Secret Key Agreement from Correlated Data, with No Prior Information

Marius Zimand
Towson University, MD, USA
http://orion.towson.edu/~mzimand/
mzimand@towson.edu

Abstract

A fundamental question that has been studied in cryptography and in information theory is whether two parties can communicate confidentially using exclusively an open channel. We consider the model in which the two parties hold inputs that are correlated in a certain sense. This model has been studied extensively in information theory, and communication protocols have been designed which exploit the correlation to extract from the inputs a shared secret key. However, all the existing protocols are not universal in the sense that they require that the two parties also know some attributes of the correlation. In other words, they require that each party knows something about the other party's input. We present a protocol that does not require any prior additional information. It uses space-bounded Kolmogorov complexity to measure correlation and it allows the two legal parties to obtain a common key that looks random to an eavesdropper that observes the communication and is restricted to use a bounded amount of space for the attack. Thus the protocol achieves complexity-theoretical security, but it does not use any unproven result from computational complexity. On the negative side, the protocol is not efficient in the sense that the computation of the two legal parties uses more space than the space allowed to the adversary.

2012 ACM Subject Classification Theory of computation → Models of computation; Mathematics of computing → Information theory; Security and privacy → Information-theoretic techniques

Keywords and phrases secret key agreement, Kolmogorov complexity, extractors

Digital Object Identifier 10.4230/LIPIcs.STACS.2020.21

Funding *Marius Zimand*: The author has been supported in part by the National Science Foundation through grant CCF 1811729.

Acknowledgements I want to thank Andrei Romashchenko for useful discussions. I also thank the anonymous referees for their observations which have helped me correct some errors and improve the presentation.

1 Introduction

The goal of a secret key agreement protocol is to allow two parties that communicate through a public channel to obtain a shared string that is secret in some reasonable sense (e.g., information-theoretical, complexity-theoretical, or some other sense) to anyone that has observed the communication. There are some well-known such protocols, such as the Diffie-Hellman protocol, or various public-key cryptosystems, that are efficient and used in the real world. However, they have the disadvantage of relying on some unproven hardness conjectures in computational complexity. Another setting is to assume that the two parties hold at the beginning of the protocol pieces of information that have a certain degree of correlation. Then, in some circumstances, it is possible to compute the shared secret key without any unproven assumption. For a simple illustration, suppose that Alice holds a line L in the 2-dimensional affine space, and Bob holds a point P which lies on L. Then Alice sends Bob the slope of L, after which Bob, *knowing that his P is on L*, can compute the intercept of L. Now, both Alice and Bob have the intercept of L, which they can use as a secret key, because the adversary has only seen the slope, which is independent of the intercept.

In this paper, we consider the latter type of secret key agreement protocols. Thus, Alice starts with a string x, Bob starts with y, and, after several rounds of interacting via messages exchanged over a public channel, they obtain at the end of the protocol a common secret key, that is a string z which is random conditioned by the transcript of the protocol. The protocol is probabilistically computable, i.e., there exists a probabilistic algorithm so that Alice computes each of her messages by running the algorithm on her input and on the messages that she has received from Bob so far, and Bob computes his messages similarly. As in the above example, if x and y are *correlated* in some way, one can hope to use the information that is common to these strings to extract with high probability a secret key.

The study of this scenario has a long history in Information Theory and the common flavor of the results is that for many interpretations of "correlated," secret key agreement is possible. Leung [7], Bennett et al. [3], Maurer [8], Ahlswede and Csiszár [1] have started an extensive research line dedicated to the case when x and y are generated by a stochastic process, whose properties describe their correlation (see the survey of Narayan and Tyagi [11]). Recently, Romashchenko and Zimand [13] have studied this problem in the very general framework of Algorithmic Information Theory using Kolmogorov complexity to gauge correlation without using any generative model for the provenance of x and y. [1]

In all these works, Alice and Bob possess at the beginning of the protocol, in addition to x and y, some information about how these strings are correlated. For instance, in the above example, Bob knows that the point P is on the line L. In the scenarios based on generative models, Alice and Bob know various attributes of the joint distribution of the two random variables (X, Y) which describe the stochastic process that generates the pair (x, y), such as entropy, ergodic properties, etc. In the algorithmic information theory setting used in [13], Alice and Bob know the complexity profile of (x, y), which is the 3-tuple $(C(x), C(y), C(x, y))$, where $C(\cdot)$ denotes Kolmogorov complexity. (Throughout this paper, $C(x)$, called the Kolmogorov complexity of x or the minimal description length of x, is the length of a shortest program that when executed by a universal Turing machine prints x.)

Can Alice and Bob agree on a secret key without any additional prior information? A disclaimer: This is not really a problem relevant for cryptography, because the protocols are not efficient. We rather view it as a question about the fundamental limits of information processing and communication. The challenge is that Alice and Bob have to detect a type of correlation of their inputs through rounds of communication without leaking too much information to the eavesdropper, so that they can compute a shared secret key of reasonable length.

