An operational characterization of mutual information in algorithmic information theory

Andrei Romashchenko, Marius Zimand

An Operational Characterization of Mutual Information in Algorithmic Information Theory

Andrei Romashchenko[1]
LIRMM, Univ Montpellier, CNRS, Montpellier, France; on leave from IITP RAS
andrei.romashchenko@lirmm.fr

Marius Zimand
Department of Computer and Information Sciences, Towson University, Baltimore, MD
http://orion.towson.edu/~mzimand

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.

2012 ACM Subject Classification Mathematics of computing → Information theory, Theory of computation → Communication complexity, Theory of computation → Probabilistic computation

Keywords and phrases Kolmogorov complexity, mutual information, communication complexity, secret key agreement

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.95

Related Version A full version of the paper is available at https://arxiv.org/abs/1710.05984.

Acknowledgements We are grateful to Bruno Bauwens for his insightful comments. We thank Tarik Kaced for attracting our attention to [5].

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 concept from 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

[1] Supported in part by ANR-15-CE40-0016-01 RaCAF grant.

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 95; pp. 95:1–95:14

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

them. As explained in the figure legend there are six important regions. The regions (1) to

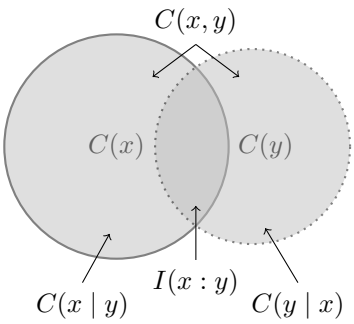

Figure 1 Two strings x and y, and their information. There are six regions that we distinguish: (1) The left solid circle represents the information in x, as given by its Kolmogorov complexity, denoted $C(x)$; (2) The right dotted circle represents the information in y, denoted $C(y)$; (3) The entire 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 $I(x:y)$.

(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 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 [15, 3, 17, 1]). In this paper, we look at this phenomenon using the very general framework of algorithmic information theory to measure the correlation of strings.

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** (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 Charles 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 Charles 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 [28], 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: we have ignored in the entire discussion some $O(\log n)$ terms). 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. 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: no probabilistic protocol can produce a longer secret key.

▶ Remark. The value $\mathrm{CO}(A, B, C)$ is understood as the communication complexity of the *omniscience problem*. However, it can be computed as a function of the Kolmogorov complexities of the involved strings, see Definition 4 below. This fact (the communication complexity of the optimal omniscience protocol depends only on the complexity profile of the inputs) is not trivial and requires a proof.

Communication complexity for secret key agreement. In the protocol in Theorem 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. To determine 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 can not 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.* [2] 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 [27] and Csiszár and Körner [7] 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 [16, 17] 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 [1] and Maurer [17] 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 [17, 1] 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 [8] 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 [1, 17, 8] and represent the AIT analogue of the results presented above. 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). Regarding the communication complexity of secret key agreement protocols, we only note here that Tiyagi [26] has shown that for memoryless sources it is equal to the difference between interactive common information in Wyner's sense and mutual information. In the full version of this paper 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 works 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 [9] 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 [19] and Romashchenko [22] 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. [6] 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 [23, Chapter 11].

Thus, previous works have 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.

1.3 The basics of algorithmic information theory

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 $C(x)$. Similarly to the complexity of x, we define the complexity of x conditioned by y as $C(x \mid y) = \min\{|p| \mid \ U \text{ on input } p \text{ and } y \text{ prints } x\}$. We list below a few basic facts about Kolmogorov complexity and introduce some notation:

- 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.)
- 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.
- 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 only depends on the universal machine U. Similarly, $a \geq^+ b$ means $a \geq b - O(\log n)$, and $a =^+ b$ means ($a \leq^+ b$ and $a \geq^+ b$).

- The chain rule (a.k.a. the Kolmogorov–Levin theorem) claims that for all sufficiently long strings x and y, $\big|C(x,y) - (C(x) + C(y \mid x))\big| \le 3(\log C(x) + \log C(y))$.
- 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 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 $(C(x_V) \mid V \subseteq [\ell], V \neq \emptyset)$. Here x_V denotes the subtuple obtained by taking the components with indices in V (for example if $V = \{1, 2, 7\}$ then $x_V = (x_1, x_2, x_7)$).

