96–113 (cit. on p. 8).

[HN23] Shuichi Hirahara and Mikito Nanashima. "Learning in Pessiland via Inductive Inference". In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2023, pp. 447–457 (cit. on pp. 2, 3).

[Hol06] Thomas Holenstein. "Strengthening key agreement using hard-core sets". PhD thesis. ETH Zurich, 2006 (cit. on pp. 5, 13).

[ILL89] Russell Impagliazzo, Leonid A Levin, and Michael Luby. "Pseudo-random generation from one-way functions". In: *Annual ACM Symposium on Theory of Computing (STOC)*. 1989, pp. 12–24 (cit. on p. 14).

[Imp95] Russell Impagliazzo. "A Personal View of Average-Case Complexity." In: *Proceedings of the Tenth Annual Structure in Complexity Theory Conference*. IEEE Computer Society, 1995, pp. 134–147 (cit. on p. 2).

[IRS22] Rahul Ilango, Hanlin Ren, and Rahul Santhanam. "Robustness of average-case meta-complexity via pseudorandomness". In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. 2022, pp. 1575–1583 (cit. on p. 12).

[Lev85] Leonid A Levin. "One-way functions and pseudorandom generators". In: *Proceedings of the seventeenth annual ACM symposium on Theory of computing*. 1985, pp. 363–365 (cit. on p. 8).

[LP20] Yanyi Liu and Rafael Pass. "On one-way functions and Kolmogorov complexity". In: *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2020, pp. 1243–1254 (cit. on pp. 2, 3, 5, 9, 10, 12, 32).

[LP21] Yanyi Liu and Rafael Pass. "Cryptography from Sublinear Time Hardness of Time-bounded Kolmogorov Complexity". In: *STOC*. 2021 (cit. on pp. 2, 3).

[LP22] Yanyi Liu and Rafael Pass. "On one-way functions from NP-complete problems". In: *Proceedings of the 37th Computational Complexity Conference*. 2022, pp. 1–24 (cit. on pp. 2, 3).

[LP23] Yanyi Liu and Rafael Pass. "On One-way Functions and the Worst-case Hardness of Time-Bounded Kolmogorov Complexity". In: *Cryptology ePrint Archive* (2023) (cit. on pp. 2–5, 7, 8, 12).

[McE78] Robert J McEliece. "A public-key cryptosystem based on algebraic". In: *Coding Thv* 4244 (1978), pp. 114–116 (cit. on p. 2).

[Rab79] Michael O Rabin. *Digitalized signatures and public-key functions as intractable as factorization*. Tech. rep. Massachusetts Inst of Tech Cambridge Lab for Computer Science, 1979 (cit. on p. 2).

[Reg09] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Journal of the ACM (JACM)* 56.6 (2009), pp. 1–40 (cit. on p. 2).

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (1978), pp. 120–126 (cit. on p. 2).

[Sho99] Peter W Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". In: *SIAM review* 41.2 (1999), pp. 303–332 (cit. on p. 2).

[Yao79] Andrew Chi-Chih Yao. "Some complexity questions related to distributive computing (preliminary report)". In: *Proceedings of the eleventh annual ACM symposium on Theory of computing*. 1979, pp. 209–213 (cit. on p. 3).

[Yao82] Andrew C. Yao. "Theory and Applications of Trapdoor Functions". In: *Annual Symposium on Foundations of Computer Science (FOCS)*. 1982, pp. 80–91 (cit. on pp. 12, 14).

[Zvo] "The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms". In: *Russian Mathematical Surveys* 25.6 (1970), p. 83 (cit. on pp. 7, 19).

A $\mathsf{GapMIK}^t\mathsf{P}$ and Average-case Hardness

We also consider the $\mathsf{GapMIK}^t\mathsf{P}$ promised problem.