What is a reasonable length of the secret key? The relevant parameter that comes into play is the *mutual information* of x and y, denoted $I(x : y)$, which intuitively represents the amount of information that is shared by x and y. In case we use Kolmogorov complexity to measure the amount of information, $I(x : y)$ is defined as $C(x) + C(y) - C(x, y)$, and, up to logarithmic precision, is also equal to $C(x) - C(x \mid y)$ and to $C(y) - C(y \mid x)$. It is shown in [13] (extending a classical result from [1] which is valid for inputs generated by memoryless processes, and which is using Shannon entropy to measure information), that no computable protocol (even probabilistic) can obtain a shared secret key longer than the mutual information of the inputs x and y. On the other hand, a protocol is presented in [13] that with high probability produces a shared secret of length $I(x : y)$ (up to logarithmic

[1] We point out that unlike the protocols based on hardness assumption (e.g., Diffie-Hellman protocol) which achieve complexity-theoretic security and are efficient, the protocols in the works above achieve information-theoretic security but do not run in polynomial time.

precison), provided, as mentioned above, the two parties know the complexity profile of the inputs. Thus, the above discussion suggests that it is natural to aim for a shared secret key whose length is equal to the mutual information of the inputs, for some concept of information that measures the detectable correlation.

Our contribution. We identify space-bounded Kolmogorov complexity as a concept of information that allows secret key agreement without any prior information or special setup (e.g., shared randomness, special extra channel) between the two parties. The space-bounded Kolmogorov complexity with space bound S of a string x, denoted $C^S(x)$, is similar to standard Kolomogorov complexity except that the universal Turing machine is restricted to use at most S cells on the working tape (see Section 1.1 for the formal definition). We show that the correlation induced by space-bounded Kolmogorov complexity can be determined without revealing much about x and y, which, in turn, allows the parties to compute a common secret key.

The protocols that we design produce a key z that is random given the transcript in the sense of space-bounded Kolmogorov complexity, where the transcript is the set of messages sent by Alice and Bob. Formally, we require that $C^S(z \mid \text{transcript})$ is close to the length of z (denoted $|z|$), for some space bound S. In other words, an eavesdropper which is bounded to use space S and knows the transcript, needs essentially $|z|$ bits to find the secret key z, which is the same as if she did not know the transcript. If $C^S(z \mid \text{transcript}) \geq |z| - \Delta$, we say that Δ is the randomness deficiency of z with respect to the transcript, and thus we want to obtain z with small randomness deficiency. We also want the length of z to be close to the mutual information of x and y, which in the case of space-bounded Kolmogorov complexity is defined as $C^{S_1}(x) - C^{S_2}(x \mid y)$ for space bounds S_1 and S_2. We next present our results.

We recall that a function $S(n)$ is fully space constructible if there is a Turing machine M that uses exactly $S(n)$ cells for every natural number n and for every input of length n.

▶ **Theorem 1.** *Let S be any fully space constructible function.such that $S(n) \geq n$.*

There is a randomized protocol that allows Alice on input x (an n-bit string) and Bob on input y (of arbitrary length) to obtain with probability $(1-\epsilon)$ a common string z such that

(i) $|z| \geq^+ C^{\lambda_1 \cdot S(n)}(x) - C^{\lambda_2 \cdot S(n)}(x \mid y)$,

(ii) $C^{S(n)}(z \mid \textit{transcript}) \geq^+ |z| - \Delta$,

where $\Delta \leq C^{\lambda_3 \cdot S(n)}(x) - C^{\lambda_4 \cdot S(n)}(x)$, $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are constants that depend only on the universal Turing machine, and $\geq^+$ hides a loss of precision bounded by $O(\log(n/\epsilon))$.

Note: The notation $a \geq^+ b$ means that $a \geq b - \alpha$, where α is the specified loss of precision.

The communication complexity of the protocol in the above theorem is $n^2 + O(n \log(1/\epsilon)) + C^{S(n)}(x \mid y)$, which is very large. The protocol in [13] (in which the parties also hold the complexity profile of (x, y)) has communication complexity roughly $C(x \mid y)$, which is shown to be optimal. Thus, in our case, it would be desirable to have a protocol with communication complexity close to $C^{S(n)}(x \mid y)$. The protocol in the next theorem has *information complexity* $C^{S(n)}(x \mid y)$ plus a polylogarithmic term and communication complexity $2C^{S(n)}(x \mid y)$ plus a polylogarithmic term.

▶ **Theorem 2** (Main Result). *Let S be a fully space constructible function such that $S(n) \geq p_0(n)$, where $p_0(n)$ is a fixed polynomial that only depends on the universal Turing machine.*

There is a randomized protocol that allows Alice on input x (an n-bit string) and Bob on input y (of arbitrary length) to obtain with probability $(1-\epsilon)$ a common string z such that

(i) $|z| \geq^+ C^{S(n)}(x) - C^{S(n)}(x \mid y)$,

(ii) $C^{S(n)}(z \mid \textit{transcript}) \geq |z| - \Delta$,

where $\Delta \leq^+ C^{\lambda^{-1}S(n)}(x) - C^{\lambda \cdot S(n)}(x)$, λ *is a constant that depends only on the universal Turing machine, and* $\geq^+$ *hides a loss of precision bounded by* $O(\log^3(n/\epsilon))$.
Furthermore, the length of the transcript is bounded by $2C^{S(n)}(x \mid y) + O(\log^3(n/\epsilon))$.

In the above theorems, the key z has Δ, the randomness deficiency conditioned by the transcript, bounded by $C^{S'(n)}(x) - C^{S''(n)}(x)$, where S' and S'' differ by a multiplicative constant. Thus, intuitively, Δ is small. A particularly favorable case is when x is a shallow string. A string x is S-shallow if $C^{S(n)}(x) =^+ C(x)$, i.e., if $S(n)$ is enough space to allow the construction of x from a description which is close to a shortest description. For every space bound S, most strings are S-shallow and in case x is such a string then $\Delta =^+ 0$.

