Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 43 (2018)

An operational characterization of mutual information in algorithmic information theory*

Andrei Romashchenko † Marius Zimand ‡

April 29, 2019

Abstract

We show that the mutual information, in the sense of Kolmogorov complexity, of any pair of strings x and y is equal, up to logarithmic precision, to the length of the longest shared secret key that two parties, one having x and the complexity profile of the pair and the other one having y and the complexity profile of the pair, can establish via a probabilistic protocol with interaction on a public channel. For $\ell > 2$, the longest shared secret that can be established from a tuple of strings $(x_1, ..., x_\ell)$ by ℓ parties, each one having one component of the tuple and the complexity profile of the tuple, is equal, up to logarithmic precision, to the complexity of the tuple minus the minimum communication necessary for distributing the tuple to all parties. We establish the communication complexity of secret key agreement protocols that produce a secret key of maximal length, for protocols with public randomness. We also show that if the communication complexity drops below the established threshold, then only very short secret keys can be obtained.

Keywords: Kolmogorov complexity; mutual information; communication complexity; secret key agreement; information inequalities; Gács–Körner common information; information reconciliation

1 Introduction

Mutual information is a concept of central importance in both information theory (IT) and algorithmic information theory (AIT), also known as Kolmogorov complexity. We show an interpretation of mutual information in AIT, which links it to a basic notion in cryptography. Even though a similar interpretation was known in the IT framework, an operational characterization of mutual information in AIT has been elusive till now.

To present our result, let us consider two strings x and y. It is common to draw a Venn-like diagram such as the one in Figure 1 to visualize the information relations between them. As explained in the figure legend there are six important regions. The regions (1) to (5) have a clear operational meaning. For instance, $C(x)$ is the length of a shortest program that prints x, $C(x \mid y)$ is the length of a shortest program that prints x when y is given to it, and so on. On the other hand, the mutual information $I(x : y)$ from region (6) is defined by a formula: $I(x : y) = C(x) + C(y) - C(x, y)$. Intuitively, it is the information shared by x and y. But, is there an operational interpretation of the mutual information? As mentioned above, we give a positive answer: The mutual information of x and y is essentially equal to the length of a longest shared secret key that two parties, one having x and the other one having y, and both parties also possessing the complexity profile of the two strings, can establish via a probabilistic protocol.

The following simple example illustrates the above concepts. Suppose that Alice and Bob want to agree on a common secret key. If they could meet face-to-face, they could just generate such a key by, say, flipping a coin. Unfortunately, they cannot meet in person and what makes the situation really troublesome is that they can only

*A preliminary version of this work has been presented at 45thInternational Colloquium on Automata, Languages, and Programming (ICALP), Prague, July 10-13, 2018.

†LIRMM, University of Montpellier & CNRS; on leave from IITP RAS. Email: andrei.romashchenko@lirmm.fr; supported in part by supported in part by the ANR through grant RaCAF ANR-15-CE40-0016-01.

‡Department of Computer and Information Sciences, Towson University, Baltimore, MD. http://orion.towson.edu/~mzimand/ ; supported in part by the National Science Foundation through grant 1811729.

ISSN 1433-8092

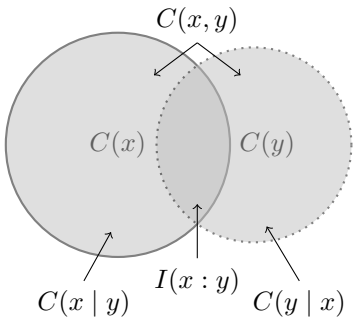

Figure 1: Two strings x and y, and their information. There are six regions that we distinguish: (1) The left solid circle represents the information in x, as given by its Kolmogorov complexity, denoted $C(x)$; (2) The right dotted circle represents the information in y, denoted $C(y)$; (3) The entire grey region (the two circles taken together) represents the information in x and y, denoted $C(x, y)$; (4) The light-grey region in the first circle represents the information in x conditioned by y, denoted $C(x \mid y)$; (5) The light-grey region in the second circle represents the information of y conditioned by x, denoted $C(y \mid x)$; and (6) the dark-grey region in the middle represents the mutual information of x and y, denoted $I(x : y)$.

communicate through a public channel. There is, however, a gleam of hope because Alice knows a random line x in the affine plane over the finite field with 2^n elements, and Bob knows a random point y on this line. The line x is specified by the slope a and the intercept b and the point y by its two coordinates c and d. Therefore each of x and y has $2n$ bits of information, but, because of the geometrical correlation, together they have $3n$ bits of information. Thus, in principle, Alice and Bob share n bits. Can they use them to obtain a common secret key? The answer is yes: Alice sends a to Bob, Bob, knowing that his point is on the line, finds x, and now they can use b as the secret key, because the adversary has only seen a, and a and b are independent.

It may appear that the geometrical relation between x and y is crucial for the above solution. In fact it is just a red herring and Alice and Bob can agree on a common secret key in a very general setting. To describe it, we consider the scenario in which Alice has a random string x and Bob has a random string y. If $x = y$, then Alice and Bob can use their common string as a secret key in an encryption scheme (such as the one-time pad) and achieve perfect information-theoretical security. What happens if x and y are not equal but only correlated? Somewhat surprisingly, for many interpretations of "correlated," they can still agree on a shared secret key via interaction on a public channel (for instances of this assertion, see [LYC76, BBR88, Mau93, AC93]). In this paper, we look at this phenomenon using the very general framework of algorithmic information theory to measure the correlation of strings.

We proceed with a description of our results.

1.1 Our contributions

Characterization of mutual information. In a secret key agreement protocol, Alice and Bob, on input x and respectively y, exchange messages and compute a common string that is random conditioned by the transcript of the protocol. Such a string is said to be a *shared secret key*. Unless specified otherwise, we use protocols having the following features:

(1) We assume that Alice and Bob also know how their x and y are correlated. In our setting this means that Alice and Bob know the complexity profile of x and y, which, by definition, is the tuple $(C(x), C(y), C(x, y))$.

(2) The protocols are effective and randomized, meaning that Alice and Bob use probabilistic algorithms to compute their messages.

Theorem 1.1 (Main Result, informal statement)**.** *1. There is a secret key agreement protocol that, for every n-bit strings x and y, allows Alice and Bob to compute with high probability a shared secret key of length equal to the mutual information of x and y (up to an $O(\log n)$ additive term).*

2. *No protocol can produce a longer shared secret key (up to an $O(\log n)$ additive term).*

Secret key agreement for three or more parties. Mutual information is only defined for two strings, but secret key agreement can be explored for the case of more strings. Let us consider again an example. Suppose that each of Alice, Bob, and Charlie have a point in the affine plane over the finite field with 2^n elements, and that the three points, which we call A, B, C, are collinear. Thus, each party has $2n$ bits of information, but together they have $5n$ bits of information, because given two points, the third one can be described with n bits. The parties want to establish a common secret key, but they can only communicate by broadcasting messages over a public channel. They can proceed as follows. Alice will broadcast a string p_A, Bob a string p_B, and Charlie a string p_C, such that each party using his/her point and the received information will reconstruct the three collinear points A, B, C. A protocol that achieves this is called an *omniscience protocol* because it spreads to everyone the information possessed at the beginning individually by each party. In the next step, each party will compress the $5n$ bits, comprising the three points, to a string that is random given p_A, p_B, p_C. The compressed string is the common secret key. We will see that up to logarithmic precision it has length $5n - (|p_A| + |p_B| + |p_C|)$. Assuming we know how to do the omniscience protocol and the compression step, this protocol produces a common secret key of length $5n - \mathrm{CO}(A, B, C)$, where $\mathrm{CO}(A, B, C)$ is the minimum communication for the omniscience task for the points A, B, C. In our example, it is clear that each one of p_A, p_B, p_C must be at least n bits long, and that any two of these strings must contain together at least $3n$ bits. Using some recent results from the reference [Zim17], it can be shown that any numbers satisfying these constraints can be used for the omniscience task. It follows that the smallest communication for omniscience is achieved when $|p_A| = |p_B| = |p_C| = 1.5n$, and thus the key has $5n - 4.5n = 0.5n$ bits. (Warning: in the entire discussion we ignore additive terms of size $O(\log n)$.) We show that this holds in general. If ℓ parties have, respectively, one component of a tuple $(x_1, \ldots, x_\ell)$ of n-bit strings, then up to $O(\log n)$ precision, they can produce a common secret key of length $C(x_1, \ldots, x_\ell) - \mathrm{CO}(x_1, \ldots, x_\ell)$, where $\mathrm{CO}(x_1, \ldots, x_\ell)$ is the minimum communication for the omniscience task (which, as we show in the paper, depends only on the complexity profile of the tuple). The protocol that produces such a key is probabilistic, and, as was the case for two strings, assumes that each party i has at the beginning of the protocol besides its input string x_i also the complexity profile of the entire tuple $(x_1, \ldots, x_\ell)$. We also show a matching (up to $O(\log n)$) upper bound, which holds for any number of rounds in the protocol.

Communication complexity for secret key agreement. In the protocol in Theorem 1.1, Alice and Bob exchange $\min(C(x \mid y), C(y \mid x)) + O(\log n)$ bits and obtain with high probability a shared secret key of length $I(x : y) - O(\log n)$. In this protocol we can assume that Alice and Bob use either private random bits or public random bits. We show that for the model with public random bits, the communication complexity of the protocol is optimal, in the sense that in any protocol with public random bits there are input strings x and y, on which Alice and Bob have to exchange at least $\min(C(x \mid y), C(y \mid x))$ bits. In fact, our lower bound is stronger: we show that, for any constants $\delta_1, \delta_2 > 0$, if Alice and Bob use a protocol with communication complexity $(1-\delta_1)\min(C(x \mid y), C(y \mid x))$ for every input pair x, y, then there are inputs for which the shared secret key that they obtain has length at most $\delta_2 I(x : y)$. That is, if the communication complexity sinks below the threshold $\min(C(x \mid y), C(y \mid x))$, then the size of the common secret key drops to virtually zero. Finding the optimal communication complexity for the model with private random bits remains an open problem.

1.2 Related previous work.

IT vs. AIT. Before reviewing existing related results in the IT and the AIT frameworks, it is useful to understand the distinction between the two theories. In computer science the attribute *random* is mainly used in two (fairly different) contexts: *random processes* and *random objects*. In short, IT, which we also call Shannon's framework, focuses on the former, whereas AIT, which we also call Kolmogorov's framework, focuses on the latter. On the one hand, we may think of an uncertain physical process with unpredictable outcomes, and employ the framework of the classic probability theory (distributions, random variables, etc.). The notion of a *random variable* formalizes the idea of a process like coin tossing. In this context we can measure the uncertainty of a random variable as a whole (by its Shannon's entropy, its min-entropy, etc.), but we cannot ask whether one specific outcome is random or not. On the other hand, people use *tables of random numbers*, which are available as specific sequences of digits, written on a disc or printed on a paper. The usefulness of such a table depends on its individual properties: frequencies of

digits, presence or absence of hidden regularities, compressibility, etc. The usual way to measure the uncertainty of an individual string of digits is Kolmogorov complexity. In both contexts, the formal measures of randomness may or may not involve computational complexity (see, e.g., different versions of pseudoentropy for distributions and the resource bounded variants of Kolmogorov complexity for individual strings). These two formalizations of randomness are connected but not interchangeable.

Both notions of randomness appear in cryptography. For example, in the one-time pad scheme, two parties share a 'random' key that remains 'secret' for the attacker. It is common to use Shannon's framework, and therefore the notions of randomness and secrecy are defined in terms of random processes. In the ideal situation both parties should have access to a common source of randomness, e.g., to the results of tossing an unbiased coin (hidden from the adversary). By tossing this coin n times we get a random variable with maximal possible entropy, and thus, in Shannon's framework, the quality of randomness is perfect. But if by chance we obtain a sequence of n zeros, then this specific one-time pad looks pretty useless in any practical application. However, Shannon's information theory provides no vocabulary to complain about this apparently non-random *individual* key. Antunes *et al.* [ALPS10] suggested to use Kolmogorov complexity to measure the 'secrecy' of individual instances of a one-time pad or a secret sharing schemes. We have in mind a similar motivation, and in this work a "secret key" is an individual string that is random in the sense of Kolmogorov complexity.

Related work. We start with a brief account of works on secret key agreement in the IT setting. The secret key agreement is a relatively well-studied problem in information theory, motivated, as the name suggests, by applications in information-theoretically secure cryptography. Wyner [Wyn75] and Csiszár and Körner [CK78] have analyzed the possibility of obtaining a shared secret key in the case when one party sends to the other party a single message on a channel from which the eavesdropper can obtain partial information. Maurer [Mau92, Mau93] considered the case of protocols with several rounds of communication and showed that interaction can be more powerful than one-way transmission. Ahlswede and Csiszár [AC93] and Maurer [Mau93] have established the tight relation between interactive secret key agreement and mutual information for *memoryless* sources. In the memoryless model, the input data is given by two random variables (X_1, X_2) obtained by n independent draws from a joint distribution, where Alice observes X_1 and Bob observes X_2. Informally stated, the references [Mau93, AC93] show that the longest shared secret key that Alice and Bob can establish via an interactive protocol with an arbitrary number of rounds is equal to the mutual information of X_1 and X_2. Csiszár and Narayan [CN04] go beyond the scenario with two parties, and consider the case of an ℓ-memoryless source $(X_1, \ldots, X_\ell)$ and ℓ parties, each one observing one component of the tuple. They show that the longest shared secret key the ℓ parties can establish via an interactive protocol with an arbitrary number of rounds is equal to the entropy $H(X_1, \ldots, X_\ell)$ of the ℓ-memoryless source from which one subtracts the minimum communication for omniscience. Their result holds also for stationary ergodic sources, which generalize memoryless sources. As one can see, our results are very similar. They have been inspired by the papers [AC93, Mau93, CN04] and represent the AIT analogue of the results presented above. As we explain in Section 8.4, our results imply their IT analogues, and can be viewed as more general because they do not require the memoryless or ergodicity properties of sources (in fact they do not require any generative model at all). The question of finding the communication complexity of secret key agreement protocols in the IT framework has been raised by Csiszár and Narayan [CN04]. Tyagi [Tya13] has shown that for memoryless sources it is equal to the difference between interactive common information in Wyner's sense and mutual information. In Section 8.4 we explain how Tyagi's result compares to our results on communication complexity in the AIT framework.

Let us now say a few words about related results from the AIT world. To the best of our knowledge, in AIT there has been no previous work on secret key agreement. However, the general idea of "materialization" of mutual information was studied extensively. Motivated by the intuition that mutual information represents the amount of shared information in two strings, researchers have explored the extent to which mutual information can be materialized more or less effectively. The relevant concept is that of *common information.* Informally, a string z is a common information string extracted from strings x and y, if z can be "computed" from x, and also from y, where "computed" is taken in a more liberal sense that allows the utilization of a few help bits. In the most common setting of parameters, we require that $C(z \mid x) = O(\log n)$ and $C(z \mid y) = O(\log n)$, where n is the length of x and y and the constant hidden in the $O(\cdot)$ notation depends only on the universal machine. Informally, the common information of x and y is the length of a longest common information string that can be extracted from x and y. It can be shown that up to logarithmic precision common information is upper bounded by mutual information. In an influential paper, Gács and Körner [GK73]

have constructed strings x and y for which the common information is much smaller than the mutual information. Moreover, the property of a pair (x, y) of having common information equal to mutual information does not depend solely on the complexity profile of x and y: There exist pairs (x_1, y_1) and (x_2, y_2) having the same complexity profile, and for (x_1, y_1) the common information and mutual information are equal, whereas for (x_2, y_2) they are not. Muchnik [Muc98] and Romashchenko [Rom00] have strengthened the Gács-Körner theorem in significant ways, by allowing a larger amount of help bits, parameterizing the mutual information of the constructed pair (x, y), and other ways. Chernov et al. [CMR$^+$02] presents alternative constructions of strings for which the common information is smaller than mutual information for several regimes of parameters. A nice, self-contained and accessible exposition of this research line can be found in the book of Shen, Vereshchagin and Uspensky [SUV17, Chapter 11].

Thus, previous work has shown negative results regarding the "materialization" of mutual information in AIT. As far as we know, ours is the first positive result. In summary, we now know that computation without communication, even enhanced with help bits, fails to extract the mutual information of two strings, while interactive computation succeeds.

Our techniques. It is common for statements in IT (in the Shannon's entropy framework) to have an analogous version in AIT (in the Kolmogorov complexity framework). However, there is no canonical way to translate a result from one setting to the other, and proofs of homologous results in these two frameworks can be drastically different. A textbook example of this phenomenon is the chain rule: it is valid for Shannon's entropy and for Kolmogorov complexity, and the formal expressions of this rule in both frameworks look very similar. However, in Shannon's case this fact is an easy corollary of the definition, while in Kolmogorov's version it requires a nontrivial argument (which is known as the Kolmogorov–Levin theorem). There are more advanced examples of parallel properties (from IT and AIT respectively), where the discrepancy between their proofs is even more striking.

This phenomenon manifests itself in this work as well. Our main results are motivated by similar ones in IT, and there is a close resemblance of statements. As discussed above, this is not surprising. In what follows we explain the relation between our proofs and the proofs of similar statements in Shannon's framework.

The positive results (the existence of communication protocols) use constructions that at the high level are akin to those from their IT counterparts [Mau93, AC93, CN04]. We employ a similar intuitive idea — manipulations with "fingerprints" of appropriate lengths, see Section 3. However, the technical machinery is different. In the AIT framework, for communication-efficient protocols, we need quite explicit constructions, while homologous results in IT are usually proven by choosing random encodings. Our constructions are based on a combination of extractors and universal hashing. Specifically, we use results from [BZ14] and [Zim17], where the "digital fingerprints" are obtained by composing certain randomness extractors and universal hashing.

Our general protocols are not time-efficient, and this is to be expected given the high generality of the type of data correlation in the AIT setting. However, for some particular types of correlation (e.g., for a pair of inputs with a bounded Hamming distance), our protocols can be modified to run in polynomial-time. In this case, we use the reconciliation technique from [Smi07, GS10, GS16].

In the negative results (upper bounds for the size of the common secret key, Theorems 4.1 and 5.5), the ideas from IT do not help. The reason is that in the AIT framework the mutual information of various strings is not exactly zero but only close to zero within some slack terms. The slack terms are small, but during the rounds of a protocol, the errors can accumulate and grow beyond control (for more details see the discussion of the limits of the "weak" upper bound in Section 3). To overcome this obstacle, we come up with a new type of inequalities for Kolmogorov complexity. These inequalities are substantially different from the classic information inequalities used in the analogous results in IT. This technique (Lemmas 4.6 and 5.10) is based on ideas similar to the *conditional information inequalities* in [KR13, KRV15]. We believe that this technique can be helpful in other cases, including applications in IT (see Section 8).

In the proof of a lower bound for communication complexity (Theorem 6.1), we use methods specific for AIT, with no apparent parallel in IT. We adapt the technique of bounds for the size of common information that goes back to An. Muchnik and use deep results regarding the contrast between *mutual information* and "extractable" *common information* for stochastic for pairs of strings [She83, MR10, Raz11], which have not been previously employed in information theory and communication complexity.