Definition A.1 ($\mathsf{GapMIK}^t\mathsf{P}$). *For functions $\sigma_Y < \sigma_N$ and t, let $\mathsf{GapMIK}^t\mathsf{P}[\sigma_Y, \sigma_N]$ denote the following promise problem:*

- $\mathcal{Y} = \{(\pi, x, y) \in (\{0,1\}^n)^3 \colon \mathsf{IK}^t(\pi; x; y) \leq \sigma_Y\}$,
- $\mathcal{N} = \{(\pi, x, y) \in (\{0,1\}^n)^3 \colon \mathsf{K}(\pi, x, y) \geq \sigma_N\}$.

In the following we prove that the average case hardness of $\mathsf{GapMIK}^t\mathsf{P}$ on some distributions imply the worst-case hardness of $\mathsf{RIK}^t\mathsf{P}$.

Definition A.2 (Hard on average). *Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a distribution ensemble with $\mathrm{Supp}(\mathcal{D}_n) \subseteq \{0,1\}^n$. We say that $\mathcal{L} = (\mathcal{Y}, \mathcal{N})$ is ϵ-hard on average on $\mathcal{D}$ if for every* PPT *algorithm A, for every large enough n with $(\mathcal{Y} \cup \mathcal{N}) \cap \{0,1\}^n \neq \emptyset$,*

$$\Pr_{x \leftarrow \mathcal{D}_n}[(x \in \mathcal{Y} \wedge \mathsf{A}(x) \neq 1) \vee (x \in \mathcal{N} \wedge \mathsf{A}(x) = 1)] \geq \epsilon(n).$$

We consider flat distributions.

Definition A.3 (Flat distribution). *A distribution $\mathcal{D}$ is* flat *if $\mathcal{D}$ is uniform over its support.*

We next prove the following theorem.

Theorem A.4. *Let $0 < c_1 < c_2$, $\beta > 0$ be constants. Let $\mathcal{D} = (\Pi = \{\Pi_n\}_{n\in\mathbb{N}}, X = \{X_n\}_{n\in\mathbb{N}}, Y = \{Y_n\}_{n\in\mathbb{N}})$ be a distribution over $\{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n$, such that for every $n \in \mathbb{N}$, Π_n is a computable, flat distribution with $\mathrm{H}_\infty(\Pi_n) = s(n)$. Assume that $\mathsf{GapMIK}^t\mathsf{P}[s(n) + c_1 \log n, s(n) + c_2 \log n]$ is $2n^{-\beta}$-hard on average over $\mathcal{D}$. Then for every constant $1 < \alpha$,*

$$\mathsf{RIK}^t\mathsf{P}[(c_1 + \beta)\log n, (c_2 - \alpha)\log n] \notin \mathsf{ioBPP}.$$

Moreover, if for some $\Delta = \Delta(n)$ and every $n \in \mathbb{N}$, $\Pr\left[\mathsf{K}^t(X_n \mid Y_n) \leq \Delta(n), \mathsf{K}^t(Y_n \mid X_n) \leq \Delta(n)\right] = 1$, then $\mathsf{RIK}^t\mathsf{P}[(c_1 + \beta)\log n, (c_2 - \alpha)\log n]|_{Q_\Delta} \notin \mathsf{ioBPP}$.

Before proving Theorem A.4, we derive the following corollary.

Corollary A.5. *Let c_1, c_2, $\beta > 0$ and $d \geq 0$ be constants such that $c_2 - 16 - 2d - \beta > c_1 \geq 0$, and $t = t(n)$ be an efficiently computable function. Let $\mathcal{D} = (\Pi = \{\Pi_n\}_{n\in\mathbb{N}}, X = \{X_n\}_{n\in\mathbb{N}}, Y = \{Y_n\}_{n\in\mathbb{N}})$ be a distribution over $\{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n$, such that for every $n \in \mathbb{N}$, Π_n is a computable, flat distribution with $\mathrm{H}_\infty(\Pi_n) = s(n)$.*

- *If $\mathsf{GapMIK}^t\mathsf{P}[s(n) + c_1 \log n, s(n) + c_2 \log n]$ is $2n^{-\beta}$-hard on average over $\mathcal{D}$, one-way functions exist.*
- *If additionally, $\Pr\left[\mathsf{K}^t(X_n \mid Y_n) \leq d \log n, \mathsf{K}^t(Y_n \mid X_n) \leq d \log n\right] = 1$, key-agreement protocols exist.*

Proof of Corollary A.5. Immediate from Theorems 3.4, 3.5 and A.4. □

In the proof of Theorem A.4, we will use the coding theorem.