1.1 Prerequisites

The S-space bounded Kolmogorov complexity of x conditioned by y with respect to a Turing machine M, denoted $C^S_M(x \mid y)$, is defined by

$$C^S_M(x \mid y) = \min\{|p| \mid M(p, y) = x \text{ and } M \text{ uses at most } S \text{ cells.}\}$$

In the case of space-bounded Kolmogorov complexity, simulation by the universal machine incurs a constant blow-up in space usage. More precisely, there exists a universal Turing machine U and a constant $\gamma > 1$ such that for any space bound S, for any Turing machine M and for all strings x, y,

$$C^{\gamma S}_U(x \mid y) \leq C^S_M(x \mid y) + O(1).$$

As usual, we fix a universal machine U, and denote more simply $C^S(\cdot)$ instead of $C^S_U(\cdot)$. Also, in case the string y used in the condition is the empty string, we drop the condition in the notation.

The *chain rules* for space-bounded Kolmogorov complexity are as follows: There exists a constant $\gamma > 1$ such that for any space bound S, it holds that:

$$\begin{array}{lcl} C^{\gamma S}(x, y) & \leq & C^S(x) + C^S(y \mid x) + O(\log(|x| + |y|)), \\ C^S(x, y) & \geq & C^{\gamma S}(x) + C^{\gamma S}(y \mid x) + O(\log(|x| + |y|)). \end{array} \tag{1}$$

To simplify the writing of expressions, we sometimes use the notation $CS^{(i)}(\dots)$ instead of $C^{\gamma^i \cdot S}(\dots)$, where S is a space bound and γ (or sometimes λ) is a constant which is clearly defined in the context. For instance, the last inequality will be written as $CS^{(0)}(x, y) \geq CS^{(1)}(x) + CS^{(1)}(y \mid x) + O(\log(|x| + |y|)$.

2 Outline of the proofs

The proofs of both Theorem 1 and Theorem 2 have the same structure. We present an outline, in which, for simplicity, we skip some technical details and ignore small factors in the quantitative relations. Recall that initially Alice holds x and Bob holds y. The protocols in both proofs have two phases: (1) *Information reconciliation*, in which Alice communicates x to Bob by sending him just enough information that allows him to obtain x given his y, and (2) *Secret key construction*, in which, separately, Alice and Bob compute the secret key z. All the communication happens in the Information reconciliation phase.

Phase 1 (Information reconciliation): First, Alice and Bob agree on a space bound $S = S(n)$. Next, Alice sends Bob a randomized hash function h. The goal is for Alice to send Bob, as a fingerprint, some prefix of $h(x)$ that permits Bob to construct Alice's string x using

the fingerprint and his string y. To avoid sending more information than what Bob needs, Alice sends the bits of $h(x)$ sequentially *one bit per round*. At each round, Bob attempts to construct x by checking if the fingerprint of some string in a set of possible candidates matches the prefix of $h(x)$ that he has received so far. More precisely, at each round j, the candidates are those strings whose S-space-bounded complexity conditioned by y is at most j. If Bob finds a string among these candidates with a fingerprint that matches the prefix of $h(x)$ sent so far by Alice, he believes that he has found x, tells Alice to stop sending further bits by sending her "1", and Phase 1 stops. Otherwise, he tells Alice that he needs more bits by sending her "0" (in which case Alice sends in the next round the next bit of $h(x)$).

Let p be the prefix of $h(x)$ that Alice sends to Bob during the entire Phase 1. Then, with high probability, at the end of Phase 1,

1. Bob has x,
2. $|p| \leq C^S(x \mid y)$, because we show that Bob can reconstruct x by round $j = C^S(x \mid y)$. In the proof of Theorem 1, a random matrix H also appears in the condition (as we explain below), but this has little impact, because H is a random.

Phase 2 (Secret key construction): After Phase 1, both Alice and Bob have x (with high probability). They both compute the shared secret key z by exhaustive searching a minimal length program of x given p in space S. So, from p and z, it is possible to construct x. It follows that z and p are independent, because otherwise z would not be minimal. But then z and the transcript of the protocol are also almost independent, because the transcript consists of p and the sequence "$0 \ldots 01$" sent by Bob, and the complexity of $0 \ldots 01$ is low (at most $\log n + O(1)$). Thus, z is a secret key. Let us now estimate the length of z. Since x can be constructed from p and z in space S, it follows that $C^S(x) \leq |p| + |z|$, and thus the length of z is at least $C^S(x) - |p|$, which, by the above bound of $|p|$, is at least $C^S(x) - C^S(x \mid y)$, which is the mutual information of x and y in the framework of space-bounded Kolmogorov complexity.

The main technical issue is finding the hash function that is used in Phase 1. In the proof of Theorem 1, this is just a random linear function given by a random matrix H, chosen by Alice. H is roughly n^2 bits long, and Alice needs to also send H to Bob. This is the reason the communication complexity is large. Also, the information-theoretical considerations in Phase 2, are somewhat more delicate, because we need to take into account H. To reduce the communication complexity, one has to use a shorter hash function. One idea is to use Newman's theorem from communication complexity, in which H is chosen from a smaller sample space. But the sample space needs to be effectively constructed, and the obvious way to do this leads to a loss of precision that is logarithmic in both the length of x and of y, which can be very damaging in case y is much longer than x. In Theorem 2, we use for hashing an explicit extractor of Raz, Reingold, and Vadhan [12], which has the special property that if we take prefixes of the output, the extractor property is preserved. These type of extractors, called *prefix extractors*, allow much more communication-efficient hashing, in the sense that Alice does not need to send the hashing function to Bob, at the cost of making Bob's reconstruction of x more complicated.