Roadmap. In Section 2 we present the necessary basic concepts and results in Kolmogorov complexity theory used in this paper, and we formally define secret key agreement protocols. Section 3 is a warm-up section and contains

a somewhat simplified version of the main result with a relatively short and self-contained proof. In Section 4 we prove our main result. Section 5 contains the results on ℓ-parties secret key agreement protocols, for $\ell \geq 3$. Section 6 presents the results on the communication complexity of secret key agreement protocols. Section 7 contains the proofs of some technical lemmas. We conclude with some loose-ends and final comments in Section 8.

2 Preliminaries

2.1 The basics of algorithmic information theory

Algorithmic Information Theory (AIT), initiated independently by Solomonoff [Sol64], Kolmogorov [Kol65], and Chaitin [Cha66], is a counterpart to the Information Theory (IT), initiated by Shannon. While in IT, a string is the realization of a random variable, AIT dispenses with the stochastic generative process, and defines the complexity of an individual string x as the length of its shortest description. For example, the string

$$x_1 = 00000000000000000000000000000000$$

has low complexity because it can be succinctly described as "2^5 zeros." The string

$$x_2 = 10110000010101110101010011011100$$

is a 32-bit string obtained using random atmospheric noise (according to random.org), and has high complexity because it does not have a short description.

Formally, given a Turing machine M, a string p is said to be a *program* (or a *description*) of a string x, if M on input p prints x. We denote the length of a binary string x by $|x|$. The *Kolmogorov complexity* of x relative to the Turing machine M is

$$C_M(x) = \min\{|p| \mid p \text{ is a program for } x \text{ relative to } M\}.$$

If U is universal Turing machine, then for every other Turing machine M there exists a string m such that $U(m, p) = M(p)$ for all p, and therefore for every string x,

$$C_U(x) \leq C_M(x) + |m|.$$

Thus, if we ignore the additive constant $|m|$, the Kolmogorov complexity of x relative to U is minimal. We fix a universal Turing machine U, drop the subscript U in $C_U(\cdot)$, and denote the complexity of x by $C(x)$. We list below a few basic facts about Kolmogorov complexity and introduce some notation:

1. For every string x, $C(x) \leq |x| + O(1)$, because a string x is trivially described by itself. (Formally, there is a Turing machine M that, for every x, on input x prints x.)

2. Similarly to the complexity of x, we define the complexity of x conditioned by y as $C(x \mid y) = \min\{|p| \mid U \text{ on input } p \text{ and } y \text{ prints } x\}$.

3. Using some standard computable pairing function $\langle \cdot, \cdot \rangle$ that maps pairs of strings into single strings, we define the complexity of a pair of strings by $C(x, y) = C(\langle x, y \rangle)$. Then we can extend this notation to tuples of larger arity.

4. We use the convenient shorthand notation $a \leq^+ b$ to mean that $a \leq b + O(\log n)$, where n is a parameter that is clear from the context and the constant hidden in the $O(\cdot)$ notation depends on the universal machine U and sometimes on some computable functions specified in the text, but not on a and b. Similarly, $a \geq^+ b$ means $a \geq b - O(\log n)$, and $a =^+ b$ means ($a \leq^+ b$ and $a \geq^+ b$). In several places we use $\leq^+, =^+, \geq^+$ to indicate a loss of precision of $O(\log(n/\epsilon))$ (instead of $O(\log n)$), where ϵ is an additional parameter that is clear from the context.

5. The chain rule in information theory states that $H(X,Y) = H(X) + H(Y \mid X)$. A similar rule holds true in algorithmic information theory: for all sufficiently long strings x and y,

$$\big|C(x,y) - (C(x) + C(y \mid x))\big| \leq 3(\log C(x) + \log C(y)). \tag{1}$$

6. The mutual information of two strings x and y is denoted $I(x:y)$, and is defined as $I(x:y) = C(x) + C(y) - C(x,y)$. The conditional mutual information is defined in the analogous way: $I(x:y \mid z) = C(x \mid z) + C(y \mid z) - C(x,y \mid z)$. It can be shown that mutual information is essentially non-negative (up to an additive term $O(\log C(x,y))$) and it satisfies the chain rule $I(x,z:y) =^+ I(z:y) + I(x:y \mid z)$ (here the n hidden in the $=^+$ is $\max(|x|,|y|,|z|)$).

7. The complexity profile of a tuple of strings $(x_1,\ldots,x_\ell)$ is given by the tuple consisting of the complexities of all non-empty subsets of the strings in the tuple, i.e., it is the tuple $(C(x_V) \mid V \subseteq [\ell], V \neq \emptyset)$. We use here and elsewhere in the paper the notation $[\ell]$ for the set $\{1,\ldots,\ell\}$ and for an ℓ-tuple $(x_1,\ldots,x_\ell)$ and V a subset of $[\ell]$, x_V denotes the subtuple obtained by taking the components with indices in V (for example if $V = \{1,2,7\}$ then $x_V = (x_1,x_2,x_7)$).

8. The *extended* complexity profile of a tuple of strings is the tuple of all conditional and unconditional complexities involving these strings. E.g., the extended complexity profile of a pair (x,y) consists of the values $C(x), C(y), C(x,y), C(x \mid y), C(y \mid x)$.

9. We use the following stronger variant, due to Bruno Bauwens [Bau18], of the main result from reference [Zim17]. It is a version of the Slepian-Wolf theorem about distributed compression of ℓ sources in the setting of Kolmogorov complexity.

   Theorem 2.1 (Distributed compression of ℓ sources)**.** *For every constant positive integer ℓ, there exist a probabilistic encoding algorithms E and a decoding algorithm D such that for every n for every tuple of n-bit strings $(x_1,\ldots,x_\ell)$ and for every tuple of integers $(n_1,\ldots,n_\ell)$,*

   - *For every $i \in [\ell]$, E on input x_i, n_i and ϵ uses $O(\log(n/\epsilon))$ random bits and outputs a string p_i of length $n_i + O(\log(n/\epsilon))$,*
   - *For every string y, if the numbers n_i satisfy the conditions $\sum_{i \in V} n_i \geq C(x_V \mid y, x_{[\ell]-V})$ for every $V \subseteq [\ell]$, then the decoding algorithm D on input y and $p_1,\ldots,p_\ell$ reconstructs $x_1,\ldots,x_\ell$ with probability $1-\epsilon$, where the probability is over the randomness used by the encoding procedures.*

   The following is the particular case of the above theorem for $\ell = 1$. A weaker version appeared in reference [BZ14].

   Theorem 2.2 (Single source compression)**.** *There exists a probabilistic algorithm that on input x, k, ϵ, where x is a string, k is a positive integer and $\epsilon > 0$, returns a string p of length $k + O(\log(n/\epsilon))$, and for every y, if $k \geq C(x \mid y)$, then with probability $(1-\epsilon)$, p is a program for x given y. The algorithm is using $O(\log(n/\epsilon))$ random bits, where n is the length of x.*

2.2 Shared secret keys and protocols for secret key agreement

Let k be a positive integer. A k-rounds two-party protocol for secret key agreement uses two computable functions A and B and runs as follows. The first party has as input a string x_A and uses private randomness r_A, the second party has as input a string x_B and uses private randomness r_B. We assume that the length of r_A (r_B) is determined by x_A

(respectively, x_B). The protocol consists of the following calculations:

$$\begin{aligned} x_1 &= A(x_A, r_A) \\ y_1 &= B(x_B, r_B, x_1) \\ x_2 &= A(x_A, r_A, y_1) \\ y_2 &= B(x_B, r_B, x_1, x_2) \\ &\vdots \\ x_k &= A(x_A, r_A, y_1, \ldots, y_{k-1}) \\ y_k &= B(x_B, r_B, x_1, \ldots, x_k). \end{aligned}$$

The algorithms A and B can handle inputs of different lengths. We also allow them to be partial (i.e., it is possible that the protocol does not converge for some pairs of inputs). Let ϵ be a positive constant and $\delta(n)$ be a constant or a slow growing function (e.g., $O(\log n)$). A protocol *succeeds* with error probability ϵ and randomness deficiency $\delta(n)$ on a pair (x_A, x_B) of n-bit strings if with probability $(1-\epsilon)$ over r_A, r_B,

$$A(x_A, r_A, t) = B(x_B, r_B, t) \stackrel{def.}{=} z, \tag{2}$$

and

$$C(z \mid t) \geq |z| - \delta(n), \tag{3}$$

where $t = (x_1, y_1, \ldots, x_k, y_k)$ is the transcript of the protocol.

The string z satisfying equation (2) and inequality (3) is called a *shared secret key* output by the protocol on input (x_A, x_B). Note that the shared secret key z is a random variable since it depends not only on the inputs x_A and x_B, but also on the randomness r_A and r_B.

The number of rounds in a protocol (parameter k) may depend on the length of the inputs.

In words, Alice and Bob start with input strings x_A and respectively x_B, use private randomness r_A, and respectively r_B and execute a protocol in which at round i, first Alice sends to Bob the string x_i, and next Bob sends to Alice the string y_i, and at the end Alice and Bob separately compute with high probability a common string z (equation (2)) such that z is random even conditioned by the transcript of the protocol (inequality (3)). Thus, z is secret to an adversary that has observed the protocol and consequently knows the transcript.

Remark 1. In our secret key agreement protocols, the inputs x_A and x_B have two components: $x_A = (x, h_A)$ and $x_B = (y, h_B)$, where the strings x and y are the main components, while h_A and h_B are short helping strings (for example, containing information about how x and y are correlated). The communication protocols designed in this paper succeed for all input pairs x_A and x_B in which $h_A = h_B =$ (the complexity profile of x and y). In case one or both of h_A and h_B are not equal to the complexity profile, the protocols still halt on every input, but the outputs may be meaningless.

Remark 2. In this paper we deal with *uniform* communication protocols: for inputs of all lengths Alice and Bob compute their messages and the final results by using the same algorithms A and B. Our main results can be easily extended to non-uniform protocols, with "custom-made" algorithms A and B for each length of inputs. In this setting the technical statements of the theorems would be more cumbersome (we would need to add the description of the protocol in the condition of all complexity terms, assuming that the protocol is known to the adversary; for details, see Remarks 4, 13), though the adjustment of the proofs is quite straightforward.

3 Warm-up: Short proofs for "light"-versions of the main results

Our main result has two parts: the *lower bound*, which consists of a protocol that produces a shared secret key of length equal (within logarithmic precision) to the mutual information of the inputs, and the *upper bound*, where we show that it is impossible to obtain a longer shared secret key. We present here *light* versions of these two parts. We call them *light* because they have weaker parameters than the full-fledged results, which are presented in Section 4. On the other hand, they admit short and self-contained proofs. Moreover, since they have a common structure, these proofs are a useful preamble for the proofs in the next section.

The *light lower bound* consists of a protocol that does allow two parties, one having the n-bit string x and the other having the n-bit string y, to obtain a shared secret key of length approximately equal to $I(x : y)$, but which communication-wise is inefficient. In Section 4 we present an improved protocol that achieves the same task, with communication complexity equal essentially to $\min(C(x \mid y), C(y \mid x))$, which we conjecture to be optimal (see Section 6). The *light upper bound* consists of a proof in which we show that no protocol with a constant number of rounds and $\mathrm{poly}(n)$ random bits can produce a secret key longer than the mutual information. In Section 4, we lift the restrictions regarding the number of rounds and random bits.

The light lower bound.[1] The setting is that Alice has the n-bit string x, and Bob has the n-bit string y. In addition, both Alice and Bob have the complexity profile of x and y. We construct a protocol that allows Alice and Bob to agree with probability $1-\epsilon$ on a shared secret key of length $I(x : y) - O(\log(n/\epsilon)$ and randomness deficiency $\delta(n) = O(\log(n/\epsilon))$. The protocol has 1 round, and actually the communication consists of a single message from Alice to Bob.

We start with an overview of the proof in a "nutshell." First, Alice chooses two *hash-functions* h_1 and h_2 from two families of hash functions which are publicly known. Alice sends h_1 and h_2 to Bob, and she also sends to Bob the hash value $h_1(x)$. Next, Bob attempts to reconstruct Alice's input: he searches over all x' such that $C(x' \mid y) \leq C(x \mid y)$ until he finds a candidate with the same hash value, $h_1(x') = h_1(x)$; in what follows he hopes that the x' he found is indeed Alice's string x. Next, Alice and Bob compute the value $h_2(x)$ and take it as the common secret key.

In this scheme the adversary gets to know the hash functions h_1, h_2 and the hash-value $h_1(x)$. For this to work, we want to guarantee that with high probability Bob finds the correct value of x (absence of collisions) and that the adversary gets very little information regarding the secret $h_2(x)$. The technical part of the proof consists in choosing appropriate families of hash functions, which guarantee the two required properties.

In this section we use a very simple hashing technique — we take h_1 and h_2 to be random linear mappings with suitable lengths of the output. In the next section we use Theorem 2.2, which provides a subtler implementation of the same idea, where the 'hashing' is based on extractors.

In what follows we present in more detail the easy version of the protocol, with linear hashing.

Step 1: Let $n_1 = C(x)$ and let $k = C(x \mid y)$. (In fact, Alice does not have k. But she can compute from the complexity profile x and y, a good approximation of k; this is a small nuisance which can be easily handled, and for the sake of simplicity we ignore it.) Alice chooses an $n_1 + \log(1/\epsilon)$-by-n random binary matrix H. Alice partitions H into two sub-matrices H_1 and H_2, where H_1 consists of the top $k + \log(1/\epsilon)$ rows of H, and H_2 consists of the bottom $n_1 - k$ rows of H. Alice calculates $q = H_1 x$, and sends q and the entire H to Bob. Let E_1 be the event that there exists $x' \neq x$ such that $C(x') \leq C(x)$ and $Hx' = Hx$ and E_2 be the event that there exists $x'' \neq x$ such that $C(x'' \mid y) \leq C(x \mid y)$ and $H_1 x'' = H_1 x$. Note that both E_1 and E_2 have probability at most ϵ. Consequently with probability at least $1 - 2\epsilon$, neither E_1 nor E_2 hold, an event that we assume henceforth. We also assume

$$C(x, y \mid H) \geq C(x, y) - O(\log(1/\epsilon)), \tag{4}$$

an event which holds with probability at least $1 - \epsilon$.

Step 2: Since x is the unique string such $H_1 x = q$ among strings that have complexity conditioned by y bounded by k, Bob reconstructs x from q and H. Next, both Alice and Bob compute the string $z = H_2 x$, which is the output of the protocol.

Let us now show that z has the desired properties. Note, first, that z has length $n_1 - k = C(x) - C(x \mid y) =^+ I(x : y)$. It remains to show that z is a shared secret key, i.e., that z is random conditioned by the transcript of the protocol, which consists of q and H. More precisely, we show that $C(z \mid q, H) \geq I(x : y) - O(\log(n/\epsilon)$. Since x can be computed from q, z and H,

$$C(x, y \mid H) \leq C(q, z, y \mid H) + O(1). \tag{5}$$

Next, by the chain rule,

$$\begin{aligned} C(q, z, y \mid H) &= C(q \mid H) + C(z \mid q, H) + C(y \mid q, z, H) + O(\log n) \\ &\leq C(x \mid y) + C(z \mid q, H) + C(y \mid x) + O(\log(n/\epsilon)). \end{aligned} \tag{6}$$

[1] The following proof is due to Bruno Bauwens (private communication, August 2017).

In the second line, we have taken into account that $C(q \mid H)$ is less than $|q| + O(1) = C(x \mid y) + \log(1/\epsilon) + O(1)$, and $C(y \mid q, z, H)$ is less than $C(y \mid x) + O(1)$ (because x can be computed from q, z, H). Combining equations (5) and (6), we obtain

$$\begin{aligned} C(z \mid q, H) &\geq C(x, y \mid H) - C(x \mid y) - C(y \mid x) - O(\log(n/\epsilon)) \\ &\geq C(x, y) - C(x \mid y) - C(y \mid x) - O(\log(n/\epsilon)) \quad \text{(using equation (4))} \\ &\geq I(x : y) - O(\log(n/\epsilon)). \end{aligned}$$

Remark 3. The defined protocol has communication complexity $O(n^2)$ (mostly due to the necessity to communicate the matrix H). The communication complexity of the protocol can be easily reduced to $O(n)$ if we take at random, not an arbitrary matrix H, but a random Toeplitz matrix of the same size. In the next section we describe a protocol with communication complexity $\min(C(x \mid y), C(y \mid x))$. (We show in Section 6 that no protocol with public randomness has smaller communication complexity, and we conjecture that the same holds for protocols with private randomness.)

The light upper bound. We show here that if k is a constant and the number of random bits is bounded by a polynomial in n, no k-round protocol can produce a shared secret key that is longer (up to logarithmic precision) than the mutual information of the inputs. More precisely, our claim is that if a shared secret key z is produced from inputs x_A, x_B by some k-round protocol with probability $1 - \epsilon$, then $C(z \mid t) \leq I(x_A : x_B) + O(k \log(n/\epsilon))$ with probability $1 - O(\epsilon)$, where t is the transcript of the protocol.

Simple warm-up: deterministic protocols without communication. Let us first consider the case $k = 0$, i.e., Alice and Bob do not interact, and also, for now, let us assume they use no randomness. Clearly, this model is very weak, and, in general, Alice and Bob cannot agree on any nontrivial common secret without communication. However, in some cases (for some very special inputs x_A and x_B) this might be possible. For instance, if x_A and x_B are n-bits strings of complexity n, and the first $n/2$ bits in these strings coincide (the i-th bit of x_A is equal to the i-th bit of x_B for $i = 1, \ldots, n/2$), then Alice and Bob can take the first $n/2$ bits of their inputs as the common key.

In this trivial example, Alice and Bob get a common key of complexity close to $n/2$, and the mutual information between x_A and x_B is also not less than $n/2$. We argue that the same relation remains true for all protocols without communication, for all pairs of inputs (x_A, x_B). That is, if Alice and Bob can extract from their inputs x_A and x_B the same string z without communication, then the complexity of this z is bounded by $I(x_A : x_B)$. This follows from a well-known inequality that is true for all x_A, x_B, z (see Lemma 7.8):

$$C(z) \leq^+ C(z \mid x_A) + C(z \mid x_B) + I(x_A : x_B). \tag{7}$$

Slightly more freedom: randomized protocols without communication. The same bound remains true if Alice and Bob use random strings r_A and r_B (still without communication). Indeed, using Lemma 7.7 (3), with probability $1 - O(\epsilon)$, we have

$$I(x_A, r_A : x_B, r_B) = I(x_A : x_B) \pm O(\log(n/\epsilon)). \tag{8}$$

Hence, we can apply the same argument to the pair $(\langle x_A, r_A \rangle, \langle x_B, r_B \rangle)$.

$$\begin{aligned} C(z) &\leq^+ C(z \mid x_A, r_A) + C(z \mid x_B, r_B) + I(x_A, r_A : x_B, r_B) \\ &=^+ C(z \mid x_A, r_A) + C(z \mid x_B, r_B) + I(x_A : x_B) \pm O(\log(n/\epsilon)) \end{aligned} \tag{9}$$

(the first inequality in (9) is always true, while the last equality is true with probability $1 - O(\epsilon)$.) In case when z is a computable function of (x_A, r_A) and a computable function of (x_B, r_B), we obtain from (9) that with probability $1 - O(\epsilon)$

$$C(z) \leq I(x_A : x_B) + O(\log(n/\epsilon)), \tag{10}$$

and we are done.