1.4 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:

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

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 us fix parameters ϵ and $\delta(n)$. (We assume ϵ is a positive constant and $\delta(n)$ is 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,

$$A(x_A, r_A, t) = B(x_B, r_B, t) \stackrel{def.}{=} z, \tag{1}$$

and

$$C(z \mid t) \ge |z| - \delta(n), \tag{2}$$

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

The string z satisfying equation (1) and inequality (2) 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.

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 (1)) such that z is random even conditioned by the transcript of the protocol (inequality (2)). Thus, z is secret to an adversary that has observed the protocol and consequently knows the transcript.

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

2 Main results

We present here our main results. We first show that there exists 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 2** (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 3** (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 (1) and (2). 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, but not on x_A and x_B.*

Theorem 3 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 3 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. 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 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. However, the proof of Theorem 2 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. If 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 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 $I(x : y) - \sigma - O(\log(n/\epsilon))$.

3 Secret key agreement for three or more parties

In this section we analyze secret key agreement for 3 parties, which we call Alice, Bob, and Charles. Alice has a string x_A, Bob has a string x_B, and Charles 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:

$$\left.\begin{array}{lll} t_1 = A(x_A, r_A), & t_2 = B(x_B, r_B), & t_3 = C(x_C, r_C), \\ t_4 = A(x_A, r_A, t[1:3]), & t_5 = B(x_B, r_B, t[1:3]), & t_6 = C(x_C, r_C, t[1:3]), \\ t_7 = A(x_A, r_A, t[1:6]), & t_8 = B(x_B, r_B, t[1:6]), & t_9 = C(x_C, r_C, t[1:6]), \\ \vdots & & \end{array}\right\} k \text{ rounds}$$

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 $(t_i, \ldots, t_j)$. 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,

$$A(x_A, r_A, t) = B(x_B, r_B, t) = C(x_C, r_C, t) \stackrel{def.}{=} z, \tag{3}$$

and $C(z \mid t) \geq |z| - \delta(n)$.

▶ **Definition 4.**

(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:

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

The constraints defining $S(x_1, x_2, x_3)$ will be referred 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.*)

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

▶ **Theorem 5** (Upper bound). *Let us consider a 3-party protocol for secret key agreement with error probability ϵ, 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 $1 - O(\epsilon)$ we have $C(z \mid t) \leq C(x_A, x_B, x_C) - \mathrm{CO}(x_A, x_B, x_C) + O(\log(n/\epsilon))$.*

▶ **Theorem 6** (Lower bound). *There exists a 3-party protocol for secret key agreement with the following characteristics. 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 x_2, the complexity profile of the tuple and ϵ, and Charles's 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.

▶ Remark. Theorem 5 and Theorem 6 remain valid for any constant number $\ell \geq 3$ of parties, with a suitable generalization of the omniscience $\mathrm{CO}(x_1, \ldots, x_\ell)$, see the full version of the paper.

4 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 1.4 (in which the random bits are private): we require that $r_A = r_B = r$ and we change equation (2) to $C(z \mid t, r) \geq |z| - \delta(n)$.

The protocol presented in the proof of Theorem 2 solves the task with communication $\min(C(x \mid y), C(y \mid x)) + O(\log n)$. This protocol can be easily modified to work in the model with public randomness. 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 7.** *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.*

5 Our techniques.

It is common for statements in IT (in the Shannon's entropy framework) to have an analogue 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 [17, 1, 8]. We employ a similar intuitive idea – manipulations with "fingerprints" of inputs of appropriate lengths[2]. 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. 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 [25, 10, 11].

In the negative results (upper bounds for the size of the common secret key, Theorems 2 and 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 detail see the discussion of the limits of the "weak" upper bound int he full version of the paper). 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 is based on ideas similar to the *conditional information inequalities* in [13, 14]. We believe that this technique can be helpful in other cases, including applications in IT (see the discussion in the full version of the paper).