In our technical approach, we were inspired by several papers. Muchnik [9] has introduced bipartite graphs similar to extractors and has used for a certain type of information reconciliation concepts similar to what we call *heavy nodes* and *poor nodes* in the proof of Theorem 2. Prefix extractors have been used for information reconciliation in [10] and [15], and the first paper analyzes the case of space-bounded Kolmogorov complexity. The application to secret-key agreement is a novel contribution of this paper. Some of the information-theoretical

estimations are similar to those in [13]. The idea of sending pieces of a fingerprint in several rounds for the problem of information reconciliation (similarly to our Phase 1) has been used before in [4, 6], and, the closest to our approach, in [5], where they study the communication complexity of this problem in terms of the Kolmogorov complexity of the two inputs. There is however a significant difference with the information reconcilation phase in our main result, because, as standard in communication complexity, the protocol in [5] is not computable, and therefore they can use random hash functions for fingerprinting.

3 Proof of Theorem 1

Phase 1: Information reconciliation. Before sending the first message, Alice takes a random matrix H with entries in the finite field GF[2], with $(n + \log(1/\delta))$ rows and n columns, where n is the length of x and $\delta = \epsilon/2n$. The random matrix H defines a random linear function h mapping n bit strings to $n + \log(1/\delta)$ bit strings (viewed as vectors over GF[2]), given by the expression $h(v) = H \cdot v$.

In Round 0, Alice sends to Bob n, H, and the first $1 + \log(1/\delta)$ bits of $h(x)$.

Then in each subsequent round, Alice sends to Bob the next bit of $h(x)$ till Bob announces that he does not need any additional bits. Thus, at round $j \geq 1$, Bob has received the first $(j + 1) + \log(1/\delta)$ bits of $h(x)$, a string which we denote p_j. Bob checks if there is a string u in $B_j = \{u \in \{0,1\}^n \mid C^{S(n)}(u \mid y, H) \leq j\}$ such that p_j is a prefix of $h(u)$. If there is such a string u, he believes that u is x, and announces that he does not need any extra bits and the information reconciliation stops here. If there is no such string u, Bob announces that he needs more bits and the protocol proceeds with the next round.

Bob may be wrong at round j, if there is a string u different from x in B_j such that the prefixes of length $(j + 1) + \log(1/\delta)$ of $h(u)$ and $h(x)$ coincide. For an arbitrary string $u \neq x$, the probability that $h(u)$ and $h(x)$ agree in the first $(j + 1) + \log(1/\delta)$ bits is $2^{-((j+1)+\log(1/\delta))} = \delta/2^{j+1}$. Since B_j has less than 2^{j+1} elements, by the union bound, the probability that Bob is wrong at round j is less than δ.

Let $k = C^{S(n)}(x \mid y, H)$. Let $\mathcal{E}$ be the event that Bob is wrong at one of the rounds $1, \ldots, k$. $\mathcal{E}$ has probability at most $k\delta \leq (n + c)\delta \leq 2n\delta = \epsilon$. Conditioned by $\mathcal{E}$ not being true, the protocol reaches round k, when Bob finds x. Thus, with probability $1 - \epsilon$, at the end of round k, Bob has obtained x, and the string $p := p_k$ is a program for x given y and H in space $\gamma' \cdot S(n)$, for some constant γ', and p has length $C^{S(n)}(x \mid y, H)$.

Phase 2: Secret key construction. By exhaustive search, Alice and (separately) Bob find z, the first program of x given p and H in space S. We show that z satisfies the conclusion of the theorem.

We denote $S := S(n)$ and we let $\geq^+$ hide a loss of precision of $O(\log(n/\epsilon))$. Recall that we use the notation $CS^{(i)}(\ldots)$ in lieu of $C^{\lambda^i \cdot S}(\ldots)$, where λ is here the maximum between the above γ' and γ (the constant from the chain rule (1)).

First, we notice that, with high probability, conditioning by a random H does not decrease complexities by too much.

▷ Claim 3. For every space bound $\mathcal{S}$, for every n-bit string u, for every string v, if H is chosen uniformly at random independent of u and v, we have

$$CS^{(0)}(u \mid v, H) \geq^+ CS^{(2)}(u \mid v) \text{ with probability } 1 - \epsilon.$$

Proof.

$$\begin{aligned} CS^{(0)}(u \mid v, H) &\geq^{+} CS^{(1)}(u, H \mid v) - CS^{(0)}(H \mid v) \\ &\geq^{+} CS^{(2)}(u \mid v) + CS^{(2)}(H \mid u, v) - CS^{(0)}(H \mid v) \\ &\geq^{+} CS^{(2)}(u \mid v) \text{ with probability } 1-\epsilon. \end{aligned} \tag{2}$$

In the first two lines, we use the chain rule, and in the last line, we use the fact that, for every i, $CS^{(i)}(H \mid u, v) \geq |H| - \log(1/\epsilon) - 1$, with probability $1-\epsilon$ (by a standard counting argument) and $CS^{(i)}(H \mid v) \leq |H| + O(1)$ for every H. ◁