The general case: randomized protocols with $O(1)$-round communication. Let us move now to the case when the number of communication rounds is $k > 0$. As announced, we claim that the upper bound remains true: $C(z \mid t) \leq I(x_A : x_B) + O(\log(n/\epsilon))$. The proof for the "light" case in which k is constant proceeds as follows.

Inequality (7) has a relativized version,

$$C(z \mid t) \leq^+ C(z \mid x_A, r_A, t) + C(z \mid x_B, r_B, t) + I(x_A, r_A : x_B, r_B \mid t), \tag{11}$$

which is true for all x_A, x_B, r_A, r_B, z, t. When z is a computable function of (x_A, r_A, t) and a computable function of (x_B, r_B, t), we obtain

$$C(z \mid t) \leq^+ I(x_A, r_A : x_B, r_B \mid t). \tag{12}$$

Thus, to prove our claim, it remains to show that for the transcript t

$$I(x_A, r_A : x_B, r_B \mid t) \leq^+ I(x_A, r_A : x_B, r_B). \tag{13}$$

and then to use inequality (8).

Notice that, in general, the conditional mutual information $I(a : b \mid c)$ can be much larger than the plain mutual information $I(a : b)$ (e.g., if a and b are independent strings of length n and c is the bitwise XOR of a and c, then $I(a : b \mid c) =^+ n \gg I(a : b) =^+ 0$). So, we should explain why (13) is true for these specific strings. The required inequality can be deduced from the following lemma.

Lemma 3.1. *(a) Let f be a computable function of one argument. Then for all a, b*

$$I(a : b \mid f(a)) \leq^+ I(a : b). \tag{14}$$

(b) Further, let g be a computable function of two arguments. Then for all a, b, c

$$I(a : b \mid g(a, c), c) \leq^+ I(a : b \mid c). \tag{15}$$

Proof. Inequality (14) follows directly from the chain rule:

$$\begin{array}{lll} I(a : b) & =^+ & I(a, f(a) : b) \qquad\qquad\qquad\quad \text{(since } f(a) \text{ can be computed from } a) \\ & =^+ & I(f(a) : b) + I(a : b \mid f(a)) \quad \text{(the chain rule)} \\ & \geq^+ & I(a : b \mid f(a)) \qquad\qquad\quad \text{(non-negativity of the mutual information)} \end{array}$$

The proof of (15) is similar, we should only substitute c as a condition to each term $I(*)$ in the inference above. The lemma is proven. □

Now we proceed with a proof of (13). Recall from the definition of a protocol, that $t = (x_1, y_1, \ldots, x_k, y_k)$ and each x_i (respectively y_i) is deterministically computed from x_A, r_A (respectively from x_B, r_B) and the previous messages. We claim that

$$\begin{array}{ll} I(x_A, r_A : x_B, r_B) & \geq^+ I(x_A, r_A : x_B, r_B \mid x_1) \\ & \geq^+ I(x_A, r_A : x_B, r_B \mid x_1, y_1) \\ & \geq^+ I(x_A, r_A : x_B, r_B \mid x_1, y_1, x_2) \\ & \vdots \\ & \geq^+ I(x_A, r_A : x_B, r_B \mid x_1, y_1, x_2, y_2, \ldots, x_k, y_k). \end{array} \tag{16}$$

The first line is a special case of (14) since x_1 can be computed from (x_A, r_A). Every next line is a special case of (15). Indeed, on each step we add to the condition of the term $I(x_A, r_A : x_B, r_B \mid *)$ another string x_i or y_i, and every time the newly added string is a function of (x_A, r_A) and the pre-history of the communication or of (x_B, r_B) and the pre-history.

All the inequalities used in the argument are valid up to an additive term $O(\log C(x_A, r_A, x_B, r_B))$, which is at most $O(\log n)$ because r_A and r_B have length bounded by a polynomial in n. Since the number of rounds k (and therefore the chain of inequalities (16)) is constant, the claimed upper bound is proved.

To prove the same inequality for a non-constant k and an arbitrary number of random bits, we will need a different technique. We discuss it in Section 4.

4 Main results

We prove here our main results. We present a secret key agreement protocol which produces a shared secret key of length equal (up to logarithmic precision) to the mutual information of the inputs, provided the two parties know the complexity profile. Next, we show that no protocol can produce a longer shared secret key. The formal statements are as follows.

Theorem 4.1 (Lower bound)**.** *There exists a secret key agreement protocol with the following property: For every n-bit strings x and y, for every constant $\epsilon > 0$, if Alice's input x_A consists of x, the complexity profile of (x, y) and ϵ, and Bob's input x_B consists of y, the complexity profile of (x, y) and ϵ, then, with probability $1 - \epsilon$, the shared secret key is a string z such that, $C(z \mid t) \geq |z| - O(\log(1/\epsilon))$ and $|z| \geq I(x : y) - O(\log(n/\epsilon))$, where t is the transcript of the protocol.*

Moreover, the communication consists of a single message sent by Alice to Bob of length $C(x \mid y) + O(\log(n/\epsilon))$, Alice uses $O(\log(n/\epsilon))$ random bits, and Bob does not use any random bits.

Theorem 4.2 (Upper bound)**.** *Let us consider a protocol for secret key agreement, let x_A and x_B be input strings of length n on which the protocol succeeds with error probability ϵ and randomness deficiency $\delta(n)$, and let z be the random string that is the shared secret key output by the protocol, i.e., a string satisfying relations (2) and (3). Then with probability at least $1 - O(\epsilon)$, if n is sufficiently large, $|z| \leq I(x_A : x_B) + \delta(n) + O(\log(n/\epsilon))$, where the constants in the $O(\cdot)$ notation depend on the universal machine and the protocol, but not on x_A and x_B.*

Theorem 4.2 establishes the upper bound claimed in the Introduction. Indeed, for any pair of n-bit strings (x, y), suppose that Alice's input x_A consists of x and the complexity profile of (x, y) and Bob's input x_B consists of y and the complexity profile of (x, y). Note that $I(x_A : x_B) =^+ I(x : y)$ because the length of the complexity profile is bounded by $O(\log n)$. Hence, Theorem 4.2 implies that secret key agreement protocols in which the two parties, besides x and respectively y, are additionally given the complexity profile of their inputs, can not produce a secret key that is longer than $I(x : y) + O(\log n)$ (provided the randomness deficiency of the key satisfies $\delta(n) = O(\log n)$).

Remark 4. In Theorem 4.1 we construct a uniform communication protocol (see Remark 2 on p. 8), i.e., our protocol is described by one program that deals with all x and y (the complexity profile of x and y is provided as an auxiliary input). Theorem 4.2 is also stated for uniform protocols. In the uniform case we can ignore whether the adversary knows the protocol (this knowledge affects all quantities of Kolmogorov complexity by at most an $O(1)$ term).

Our negative result (Theorem 4.2) can be extended to non-uniform protocols. In the non-uniform setting the statement of Theorem 4.2 has to be adjusted: we should assume that the protocol is "known" to the adversary, and all complexity terms are "relativized" with respect to the protocol. Technically this means that we append the description of the protocol to the condition of all complexity quantities involved in the theorem and in the value of $C(z \mid t)$ in the definition of the randomness deficiency.

Proof of Theorem 4.1. We first give an overview of the protocol (an adaptation of the scheme "in a nutshell" explained on p. 9).

In Step 1, Alice computes $p_{x|y}$ a program for x given y of length $C(x \mid y) + O(\log n)$. She sends to Bob the string $p_{x|y}$ and also random strings r and s that both Alice and Bob will use in Step 2. The strings r and s have length $O(\log(n/\epsilon))$.

In Step 2, Bob uses $p_{x|y}$ and his string y, to construct x. Next, both Alice and Bob construct a $O(\log n)$-short program $\tilde{z}$ for x given $p_{x|y}$ using the random string r. Then, using s, they modify $\tilde{z}$ and obtain the common output z of the protocol.

Let us now see the details.

Step 1. Alice, using the complexity profile of x and y, first computes $k = C(x, y) - C(x) + 4 \log n$. Taking into account relation (1), if n is sufficiently large, $C(x \mid y) + 8 \log n \geq k \geq C(x \mid y)$. Next, using the algorithm from Theorem 2.2, Alice on input (x, k, ϵ) computes a string $p_{x|y}$ that, with probability $1 - \epsilon$, is a program for x given y of length $k + O(\log(n/\epsilon)) \leq C(x \mid y) + O(\log(n/\epsilon))$. In what follows, we assume that $p_{x|y}$ has this property (thus, we exclude an event that has probability at most ϵ).

Note that $C(p_{x|y}) \geq C(x \mid y)$ (because $p_{x|y}$ is a program for x given y), and $C(p_{x|y}) \leq |p_{x|y}| + O(1) \leq k + O(\log(n/\epsilon)) \leq C(x \mid y) + O(\log(n/\epsilon))$. Thus, $C(p_{x|y}) =^+ C(x \mid y)$.

Next,

$$\begin{aligned} C(x \mid p_{x|y}) &=^+ C(x, p_{x|y}) - C(p_{x|y}) \quad \text{(chain rule)} \\ &=^+ C(x) - C(x \mid y) \quad (p_{x|y} \text{ is computed from } x \text{ and } O(\log(n/\epsilon)) \text{ randomness}) \\ &=^+ I(x:y). \end{aligned} \tag{17}$$

Alice sends to Bob the string $p_{x|y}$ and also random strings r and s of length $O(\log(n/\epsilon))$ that will be used by both parties in Step 2.

Step 2. First, Bob reconstructs x from y and $p_{x|y}$.

Next, both Alice and, separately, Bob do the following. In the first phase, they use the random string r (from Step 1) and compute a string $\tilde{z}$, which almost has the desired properties, except that its randomness deficiency is a little too large. In a second phase, they use the random string s (also from Step 1), to finally obtain z with all the desired properties.

Phase 1. Using the complexity profile of x and y and based on relation (17), Alice and Bob compute $k' = I(x:y) + c' \log(n/\epsilon)$, for a constant c' such that $C(x|p_{x|y}) \leq k'$. Then, using the algorithm from Theorem 2.2 on input (x, k', ϵ), with randomness given by the random string r from Step 1, they compute a string $\tilde{z}$ of length $k' + O(\log(n/\epsilon))$. With probability (over r) at least $1 - O(\epsilon)$, $\tilde{z}$ is a program for x given $p_{x|y}$. In what follows, we assume that $\tilde{z}$ has this property (thus, we exclude another event that has probability at most $O(\epsilon)$).

Note that $|\tilde{z}| = k' + O(\log n) =^+ I(x:y)$ and $C(\tilde{z}) \leq |\tilde{z}| + O(1) \leq^+ I(x:y)$. Also

$$\begin{aligned} C(\tilde{z} \mid p_{x|y}) &\geq^+ C(x \mid p_{x|y}) && \text{(because } \tilde{z} \text{ is a program for } x \text{ given } p_{x|y}) \\ &=^+ I(x:y). && \text{(by using inequality (17))} \end{aligned}$$

Combining the last three relations, it follows that

$$|\tilde{z}| =^+ C(\tilde{z}) =^+ C(\tilde{z} \mid p_{x|y}) =^+ I(x:y).$$

The transcript of the protocol (without s which has not been used yet) is $t = (p_{x|y}, r)$, and since $|r| = O(\log(n/\epsilon))$, we get

$$C(\tilde{z} \mid t) \geq I(x:y) - O(\log(n/\epsilon)) \text{ and } |\tilde{z}| = I(x:y) - O(\log(n/\epsilon)). \tag{18}$$

Thus, we are almost done, except for the fact that the randomness deficiency of $\tilde{z}$ is $O(\log(n/\epsilon))$, which is slightly too large.

Phase 2. To reduce the randomness deficiency, we need to use a strong extractor. In what follows we use the standard definitions of extractors that can be found, e.g., in [Sha02]. Recall that a function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ϵ) extractor if for any set $B \subseteq \{0,1\}^n$ of size $|B| \geq 2^k$, and for any $A \subseteq \{0,1\}^m$,

$$\text{Prob}(E(U_B, U_{\{0,1\}^d}) \in A) \in \big(\frac{|A|}{M} - \epsilon, \frac{|A|}{M} + \epsilon\big),$$

where $U_B, U_{\{0,1\}^d}$ denote independent random variables uniformly distributed on B, and respectively $\{0,1\}^d$, and $M = 2^m$.

A function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ϵ) *strong* extractor if the function E_1 defined by $E_1(x, y) = (y, E(x, y))$ is a (k, ϵ) extractor. Using the probabilistic method, one can show that there exist strong extractors with $m = k - 2\log(1/\epsilon) - O(1)$ and $d = \log n + 2\log(1/\epsilon) + O(1)$ [Vad12, Theorem 6.17]. Alice and Bob compute by brute-force such a strong extractor E.

For some value of $k = I(x:y) - O(\log(n/\epsilon))$ (the exact value of k is specified in the proof of the next lemma), Alice and Bob use the strong (k, ϵ) extractor $E : \{0,1\}^{|\tilde{z}|} \times \{0,1\}^d \to \{0,1\}^m$ (with d and m as above), and the random string s from Step 1, and take $z = E(\tilde{z}, s)$. Intuitively, by the properties of a strong extractor, the output z is random even conditioned by the seed s, which is exactly what we need here. The following lemma, whose proof is given in Section 7.1, shows that indeed z has the desired properties.

Lemma 4.3. *Let E be the above strong extractor, and suppose $\tilde{z}$ satisfies inequalities* (18)*. If s is chosen uniformly at random in $\{0,1\}^d$, and $z = E(\tilde{z}, s)$, then with probability $1 - O(\epsilon)$, $C(z \mid t, s) \geq |z| - O(\log(1/\epsilon))$ and $|z| = I(x:y) - O(\log(n/\epsilon))$.*

This concludes the proof of the theorem. □

Remark 5. The proof of Theorem 4.1 can be adapted to the situation where Alice and Bob are not given the exact value of the complexity profile of (x, y) but only an approximation of this profile. Indeed, assume that Alice and Bob are given upper and lower bounds for each component of the complexity profile of (x, y) with precision $\leq \sigma$, for some integer σ. Then we can use the communication protocol from the proof of Theorem 4.1 where the length of $p_{x|y}$ is chosen in accordance with the *upper bound* on $C(x \mid y)$, and the length of the resulting key z is chosen in accordance with the *lower bound* on the value of $I(x : y)$. As a result, with probability $1 - O(\epsilon)$, Alice and Bob agree on a common secret z that is incompressible (i.e., $C(z \mid t) \geq |z| - O(\log(1/\epsilon))$ where t is the transcript of the protocol), and the length of z is greater than $I(x : y) - \sigma - O(\log(n/\epsilon))$.

Remark 6. In the protocol in the proof of Theorem 4.1, we use "digital fingerprints" of x computed by a combination of randomness extractors and universal hashes (a part of the construction of these fingerprints is hidden in the proof of Theorem 2.2). We apply "digital fingerprints" in two significantly different contexts: in Step 1, we use a fingerprint of x to identify this string given the partial information on x held by Bob, and in Step 2, we use another fingerprint to extract from x (almost) pure randomness, even conditional on the information on x held by the eavesdropper. In the first case, we need to extract from x all the information about this string that Bob misses (with possibly a minor overhead, which contains a part of information that Bob already knows). In the second case, we need to extract from x the information that the eavesdropper misses after observing Step 1 (possibly with a minor shortage, so that the randomness contained in x remains partially unrevealed). Both ideas have been used in other secret key agreement protocols. However, their technical implementation in the AIT framework requires specific constructions needed for this setting, such as, to give just one example, *prefix extractors*, which were first introduced explicitly in [RRV02], and in AIT in [MRS11].

Proof of Theorem 4.2. Similarly to the argument in the *light upper bound* of the previous section, we are going to combine inequalities (12) and (13). All we need to do is to establish inequality (13) for the case when the number of rounds k is non-constant. The challenge is that we cannot iterate k applications of Lemma 3.1. If k is not a constant, then the sum of k copies of $O(\log n)$ terms in (16) becomes too large. Thus, we need to handle the transcript t "in one piece," without splitting it in k messages. To this end we have to use instead of Lemma 3.1 some other tool.

We start with a reminder of the notion of a combinatorial rectangle (quite standard in communication complexity).

Definition 4.4. *A set $S \subset \{0,1\}^* \times \{0,1\}^*$ is a* combinatorial rectangle*, if it can be represented as a Cartesian product $S = A \times B$ for some $A, B \subset \{0,1\}^*$.*

For example, in Fig. 2 the black cells in the table form a combinatorial rectangle (the black cells occupy the intersection of columns a_1, a_2, a_3, a_4 and rows b_1, b_2, b_3). We use the notion of a combinatorial rectangle in the

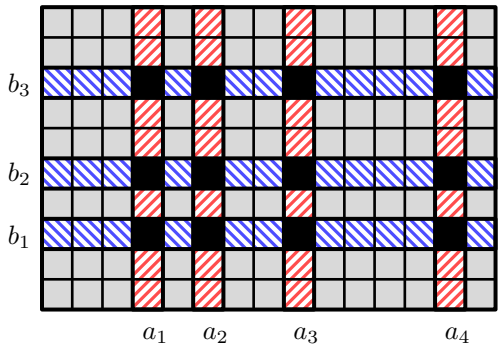

Figure 2: Combinatorial rectangles.

following (less conventional) definition.

Definition 4.5. *We say that a computable function $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is* transcript-like*, if for every t, the pre-image $f^{-1}(t)$ is a combinatorial rectangle.*

For example, the cells in the table in Fig. 2 are colored in four different colors (solid black, solid grey, and two types of hatching), and cells of each color form a combinatorial rectangle. Thus, the function that maps each cell to its color is transcript-like.

By definition, for every transcript-like function f, if $f(a,b') = f(a',b) = t$ for some a, a', b, b', then we have also $f(a,b) = t$. Notice that for every deterministic communication protocol with two parties (Alice and Bob) the mapping

$$f : [\text{input of Alice } x_A] \times [\text{input of Bob } x_B] \mapsto [\text{the transcript of the protocol}]$$

is transcript-like. A similar statement holds for randomized protocols with private source of randomness, we should only join the random bits given to Alice and Bob with their inputs : the mapping

$$f : \langle x_A, r_A\rangle \times \langle x_B, r_B\rangle \mapsto [\text{the transcript of the protocol}]$$

is also transcript-like.

Now we formulate a generalization of Lemma 3.1 (a), which can work with the transcript of the protocol 'in one piece'.

Lemma 4.6. *If $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is transcript-like computable function, then for all a, b*

$$I(a : b \mid f(a,b)) \leq^+ I(a : b). \tag{19}$$

Remark 7. If f depends only on a or only on b, then f is transcript-like. Thus, Lemma 3.1 (a) is a special case of Lemma 4.6.

Proof. Let us fix a pair (a,b) and denote $t := f(a,b)$. We say that a string a' is an A-*clone* of a (conditional on t), if

(i_A) there exists another string b' such that $f(a',b') = t$, and

(ii_A) $C(a') \leq C(a)$.