Lemma A.6 (Coding theorem). *Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a computable distribution ensemble over $\{0,1\}^n$. Then for every $n \in \mathbb{N}$ and every $x \in \mathrm{Supp}(\mathcal{D}_n)$, it holds that*

$$\mathsf{K}(x) \leq -\log(\Pr_{\mathcal{D}}[x]) + \log n + 2\log\log n + O(1).$$

Proof of Theorem A.4. Let $c_1, c_2, \delta, \alpha, s$ and $\mathcal{D}$ be as in Theorem A.4. We start by bounding the Kolmogorov complexity of a random sample from Π. By Lemma 2.23, with probability at least $1 - n^{-\beta}$ over $\pi \leftarrow \Pi_n$,

$$\mathsf{K}(\pi) \geq s(n) - \beta \log n.$$

By Lemma A.6, and since Π is computable and flat,

$$\mathsf{K}(\pi) \leq s(n) + \log n + 2\,\mathrm{loglog}\, n + O(1) \leq s(n) + \alpha \log n.$$

Next, assume toward a contradiction that $\mathsf{RIK}^t\mathsf{P}[(c_1 + \beta)\log n, (c_2 - \alpha)\log n] \in \mathsf{ioBPP}$, and let A be the algorithm that, for infinite many n's, decides $\mathsf{RIK}^t\mathsf{P}[(c_1 + \beta)\log n, (c_2 - \alpha)\log n]$ with error at most $n^{-\beta}$ on every input of length $3n$. In the following we show that A solves $\mathsf{GapMIK}^t\mathsf{P}[s(n) + c_1 \log n, s(n) + c_2 \log n]$ on (Π_n, X_n, X_n) with small error for any such n. Indeed, fix such n and fix $(\pi, x, y) \in \mathrm{Supp}(\Pi_n, X_n, Y_n)$ such that

$$s(n) - \beta \log n \leq \mathsf{K}(\pi) \leq s(n) + \alpha \log n$$

. Observe that if $\mathsf{IK}^t(\pi; x; y) \leq s(n) + c_1 \log n$, then it holds that

$$\mathsf{IK}^t(\pi; x; y) \leq s(n) + c_1 \log n \leq \mathsf{K}(\pi) + \beta \log n + c_1 \log n,$$

and thus A must answer correctly for any such π, x, y with probability at least $1 - n^{-\delta}/2$. Similarly, if $\mathsf{K}(\pi, x, y) \geq s(n) + c_2 \log n$, then,

$$\mathsf{K}(\pi, x, y) \geq s(n) + c_2 \log n \geq \mathsf{K}(\pi) - \alpha \log n + c_2 \log n,$$

and thus A must answer correctly for any such π, x, y with probability at least $1 - n^{\beta}$. Overall, with probability at least $1 - n^{-\beta}$ over Π_n, A error with probability at most $n^{-\beta}$. Thus, the error probability of A over (Π_n, X_n, Y_n) is at most $2n^{-\beta}$, which is a contradiction.

The moreover part of the theorem follows by the same proof, by noticing that it is enough to assume that A succeed on the support of the distribution (Π, X, Y). □

B Missing proofs

B.1 Proving Lemma 2.3

Lemma B.1 (Lemma 2.3, restated)**.** *Let X_1 and X_2 be distributions over a set Ω, such that $\mathrm{SD}(X_1, X_2) = \epsilon$. Then there exist random variables W_1 and W_2, jointly distributed with X_1 and X_2 respectively, such that $\Pr[W_1 = 1] = \Pr[W_2 = 1] = 1 - \epsilon$, and $X_1|_{W_1=1} \equiv X_2|_{W_2=1}$.*

Proof. Let P_1 be a random variable over $[0, 1]$, jointly distributed with X_1, defined as follows. For every x, $P_1|_{X_1=x}$ is uniformly distributed over the interval $[0, \Pr[X_1 = x]]$. Define the random variable P_2 (jointly distributed with X_2) symmetrically with respect to X_2.