In the proof of a lower bound for communication complexity (Theorem 7), 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 stochastic strings [24, 20, 21], which have not been previously employed in information theory and communication complexity.

6 Final comments

On time-efficient secret key agreement protocols. The secret key agreement protocol in the proof of Theorem 2 is computable but highly non-efficient. The only slow stage is when Bob reconstructs x given his input string y and the fingerprint of x 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. All 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 $\mathrm{poly}(\log n)$, but this is still negligible compared to the size of the fingerprint, which is the dominating term in the communication complexity of the protocol; for details see [28]).

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

[2] On the high level this protocol consists of three stages: (i) Alice sends to Bob a suitable "fingerprint" of her input $p_1(x)$; (ii) Bob uses y and $p_1(x)$ to recover x; (iii) then both Alice and Bob independently compute another fingerprint $p_2(x)$, which is used as a common secret key. The construction of the fingerprints guarantees that the adversary (who eavesdrops p_1) obtains virtually no information about p_2.

Example 1. (Discussed in Introduction, p. 2.) 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 $C(x \mid y) = h(\delta)n \pm O(\log n)$ and $I(x : y) = (1 - h(\delta))n \pm O(\log n)$, 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 [25, 10, 11]. Plugging this protocol in our proof of Theorem 3 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.

On using our approach for "one-shot" sources. Most known results for secret key agreement in Shannon's framework are proven under the assumption that the input data available to Alice and Bob is generated by i.i.d. or at least stationary ergodic sources. These results can be derived from Theorem 2 and Theorem 3, using the well-known relation between Shannon entropy and Kolmogorov complexity for the above type of sources [18, 12]. But actually Theorem 2 and Theorem 3 apply in more general settings. We can prove similar bounds for random inputs obtained *in one shot*, without the property of ergodicity.

This is useful because in many natural instances of the secret key agreement problem the input data are far from being ergodic, and therefore the classic technique does not apply. For instance, *Example 1* and *Example 2* discussed above illustrate this situation if we reformulate them in the probabilistic setting (i.e., we 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 2 and Theorem 3.

On the error probability. The standard results on secret key agreement deal with the paradigm that the protocol works properly for most randomly chosen inputs (which is typical for the 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).

7 Open problems and acknowledgements

▶ **Open Question 1.** *In Theorem 7 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.** *Our communication protocols are randomized. On the other hand they use unusually few random bits, only $O(\log n)$. 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 likely requires a better understanding of the nature of non-stochastic objects (such as Chaitin's Omega number, [4], see also [24] and [23]).*

References

1 Rudolf Ahlswede and Imre Csiszár. Common randomness in information theory and cryptography - I: secret sharing. *IEEE Trans. Information Theory*, 39(4):1121–1132, 1993. doi:10.1109/18.243431.

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

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

4 Gregory J Chaitin. A theory of program size formally identical to information theory. *Journal of the ACM (JACM)*, 22(3):329–340, 1975.

5 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.

6 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. doi:10.1016/S0304-3975(01)00032-9.

7 Imre Csiszár and János Körner. Broadcast channels with confidential messages. *IEEE Trans. Information Theory*, 24(3):339–348, 1978. doi:10.1109/TIT.1978.1055892.

8 Imre Csiszár and Prakash Narayan. Secrecy capacities for multiple terminals. *IEEE Trans. Information Theory*, 50(12):3047–3061, 2004. doi:10.1109/TIT.2004.838380.

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

10 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*, pages 723–732, 2010. doi:10.1109/FOCS.2010.74.

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

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

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

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

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

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

17 Ueli M. Maurer. Secret key agreement by public discussion from common information. *IEEE Trans. Information Theory*, 39(3):733–742, 1993. doi:10.1109/18.256484.

18 Li Ming and Paul M.B. Vitányi. *Kolmogorov complexity and its applications*. Elsevier, 2014.

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

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

21 Ilya Razenshteyn. Common information revisited. *arXiv preprint arXiv:1104.3207*, 2011.

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

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

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

25 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.

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

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

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