Similarly, we say that b' is a B-*clone* of b (conditional on t), if

(i_B) there exists another tuple a' such that $f(a',b') = t$, and

(ii_B) $C(b') \leq C(b)$.

Remark 8. The A- and B-clones are in some sense "similar" to a and b respectively. In what follows we show that the definition of clones is not too restrictive: there exist many clones of a and b.

Denote the sets of A-clones and B-clones by Clones_A and Clones_B respectively. By definition we have $a \in \mathrm{Clones}_A$ and $b \in \mathrm{Clones}_B$. Let us show that in fact there exist many other A- and B-clones.

Claim 4.7. *If complexities of a and b are bounded by n, then $|\mathrm{Clones}_A| \geq 2^{C(a|f(a,b))-O(\log n)}$ and $|\mathrm{Clones}_B| \geq 2^{C(b|f(a,b))-O(\log n)}$.*

Proof of the claim: Notice that for every computable function f the property

a string a' is a clone of some a of complexity k conditional on t

is a semi-decidable predicate of the triple $\langle a', t, k\rangle$. Therefore, given t and the value of $C(a)$, we can run the process of enumeration of all A-clones of a. Thus, to identify some specific A-clone a' of a given t, we need to know the complexity of a and the index of appearance of this specific clone in the enumeration of all A-clones. Hence, $C(a' \mid t)$ is not greater than $\log|\mathrm{Clones}_A| + O(\log n)$. In particular, this inequality is true for a itself. Similarly, $C(b \mid t)$ is not greater than $\log|\mathrm{Clones}_B| + O(\log n)$. Taking the exponent of both sides of these inequalities, we obtain the Claim.

Let us take a pair of "typical" clones $(a', b') \in \text{Clones}_A \times \text{Clones}_B$ that maximizes $C(a', b' \mid t)$. Due to the claim above, this maximum is

$$\begin{array}{rcl} C(a', b' \mid t) & =^+ & \log |\text{Clones}_A \times \text{Clones}_B| \\ & \geq^+ & C(a \mid t) + C(b \mid t). \end{array}$$

Thus, we have

$$C(a', b', t) =^+ C(t) + C(a', b' \mid t) \geq^+ C(t) + C(a \mid t) + C(b \mid t). \tag{20}$$

By definition of clones, a' and b' belong to the projection of the pre-image $f^{-1}(t)$. Since f is a transcript-like function, it follows that $f(a', b') = t$. That is, given a' and b' we can compute t, so $C(t \mid a', b') =^+ 0$. Hence, the left-hand side of (20) can be upper bounded as

$$C(a', b', t) \leq^+ C(a') + C(b') \leq^+ C(a) + C(b) \tag{21}$$

(in the last inequality we used again the definition of clones). Combining (20) and (21) we obtain

$$C(t) + C(a \mid t) + C(b \mid t) \leq^+ C(a) + C(b). \tag{22}$$

Given $C(t \mid a, b) =^+ 0$, inequality (22) rewrites to $I(a : b \mid t) \leq^+ I(a : b)$, and lemma is proven. □

Now we are ready to prove the theorem. Denote by π the function

$$\pi : \langle x'_A, r'_A \rangle \times \langle x'_B, r'_B \rangle \mapsto [\text{the transcript of the protocol}]$$

mapping the input data of Alice and Bob to the corresponding transcript of the communication protocol. Clearly, π is a transcript-like function (as the outcome of any communication protocol). By applying Lemma 4.6 to the function π we get inequality (13). This concludes to proof of the theorem. □

Remark 9. The proof of inequality (13) applies even to "invalid" input data x_A and x_B, when Alice and Bob are given false information on the complexity profile of the main input components x and y. (See Remark 1).

Remark 10. In some sense the proof of the weaker 'light upper bound' in section Section 3 is more robust than the proof of Theorem 4.2, since it applies to protocols with small advice obtained from a trustable all-powerful helper, see Section 8.5.

5 Secret key agreement for 3 or more parties

We analyze secret key agreement for 3 or more parties. All the results in this section are valid for any constant number $\ell \geq 3$ of parties. For notational convenience, we handle the case of $\ell = 3$ parties, but the proofs with straightforward adaptations are valid for any constant ℓ.

Thus, this time there are three parties, which we call Alice, Bob, and Charlie. Alice has a string x_A, Bob has a string x_B, and Charlie has a string x_C. They also have private random bits r_A, respectively r_B and r_C. They run a k-round protocol. In each of the k rounds, each party broadcasts a message to the other two parties, where the message is a string computed from the party's input string and private random bits, and the messages from the previous rounds. After the completion of the k rounds, each party computes a string. The requirement is that with probability at least $1 - \epsilon$, they compute the same string, and that this string is random conditioned by the transcript of the protocol.

Formally, a k-round 3-party protocol for secret key agreement uses three computable functions A, B, C, and runs as follows. The first party has as input an n-bit string x_A and uses private randomness r_A, the second party has as input an n-bit string x_B and uses private randomness r_B, and the third party has as input an n-bit string x_C and uses private randomness r_C. The protocol consists of the following calculations:

$$\begin{array}{lll} t_1 = A(x_A, r_A), & t_2 = B(x_B, r_B), & t_3 = C(x_C, r_C), \\ t_4 = A(x_A, r_A, t[1:3]), & t_5 = B(x_B, r_B, t[1:3]), & t_6 = C(x_C, r_C, t[1:3]), \\ \vdots & & \\ t_{3k-2} = A(x_A, r_A, t[1:3(k-1)]), & t_{3k-1} = B(x_B, r_B, t[1:3(k-1)]), & t_{3k} = C(x_C, r_C, t[1:3(k-1)]) \end{array}$$

Each row corresponds to one round and shows the messages that are broadcast in that round, and we use the notation $t[i:j]$ to denote the tuple of messages $(t_i, \dots, t_j)$. We also denote $t = t[1:3k]$, the entire transcript of the protocol. The protocol succeeds with probability error ϵ and randomness deficiency $\delta(n)$ on the 3-tuple input (x_A, x_B, x_C) if with probability $(1-\epsilon)$ over r_A, r_B, r_C,

$$A(x_A, r_A, t) = B(x_B, r_B, t) = C(x_C, r_C, t) \stackrel{def.}{=} z, \tag{23}$$

and

$$C(z \mid t) \geq |z| - \delta(n). \tag{24}$$

5.1 Omniscience: definitions and explicit expressions

We transfer to AIT the notion of *omniscience* introduced in IT (see the discussion in Section 1.1):

Definition 5.1. *(1) For each triple of strings (x_1, x_2, x_3), we denote by $S(x_1, x_2, x_3)$ the set of all triples of integers (n_1, n_2, n_3) that satisfy the following inequalities:*

$$\begin{array}{ccc} n_1 \geq C(x_1 \mid x_2, x_3), & n_2 \geq C(x_2 \mid x_1, x_3), & n_3 \geq C(x_3 \mid x_1, x_2), \\ n_1 + n_2 \geq C(x_1, x_2 \mid x_3), & n_1 + n_3 \geq C(x_1, x_3 \mid x_2), & n_2 + n_3 \geq C(x_2, x_3 \mid x_1). \end{array} \tag{25}$$

The constraints defining $S(x_1, x_2, x_3)$ will be referred to as the Slepian-Wolf *constraints.*

*(2) We define $\mathrm{CO}(x_1, x_2, x_3)$ to be the minimal value of $n_1 + n_2 + n_3$ subject to n_1, n_2, n_3 satisfying the Slepian-Wolf constraints. (*CO *stands for communication for omniscience.)*

This definition has a straightforward (though rather cumbersome) generalization for the tuples of size $\ell > 3$:

Definition 5.2. *(1) For each tuple of strings $(x_1, x_2, \dots, x_\ell)$, we denote by $S(x_1, x_2, \dots, x_\ell)$ the set of all tuples of integers $(n_1, n_2, \dots, n_\ell)$ that satisfy the following inequalities: for every splitting of the set of all indices into two disjoint nonempty sets, $\{1, \dots, \ell\} = \mathcal{I} \cup \mathcal{J}$, we have*

$$\sum_{i \in \mathcal{I}} n_i \quad \geq \quad C(x_{\mathcal{I}} \mid x_{\mathcal{J}}). \tag{26}$$

Similar to the case of $\ell = 3$, the constraints defining $S(x_1, x_2, \dots, x_\ell)$ are referred to as the Slepian-Wolf *constraints.*

(2) We define $\mathrm{CO}(x_1, x_2, \dots, x_\ell)$ to be the minimal value of $n_1 + \dots + n_\ell$ subject to $n_1, \dots, n_\ell$ satisfying the Slepian-Wolf constraints.

The value of $\mathrm{CO}(x_1, x_2, x_3)$ can be viewed as the solution of a problem of linear programming defined by constraints (25). This solution can be computed explicitly, as shown in the following proposition.

Proposition 5.3 (see Example 3 in [CN04])**.** *The value of $\mathrm{CO}(x_1, x_2, x_3)$ is equal to the maximum of the following four quantities*

$$\begin{array}{l} C(x_1 \mid x_2, x_3) + C(x_2, x_3 \mid x_1), \\ C(x_2 \mid x_1, x_3) + C(x_1, x_3 \mid x_2), \\ C(x_3 \mid x_1, x_2) + C(x_1, x_2 \mid x_3), \\ \frac{1}{2}\left[C(x_1, x_2 \mid x_3) + C(x_1, x_3 \mid x_2) + C(x_2, x_3 \mid x_1)\right]. \end{array}$$

Accordingly, the difference $C(x_1, x_2, x_3) - \mathrm{CO}(x_1, x_2, x_3)$ is equal to the minimum of the following four quantities:

$$\begin{array}{l} I(x_1 : x_2, x_3), \\ I(x_2 : x_1, x_3), \\ I(x_3 : x_1, x_2), \\ \frac{1}{2}\left[C(x_1) + C(x_2) + C(x_3) - C(x_1, x_2, x_3)\right]. \end{array} \tag{27}$$

This result can be extended to the case $\ell > 3$ as follows.

Proposition 5.4 ([CABE+15])**.** *For every tuples of strings $(x_1, \ldots, x_\ell)$ the difference $C(x_1, \ldots, x_\ell) - \mathrm{CO}(x_1, \ldots, x_\ell)$ is equal to the minimum of the quantities*

$$\tfrac{1}{s-1}\left[C(x_{\mathcal{J}_1}) + \ldots + C(x_{\mathcal{J}_s}) - C(x_1, \ldots, x_\ell)\right]. \tag{28}$$

over all splittings $\{1, \ldots, \ell\} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \ldots \cup \mathcal{J}_s$, where $\mathcal{J}_1, \ldots, \mathcal{J}_s$ are nonempty and disjoint.

Remark 11. It is easy to see that the minimum of four values (27) is a special case of the minimum of (28) (for $\ell = 3$).

In what follows, we show that there exists a protocol that on every input tuple (x_A, x_B, x_C) produces with high probability a secret key of length $C(x_A, x_B, x_C) - \mathrm{CO}(x_A, x_B, x_C) - O(\log n)$ (provided the parties have the complexity profile of the input tuple), and that no protocol can produce a secret key of length larger than $C(x_A, x_B, x_C) - \mathrm{CO}(x_A, x_B, x_C) + O(\log n)$. We prove the upper bound in Section 5.2 and the lower bound in Section 5.3.

5.2 Negative result for multi-party protocols

In this section we prove the upper bound on the length of the longest secret key that Alice, Bob, and Charlie can agree upon. A weak version of this bound (for the protocols with $O(1)$ rounds) can be obtained with the technique of classic information inequalities, very similar to the proof of an analogous statement in [CN04] in the setting of IT. We only need to "translate" the information inequalities for Shannon's entropy from the proof in [CN04] in homologous inequalities for Kolmogorov complexity, see Section 8.6. However, this technique does not work for protocols with non-constant number of rounds. So, to prove our next theorem we use a quite different method, employing the method of "clones" and constraint (non classic) information inequalities. This argument generalizes the proof of Theorem 4.2.

Theorem 5.5. *Let us consider a 3-party protocol for secret key agreement with error probability ϵ. Let (x_A, x_B, x_C) be a 3-tuple of n-bit strings on which the protocol succeeds. Let z be the random variable which represents the secret key computed from the input (x_A, x_B, x_C) and let t be the transcript of the protocol that produces z. Then, for sufficiently large n, with probability $1 - O(\epsilon)$,*

$$C(z \mid t) \le C(x_A, x_B, x_C) - \mathrm{CO}(x_A, x_B, x_C) + O(\log(n/\epsilon)),$$

where the constants in the $O(\cdot)$ notation depend on the universal machine and the protocol, but not on (x_A, x_B, x_C).

Proof. Assume that Alice, Bob, and Charlie agreed on a common key z. Denote by t the transcript of the communication protocol. We need to show that $C(z \mid t)$ is not greater than the minimum of four quantities (27).

Trivial case: no communication nor randomness. Similarly to the *light upper bound* in Section 3, we start with a naive question: on which z, can Alice, Bob, and Charlie agree without any communication? The upper bound from Section 3 applies to the case of three participants. Indeed, inequality (7) specializes to

$$C(z) \le^+ C(z \mid x_A, x_B) + C(z \mid x_C) + I(x_A, x_B : x_C).$$

Hence, if z can be computed by Alice, Bob, and Charlie given each of the strings x_A, x_B, x_C, then

$$C(z) \le^+ I(x_A, x_B : x_C). \tag{29}$$

Similarly, we have

$$C(z) \le^+ I(x_A, x_C : x_B) \tag{30}$$

and

$$C(z) \le^+ I(x_B, x_C : x_A). \tag{31}$$

In addition to the three mutual information quantities handled above, the maximum in (27) involves another quantity, which is specific for protocols with three participants. We treat it separately:

Lemma 5.6. *If z can be computed from each of the strings x_A, x_B, x_C, then*

$$C(z) \leq^+ \frac{1}{2}\left(C(x_A) + C(x_B) + C(x_C) - C(x_A, x_B, x_C)\right). \tag{32}$$

(See the proof in Section 7.)

The combination of (29), (30), (31), (32) implies that if a string z can be "extracted" from x_A, x_B, and x_C (without communication), then its complexity $C(z)$ is not greater than (27).

Randomized protocol without communication. We join x_A, x_B, and x_C with strings of random bits r_A, r_B, r_C, and repeat the same arguments for the "common information" z, which is extracted from the triple $\langle x_A, r_A\rangle$, $\langle x_B, r_B\rangle$, and $\langle x_C, r_C\rangle$. We obtain that $C(z)$ is not greater than the minimum of

$$\begin{array}{l} I(x_A, r_A : x_B, r_B, x_C, r_C), \\ I(x_B, r_B : x_A, r_A, x_C, r_A), \\ I(x_C, r_C : x_A, r_A, x_B, r_B), \\ \frac{1}{2}\left[C(x_A, r_A) + C(x_B, r_B) + C(x_C, r_C) - C(x_A, r_A, x_B, r_B, x_C, r_C)\right]. \end{array} \tag{33}$$

Then we notice that for randomly chosen r_A, r_B, and r_C, with probability $1 - \epsilon$ the difference between (33) and (27) is only $O(\log n/\epsilon)$.

The general case: protocols with any number of rounds. The argument from the case of *zero communication* examined above easily relativizes: if Alice, Bob, and Charlie agree (after some communication) on a common key z, and the transcript of the communication is t, then $C(z \mid t)$ is not greater than the minimum of

$$\begin{array}{l} I(x_A, r_A : x_B, r_B, x_C, r_C \mid t), \\ I(x_B, r_B : x_A, r_A, x_C, r_C \mid t), \\ I(x_C, r_C : x_A, r_A, x_B, r_B \mid t), \\ \frac{1}{2}\left[C(x_A, r_A \mid t) + C(x_B, r_B \mid t) + C(x_C, r_C \mid t) - C(x_A, r_A, x_B, r_B, x_C, r_C \mid t)\right]. \end{array} \tag{34}$$

Let us prove that (34) is not greater than (33).

From the proof of Theorem 4.2 we know that the mutual information between parties' data cannot increase when we relativize it with the transcript t. Thus, the first three quantities in (33) cannot become greater when we add to these terms the transcript t as a condition.

It remains to prove a similar inequality between the information quantity

$$J(x_A, x_B, x_C) := \frac{1}{2}\left[C(x_A) + C(x_B) + C(x_C) - C(x_A, x_B, x_C)\right]$$

and its relativized version

$$J(x_A, x_B, x_C \mid t) := \frac{1}{2}\left[C(x_A \mid t) + C(x_B \mid t) + C(x_C \mid t) - C(x_A, x_B, x_C \mid t)\right].$$

Notice that, in general (for an arbitrary t), the inequality $J(x_A, x_B, x_C \mid t) \leq^+ J(x_A, x_B, x_C)$ can be *false*. So we need to employ the fact that t is the transcript of a communication protocol. Here we reuse the technique from Theorem 4.2. We adapt the notion of a transcript-like function to the protocols with three parties.

Definition 5.7. *A set $S \subset \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$ is a* combinatorial parallelepiped*, if it can be represented as a Cartesian product $S = A \times B \times C$ for some $A, B, C \subset \{0,1\}^*$.*

Definition 5.8. *We say that a computable function $f : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is* transcript-like*, if for every t, the pre-image $f^{-1}(t)$ is a combinatorial parallelepiped.*

Similarly to the proof of Theorem 4.2, we will use the trick with "clones" of inputs. Let f be any transcript-like function of three arguments. We fix a triple of strings (a, b, c) (of length at most n) and denote $t := f(a, b, c)$. We say that a string a' is an A-*clone* of a (conditional on t), if

(i) there exist some strings b', c' such that $f(a', b', c') = t$, and

(ii) $C(a') \le C(a)$.

Denote the sets of A-clones by Clones_A. Similarly, we define the set of B-*clones* of b (denoted Clones_B) and C-*clones* of c (denoted Clones_C), conditional on the fixed value $t = f(a, b, c)$.