Now we can show part (i) of Theorem 1.

$$\begin{aligned} |z| \quad = CS^{(0)}(x \mid p, H) \quad &\geq^{+} CS^{(1)}(x, p \mid H) - CS^{(0)}(p \mid H) \\ &\qquad \text{(chain rule)} \\ &\geq^{+} CS^{(2)}(x \mid H) - |p| \\ &\qquad \text{(because } |p| \geq^{+} CS^{(0)}(p \mid H)) \\ &\geq^{+} CS^{(2)}(x \mid H) - CS^{(0)}(x \mid y, H) \text{ with probability } 1-\epsilon \\ &\qquad \text{(because } |p| = CS^{(0)}(x \mid y, H)) \\ &\geq^{+} CS^{(4)}(x) - CS^{(0)}(x \mid y) \text{ with probability } 1-2\epsilon \\ &\qquad \text{(by Claim 3)} \end{aligned} \tag{3}$$

Next we move to part (ii), where we need to show that the complexity of the secret key z, conditioned by the transcript of the protocol, is close to the length of z. The transcript consists of p, H, n (all sent by Alice to Bob) and of Bob's sequence of responses $s = 000\ldots01$ of length $\ell = k + 1 + \log(1/\epsilon)$. Bob's sequence has complexity bounded by $\log \ell + O(1) = O(\log(n/\epsilon))$, and therefore, for every i we have $CS^{(i)}(z \mid s, p, H, n) =^{+} CS^{(i)}(z \mid p, H)$. Thus we can ignore s and n in the condition and it is enough to bound from below $CS^{(i)}(z \mid p, H)$. We show the following estimation, which ends the proof of the theorem.

▷ Claim 4. With probability $1-2\epsilon$, $CS^{(4)}(z \mid p, H) \geq^{+} |z| - \Delta$, where $\Delta = CS^{(-2)}(x) - CS^{(8)}(x)$.

Proof. We need an upper bound of $|z|$:

$$\begin{aligned} |z| \quad = CS^{(0)}(x \mid p, H) \quad &\leq^{+} CS^{(-1)}(x, p \mid H) - CS^{(0)}(p \mid H) \\ &\qquad \text{(chain rule)} \\ &\leq^{+} CS^{(-2)}(x \mid H) - CS^{(1)}(x \mid y, H) \text{ with probability } 1-\epsilon \\ &\qquad (p \text{ can be computed from } x,\ H \text{ and its length; and } x \text{ from } p, y, H) \\ &\leq^{+} CS^{(-2)}(x) - CS^{(3)}(x \mid y) \text{ with probability } 1-2\epsilon \\ &\qquad \text{(by Claim 3)} \end{aligned} \tag{4}$$

Next,

$$\begin{aligned} CS^{(5)}(p, z \mid H) &\geq^{+} CS^{(6)}(x \mid H) \text{ with probability } 1-\epsilon \\ &\qquad (x \text{ can be computed from } p, z, H) \\ &\geq^{+} CS^{(8)}(x) \text{ with probability } 1-2\epsilon \qquad \text{(by Claim 3)}, \end{aligned} \tag{5}$$

and

$$\begin{aligned} CS^{(5)}(p, z \mid H) &\leq^{+} CS^{(4)}(p \mid H) + CS^{(4)}(z \mid p, H) \\ &\qquad \text{(chain rule)} \\ &\leq^{+} CS^{(3)}(x \mid y, H) + CS^{(4)}(z \mid p, H) \\ &\qquad (p \text{ can be computed from } x, H \text{ and its length}) \\ &\leq^{+} CS^{(3)}(x \mid y) + CS^{(4)}(z \mid p, H) \end{aligned} \tag{6}$$

Combining inequalities (6) and (5), we obtain

$$CS^{(4)}(z \mid p, H) \geq^{+} CS^{(8)}(x) - CS^{(3)}(x \mid y) \text{ with probability } 1 - 2\epsilon. \tag{7}$$

Using inequality (4), we finally obtain

$$CS^{(4)}(z \mid p, H) \geq^{+} |z| - \Delta \text{ with probability } 1 - 4\epsilon, \tag{8}$$

where $\Delta = CS^{(-2)}(x) - CS^{(8)}(x)$. The conclusion follows after rescaling ϵ. ◁

4 Proof of Theorem 2

We first present *extractors*, which have been studied in the theory of pseudorandomness (for example, see [14]). A particular type of extractor, *prefix extractor*, is used in the protocol in the proof of Theorem 2 for hashing.

We recall that a (k, ϵ) extractor is a function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with the property that for every subset $B \subseteq \{0,1\}^n$ of size at least 2^k and for every subset $A \subseteq \{0,1\}^m$:

$$\left|\mathrm{Prob}[\mathrm{E}(\mathrm{U_B}, \mathrm{U_d}) \in \mathrm{A}] - \frac{|\mathrm{A}|}{\mathrm{M}}\right| < \epsilon, \tag{9}$$

where U_B and U_d are independent random variables that are uniformly distributed over B and, respectively, $\{0,1\}^d$.

It is useful to view an extractor E as a bipartite graph G, whose set of left nodes is $\{0,1\}^n$, the set of right nodes is $\{0,1\}^m$, and each left node x has 2^d (not necessarily distinct) right neighbors $\{E(x,w) \mid w \in \{0,1\}^d\}$. The right node $E(x,w)$, for random $w \in \{0,1\}^d$, is viewed as the random fingerprint of the left node x.