For every $x \in \Omega$, let $p_x = \min\{\Pr[X_1 = x], \Pr[X_2 = x]\}$, and let W_1 be a random variable such that $W_1 = 1$ iff $P_1 \leq p_{X_1}$, and W_2 be such that $W_2 = 1$ iff $P_2 \leq p_{X_2}$. Then it holds that,

$$\begin{aligned}
\Pr[W_1 = 1] &= \sum_{x \in \Omega} \Pr[X_1 = x] \frac{p_x}{\Pr[X_1 = x]} \\
&= \sum_{x \in \Omega} \min\{\Pr[X_1 = x], \Pr[X_2 = x]\} \\
&= \sum_{x \in \Omega} \min\{\Pr[X_1 = x], \Pr[X_2 = x]\} \\
&\quad + 1/2 \cdot \sum_{x \in \Omega} \max\{\Pr[X_1 = x], \Pr[X_2 = x]\} - 1/2 \cdot \sum_{x \in \Omega} \max\{\Pr[X_1 = x], \Pr[X_2 = x]\} \\
&= 1/2 \cdot \sum_{x \in \Omega} \min\{\Pr[X_1 = x], \Pr[X_2 = x]\} + 1/2 \cdot \sum_{x \in \Omega} \max\{\Pr[X_1 = x], \Pr[X_2 = x]\} \\
&\quad + 1/2 \cdot \sum_{x \in \Omega} \min\{\Pr[X_1 = x], \Pr[X_2 = x]\} - 1/2 \cdot \sum_{x \in \Omega} \max\{\Pr[X_1 = x], \Pr[X_2 = x]\} \\
&= 1/2 \cdot \sum_{x \in \Omega} (\Pr[X_1 = x] + \Pr[X_2 = x]) - 1/2 \cdot \sum_{x \in \Omega} |\Pr[X_1 = x] - \Pr[X_2 = x]| \\
&= 1 - \epsilon,
\end{aligned}$$

and similarly $\Pr[W_2 = 1] = 1 - \epsilon$. Moreover, for every $x \in \Omega$,

$$\begin{aligned}
\Pr[X_1 = x \mid W_1 = 1] &= \frac{\Pr[X_1 = 1] \Pr[P_1 \leq p_x]}{\Pr[W_1 = 1]} = \frac{p_x}{\Pr[W_1 = 1]} \\
&= \frac{p_x}{\Pr[W_2 = 1]} = \frac{\Pr[X_2 = 1] \Pr[P_2 \leq p_x]}{\Pr[W_2 = 1]} = \Pr[X_2 = x \mid W_2 = 1].
\end{aligned}$$

□

B.2 Proving Lemma 2.13

Lemma B.2 (Lemma 2.13, restated)**.** *The following holds for every constants $\alpha > \beta$. Assume there exists an n-bit, $(1-n^{-\alpha}, 1-n^{-\beta})$-key agreement protocol. Then, there exists a key-agreement protocol.*

To prove the above lemma, we use the following weak version of GL.

Lemma B.3. *There exists a* PPT *oracle-aided algorithm* Dec *such that the following holds. Let $n \in N$ be a number, $x \in \{0,1\}^n$, and and let* Pred *be an algorithm such that*

$$\Pr_{r \leftarrow \{0,1\}^n}[\mathsf{Pred}(r) = \mathrm{GL}(x, r)] > 3/4 + 0.01,$$

where $\mathrm{GL}(x, r) := \langle x, r \rangle$ is the Goldreich-Levin predicate. Then $\Pr\left[\mathsf{Dec}^{\mathsf{Pred}}(1^n) = x\right] = 1 - \mathrm{neg}(n)$.