Claim 5.9. *Assuming that complexities of the strings a, b, and c are bounded by an integer n, we get $|\mathrm{Clones}_A| \ge 2^{C(a|f(a,b,c))-O(\log n)}$, $|\mathrm{Clones}_B| \ge 2^{C(b|f(a,b,c))-O(\log n)}$, and $|\mathrm{Clones}_C| \ge 2^{C(c|f(a,b,c))-O(\log n)}$.*

Proof. Similar to the proof of Claim 4.7. □

Lemma 5.10. *If $f : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is transcript-like function, then for all a, b, c we have $J(a, b, c \mid f(a, b, c)) \le J(a, b, c)$.*

Proof. Fix a, b, c and denote $t := f(a, b, c)$. We choose a triple

$$(a', b', c') \in \mathrm{Clones}_A \times \mathrm{Clones}_B \times \mathrm{Clones}_C$$

that maximizes $C(a', b', c' \mid t)$. Due to the Claim above we have

$$\begin{array}{rcl} C(a', b', c', t) & =^+ & C(t) + \log|\mathrm{Clones}_A \times \mathrm{Clones}_B \times \mathrm{Clones}_C| \\ & \ge^+ & C(t) + C(a \mid t) + C(b \mid t) + C(c \mid t). \end{array}$$

On the other hand, t can be deterministically computed as $f(a', b', c')$, so

$$C(a', b', c', t) \le^+ C(a') + C(b') + C(c') \le^+ C(a) + C(b) + C(c).$$

Combining the last two inequalities we obtain

$$C(t) + C(a \mid t) + C(b \mid t) + C(c \mid t) \le^+ C(a) + C(b) + C(c),$$

which rewrites (assuming $C(t \mid a, b, c) =^+ 0$) to

$$C(a \mid t) + C(b \mid t) + C(c \mid t) - C(a, b, c \mid t) \le^+ C(a) + C(b) + C(c) - C(a, b, c).$$

Hence, $J(a, b, c \mid t) \le J(a, b, c)$, and the lemma is proven. □

Now we conclude the proof of the theorem. Denote by π the function

$$\pi : \langle x'_A, r'_A\rangle \times \langle x'_B, r'_B\rangle \times \langle x'_C, r'_C\rangle \mapsto [\text{the transcript of the protocol}]$$

mapping the input data of Alice, Bob, and Charlie to the transcript of the communication protocol. This is a transcript-like function (as the outcome of any communication protocol with three parties in the model *input in the hand*). So we can apply Lemma 5.10 to this mapping π, and we are done. □

Remark 12. The inequality $J(a, b, c \mid t) \le^+ J(a, b, c)$ can be easily proven in case when t is a deterministic function of one of the arguments a, b or c. This observation (plus the idea to split the transcript t into separate messages) essentially gives the proof of Theorem 8.2 (for a constant number of rounds).

The case of $\ell > 3$ parties. Let us sketch the proof of a general version of Theorem 5.5 for any constant number of parties $\ell \geq 3$. To this end we should generalize the information quantities $J(a,b,c)$, $J(a,b,c \mid t)$ defined above. We need to consider the s-valent version of J for all $s = 2, \ldots, \ell$,

$$J(x_1, x_2, \ldots, x_s) := \frac{1}{s-1}\big[C(x_1) + \ldots + C(x_s) - C(x_1, \ldots, x_s)\big]$$

and

$$J(x_1, x_2, \ldots, x_s \mid t) := \frac{1}{s-1}\big[C(x_1 \mid t) + \ldots + C(x_s \mid t) - C(x_1, \ldots, x_s \mid t)\big].$$

(Observe that these quantities appear in (28).)

The notions of a transcript-like function can be extended to the functions with $\ell > 3$ arguments. The technique of "clones" permits to generalize Lemma 5.10 to this case. It follows that for all transcript-like function $f = f(x_1, x_2, \ldots, x_s)$ it holds

$$J(x_1, x_2, \ldots, x_s \mid f(x_1, x_2, \ldots, x_s)) \leq J(x_1, x_2, \ldots, x_s).$$

Combining this observation with Proposition 5.4 we obtain the generalization of Theorem 5.5 for any number of parties $\ell > 3$. We omit the details of this proof (which is a rather straightforward but cumbersome generalization of the proof of Theorem 5.5).

Remark 13. Similarly to the case of two party protocols, the upper bounds for multi-party protocols can be extended to the setting with non-uniform communication protocols (see Remarks 2, 4). The generalization of the proof is quite straightforward, though the statement of the theorem becomes more technical (we have to append a description of the protocol to the condition of all complexity terms involved in the theorem).

5.3 Positive results for multi-party protocols

In this section we prove the positive statement: we describe a 3-party protocol that on every input tuple (x_A, x_B, x_C) produces with high probability a secret key of length

$$C(x_A, x_B, x_C) - \mathrm{CO}(x_A, x_B, x_C) - O(\log n)$$

assuming that the parties are given the complexity profile of the input tuple.

Theorem 5.11. *There exists a 3-party protocol for secret key agreement with the following characteristics:*

(1) For every n, for every tuple (x_1, x_2, x_3) of n-bit strings, for every $\epsilon > 0$, if Alice's input x_A consists of x_1, the complexity profile of the tuple and ϵ, Bob's input x_B consists of x_2, the complexity profile of the tuple and ϵ, and Charlie' input x_C consists of x_3, the complexity profile of the tuple and ϵ, then, at the end, the three parties compute with probability $1 - O(\epsilon)$ a common string z such that

$$C(z \mid t) \geq |z| - O(\log(1/\epsilon)) \text{ and } |z| \geq C(x_1, x_2, x_3) - \mathrm{CO}(x_1, x_2, x_3) - O(\log(n/\epsilon)),$$

where t is the transcript of the protocol, and the constants in the $O(\cdot)$ notation depend only on the universal machine.

(2) The protocol has 1 round and each party uses $O(\log(n/\epsilon))$ random bits.

Proof. We first give an overview of the protocol.

In the first step, the three parties compute an optimal tuple (n_1, n_2, n_3) for the optimization problems that defines $\mathrm{CO}(x_1, x_2, x_3)$ (see Definition 5.1), and then Alice computes p_A of length $n_1 + O(\log n)$, Bob computes p_B of length $n_2 + O(\log n)$, and Charlie computes p_C of length $n_3 + O(\log n)$, such that with high probability (x_A, p_B, p_C) is a program for (x_1, x_2, x_3), (p_A, x_B, p_C) is a program for (x_1, x_2, x_3), and (p_A, p_B, x_C) is a program for (x_1, x_2, x_3). Next, Alice broadcasts p_A, Bob broadcasts p_B, and Charlie broadcasts p_C. Alice also broadcasts a random string r of length $O(\log n)$ bits, which will be used in the second step.

In the second step, each party first computes (x_1, x_2, x_3). Next, each party, using the randomness r, computes a string z that with high probability is a short program for (x_1, x_2, x_3) given (p_A, p_B, p_C). We will show that z has the desired properties.

We proceed with the details.

Step 1. Using the complexity profile of (x_1, x_2, x_3), each party computes a common tuple (n_1, n_2, n_3) which is optimal for the optimization problem that defines $\mathrm{CO}(x_1, x_2, x_3)$ (there may be more optimal tuples, but the parties agree on choosing the same tuple in some canonical way).

Next, Alice uses the encoding procedure E from Theorem 2.1 for $\ell = 2$, on input x_1, n_1, ϵ and obtains p_A. The length of p_A is bounded by $n_1 + O(\log(n/\epsilon))$. Similarly, Bob and Charlie obtain p_B and respectively p_C, from x_2, n_2, and respectively x_3, n_3. The parties broadcast p_A, p_B, p_C as indicated in the overview.

Step 2. Alice uses the decoding algorithm D from Theorem 2.1 for $\ell = 2$, on input x_1, p_B, p_C. Since n_2, n_3 satisfy the Slepian-Wolf constraints (i.e., $n_2 + n_3 \geq C(x_2, x_3 \mid x_1)$, $n_2 \geq C(x_2 \mid x_1, x_3)$, $n_3 \geq C(x_3 \mid x_1, x_2)$), it follows that with probability $1 - O(\epsilon)$, Alice can reconstruct x_2 and x_3, and since she has x_1, she obtains (x_1, x_2, x_3). The similar facts hold for Bob and Charlie. Therefore, with probability $1 - O(\epsilon)$, each party obtains the tuple (x_1, x_2, x_3), an event which henceforth we assume to hold.

For the next operation, the plan is to use the encoding procedure given in Theorem 2.2 and compute a short program for (x_1, x_2, x_3) given (p_A, p_B, p_C). For that, the parties need to have a tight upper bound for $C(x_1, x_2, x_3 \mid p_A, p_B, p_C)$. Such an upper bound is provided in the following claim, which will be proved later.

Claim 5.12. *There exists a constant d such that*

$$C(x_1, x_2, x_3 \mid p_A, p_B, p_C) \leq C(x_1, x_2, x_3) - \mathrm{CO}(x_1, x_2, x_3) + d \log n.$$

Now, using the upper bound provided by Claim 5.12, Alice (and also Bob, Charlie) use the encoding procedure from Theorem 2.2 and on input (x_1, x_2, x_3), upper bound $k = C(x_1, x_2, x_3) - \mathrm{CO}(x_1, x_2, x_3) + d \log n$ and randomness r, computes a string z, of length $|z| = k + c \log(n/\epsilon)$ for some constant c, that, with probability $1 - O(\epsilon)$ is a program for (x_1, x_2, x_3) given (p_A, p_B, p_C). The transcript of the protocol is $t = (p_A, p_B, p_C, r)$. We have

$$\begin{aligned}
C(z \mid t) &= C(z \mid p_A, p_B, p_C, r) =^+ C(z \mid p_A, p_B, p_C) \quad \text{(because } |r| = O(\log n)\text{)} \\
&\geq^+ C(x_1, x_2, x_3 \mid p_A, p_B, p_C) \quad \text{(because } z \text{ is a program for } (x_1, x_2, x_3) \text{ given } p_A, p_B, p_C\text{)} \\
&=^+ C(x_1, x_2, x_3) - C(p_A, p_B, p_C) \\
&\qquad \text{(because } p_A \text{ is computed from } x_1 \text{ and } O(\log n) \text{ bits, similarly for } p_B, p_C\text{)} \\
&\geq^+ C(x_1, x_2, x_3) - (n_1 + n_2 + n_3) \quad (\ |p_A| \leq^+ n_1, \text{ similarly for } p_B, p_C) \\
&= C(x_1, x_2, x_3) - \mathrm{CO}(x_1, x_2, x_3) \\
&\geq^+ |z|,
\end{aligned}$$

which shows that z is a secret key with randomness deficiency $O(\log(n/\epsilon))$. Using randomness extractors and an additional random string s broadcast by Alice in Step 1, similar to the proof of Theorem 4.1, the randomness deficiency of z can be reduced to $O(\log(1/\epsilon))$.

It remains to prove Claim 5.12.

Proof of Claim 5.12. Suppose that Alice, Bob, and Charlie know $s = C(x_1, x_2, x_3 \mid p_A, p_B, p_C)$. Since each of x_1, x_2 and x_3 are n-bits long, $C(x_1, x_2, x_3 \mid p_A, p_B, p_C) \leq 3n + O(1)$, and thus the length of s in binary representation is bounded by $\log n + O(1)$ bits. Suppose s is given as part of the input, i.e., Alice's input is (x_1, s) instead of just x_1, and similarly s is given to Bob and Charlie. Then Alice, Bob and Charlie can use the encoding procedure given in Theorem 2.2: From (x_1, x_2, x_3), the upper bound s and the randomness r, they compute a string z' that, with probability $1 - 1/n$ is a program for (x_1, x_2, x_3) given (p_A, p_B, p_C) and z' has length $|z'| \leq s + c \log n = C(x_1, x_2, x_3 \mid p_A, p_B, p_C) + c \log n$ for some constant c. We have

$$\begin{aligned}
C(z' \mid p_A, p_B, p_C) &\geq C(x_1, x_2, x_3 \mid p_A, p_B, p_C) \\
&\qquad \text{(because } z' \text{ is a program for } (x_1, x_2, x_3) \text{ given } (p_A, p_B, p_C) \\
&\geq |z'| - c \log n.
\end{aligned}$$

The transcript of the protocol (that uses s as part of the input) is $t = (p_A, p_B, p_C, r)$ and since the length of r is only $O(\log n)$, it follows that $C(z' \mid t) =^+ C(z' \mid p_A, p_B, p_C)$ and therefore $C(z' \mid t) \geq^+ |z'|$. It follows that z' is a secret key that Alice, Bob, and Charlie have obtained via the protocol with s added to the input. It follows from Theorem 5.5 that we get the upper bound $C(z' \mid t) \leq C(x_1, x_2, x_3, s) - \mathrm{CO}(x_1, x_2, x_3) + d' \log n$, for some constant d'. Since s is only $\log n + O(1)$ bits long, we can drop s at the cost of increasing the constant d'. The conclusion follows, because $C(x_1, x_2, x_3 \mid p_A, p_B, p_C) \leq C(z' \mid p_A, p_B, p_C) \leq C(z' \mid t)$. □

(End of proof of Claim 5.12.) □

6 Communication complexity of secret key agreement

It is of interest to find the communication complexity for the task of finding a shared secret key having the optimal length of $I(x : y)$. We solve this problem in the model of randomized protocols with *public random bits*, visible to Alice, Bob, and the adversary. This model is obtained by modifying slightly the definition from Section 2.2 (in which the random bits are private): we require that $r_A = r_B = r$ and we change equation (3) to $C(z \mid t, r) \geq |z| - \delta(n)$.

The protocol presented in the proof of Theorem 4.1 solves the task with communication $\min(C(x \mid y), C(y \mid x)) + O(\log n)$. As presented, that protocol uses private random bits, but it can be easily modified to work in the model with public randomness. Indeed, Alice and Bob use only $O(\log n)$ private random bits. So, if we make these bits known to the adversary, this will not affect much the secrecy of the common key z produced in the protocol. That is, we still have the inequality

$$C(z \mid [\text{everything known to the adversary}]) \geq |z| - O(\log n),$$

only the constant in the term $O(\log n)$ becomes bigger (compared to equation (18)). Thus, the assumption that all participants of the protocol have access to a public source of random bits, only makes the protocol simpler, and with decreased communication complexity by $O(\log n)$.

We argue that within the model with public randomness the communication complexity of this protocol is optimal, up to the $O(\log n)$ term. In what follows, we assume as usual that Alice is given a string x and Bob is given a string y, and both parties know the complexity profile of (x, y).

Theorem 6.1. *Let $\epsilon, \delta_1, \delta_2$ be arbitrary positive real constants. There is no secret key agreement protocol with public random bits such that for all inputs x and y,*

1. *the communication complexity of the protocol (the total number of all bits sent by Alice and Bob) is less than $(1 - \delta_1) \min\{C(x \mid y), C(y \mid x)\}$,*
2. *Alice and Bob agree with probability $> \epsilon$ on a common key z such that $C(z \mid t, r) > \delta_2 I(x : y)$, where r is the public randomness and $t = t(x, y, r)$ is the transcript of the protocol.*

Specifying the parameters. To prove the theorem, we find a 'hard' pair (x, y), on which any communication protocol with public randomness fails. More specifically, we will choose a 'hard' pair with the profile

$$C(x) =^+ 2n,\ C(y) =^+ 2n,\ C(x, y) =^+ 3n \tag{35}$$

(for large enough n); notice that (35) implies $I(x : y) =^+ n$. We will show that if the protocol satisfies condition (1) from Theorem 6.1, then, for the chosen 'hard' strings (x, y), condition (2) from this theorems is false.

Remark 14. The construction that we present in this section can be adapted to construct a 'hard' pair (x, y) with any given nontrivial complexity profiles (we only need that $C(x|y)$ and $C(y|x)$ are both not too small). We could make Theorem 6.1 stronger by replacing the constant values δ_i by $\delta_i = \lambda_i \log n/n$ with large enough constant coefficients $\lambda_i > 0$ $(i = 1, 2)$. We sacrifice the generality to simplify the notation.

Digression: common information. Before we prove Theorem 6.1, we remind some results on common information. We say (somewhat informally) that strings x and y share k bits of common information, if there exists a string w of complexity k such that w can be easily "extracted" from x and from y, i.e., $C(w \mid x) \approx 0$ and $C(w \mid y) \approx 0$. To make this statement formal, we must specify the precision of the equalities $C(w \mid x) \approx 0$ and $C(w \mid y) \approx 0$. In the next definition we introduce a suitable notation.

Definition 6.2. *We say that k bits of common information can be extracted from x and y with inaccuracies (k_A, k_B), if there exists a string w such that*

$$\begin{cases} C(w) & \geq & k, \\ C(w \mid x) & \leq & k_A, \\ C(w \mid y) & \leq & k_B. \end{cases}$$

Denote by $\mathrm{ComInf}(x, y)$ *the set of all triples (k, k_A, k_B) such that k bits of common information can be extracted from x and y with inaccuracies (k_A, k_B).*

It is know that $\mathrm{ComInf}(x, y)$ is not uniquely defined by the complexity profile of (x, y). That is, pairs (x, y) and (x', y') with very similar (or even identical) complexity profiles can have very different properties of "extractability" of the mutual information, see [CMR$^+$02]. The next theorem describes the minimal possible set of triples $\mathrm{ComInf}(x, y)$ for a pair with complexity profile (35).

Theorem 6.3. *[Muc98] For every $\delta > 0$ and for all large enough n there exists a pair of strings (x, y) such that $C(x) =^+ 2n$, $C(y) =^+ 2n$, $C(x, y) =^+ 3n$, and* no *triple (k, k_A, k_B) satisfying*

$$\begin{cases} k_A \leq (1 - \delta)n, \\ k_B \leq (1 - \delta)n, \\ k_A + k_B + \delta n \leq k \end{cases}$$

belongs to $\mathrm{ComInf}(x, y)$.

I. Razenshteyn proposed a more constructive version of Theorem 6.3. Though we cannot algorithmically construct a pair (x, y) satisfying Theorem 6.3 (there is obviously no short algorithmic description of any object of high Kolmogorov complexity), we can immerse such a pair in a large constructive set where the majority of elements have the desired property:

Theorem 6.4. *[Raz11] For every $\delta > 0$ there exists an algorithm that for all large enough n generates some list of pairs S_n such that for the majority of $(x, y) \in S_n$ $C(x) =^+ 2n$, $C(y) =^+ 2n$, $C(x, y) =^+ 3n$, and* no *triple (k, k_A, k_B) satisfying*

$$\begin{cases} k_A \leq (1 - \delta)n, \\ k_B \leq (1 - \delta)n, \\ k_A + k_B + \delta n \leq k \end{cases}$$

belongs to $\mathrm{ComInf}(x, y)$.

Remark 15. Technically, [Raz11] formulates Theorem 6.4 only in the symmetric setting $k_A = k_B$. However Razenshteyn's proof applies in the general case.

Remark 16. When we claim in Theorem 6.4 that an algorithm *generates* a set S_n, we mean that on the input n the algorithm prints the list of elements of this set and stops. It is important that the enumeration of S_n has a distinguishable completion.

The weak version of "constructiveness" that appear in Theorem 6.4 is called *stochasticity*, [She83]. An individual string x is called (α, β)-stochastic, if there exists an algorithm of complexity $< \alpha$ that prints the list of elements of a set S such that $x \in S$ and $C(x) \geq \log |S| - \beta$. That is, if α and β are small, then an (α, β)-stochastic string x is, so to say, a *typical* element in a *simple* set.

The notion of (α, β)-stochasticity can be easily adjusted to deal with pairs of strings. Thus, Theorem 6.4 asserts that there exist $(O(\log n), O(\log n))$-stochastic pairs with the worst possible extractability of the mutual information.

On the uselessness of random oracles. In computability theory, it is very common to study "relativized" computations, i.e., computations with an oracle. In the theory of Kolmogorov complexity, a natural version of relativization consists in adding an oracle (which can be a finite or infinite sequence of bits) to the "condition" part of all complexity terms. So, the property of extracting of k bits of common information from x and y with an oracle r can be formulated

as follows: there exists a string w such that $C(w \mid r) = k$, while $C(w \mid x, r) \approx 0$ and $C(w \mid y, r) \approx 0$. More generally, we define $\mathrm{ComInf}(x, y \mid r)$ as the set of all triples (k, k_A, k_B) such that

$$\begin{cases} C(w \mid r) & \geq & k, \\ C(w \mid x, r) & \leq & k_A, \\ C(w \mid y, r) & \leq & k_B. \end{cases}$$

If a string r is independent of (x, y), i.e., $I(x, y : r) \approx 0$, then it seems natural to conjecture that all reasonable properties of Kolmogorov complexity concerning (x, y) do not change if we relativize them with r. In particular, it seems plausible that the difference between the sets $\mathrm{ComInf}(x, y|r)$ and $\mathrm{ComInf}(x, y)$ must be negligible. Surprisingly, this conjecture remains unproven. However, a version of this conjecture for stochastic pairs is known to be true.