As usual, we use *explicit* extractors. An explicit extractor is a family of extractors $\{E_n\}_{n\in\mathbb{N}}$ as above, indexed by n, and with the rest of the parameters k, d, m, ϵ being functions of n, such that there exists an algorithm that computes $E_n(x,w)$ in time polynomial in n. Actually, for us it is more important the space complexity of the algorithm that computes the extractor.

We denote $D = 2^d, M = 2^m$. Let B be a set of left nodes. The average numbers of neighbors in B of a right node (called the average B-degree) is $avg = |B| \cdot D/M$. We say that a right node p is ϵ-heavy for B if it has more $(1/\epsilon) \cdot avg$ left neighbors in B. We say that a left node $u \in \{0,1\}^n$ is ϵ-poor for B if a fraction larger than 2ϵ of its right neighbors are ϵ-heavy for B. Intuitively, a heavy p is a fingerprint that causes many collisions, and u is poor if many of its fingerprints produce many collisions.

The relevant property of extractors is presented in the next lemma. The point is that an ϵ-poor string is difficult to handle because a random fingerprint of it produces many collisions. The lemma gives a criterion which guarantees that a string is not ϵ-poor.

▶ **Lemma 5.** *There exist constants $\lambda > 1$ and c with the following property:*

Let $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a $(k-c, \epsilon)$ extractor computable in space $S(n)$ (in the above sense). Let x be an n-bit string (which in the protocol is Alice's input) and y be a string (which is Bob's input), such that $C^{S(n)}(x \mid y, n, k) \leq k$ and $C^{\lambda S(n)}(x \mid y, n, k-1) > k-1$, and let $B = \{u \in \{0,1\}^n \mid C^{S(n)}(u \mid y, n, k) \leq k\}$. Then x is not ϵ-poor for B.

Proof. Let A be the set of strings that are ϵ-heavy for B. By counting the edges between B and A from left-to-right and from right-to-left, we obtain that $|A|/M \leq \epsilon$. Let POOR be the set of nodes that are ϵ-poor for B. Note that

$$\mathrm{Prob}(\mathrm{E}(\mathrm{U_{POOR}}, \mathrm{U_d}) \in \mathrm{A}) > 2\epsilon \geq |\mathrm{A}|/\mathrm{M} + \epsilon,$$

It follows that POOR has size less than 2^{k-c}, because otherwise the set POOR would violate the property that E is a $(k-c,\epsilon)$-extractor.

Given y, n, k, c, the set POOR can be enumerated using space $S(n)+O(n)$ (we need the second term to maintain several counters which require $O(n)$ space). Taking into account the additional space needed by the universal machine, it follows that for some constant λ, if u is an ϵ-poor node then

$$C^{\lambda S(n)}(u \mid y, n, k, c) \leq k - c + O(1),$$

which implies $C^{\lambda S(n)}(u \mid y, n, k-1) \leq k-1$, for sufficiently large c. It follows that x is not ϵ-poor, which proves the lemma. ◀

We need to use a *prefix extractor*, which is a a function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^n$ with the property that for every $k \leq n$, the function E_k obtained by retaining only the prefix of length k of $E(x,w)$ is a (k,ϵ) randomness extractor. Raz, Reingold and Vadhan [12] have obtained an explicit extractor E_{RRV} of this type with $d = O(\log^3(n/\epsilon))$. $E_{\text{RRV}}(x,w)$ can be computed in time polynomial in n (recall that $n = |x|$). Let $p_0(n)$ be the polynomial that bounds the *space* used in the computation of $E_{\text{RRV}}(x,w)$.

In the protocol, we use the Raz-Reingold-Vadhan prefix extractor E_{RRV}. We denote by E_k, the k-prefix of E_{RRV}, and, abusing notation, also the bipartite graph corresponding to the (k,ϵ) extractor E_k.

In addition to E_{RRV}, we use a hash function h, based on congruences modulo prime numbers. We view a string x as an integer (in some canonical way) and define $h_t(x) = (x \bmod q, q)$, where q is a prime number chosen at random among the first t prime numbers. The properties of h_t follow from the following lemma.

▶ **Lemma 6** ([2]). *Let $x_1, x_2 \ldots, x_s$ be distinct n-bit strings, which we view in some canonical way as integers $< 2^{n+1}$. Let $t = (1/\epsilon) \cdot s \cdot n$. Let q be a prime number chosen uniformly at random among the first t prime numbers. Then, with probability $(1-\epsilon)$,*

$$x_1 \bmod q \notin \{x_2 \bmod q, \ldots, x_s \bmod q\}.$$

We now present the protocol. Recall that at the beginning of the protocol, Alice holds an n-bit string x, and Bob holds a string y. We fix the parameters as follows. Let λ and c be the constants guaranteed by Lemma 5, let $s = (1/\epsilon) \cdot 2^{c+1} \cdot D$, where $D = 2^d = 2^{O(\log^3(n/\epsilon))}$ is the degree of the E_{RRV} extractor, and let $t = (1/\epsilon) \cdot s \cdot n^2$. We use the space bound $S(n)$ and the constant $\lambda > 1$, given by Lemma 5 applied to the E_{RRV} extractor. We assume that the polynomial p_0 and the constant c, promised by Lemma 5, are large enough so that for every string x and every condition string u, $C^{p_0(|x|)}(x \mid u) \leq |x| + c$. As we did earlier, we use the abbreviated notation $CS^{(i)}(\ldots)$ for $C^{\lambda^i \cdot S(n)}(\ldots)$.