Proof of Lemma B.3. We use Pred to decode each bit of x separately. For every i, let e_i be the vector that has 1 in the i-th entry, and 0's everywhere else. Observe that, for a uniformly chosen $R \leftarrow \{0,1\}^n$,

$$\Pr[\mathsf{Pred}(R) = \mathrm{GL}(x, R) \wedge \mathsf{Pred}(R \oplus e_i) = \mathrm{GL}(x, R \oplus e_i)] \geq 1/2 + 0.01.$$

Thus,

$$\Pr[\mathsf{Pred}(R) \oplus \mathsf{Pred}(R \oplus e_i) = \mathrm{GL}(x, R) \oplus \mathrm{GL}(x, R \oplus e_i)] \geq 1/2 + 0.01.$$

By linearity of the inner product we get that,

$$\Pr[\mathsf{Pred}(R) \oplus \mathsf{Pred}(R \oplus e_i) = x_i] \geq 1/2 + 0.01.$$

Let Dec be the algorithm that for every i, computes $\mathsf{Pred}(R) \oplus \mathsf{Pred}(R \oplus e_i)$ for n random values of R, and let x'_i to be the majority of the outputs. Then, Dec outputs $x' = x'_1, \ldots, x'_n$. By Chernoff bound, x'_i is equal to x_i with all but negligible probability. By the union bound, the above is true for all i's simultaneously with all but negligible probability, as we wanted to show. □

Proof of Lemma 2.13. Let (A, B) be a n-bit, $(1 - n^{-\alpha}, 1 - n^{-\beta})$-key agreement protocol. We will construct a 1-bit, $(1 - n^{-\alpha}, 1 - n^{-\beta}/10)$-key agreement protocol. The lemma than follows by Lemma 2.12 by choosing $(\alpha', \beta') = (\alpha, \beta + (\alpha - \beta)/2)$.

Let $(\mathsf{A}', \mathsf{B}')(1^n)$ be the protocol in which the parties simulates $(\mathsf{A}, \mathsf{B})(1^n)$ to get transcript π and outputs x and y respectively. Then, A' samples $r \in \{0,1\}^n$ and sends it to B'. Finally, A' outputs $\langle r, x \rangle$ and B' outputs $\langle r, y \rangle$.

Clearly, the agreement probability is at least as the agreement probability of (A, B). For leakage, assume toward a contradiction that there exists a PPT algorithm Eve that, given π, r guesses x with probability larger than $1 - n^{-\beta}/10$, for infinitely many n's. Fix such n, and let Π, X, Y, R be the values of π, x, y, r in a random execution of $(\mathsf{A}', \mathsf{B}')(1^n)$. Let $\mathcal{B} = \big\{(\pi, x) \colon \Pr_{r \leftarrow \{0,1\}^n}[\mathsf{Eve}(1^n, \pi, R) = \langle r, x \rangle] \geq 7/8\big\}$. Then,

$$\begin{aligned} 1 - n^{-\beta}/10 < \Pr[\mathsf{Eve}(1^n, \Pi, R) = \langle R, X \rangle] \leq & 7/8 \cdot (\Pr[(\Pi, X) \notin \mathcal{B}]) + \Pr[(\Pi, X) \in \mathcal{B}] \\ & = 7/8 + 1/8 \cdot \Pr[(\Pi, X) \in \mathcal{B}] \end{aligned}$$

Thus, $\Pr[(\Pi, X) \in \mathcal{B}] > 1 - 8/10 \cdot n^{-\beta}$. Let Dec be the algorithm promised by Lemma B.3, and let $\mathsf{Eve}_\pi(r) = \mathsf{Eve}(1^n, \pi, r)$. By Lemma B.3, for every $(\pi, x) \in \mathcal{B}$ it holds that $\Pr\big[\mathsf{Dec}^{\mathsf{Eve}_\pi}(1^n) = x\big] \geq 1 - \mathrm{neg}(n)$. We get that,

$$\Pr\Big[\mathsf{Dec}^{\mathsf{Eve}_\Pi}(1^n) = X\Big] \geq \Pr[(\pi, X) \in \mathcal{B}](1 - \mathrm{neg}(n)) \geq 1 - 8/10 \cdot n^{-\beta}(1 - \mathrm{neg}(n)) > 1 - n^{-\beta},$$

with contradiction to the leakage assumption of (A, B). □