Theorem 6.5. *[MR10] If (x, y) is $(O(\log n), O(\log n))$-stochastic, then the set $\mathrm{ComInf}(x, y \mid r)$ belongs to an $O(\log n)$-neighborhood of $\mathrm{ComInf}(x, y)$ and vice-versa.*

Now we are ready to prove Theorem 6.1.

Proof of Theorem 6.1. We fix some $\epsilon, \delta_1, \delta_2 > 0$. Assume for the sake of the argument that in a protocol with communication complexity $< (1-\delta_1)n$, Alice and Bob agree with a probability greater than ϵ on a common secret key z such that $C(z \mid t, r) > \delta_2 n$. We show that this assumption leads to a contradiction. We will provide a counterexample, i.e., a pair (x, y) on which the protocol fails. First, we consider a degenerate case — a communication protocol without randomness.

(a) The case of deterministic protocols. At first, we focus on a very special case: we assume that the communication protocol is *deterministic* (Alice and Bob do not use the source of random bits). In this case $C(z \mid t, r) > \delta_2 n$ simplifies to $C(z \mid t) > \delta_2 n$.

Denote $w := \langle t, z \rangle$. We look at this w as sort of "common information" extracted from x and y. Let us estimate the parameters of this "shared part" of two strings.

- [a trivial calculation] $C(w \mid x) \leq^+ C(t) + C(z \mid t, x) \leq^+ (1 - \delta_1)n$,
- [a similar calculation] $C(w \mid y) \leq^+ C(t) + C(z \mid t, y) \leq^+ (1 - \delta_1)n$,
- [a less trivial but also straightforward calculation]

$$\begin{array}{rcll} C(w) & =^+ & C(w \mid x) + C(w \mid y) + I(x : y) - I(x : y \mid w) & \text{[Section 7, Lemma 7.9(a)]} \\ & =^+ & C(w \mid x) + C(w \mid y) + I(x : y) - I(x : y \mid t) + C(z \mid t) & \text{[Section 7, Lemma 7.9(b)]} \\ & \geq^+ & C(w \mid x) + C(w \mid y) + C(z \mid t) & \\ & \geq^+ & C(w \mid x) + C(w \mid y) + \delta_2 n. & \end{array}$$

For the transition from the second to the third line we used the inequality $I(x : y) - I(x : y \mid t) \geq^+ 0$, which can be false for arbitrary x, y, t. Fortunately, it is true if t is the transcript of a communication protocol with inputs x and y (Lemma 4.6).

We know from Theorem 6.3 that the three inequalities above cannot hold together, if (x, y) is a pair with a minimal $\mathrm{ComInf}(x, y)$. Thus, we get a contradiction.

(b) Randomized protocols. Now we want to extend the result from the previous paragraph to randomized protocol. We assume now that there exists a communication protocol with public random bits r such that for most r the protocol produces a transcript t of size $< (1-\delta_1)n$ that permits to extract from x and y a common key z such that $C(z \mid t, r) \geq \delta_2 n$.

Denote again $w := \langle t, z \rangle$. Then the triple $(C(w \mid r), C(w \mid x, r), C(w \mid y, r))$ belongs to $\mathrm{ComInf}(x, y \mid r)$. We can repeat the calculations from Case (a), adding the string r to the condition of all complexities:

- $C(w \mid x, r) \leq^+ (1 - \delta_1)n$,
- $C(w \mid y, r) \leq^+ (1 - \delta_1)n$,

- $C(w \mid r) \geq^+ C(w \mid x, r) + C(w \mid y, r) + \delta_2 n$

We may assume that r is independent of (x, y), i.e., $I(x, y : r) =^+ 0$. If (x, y) is a stochastic pair, then from Theorem 6.5 it follows that an almost the same triple of integers

$$(C(w \mid r) \pm O(\log n), C(w \mid x, r) \pm O(\log n), C(w \mid y, r) \pm O(\log n))$$

must belong to $\mathrm{ComInf}(x, y)$. Due to Theorem 6.4 this is false for some stochastic (x, y), and we get a contradiction. □

7 Several technical lemmas

7.1 Proof of Lemma 4.3

Let us fix a $\tilde{z}$ and t that satisfy the relations (18). We introduce some notation:

$$n' = |\tilde{z}|, \quad \tilde{t} = (t, n', \epsilon), \quad \alpha = C(\tilde{z} \mid \tilde{t}).$$

By the relations (18) and the fact that $n' \leq n$, we have, for some constant c,

$$C(\tilde{z} \mid \tilde{t}) \geq |\tilde{z}| - c \log(n/\epsilon).$$

Let $k = |\tilde{z}| - c \log(n/\epsilon)$. Thus, $C(\tilde{z} \mid \tilde{t}) \geq k$.

We use $E : \{0,1\}^{n'} \times \{0,1\}^d \to \{0,1\}^m$, a $(k - c_1, \epsilon)$- strong extractor (for some constant c_1 that will be specified later), having $m = k - c_1 - 2\log(1/\epsilon) - O(1)$ and $d = \log n' + 2\log(1/\epsilon) + O(1)$.

In the following lemmas, s is a seed of the above strong extractor E chosen uniformly at random and $z = E(\tilde{z}, s)$.

Lemma 7.1. *With probability* $1 - O(\epsilon)$ *over the choice of* s,

$$C(\tilde{z}, s \mid \tilde{t}, \alpha) \leq C(z, s \mid \tilde{t}) + C(\tilde{z} \mid \tilde{t}) - m + \log(1/\epsilon) + O(1).$$

Lemma 7.2. *With probability* $1 - O(\epsilon)$ *over the choice of* s,

$$C(\tilde{z}, s \mid \tilde{t}, \alpha) \geq C(\tilde{z} \mid \tilde{t}, \alpha) + |s| - O(\log(1/\epsilon)) - O(1).$$

Lemma 7.3. $C(\tilde{z} \mid \tilde{t}, \alpha) \geq C(\tilde{z} \mid \tilde{t}) - O(1)$.

Lemma 7.4. *For every* s, $C(z, s \mid \tilde{t}) \leq |s| + C(z \mid s, \tilde{t}) + O(1)$.

Combining the four lemmas, we obtain that, with probability $1 - O(\epsilon)$, $C(z \mid s, \tilde{t}) \geq m - O(\log(1/\epsilon)) - O(1)$. Since $|z| = m = I(x : y) - O(\log n) - O(\log(1/\epsilon))$ and taking into account that $\tilde{t}$ includes t, we obtain $C(z \mid t, s) \geq |z| - O(\log(1/\epsilon)) - O(1)$, and that the length of z is $I(x : y) - O(\log n) - O(1/\epsilon)$.

It remains to prove the lemmas.

Proof of Lemma 7.1. Recall that the function $E_1 : \{0,1\}^{n'} \times \{0,1\}^d \to \{0,1\}^d \times \{0,1\}^m$ defined by $E_1(w, s) = (s, E(w, s))$ is a $(k - c_1, \epsilon)$ extractor. We view E_1 also as a bipartite graph, where the set of left nodes is $\{0,1\}^{n'}$, the set of right nodes is $\{0,1\}^d \times \{0,1\}^m$, and $(w, (s, z))$ is an edge if $E(w, s) = z$ (or, equivalently, $E_1(w, s) = (s, z)$). Let

$$B = \{w \in \{0,1\}^{n'} \mid C(w \mid \tilde{t}) \leq C(\tilde{z} \mid \tilde{t}\}.$$

We say that a right node (s, z) in the above graph is *heavy*, if its B-degree is $\geq (1/\epsilon)\frac{|B|}{M}$, where $M = 2^m$ (the B-degree of a node is the number of edges coming from B into the node). Let HEAVY be the set of nodes that are heavy. By counting from the left and also from the right the edges that go out from B, we get

$$|\mathrm{HEAVY}| \cdot (1/\epsilon) \cdot \frac{|\mathrm{B}|}{\mathrm{M}} \leq |\mathrm{B}| \cdot \mathrm{D},$$

and therefore,

$$\frac{|\mathrm{HEAVY}|}{M \cdot D} \leq \epsilon.$$

A left node $w \in B$ is *poor* if for more than 2ϵ fraction of $s \in \{0,1\}^d$, $E_1(w,s)$ is heavy.

Claim 7.5. *The number of poor nodes is less than* 2^{k-c_1}.

Proof. Let POOR be the set of poor nodes. Since

$$\begin{aligned}\mathrm{Prob}(E_1(U_{\mathrm{POOR}}, U_{\{0,1\}^d}) \in \mathrm{HEAVY}) &> 2\epsilon \\ &\geq \tfrac{|\mathrm{HEAVY}|}{M \cdot D} + \epsilon,\end{aligned}$$

the number of poor nodes has to be less than 2^{k-c_1} because otherwise the extractor property of E_1 would be contradicted. □

Claim 7.6. *$\tilde{z}$ is not poor.*

Proof. Suppose $\tilde{z}$ is poor. Then $C(\tilde{z} \mid \tilde{t}, \alpha) \leq k - c_1$, because, given $\tilde{t}, \alpha$, the string tz is described by its index in a canonical enumeration of B and the size of B is bounded by 2^{k-c_1} (by Claim 7.5). It is known that for any z', t', $C(z' \mid t', C(z' \mid t')) \geq C(z' \mid t') - O(1)$ (this is the relativized version of the known inequality $C(z' \mid C(z')) \geq C(z') - O(1)$, see [SUV17, Exercise 44]). It follows that

$$C(\tilde{z} \mid \tilde{t}) \leq C(\tilde{z} \mid \tilde{t}, \alpha) + O(1) \leq k - c_1 + O(1). \tag{36}$$

On the other hand, $C(\tilde{z} \mid \tilde{t}) \geq k$. If c_1 is large enough, we get a contradiction. □

Claim 7.6 says that for a fraction of $1 - 2\epsilon$ of $s \in \{0,1\}^d$, the node (s,p) is non-heavy, where $z = E(\tilde{z}, s)$. If (s,z) is non-heavy, then the number of its left neighbors in B is bounded by

$$(1/\epsilon)\frac{|B|}{M} \leq (1/\epsilon)\frac{2^{C(\tilde{z}|\tilde{t})+1}}{M} = 2^{C(\tilde{z}|\tilde{t})-m+\log(1/\epsilon)+1}.$$

It follows that, given $\tilde{t}, \alpha$, $\tilde{z}$ can be described by a non-heavy (s,z) and its index in an enumeration of the left nodes of (s,z) in B. Thus, with probability $1 - 2\epsilon$ over the choice s,

$$C(\tilde{z}, s \mid \tilde{t}, \alpha) \leq C(z, s \mid \tilde{t}) + (C(\tilde{z} \mid \tilde{t}) - m + \log(1/\epsilon) + 1) + O(1).$$

This concludes the proof of the lemma. □

Proof of Lemma 7.2. Let $w = C(\tilde{z} \mid \tilde{t}, \alpha) + |s| - C(\tilde{z}, s \mid \tilde{t}, \alpha)$. Note that w is a random variable because it depends on s. We need to show that $w = O(\log(1/\epsilon))$, with probability $1 - \epsilon$.

By the standard counting argument, we have that with probability $1 - \epsilon$,

$$C(s \mid \tilde{t}, \alpha) \geq |s| - \log(1/\epsilon). \tag{37}$$

Let us fix an s that satisfies inequality (37). We show that the w that corresponds to this s satisfies $w = O(\log(1/\epsilon))$, from which the conclusion follows.

If $w \leq 0$, we are done. So let us suppose that $w > 0$.

Let $u = C(\tilde{z}, s \mid \tilde{t}, \alpha) = C(\tilde{z} \mid \tilde{t}, \alpha) + |s| - w = (\alpha - O(1)) + |s| - w$ (the last equality holds because $C(\tilde{z} \mid \tilde{t}, \alpha) = C(\tilde{z} \mid \tilde{t}) - O(1)$, by the first part of inequality (36)).

Let $A = \{(z', s') \mid C(z', s' \mid \tilde{t}, \alpha) \leq u\}$. Note that $|A| \leq 2^{u+1}$. For every z', we define $A_{z'} = \{s' \mid (z', s') \in A\}$. Let e be the integer for which $2^{e-1} < |A_{\tilde{z}}| \leq 2^e$. Since $s \in A_{\tilde{z}}$ and $A_{\tilde{z}}$ can be enumerated given $w, \tilde{t}, \alpha$ and $O(1)$ bits, it follows that

$$C(s \mid \tilde{t}, \alpha) \leq e + 2\log w + O(1). \tag{38}$$

Let $H = \{z' \mid |A_{\tilde{z}}| > 2^{e-1}\}$. Note that $|H| \leq |A|/2^{e-1} \leq 2^{u-e+2}$ and that $\tilde{z}$ is in H. We can write the index of $\tilde{z}$ in an enumeration of H on exactly $u - e + 2$ bits and taking into account that u can be computed from $\tilde{z}, \alpha, w$ and $O(1)$ bits, it follows that e can be derived as well from the index of $\tilde{z}$ and the information mentioned above. In this way, we have all the information needed to enumerate H and we obtain

$$C(\tilde{z} \mid \tilde{t}, \alpha) \leq u - e + 2 + 2\log w + O(1). \tag{39}$$

Adding inequalities (38) and (39), we get

$$\begin{aligned} C(\tilde{z} \mid \tilde{t}, \alpha) + C(s \mid \tilde{t}, \alpha) &\leq u + 4\log w + O(1) \\ &\leq (C(\tilde{z} \mid \tilde{t}, \alpha) + |s| - w) + 4\log w + O(1). \end{aligned}$$

Taking into account inequality (37), we obtain $w - 4\log w \leq \log(1/\epsilon) + O(1)$, which implies $w = O(\log(1/\epsilon))$, as desired. □

Proof of Lemma 7.3. This is the first part in the chain of inequalities (36). □

Proof of Lemma 7.4. This follows from the fact that given $\tilde{t}$ (which contains n') we can find the length of s, and therefore we do not need delimiters to concatenate s and a description of z given s and $\tilde{t}$. □

7.2 Useful information inequalities

Lemma 7.7. *Let f be a computable function, and suppose that for every string x, we choose a string r uniformly at random among the strings of length $f(|x|)$. Then,*

(1) with probability $1 - \epsilon$,

$$\big|C(x, r) - (C(x) + C(r \mid x))\big| \leq O(\log(|x|/\epsilon)),$$

(2) with probability $1 - \epsilon$,

$$\big|C(x, r) - (C(r) + C(x \mid r))\big| \leq O(\log(|x|/\epsilon))$$

and

(3) with probability $1 - \epsilon$,

$$C(x, r) = C(x) + |r| \pm O(\log(|x|/\epsilon)).$$

Proof. Note that part (3) follows, after a rescaling of ϵ, from part (1) and the fact that with probability $1 - \epsilon$, $C(r \mid x) \geq |r| - \log(1/\epsilon)$.

We prove (1). The proof of (2) is similar.

Clearly, $C(x, r) \leq C(x) + C(r \mid x) + 2\log|x| + O(1)$ by using the standard way of appending two strings in a self-delimiting manner.

For the other inequality, we tweak the proof of the chain rule. We fix x and r and let $t = C(x, r) - |r| + \log(1/\epsilon)$. By the standard counting argument, with probability $1 - \epsilon$, t is positive, which we assume from now on. Since $C(x, r) \leq |x| + |r| + 2\log|x| + O(1)$ and $C(x, r) = t + |r| - \log(1/\epsilon)$, we get $t \leq |x| + 2\log|x| + \log(1/\epsilon) + O(1)$, and from here $\log t \leq O(\log(|x|/\epsilon))$, a fact which we will use later.

Now, let $A = \{(x', r') \mid C(x', r') \leq t + |r|\}$. For each x', let $A_{x'} = \{r' \mid (x', r') \in A\}$. Let e be such that $2^{e-1} < |A_x| \leq 2^e$. Note that by taking into account that $|r| = f(|x|)$,

$$C(r \mid x) \leq e + 2\log t + O(1). \tag{40}$$

Next, let $H = \{x' \mid A_{x'} > 2^{e-1}\}$. We have $|H| \leq |A|/2^{e-1} \leq 2^{t+|r|+1}/2^{e-1} = 2^{t+|r|-e+2}$. Note that x can be described by its rank in an enumeration of H, and for the enumeration of H we need t and e. By writing the rank of x on exactly $t + |r| - e + 2$ bits, and keeping in mind that $|r|$ can be computed from $|x|$, we obtain

$$C(x) \leq t + |r| - e + 2 + 2\log t + 2\log|x| + O(1). \tag{41}$$

By adding equations (40) and (41), we obtain $C(x) + C(r \mid x) \leq t + |r| + 4\log t + 2\log|x| + O(1)$. Therefore,

$$\begin{aligned} C(x,r) &= t + |r| - \log(1/\epsilon) \\ &\geq C(x) + C(r \mid x) - 4\log t - 2\log|x| - \log(1/\epsilon) - O(1) \\ &\geq C(x) + C(r \mid x) - O(\log(|x|/\epsilon)). \end{aligned}$$

□

Lemma 7.8. *For all x_A, x_B, z, $C(z) \leq^+ C(z \mid x_A) + C(z \mid x_B) + I(x_A : x_B)$.*

Proof. For all x_A, x_B, z we have

$$C(x_A, x_B \mid z) \leq^+ C(x_A \mid z) + C(x_B \mid z).$$

We add $2C(z)$ to both sides of this inequality and get

$$C(z) + C(x_A, x_B, z) \leq^+ C(x_A, z) + C(x_B, z).$$

Now we make it slightly weaker,

$$C(z) + C(x_A, x_B) \leq^+ C(x_A, z) + C(x_B, z).$$

It follows

$$C(z) \leq^+ C(z \mid x_A) + C(z \mid x_B) + C(x_A) + C(x_B) - C(x_A, x_B),$$

which is equivalent to (7), ending the proof. □

Lemma 7.9. *(a) For all strings w, x, y of length $O(n)$*

$$C(w \mid x) + C(w \mid y) + I(x : y) - I(x : y \mid w) - C(w \mid x, y) =^+ C(w)$$

(b) For all strings x, y, z, t of length $O(n)$, if $C(z \mid x, t) =^+ C(z \mid y, t) =^+ 0$, then

$$I(x : y \mid z, t) =^+ I(x : y \mid t) - C(z \mid t)$$

Proof. (a) A straightforward calculation gives

$$\begin{aligned} &C(w \mid x) + C(w \mid y) + I(x : y) - I(x : y \mid w) - C(w \mid x, y) \\ &=^+ C(x,w) - C(x) + C(y,w) - C(y) + C(x) + C(y) - C(x,y) \\ &\quad - C(x,w) - C(y,w) + C(x,y,w) + C(w) - C(x,y,w) + C(x,y) = C(w) \end{aligned}$$

(b) We have

$$\begin{aligned} I(x : y \mid z, t) &=^+ C(x,z,t) + C(y,z,t) - C(x,y,z,t) - C(z,t) && \\ &=^+ C(x,t) + C(y,t) - C(x,y,t) - C(z,t) && \text{since } C(z \mid x,t) =^+ C(z \mid y,t) =^+ 0 \\ &=^+ C(x,t) + C(y,t) - C(x,y,t) - C(t) - C(z \mid t) && \text{the chain rule} \\ &=^+ I(x : y \mid t) - C(z \mid t) && \end{aligned}$$

□

7.3 Proof of Lemma 5.6

For all x_A, x_B, x_C, z we have

$$C(x_A, x_B, x_C \mid z) \leq^+ C(x_A \mid z) + C(x_B \mid z) + C(x_C \mid z),$$

which rewrites to

$$2C(z) + C(x_A, x_B, x_C, z) \leq^+ C(x_A, z) + C(x_B, z) + C(x_C, z).$$

Now we use the assumption that z is a deterministic function of x_A, of x_B, and of x_C :

$$2C(z) \leq^+ C(x_A) + C(x_B) + C(x_C) - C(x_A, x_B, x_C),$$

and the lemma is proven.