Phase 1: Information reconciliation. In Round 0, Alice sends to Bob, n and $h_t(x)$, where h_t is the hash function introduced above.

Next, Alice computes $p' = E_{\text{RRV}}(x,w)$ for a random $w \in \{0,1\}^d$.

Alice sends to Bob the string p' (or rather a prefix of it), one bit per round, till Bob announces that he does not need more bits.

Suppose we are at round k, after Alice has sent the k-th bit of p'. Thus, by now Bob has received p_k, the k-th bit long prefix of p'. He calculates, as we explain next, a set of candidate strings, which he thinks might be x. A string x' is a candidate at round k if

1. $x' \in B = \{u \in \{0,1\}^n \mid CS^{(n-k)}(u \mid y, n, k+c) \leq k+c\}$, and
2. x' is a neighbor of p_k, when viewing x' as a left node and p_k as a right node in the graph E_k, and

3. x' is among the first (in some canonical order) s strings with the above two properties. If no candidate has the fingerprint $h_t(x)$, then Bob asks for the next bit of p'. Otherwise, there is one candidate string x' so that $h_t(x') = h_t(x)$. Then Bob believes that x' is Alice's x, and he responds to Alice that he does not need further bits. The Phase 1 (information reconciliation) of the protocol is over.

We now analyze Phase 1 (information reconciliation). We show that with high probability, at the end of Phase 1, Bob obtains x.

Let $k^* = \min\{k \mid CS^{(n-k)}(x \mid y, n, k+c) \leq k+c\}$. By the above largeness assumptions for c and $p_0(n)$, it follows that $k^* \leq n$. Let $\mathcal{E}$ be the event that there exists x' other than x that is a candidate at one of the rounds $1, 2, \ldots, k^*$ and has the same fingerprint as x (i.e., $h_t(x') = h_t(x)$). The total number of candidates from rounds $1, 2, \ldots, k^*$ is at most $k^* \cdot s \leq n \cdot s$. It follows from Lemma 6, that $\mathcal{E}$ has probability at most ϵ. Conditioned on $\mathcal{E}$ not holding, either Bob finds correctly x before round k^* (this happens if x is a candidate at one of these earlier rounds), in which case we are done, or Phase 1 reaches round k^*.

Suppose Phase 1 reaches round k^*. Let $B = \{u \in \{0,1\}^n \mid CS^{(n-k^*)}(u \mid y, n, k^*+c) \leq k^*+c\}$. Clearly, by the definition of k^*,

$$CS^{(n-k^*)}(x \mid y, n, k^*+c) \leq k^*+c$$

and

$$CS^{(n-k^*+1)}(x \mid y, n, k^*+c-1) > k^*+c-1.$$

Now we use Lemma 5 for the pair (x, y), the (k^*, ϵ) extractor $E_{k^*} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{k^*}$ and the set B. The size of B is less than 2^{k^*+c+1} and the average B-degree of a right node is $avg = |B| \cdot D/2^{k^*} \leq 2^{c+1} \cdot D$. By Lemma 5 and the two inequalities above, x is not ϵ-poor, which means that with probability $1-2\epsilon$, p_{k^*} is a right neighbor of x that is not heavy, i.e., it has at most $(1/\epsilon) \cdot avg \leq (1/\epsilon) \cdot 2^{c+1} \cdot D = s$ neighbors in B. Therefore, conditioned on non $\mathcal{E}$, with probability $1-2\epsilon$, x is a candidate at round k^*, and Bob finds it. We conclude that with probability larger than $1-3\epsilon$, Bob correctly obtains x.

Let p be the part of the protocol's transcript that Alice has sent to Bob. For the analysis of Phase 2, we need to evaluate the length of p. The string p consists of n, $h_t(x)$ and the prefix of p' that Alice has sent bit-by-bit before Bob told her that he does not need any further bits. By the analysis above, with probability $1-3\epsilon$, the length of the prefix of p' is at most k^*. Let $k = C^{S(n)}(x \mid y) - c$. Note that $k \leq n$. Since

$$CS^{(n-k)}(x \mid y, n, k+c) \leq CS^{(0)}(x \mid y) = k+c,$$

it follows from the definition of k^* that $k^* \leq k$. Next, the length of n and $h_t(x)$ is $O(\log^3(n/\epsilon))$ because the t-th largest prime number is less than $t \log t$. We conclude that

$$|p| \leq^+ CS^{(0)}(x \mid y). \tag{10}$$

The communication complexity is $2|p|$, because it consists of p and of Bob's responses $00\ldots01$.

Phase 2: Secret key construction. Alice and Bob compute by exhaustive search from x and p a program z of x given p in space $S(n)$ of minimal length $CS^{(0)}(x \mid p)$.

We now show that the protocol satisfies the requirements of Theorem 2, and we start with part (i). We let $\geq^+$ hide a loss of precision of $O(\log^3(n/\epsilon))$. We have

$$\begin{aligned} CS^{(0)}(x) &\leq^+ |p| + |z| && \text{(because } x \text{ is computed from } p \text{ and } z \text{ in space } S(n)) \\ &\leq^+ CS^{(0)}(x \mid y) + |z| && \text{(by (10))} \end{aligned}$$

Hence, $|z| \geq^+ CS^{(0)}(x) - CS^{(0)}(x \mid y)$.