8 Final comments

We make several considerations regarding secret key agreement protocols. They are not required for an understanding of the main results. The next subsections can be read independently.

8.1 Time-efficient secret key agreement protocols.

The secret key agreement protocol in the proof of Theorem 4.1 is computable but highly non-efficient. The only slow stage of the protocol is Step 2, where Bob reconstructs x given his input string y and the fingerprint $p_{x|y}$ obtained from Alice. At this stage, Bob has to simulate all programs of size $C(x \mid y)$ until he obtains a string matching the fingerprint $p_{x|y}$. All the other stages of the protocol can be implemented in polynomial time (to this end we need to use an effective version of an extractor in the definition of fingerprints; this increases the overhead in communication complexity from $O(\log n)$ to $\mathrm{poly}(\log n)$, which is still negligible comparative to the size of $p_{x|y}$; for details see [Zim17]).

We cannot make Bob's computation effective in general, but we can do it for some specific pairs of inputs (x, y). Actually, we can make the entire communication protocol fast, if there is a way to communicate x from Alice to Bob so that

- communication complexity of this stage (and therefore the information revealed to the adversary) remains about $C(x \mid y)$,
- all computations are performed by Alice and Bob in time $\mathrm{poly}(n)$.

Example 1. (Discussed in Introduction, p. 1.) Let Alice get a random line x in the affine plane over the finite field with 2^n elements, and Bob get a random point y on this line. For *most* inputs of this type we have $C(x \mid y) = n \pm O(\log n)$, and there exists a simple way to transfer x from Alice to Bob with communication complexity n (Alice just sends to Bob the slope of her affine line, and Bob draws a line with this slope incident to his point). Thus, we see once again that for this simple example there exists an effective (polynomial-time) communication protocol to agree on common secret key of size $\approx n$ bits.

Example 2. Let Alice and Bob get n-bits strings x and y respectively, and the Hamming distance between these strings is at most δn for some constant $\delta < 1/2$. For *most* inputs of this type we have

$$C(x \mid y) = h(\delta)n \pm O(\log n),\ I(x : y) = (1 - h(\delta))n \pm O(\log n),$$

where $h(\delta) = \delta \log \frac{1}{\delta} + (1-\delta) \log \frac{1}{1-\delta}$. Can we transfer x from Alice to Bob with communication complexity $h(\delta)n + o(n)$? It turns out that such a protocol exists; moreover, there exists a communication protocol with asymptotically optimal communication complexity and polynomial time computations, see [Smi07, GS10, GS16]. Plugging this protocol in our proof of Theorem 4.2 we conclude that on most pairs of inputs (x, y) of this type Alice and Bob can agree on a common secret key of size $(1 - h(\delta))n - o(n)$, with poly-time computations for both parties.

8.2 On pseudo-deterministic protocols for key agreement

The communication protocols that we have analyzed (defined in Section 2.2) are randomized. In particular, this means that, for the same pair of inputs x_A, x_B, Alice and Bob can end up with different values of the common secret key z, since the result depends on the randomness used in the protocol. It seems that for all practical reasons, running the same communication protocol twice on the same pair of inputs (but with different realizations of random bits) is useless and even harmful: in two independent conversations, Alice and Bob would divulge too much information about x and y. However, there remains a natural theoretical question: can we adjust the communication protocol so that for most values of random bits, Alice and Bob agree on one and the same value of the common random key z? Such a protocol could be called *pseudo-deterministic* or *Bellagio* protocol, similarly to the pseudo-deterministic algorithms introduced in [Gol12].

In what follows we show that in some sense the answer to this question is positive. More precisely, we can prove a version of Theorem 4.1 with only slightly weaker property of secrecy (we divulge $O(\log n)$ bits of information regarding the resulting secret key) with a protocol where Alice and Bob with high probability agree on one "canonical" value of z.

Theorem 8.1 (Lower bound with a unique key)**.** *There exists a secret key agreement protocol with the following property: For every n-bit strings x and y, for every constant $\epsilon > 0$, if Alice's input x_A consists of x, the complexity profile of (x, y) and ϵ, and Bob's input x_B consists of y, the complexity profile of (x, y) and ϵ, then, with probability $1 - \epsilon$, the shared secret key is a string z such that, $C(z \mid t) \geq |z| - O(\log(n/\epsilon))$ and $|z| \geq I(x : y) - O(\log(n/\epsilon))$, where t is the transcript of the protocol.*

Moreover, with probability $1 - \epsilon$, Alice and Bob agree on one "canonical" value of $z = z(x, y)$.

Sketch of the proof. Let us start with the simplified scheme "in a nutshell" explained on p. 9. In this protocol, we use two randomized "digital fingerprints" of x denoted $h_1(x)$ (which is sent by Alice to Bob) and $h_2(x)$ (which is taken as the common secret key). In the argument on p. 9 both these digital fingerprints were computed as random linear mappings of x. Observe that the scheme works fine if the second digital fingerprint is computed in a different way, without randomization. For example, we may assume that Alice and Bob search for the first stopping program of length $C(x)$ that prints x, and then take the first $C(x) - C(x \mid y)$ bits of this program as the secret key.

Clearly, in every execution of the new protocol, Alice and Bob agree on one and the same value of z assuming that, given y and $h_1(x)$ (and the complexity profile of (x, y)), Bob reconstructs correctly Alice's input x, which is true with probability close to 1. Moreover, it can be shown that with high probability the value of $h_1(x)$ contains only $O(\log(n/\epsilon))$ bits of information regarding this "canonical" value of z, so the secrecy condition is preserved.

The same idea can be used to adjust the proof of Theorem 4.1, where the value of h_1 is computed in a subtler way (with randomness extractors involved in the proof of Theorem 2.2). Roughly speaking, the random "fingerprint" of x computed in the proof of Theorem 2.2 with high probability has only negligibly small mutual information with the fixed in advance "canonical" value of z. We omit the details. □

We stress that the "secret key uniqueness" achieved in Theorem 8.1 is a property of a specific protocol, and this property does not follow automatically from the definition of shared secret key agreement. Moreover, we can construct a communication protocol so that on each new application of the protocol, Alice and Bob agree on a random value of the secret key z that is *uniformly distributed* on the set of $\{0, 1\}^k$ with $k = I(x : y)$, and, therefore, the instances of the key z obtained in different executions of the protocol are with high probability independent with each other. (The protocol still has a positive error probability, which means that with a small probability $\epsilon > 0$ Alice and Bob do not agree on a common key and accomplish the communication with different values of z.) To achieve this property, we should modify the protocol from Theorem 4.1 as follows. First of all, Alice and Bob should agree on some common key z using the old version of the protocol (and this z may be distributed non-uniformly). Then, Alice choses a random bit string w of length $|z|$ and openly sends it to Bob. At last, Alice and Bob XOR the bits of z with the bits of w. The resulting string z' is a uniformly distributed common secret key.

In the arguments above we used specially designed protocols. In the "standard" communication protocol defined in the proof of Theorem 4.1 the key z is not uniquely defined by x, y, and it is not uniformly distributed either. The specific attributes of the shared key produced by a protocol are determined by the details of the construction and by the ratio between $C(x)$ and $I(x : y)$, and are positioned between the two extreme cases that we have presented above. A more detailed analysis of generic communication protocols (e.g., a more precise estimation of $I(z_1 : z_2)$ for two keys z_1 and z_2 obtained in two independent realizations of the protocol from the proof of Theorem 4.1) requires subtler considerations, and we do not discuss this question here.

8.3 Secret key agreement with an additional private communication channel

In this section, we discuss a slightly more general model of communication that combines a public and private channels. We assume that the *private* communication channel is not visible to the adversary. There is a naive and straightforward way to gain from such a channel: Alice tosses her private random coin, sends the obtained bits to Bob via this private channel, and afterwards Alice and Bob can use these bits as a common secret key. This naive idea is actually the optimal usage of the private channel. There is no better way to use the secret communication: if Alice and Bob are given inputs x_A and x_B, and they send to each other s bits of information via the private communication, they cannot agree (with probability $1 - \epsilon$) on a common secret key of complexity greater than $I(x_A : x_B) + s + O(\log(n/\epsilon))$. Indeed, denote by t_p the transcript of the communications via the private channel. Then the common secret z can be

computed from (x_A, r_A, t_p) as well as from (x_B, r_A, t_p). It follows that

$$\begin{array}{rcl} C(z) & \leq^+ & C(z \mid x_A, r_A, t_p) + C(z \mid x_B, r_B, t_p) + I(x_A, r_A, t_p : x_B, r_B, t_p) \\ & \leq^+ & I(x_A, r_A, t_p : x_B, r_B, t_p) \leq^+ I(x_A, r_A : x_B, r_B) + C(t_p) \\ & \leq & I(x_A : x_B) + C(t_p) + O(\log(n/\epsilon)) \\ & \leq & I(x_A : x_B) + s + O(\log(n/\epsilon)) \end{array}$$

(the third inequality holds with probability $1 - \epsilon$).

Let us consider now a model where Alice and Bob communicate via two channels: via an 'expensive' private channel of capacity s bits and a 'cheap' public channel of unbounded capacity (accessible to the adversary). As above, let x_A and x_B be input strings of length n on which the protocol succeeds with error probability ϵ and randomness deficiency $\delta(n) = O(\log n)$, and let z be the random string that is the shared secret key output by the protocol, i.e., a string satisfying relations (2) and (3). We combine the simple observation above with the proof of Theorem 4.2 and conclude with the following result: with probability at least $1 - O(\epsilon)$, if n is sufficiently large, $|z| \leq I(x_A : x_B) + s + O(\log(n/\epsilon))$.

The 'light upper bound' from Section 3 does not apply to the model with an additional private channel. Actually, the traffic of size s over the private channel in some sense can be counted as an "overhead" in inequalities (14) and (15). More precisely, we can prove a version of Lemma 3.1 with an extra term corresponding to the private channel. If a message sent by Alice is computed as a function of its own input x_A and of the data t_{priv} sent via the private channel, than similarly to (14) we can prove that

$$I(x_A : x_B \mid f(x_A, t_{\mathrm{priv}})) \leq^+ I(x_A : x_B) + C(t_{\mathrm{priv}}).$$

Also, if a message sent by Alice is computed as a function of her own input x_A, the publicly visible "prehistory" of the protocol t_{pub} (i.e., the bits sent earlier via the public channel), and of the secret part of the transcript t_{priv} (sent via the private channel), then similarly to (15) we can prove that

$$I(x_A : x_B \mid g(x_A, t_{\mathrm{pub}}, t_{\mathrm{priv}}), t_{\mathrm{pub}}) \leq^+ I(x_A : x_B \mid t_{\mathrm{pub}}) + C(t_{\mathrm{priv}}).$$

Using these inequalities and keeping in mind that the size of the private part of the communication $C(t_{\mathrm{priv}})$ is bounded by s, we can adapt the argument from p. 11 with a suitable correction: in the chain of inequalities (16) we need to count s bits of the "overhead" for each round of communication via the public channel. For a k-round protocol this would give the bound $|z| \leq I(x_A : x_B) + ks + O(\log(n/\epsilon))$, which is too weak for non-constant k.

We revisit these observations in Section 8.4.

8.4 Secret key agreement: the Shannon framework vs. the Kolmogorov framework

Upper and lower bounds on the size of the shared secret key. In the Introduction we discussed the general connections between IT and AIT. In what follows we discuss some more precise and formal relations between Shannon-type and Kolmogorov-type theorems on secret key agreement.

Actually the upper and lower bounds on the size of the common key in the framework of Kolmogorov complexity (Theorem 4.1 and Theorem 4.2) formally imply similar bounds in Shannon's framework for i.i.d. pairs variables and even for stationary ergodic sources (Theorem 1 in [Tya13], see also [LYC76, BBR88, Mau93, AC93]). Indeed, it is well known that a sequence of i.i.d. random variables (or, more generally, the outcome of a stationary ergodic source) gives with a high probability a string of letters whose Kolmogorov complexity is close to Shannon's entropy of the random source (this statement was announced by L. Levin in [ZL70, Proposition 5.1] and published with proof in [Bru82]; see also [Hor03]). Hence, if Alice gets a value of $(X_1, \ldots, X_n)$, Bob gets a value of $(Y_1, \ldots, Y_n)$, and the sequence of pairs (X_i, Y_i) is a stationary ergodic source, then with high probability the mutual information (in the sense of Kolmogorov complexity) between Alice's and Bob's individual inputs is close to the mutual information between these two random sequences (in the sense of Shannon's entropy).

Thus, to prove the positive statement in Shannon's framework (Alice and Bob can agree on a secret key of size about $I(X_1, \ldots, X_n : Y_1, \ldots, Y_n)$) we can apply the communication protocol from the proof of Theorem 4.1. Notice that in Theorem 4.1 Alice and Bob need to know the complexity profile of the input data; we cannot know the *exact*

values of Kolmogorov complexity for a pair of randomly chosen random strings, but it is enough to provide Alice and Bob with the known *expected* value of the complexity profile of their input data (see Remark 5 on p. 14). In the case when the complexity profile of the values of $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$ is 'typical' (close to the average values), Theorem 4.1 guarantees that the resulting common key z has a large enough Kolmogorov complexity conditional on the transcript of the protocol. Otherwise, (in case when the complexity profile of the input data is not typical) the result of the protocol can be meaningless; however, this happens with only a small probability. So, assuming that Alice and Bob are given random values of $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$ respectively, we conclude that the outcome of the protocol (the common secret key) is distributed mostly on strings of high Kolmogorov complexity. It is not hard to show that such a distribution must have Shannon's entropy (conditional on the value of the transcript) close to $I(X_1, \ldots, X_n : Y_1, \ldots, Y_n)$.

In a similar way, our negative result (the secret common key cannot be made bigger, Theorem 4.2) implies a similar result in Shannon's setting. Indeed, assume that there exists a communication protocol that permits to Alice and Bob to agree on a common secret on random inputs $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$ with a probability close to 1. If the pair of random sources is stationary and ergodic, then with high probability the pair of input values has a complexity profile close to the corresponding values of Shannon's entropy. It follows from Theorem 4.2 that the Kolmogorov complexity of the secret key is not much bigger than the mutual information between $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$. It remains to notice that the Shannon's entropy of the common secret key Z (as a random variable) cannot be much bigger than the average Kolmogorov complexity of its value (as a binary string).

In Shannon's framework, we can consider a hybrid communication model (similar to those in Section 8.3), where Alice and Bob can communicate via a private channel of bounded capacity (hidden from the adversary) and a public channel of unbounded capacity (accessible to the adversary). The connection between Shannon's and Kolmogorov's settings discussed above combined with the observation made in Section 8.3 implies that, in this model, Alice and Bob cannot agree on a common secret key of entropy greater than

$$I(X_1, \ldots, X_n : Y_1, \ldots, Y_n) + [\text{capacity of the private channel}].$$

Communication complexity of the protocol. Another issue that can be studied in both settings (for Shannon's and Kolmogorov's approaches to the key agreement problem) is the optimal communication complexity of the protocol. Tyagi investigated in [Tya13] this problem for the Shannon's framework (for i.i.d. sources) and showed that in some cases (for some distributions (X_i, Y_i)) Alice and Bob need to send to each other at least

$$\min(H(X_1, \ldots, X_n \mid Y_1, \ldots, Y_n), H(Y_1, \ldots, Y_n \mid X_1, \ldots, X_n))$$

bits of information to agree on a common secret key of size $I(X_1, \ldots, X_n : Y_1, \ldots, Y_n)$. Moreover, Tyagi found a complete description (a so-called *one letter characterization*) of the optimal communication complexity for an arbitrary distribution on Alice's and Bob's inputs. This result is similar to our Theorem 6.1, which claims that Alice and Bob need to send at least $\min(C(x \mid y), C(y \mid x))$ bits of information to agree even on a common secret key of size $\delta I(x : y)$. However, these results are incomparable with each other (one does not follow from another):

- Tyagi investigated protocols that agree on a common secret of maximal size $I(X_1, \ldots, X_n : Y_1, \ldots, Y_n)$, while Theorem 6.1 applies even if Alice and Bob agree on a secret of size $\delta I(x : y)$ for a small $\delta > 0$.
- [Tya13] proposed a one letter characterization of the communication complexity in terms of the distribution (X_i, Y_i), which arguably cannot be reformulated in the setting of Kolmogorov complexity.
- The result in [Tya13] applies to communication protocols with private randomness, while Theorem 6.1 is proven only for protocols with public randomness.

One-shot paradigm. Most known results on the protocols for secret key agreement in Shannon's framework are proven with the assumption that the input data available to Alice and Bob are i.i.d. random variables or at least stationary ergodic random sources. We mentioned above that for this class of inputs (stationary ergodic random sources) the bounds on the optimal size of the secret in Shannon's framework can be deduced from analogous results on Kolmogorov complexity. But, actually Theorem 4.1 and Theorem 4.2 apply in more general settings. We can prove similar bounds for random inputs obtained *in one shot*, without the property of ergodicity.

In many natural instances of the secret key agreement problem, the input data are far from being ergodic, so the classic technique does not apply. In Section 8.1, we discussed two examples of this kind: in the first one, Alice is given a line in the affine plane, and Bob is given a point in this line; in the second one, Alice and Bob get n-bits strings x and y respectively, and the Hamming distance between these strings is at most δn for some constant $\delta < 1/2$. These examples can be naturally reformulated in the probabilistic setting: we can introduce the uniform distribution on the set of all valid pairs of inputs. For the probabilistic versions of these examples, the matching upper and lower bounds on the size of the common secret key can be easily deduced from Theorem 4.1 and Theorem 4.2.

To conclude, the standard results on the secret key agreement deal with the paradigm where *the protocol works properly for most randomly chosen inputs* (which is typical for information theory), while in our approach we prove a somewhat stronger statement : *for each valid pair of input data the protocol works properly with high probability* (which is typical for the theory of communication complexity).

8.5 Communication protocols with logarithmic advice

In this section we show that in some setting the 'light upper bound' in Section 3 is more robust and flexible than the more technical proof of Theorem 4.2.

Despite many prominent parallels between Shannon's and Kolmogorov's variants of information theory, there is no canonical way to translate the basic properties from one framework to another. The rule of thumb is to substitute Kolmogorov complexity instead of Shannon's entropy and to change all constraints of type '*a random variable X is a function of a random variable Y*' in Shannon's case to '*a string x has small Kolmogorov complexity conditional on y*' in Kolmogorov's version. For example, the classic Slepian–Wolf theorem essentially claims that, for jointly distributed X, Y (that technically must be a sequence of i.i.d. pairs), there exists another random variable Z such that

$$\begin{cases} Z \text{ is a deterministic function of } X, \\ \text{number of bits in } Z \approx H(X \mid Y), \\ X \text{ can be recovered from } Y \text{ and } Z \text{ with high probability.} \end{cases}$$

The counterpart of this result for Kolmogorov complexity is known as Muchnik's theorem, [Muc98]: for all strings x, y of length n there exists a string z such that

$$\begin{cases} C(z \mid x) = O(\log n), \\ |z| = C(x \mid y) + O(\log n), \\ C(x \mid y, z) = O(\log n). \end{cases}$$

We can use a similar logic to translate in the language of Kolmogorov complexity the general notion of an interactive communication protocol. A protocol adapted to the framework of Kolmogorov complexity can be viewed as a protocol where on each step Alice and Bob get logarithmic 'advice' strings from a trustable Merlin.

In what follows we give a more formal definition of a protocol with advice bits. Let k be a positive integer. We say that Alice and Bob given inputs x_A an x_B compute a common value z in a k-rounds *protocol with q bits of advice on each round*, given inputs x_A and x_B, if there exists is a sequence of strings ('messages') $x_1, y_1, x_2, y_2, \ldots, x_k, y_k$ such that

$$\begin{aligned} &C(x_1 \mid x_A) \le q, \\ &C(y_1 \mid x_B, x_1) \le q, \\ &C(x_2 \mid x_A, x_1, y_1) \le q, \\ &C(y_2 \mid x_B, x_1, y_1, x_2) \le q, \\ &\vdots \\ &C(x_k \mid x_A, x_1, y_1, \ldots, x_{k-1}, y_{k-1}) \le q, \\ &C(y_k \mid x_B, x_1, y_1, \ldots, x_{k-1}, y_{k-1}) \le q, \\ &C(z \mid x_A, x_1, y_1, \ldots, x_k, y_k) \le q, \\ &C(z \mid x_B, x_1, y_1, \ldots, x_k, y_k) \le q. \end{aligned}$$

In a similar way, we can define the *randomized* protocols with advice, where Alice and Bob are given in addition random r_A and r_B. For $c > 0, \epsilon > 0$, we say that Alice and Bob (ϵ, c)-*succeed* on an input pair (x_A, x_B) to agree on

a common secret key in a k-rounds protocol with q bits of advice, if with probability $(1-\epsilon)$ over r_A, r_B, there exists a string z such that Alice and Bob can compute common value z in a k-rounds protocol with advice as defined above, and

$$C(z \mid t) \geq^{+} |z| - c \log n,$$

where $t := (x_1, y_1, \ldots, x_k, y_k)$ is a transcript of this protocol.

Remark 17. In a conventional definition of a communication protocol, every next message sent by Alice or Bob is computed as a function of all previous messages and of the input given to Alice or Bob respectively (this function is one and the same for all pairs of inputs). Thus, every next message is "simple" conditional on the previous messages and one party's input in a very strong sense. Such a schema is unusual in the context of algorithmic information theory, where the relation "a is simple conditional on b" is typically understood as $C(a \mid b) \sim \log(C(a)+C(b))$. This observation motivates the given above definition of a communication protocol with advice. As a matter of fact, we substitute the standard definition of a communication protocol by a chain of conditions "every new message is simple conditional on the prehistory of the communication and one party's input." Considering this motivation, we believe that a natural measure if "simplicity" is logarithmic (i.e., on each round of the protocol we can use $O(\log n)$ bits of advice, where n is the length of the inputs).

Formally speaking, we might consider another communication model where the advice is provided only for some selected rounds, while on other steps of the protocol the messages are computed in a conventional way, by a fixed in advance deterministic function. In this case we should bound *the total* number of advice bits distributed over all rounds. However, such a mixed model lacks a natural motivation.

Any conventional communication protocol (without advice) can serve as a protocol with advice strings of size $q = O(1)$, so the lower bound from Theorem 4.1 remains true in the new setting. Also the proof of the 'light' upper bound form Section 3 applies to the protocols with advice strings of size $q = O(\log n)$, and we conclude that in the new setting Alice and Bob still cannot agree on a common secret key of size greater than $I(x_A : x_B)$. Notice that the proof of Theorem 4.2 does not apply to protocol with advice, so we cannot extend this upper bound to non-constant number of protocols. But this is not a real loss: the protocols with advice probably make no sense for non-constant number of rounds (since too much information can be hidden in the bits of advice).

8.6 Weak version of negative results for multi-party protocols

In this section we prove a weak version of the upper bound on the length of the longest secret key that three parties can agree upon. The proposed argument follows the ideas from the proof of a similar result in [CN04] in Shannon's setting (it is based on classic information inequalities). While this technique works pretty well in IT, in our case it covers only the protocol with $O(1)$ rounds and with the number of random bits bounded polynomially in the input length. Thus, the result proven in this section is much weaker than Theorem 5.5. However, it is instructive to see how far we can go in the setting of AIT with the classic technique borrowed from IT.

Theorem 8.2. *Let us consider a 3-party protocol for secret key agreement with error probability ϵ, having a constant number of rounds, and where the number of random bits is bounded polynomially in the input length. Let (x_A, x_B, x_C) be a 3-tuple of n-bit strings on which the protocol succeeds. Let z be the random variable which represents the secret key computed from the input (x_A, x_B, x_C) and let t be the transcript of the protocol that produces z. Then, for sufficiently large n, with probability $1 - O(\epsilon)$,*

$$C(z \mid t) \leq C(x_A, x_B, x_C) - \mathrm{CO}(x_A, x_B, x_C) + O(\log(n/\epsilon)),$$

where the constants in the $O(\cdot)$ notation depend on the universal machine, the number of rounds and the protocol, but not on (x_A, x_B, x_C).

Proof. We fix a triplet of n-bit input strings (x_A, x_B, x_C) and we consider some randomness $r = (r_A, r_B, r_C)$, on which the protocol produces a string z which satisfies the relations (23) and (24). Recall that the set of such r's has probability $1 - \epsilon$. To simplify the notation, we merge the input of each participant with the corresponding random strong and denote

$$x'_A := \langle x_A, r_A \rangle,\ x'_B := \langle x_B, r_B \rangle,\ x'_C := \langle x_C, r_C \rangle.$$

Observe that with high probability the random strings r_A, r_B, r_C are independent of x_A, x_B, x_C, and, therefore,

$$C(x'_A, x'_B, x'_C) =^+ C(x_A, x_B, x_C) + |r_A| + |r_B| + |r_C|$$

and

$$\mathrm{CO}(x'_A, x'_B, x'_C) =^+ \mathrm{CO}(x_A, x_B, x_C) + |r_A| + |r_B| + |r_C|.$$

Thus, to prove the theorem, it is enough to show that

$$C(z \mid t) \leq C(x'_A, x'_B, x'_C) - \mathrm{CO}(x'_A, x'_B, x'_C) + O(\log(n/\epsilon)).$$

To this end we transform the value of $C(x'_A, x'_B, x'_C)$ as follows:

$$\begin{array}{rcl} C(x'_A, x'_B, x'_C) & =^+ & C(t, z, x'_A, x'_B, x'_C) \\ & =^+ & C(t_1) + C(t_2 \mid t_1,) + C(t_3 \mid t[1:2]) + \ldots + C(t_{3k} \mid t[1:3k-1]) \\ & + & C(z \mid t) + C(x'_A \mid z, t) + C(x'_B \mid x'_A, z, t) + C(x'_C \mid x'_A, x'_B, z, t). \end{array} \tag{42}$$

The first line follows because t and z are computed from x_A, x_B, x_C and r, and the second line uses the chain rule. Next, we break the sum in the right hand side of equation (42) into the sum $q_1(r) + q_2(r) + q_3(r)$, where the terms in the latter sum are defined as

$$\begin{array}{rcl} q_1(r) & := & C(t_1) + C(t_4 \mid t[1:3]) + \ldots + C(t_{3k-2} \mid t[1:3k-3]) + C(x'_A \mid z, t) \\ q_2(r) & := & C(t_2 \mid t_1) + C(t_5 \mid t[1:4]) + \ldots + C(t_{3k-1} \mid t[1:3k-2]) + C(x'_B \mid x'_A, z, t), \\ q_3(r) & := & C(t_3 \mid t[1:2]) + C(t_6 \mid t[1:5]) + \ldots + C(t_{3k} \mid t[1:3k-1]) + C(x'_C \mid x'_A, x'_B, z, t). \end{array}$$

By definition of $q_1(r), q_2(r), q_3(r)$ and (42) we have

$$C(x'_A, x'_B, x'_C) =^+ C(z \mid t) + (q_1(r) + q_2(r) + q_3(r)). \tag{43}$$

The following claim shows that $q_1(r), q_2(r)$ and $q_3(r)$ satisfy relations similar to those that define $S(x_A, x_B, x_C)$.

Claim 8.3.

(1) $q_1(r) \geq^+ C(x'_A \mid x'_B, x'_C)$, $q_2(r) \geq^+ C(x'_B \mid x'_A, x'_C)$, $q_3(r) \geq^+ C(x'_C \mid x'_A, x'_B)$.

(2) $q_1(r) + q_2(r) \geq^+ C(x'_A, x'_B \mid x'_C)$, $q_1(r) + q_3(r) \geq^+ C(x'_A, x'_C \mid x'_B)$, $q_2(r) + q_3(r) \geq^+ C(x'_B, x'_C \mid x'_A)$.

Proof of Claim 8.3. We show the first relation in (1).

$$\begin{array}{rcl} C(x'_A \mid x'_B, x'_C) & =^+ & C(t, z, x'_A \mid x'_B, x'_C) \\ & =^+ & C(t_1 \mid x'_B, x'_C) + C(t_2 \mid t_1, x'_B, x'_C) + C(t_3 \mid t[1:2], x'_B, x'_C) \\ & & + C(t_4 \mid t[1:3], x'_B, x'_C) + C(t_5 \mid t[1:4], x'_B, x'_C) + C(t_6 \mid t[1:5], x'_B, x'_C) \\ & & \vdots \\ & & + C(t_{3k-2} \mid t[1:3k-3], x'_B, x'_C) + C(t_{3k-1} \mid t[1:3k-2], x'_B, x'_C) \\ & & + C(t_{3k} \mid t[1:3k-1], x'_B, x'_C) \\ & & + C(z \mid t, x'_B, x'_C) + C(x_A \mid t, z, x'_B, x'_C). \end{array} \tag{44}$$

The first line follows from the fact that t and z are computed from x_A, x_B, x_C and r, and the second line follows from the chain rule.

For every $\ell \in \{1, \ldots, k\}$, $C(t_{3\ell-1} \mid t[1:3\ell-2], x'_B, x'_C) =^+ 0$, because $t_{3\ell-1}$ is computed from $t[1:3\ell-2], x'_B, x'_C)$. For the same reason, $C(t_{3\ell} \mid t[1:3\ell-1], x'_B, x'_C) =^+ 0$ and $C(z \mid t, x'_B, x'_C) =^+ 0$. Eliminating the terms that are $=^+ 0$ in equation (44) and also the conditioning on x'_B and x'_C in all terms on the right hand side, we obtain

$$\begin{array}{rcl} C(x'_A \mid x'_B, x'_C) & \leq^+ & C(t_1) + C(t_4 \mid t[1:3]) + \ldots + C(t_{3k-2} \mid t[1:3k-3]) + C(x'_A \mid z, t) \\ & = & q_1(r). \end{array}$$

The other relations in the claim are proved in a similar way. □

Claim 8.3 implies that $q_1(r) + q_2(r) + q_3(r) \geq \mathrm{CO}(x'_A, x'_B, x'_C)$. Combining this fact with relation (43), we obtain $C(z \mid t) \leq^+ C(x'_A, x'_B, x'_C) - \mathrm{CO}(x'_A, x'_B, x'_C)$, and we are done. □

8.7 Open problems

Open Question 1. In Theorem 6.1 we establish a lower bound on how many bits Alice and Bob must communicate to agree on a common secret key. Our proof is valid only for communication protocols with public randomness. Is the same bound true for protocols with private sources of random bits?

Open Question 2. All our communication protocols are randomized. It is natural to ask whether we can get rid of external randomness. We conjecture that for $(O(\log n), O(\log n))$-stochastic tuples of inputs the protocol can be made purely deterministic (though it would require very high computational complexity), but this cannot be done in the general case. The proof of this fact would require a better understanding of the nature of non-stochastic objects (for a discussion of non-stochastic objects see [She83] and [SUV17, Section 14.2]).

9 Acknowledgments

We are grateful to Bruno Bauwens for his insightful comments. In particular we thank him for allowing us to reproduce his proof of the "light lower bound" in Section 3 and for bringing to our attention the improved Kolmogorov complexity version of the Slepian-Wolf theorem 2.1. We thank Tarik Kaced for attracting our attention to [CABE$^+$15].

References

[AC93] Rudolf Ahlswede and Imre Csiszár. Common randomness in information theory and cryptography - I: secret sharing. *IEEE Trans. Information Theory*, 39(4):1121–1132, 1993.

[ALPS10] Luis Antunes, Sophie Laplante, Alexandre Pinto, and Liliana Salvador. Cryptographic security of individual instances. *ICITS*, pages 195–210, 2010.

[Bau18] Bruno Bauwens. Optimal probabilistic polynomial time compression and the slepian-wolf theorem: tighter version and simple proofs. *arXiv preprint arXiv:1802.00750*, 2018.

[BBR88] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[Bru82] A. A. Brudno. Entropy and the complexity of the trajectories of a dynamic system. *Trudy Moskovskogo Matematicheskogo Obshchestva*, 44:124–149, 1982.

[BZ14] Bruno Bauwens and Marius Zimand. Linear list-approximation for short programs (or the power of a few random bits). In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 241–247. IEEE, 2014.

[CABE$^+$15] Chung Chan, Ali Al-Bashabsheh, Javad B Ebrahimi, Tarik Kaced, and Tie Liu. Multivariate mutual information inspired by secret-key agreement. *Proceedings of the IEEE*, 3(10):1883–1913, 2015.

[Cha66] Gregory J. Chaitin. On the length of programs for computing finite binary sequences. *Journal of the ACM*, 13:547–569, 1966.

[CK78] Imre Csiszár and János Körner. Broadcast channels with confidential messages. *IEEE Trans. Information Theory*, 24(3):339–348, 1978.

[CMR$^+$02] Alexey V. Chernov, Andrei A. Muchnik, Andrei E. Romashchenko, Alexander Shen, and Nikolai K. Vereshchagin. Upper semi-lattice of binary strings with the relation "x is simple conditional to y". *Theor. Comput. Sci.*, 271(1-2):69–95, 2002.

[CN04] Imre Csiszár and Prakash Narayan. Secrecy capacities for multiple terminals. *IEEE Trans. Information Theory*, 50(12):3047–3061, 2004.

[GK73] Peter Gács and János Körner. Common information is far less than mutual information. *Probl. Control Inf. Theory*, 2(2):149–162, 1973.

[Gol12] Shafi Goldwasser. Pseudo-deterministic algorithms. In *STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)*, volume 14, pages 29–29. LIPIcs, 2012.

[GS10] Venkatesan Guruswami and Adam Smith. Codes for computationally simple channels: Explicit constructions with optimal rate. *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, (723–732), 2010.

[GS16] Venkatesan Guruswami and Adam Smith. Optimal rate code constructions for computationally simple channels. *Journal of the ACM (JACM)*, 63(4):35, 2016.

[Hor03] Yasuichi Horibe. A note on Kolmogorov complexity and entropy. *Applied mathematics letters*, 16(7):1129–1130, 2003.

[Kol65] Andrei Nikolaevich Kolmogorov. Three approaches to the quantitative definition of information. *Problems Inform. Transmission*, 1(1):1–7, 1965.

[KR13] Tarik Kaced and Andrei Romashchenko. Conditional information inequalities for entropic and almost entropic points. *IEEE Transactions on Information Theory*, 59(11):7149–7167, 2013.

[KRV15] Tarik Kaced, Andrei Romashchenko, and Nikolay Vereshchagin. Conditional information inequalities and combinatorial applications. *arXiv preprint arXiv:1501.04867*, 2015.

[LYC76] Sik Kow Leung-Yan-Cheong. Multi-user and wiretap channels including feedback, July 1976. Tech. Rep. No. 6603-2, Stanford Univ.

[Mau92] Ueli M. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1):53–66, 1992.

[Mau93] Ueli M. Maurer. Secret key agreement by public discussion from common information. *IEEE Trans. Information Theory*, 39(3):733–742, 1993.

[MR10] Andrei A. Muchnik and Andrei E. Romashchenko. Stability of properties of Kolmogorov complexity under relativization. *Problems of information transmission*, 46(1):38–61, 2010.

[MRS11] D. Musatov, A. E. Romashchenko, and A. Shen. Variations on Muchnik's conditional complexity theorem. *Theory Comput. Syst.*, 49(2):227–245, 2011.

[Muc98] Andrei A. Muchnik. On common information. *Theor. Comput. Sci.*, 207:319–328, 1998.

[Raz11] Ilya Razenshteyn. Common information revisited. *arXiv preprint arXiv:1104.3207*, 2011.

[Rom00] Andrei Romashchenko. Pairs of words with nonmaterializable mutual information. *Problems of Information Transmission*, 36(1):3–20, 2000.

[RRV02] Ran Raz, Omer Reingold, and Salil P. Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. *J. Comput. Syst. Sci.*, 65(1):97–128, 2002.

[Sha02] Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77(67-95):10, 2002.

[She83] Alexander Kh. Shen. The concept of (α, β)-stochasticity in the Kolmogorov sense, and its properties. *Soviet Math. Dokl.*, 28(1):295–299, 1983.

[Smi07] Adam D. Smith. Scrambling adversarial errors using few random bits, optimal information reconciliation, and better private codes. *Symposium on Discrete Algorithms: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 7(09):395–404, 2007.

[Sol64] Ray J. Solomonoff. A formal theory of inductive inference. *Information and Control*, 7:224–254, 1964.

[SUV17] Alexander Shen, Vladimir Uspensky, and Nikolay Vereshchagin. *Kolmogorov complexity and algorithmic randomness*. American Mathematical Society, 2017.

[Tya13] Himanshu Tyagi. Common information and secret key capacity. *IEEE Transactions on Information Theory*, 59(9):5627–5640, 2013.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1-3):1–336, 2012.

[Wyn75] Aaron D. Wyner. The wire-tap channel. *Bell Syst. Tech J.*, 54(8):1355–1387, 1975.

[Zim17] Marius Zimand. Kolmogorov complexity version of Slepian-Wolf coding. In *STOC 2017*, pages 22–32. ACM, June 2017.

[ZL70] Alexander Zvonkin and Leonid Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25(6):83–124, 1970.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il