Next we show part (ii) in Theorem 2. First notice that, by the chain rule,

$$|z| = CS^{(0)}(x \mid p) \leq^+ CS^{(-1)}(x,p) - CS^{(0)}(p). \tag{11}$$

Next,

$$\begin{aligned} CS^{(0)}(z \mid p) &\geq^+ CS^{(1)}(z,p) - CS^{(0)}(p) \\ &\qquad \text{(chain rule)} \\ &\geq^+ CS^{(1)}(x,p) - CS^{(0)}(p) \\ &\qquad \text{(because } x \text{ can be computed from } z \text{ and } p \text{ in space } S(n)) \\ &= CS^{(-1)}(x,p) - CS^{(0)}(p) - (CS^{(-1)}(x,p) - CS^{(1)}(x,p)) \\ &\geq^+ |z| - \Delta, \end{aligned}$$

where $\Delta = CS^{(-1)}(x,p) - CS^{(1)}(x,p)$. Since p can be computed from x and the seed of the extractor and the random prime number q used by h_t in space $p_0(n) \leq S(n)$, we have

$$\Delta \leq^+ CS^{(0)}(x) - CS^{(1)}(x).$$

The transcript of the protocol consists of p and Bob's sequence of responses $00\ldots01$, which has complexity bounded by $\log n$. Therefore

$$CS^{(0)}(z \mid \mathit{transcript}) \geq^+ CS^{(0)}(z \mid p) \geq^+ |z| - \Delta,$$

which proves part (ii) of Theorem 2. ◀

5 Final comments

As we have mentioned in the Introduction, the main results are of theoretical, rather than practical, relevance. The secret key agreement protocols in Theorem 1 and Theorem 2 produce a key that looks random to an adversary whose computation is space-bounded by $S(n)$, and, on the other hand, in both theorems, the two legal parties (i.e., Alice and Bob) execute the protocol in space larger than $S(n)$. For this reason, the protocols do not seem to be suitable for real cryptographic applications.

Another observation regards the key length. In Theorem 2, the protocol, on inputs the n-bit string x and the string y, runs in space bounded by $\lambda^n S(n)$ (we take into account the space used by the two parties combined) for some constant $\lambda > 1$ and produces a secret key z of length $|z| \approx C^{S(n)}(x) - C^{S(n)}(x \mid y)$ and having the randomness deficiency of z conditioned by the trancript as stated in the theorem. Recall that the randomness deficiency Δ is defined by $\Delta = |z| - C^{S(n)}(z \mid \mathit{transcript})$. Is the length of z optimal? It is known from [13], that no computable protocol can produce a key longer than $C(x) - C(x \mid y)$, the mutual information of the inputs hold by the two parties. We have not been able to obtain a similarly clean result for protocols that run in space $S(n)$. By adapting the arguments in [13], it can be shown, that if a protocol runs in space $S(n)$ then, for every pair of inputs (x,y) with length bounded by n, it produces a key z with $C^{\lambda^2 S(n)}(z \mid \mathit{transcript}) \leq C^{S(n)}(x) - C^{\lambda^3 S(n)}(x \mid y)$, for some constant $\lambda > 1$. Thus we obtain the following upper bound: If a secret key agreement protocol runs in space $S(n)$ and on input (x,y), with $|x|, |y| \leq n$, it produces a secret key z with randomness deficiency Δ, then

$$|z| \leq C^{S(n)}(x) - C^{\lambda^3 S(n)}(x \mid y) + \Delta + \Delta_1,$$

where $\Delta_1 = C^{(S(n)}(z \mid \mathit{transcript}) - C^{\lambda^2 S(n)}(z \mid \mathit{transcript})$ and $\lambda > 1$ is a constant.

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 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. doi:10.1109/CCC.2014.32.

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 Mark Braverman and Anup Rao. Information equals amortized communication. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 748–757. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.86.

5 Harry Buhrman, Michal Koucký, and Nikolai K. Vereshchagin. Randomised individual communication complexity. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity, CCC 2008, 23-26 June 2008, College Park, Maryland, USA*, pages 321–331. IEEE Computer Society, 2008. doi:10.1109/CCC.2008.33.

6 Alexander Kozachinskiy. On Slepian-Wolf theorem with interaction. *Theory Comput. Syst.*, 62(3):583–599, 2018. doi:10.1007/s00224-016-9741-x.

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

8 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.

9 Andrei A. Muchnik. Conditional complexity and codes. *Theor. Comput. Sci.*, 271(1-2):97–109, 2002. doi:10.1016/S0304-3975(01)00033-0.

10 D. Musatov, A. E. Romashchenko, and A. Shen. Variations on Muchnik's conditional complexity theorem. *Theory Comput. Syst.*, 49(2):227–245, 2011. doi:10.1007/s00224-011-9321-z.

11 Prakash Narayan and Himanshu Tyagi. Multiterminal secrecy by public discussion. *Foundations and Trends in Communications and Information Theory*, 13(2-3):129–275, 2016. doi:10.1561/0100000072.

12 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. doi:10.1006/jcss.2002.1824.

13 Andrei E. Romashchenko and Marius Zimand. An operational characterization of mutual information in algorithmic information theory. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pages 95:1–95:14, 2018.

14 Salil P. Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1-3):1–336, 2012. doi:10.1561/0400000010.

15 Marius Zimand. Kolmogorov complexity version of Slepian-Wolf coding. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 22–32. ACM, 2017. doi:10.1145/3055399.3055421.