Low Communication Complexity Protocols, Collision Resistant Hash Functions and Secret Key-Agreement Protocols* **

Shahar P. Cohen[1,2][0000−0002−4609−5952] and Moni Naor[1,3][0000−0003−3381−0221]

[1] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel.
[2] shahar.cohen@weizmann.ac.il
[3] moni.naor@weizmann.ac.il

Abstract. We study communication complexity in computational settings where bad inputs may exist, but they should be hard to find for any computationally bounded adversary.
We define a model where there is a source of public randomness but the inputs are chosen by a computationally bounded adversarial participant *after seeing the public randomness*. We show that breaking the known communication lower bounds of the private coins model in this setting is closely connected to known cryptographic assumptions. We consider the simultaneous messages model and the interactive communication model and show that for any non trivial predicate (with no redundant rows, such as equality):

1. Breaking the $\Omega(\sqrt{n})$ bound in the simultaneous message case or the $\Omega(\log n)$ bound in the interactive communication case, implies the existence of distributional collision-resistant hash functions (dCRH). This is shown using techniques from Babai and Kimmel [BK97]. Note that with a CRH the lower bounds can be broken.
2. There are no protocols of constant communication in this preset randomness settings (unlike the plain public randomness model).

The other model we study is that of a stateful "free talk", where participants can communicate freely *before* the inputs are chosen and may maintain a state, and the communication complexity is measured only afterwards. We show that efficient protocols for equality in this model imply secret key-agreement protocols in a constructive manner. On the other hand, secret key-agreement protocols imply optimal (in terms of error) protocols for equality.

1 Introduction

What does a lower bound mean if it is not feasible to find the bad inputs? In other words, can we bypass it, if we assume that the choice of inputs is done by

* Research supported in part by grants from the Israel Science Foundation (no. 2686/20) and by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness. The second author is incumbent of the Judith Kleeman Professorial Chair.
** The full version of this paper is available at ia.cr/2022/312.

a process that is computationally limited? In this work we study this issue in the setting of communication complexity.

The study of communication complexity deals with proving bounds on the amount of communication that is required to perform certain tasks when the input is separated: two parties, Alice and Bob, have as inputs $x \in X$ and $y \in Y$ respectively; how many bits do they have to send each other (as a function of $|X|$ and $|Y|$) for computing the value of a function $f(x, y)$? An answer for such a question depends, of course, on the exact model details: Do Alice and Bob have any limitation in the communication? Do they communicate directly or through a third party? What predicates f are they trying to compute? See Kushilevitz and Nisan [KN96] and Rao and Yehudayoff [RY20] for extensive background on communication complexity.

There are several models of communication that differ mainly on two properties: whether the strategy of the participants can be probabilistic and the exact communication settings (network layout). The participants of those models do not have a bound on their running time, however, they are required to be correct[4] for every input in the space.

When the participants are allowed to be probabilistic there is an important distinction: whether they share common random bits (public coins) or not (private coins). It is important to note that the random bits (in both options) and the problem's inputs (x and y) are *independent*. This can be seen as uniform random bits that are chosen *after* the (worst-case) input was chosen.

By definition, the private coins model is no stronger than the public coins model and indeed some tasks can be done in the latter but cannot be done in the former with the same communication complexity (see below). On the other hand, the private coins model may be considered more *realistic*, where there is no assumption of *independent* public random string.

However, both the public and private coins models are known to be 'better' than the deterministic model in the sense that they have more efficient protocols in terms of the communication complexity: for instance, as proved by Yao, the deterministic communication complexity of many predicates is $\Omega(n)$ (Alice and Bob can do nothing better than just sending their full inputs), while in the probabilistic world there is quite a lot to be done. Equality is a prominent example, with complexity of $\Theta(\log n)$ for the private coins model and $\Theta(1)$ for the public coins model.

We examine another relaxation that can help us: limiting all parties, including the one who selects the inputs (the adversary), to a computationally bounded world. We will not require that Alice and Bob be correct for every input in the space, but only on inputs that are chosen by a computationally bounded adversary. Note that the new definition is by nature relevant only to probabilistic algorithms.

Considering a polynomially bounded adversary raises the question of whether there are benefits from different computational hardness assumptions: *can we re-*

[4] For the probabilistic version they are required to succeed with constant high probability.

duce the communication complexity of protocols by assuming that certain tasks cannot be performed efficiently? That is, given that Alice and Bob in our new definition do not have to be correct for every input in the input space, a computational hardness assumption can be used for proving that no efficient adversary can find bad inputs with non-negligible probability.

Back to the relationship between the public and private coins models: we propose a new model that is, in general, more powerful than the private coins but still realistic. Also, in contrast to the above mentioned models, our model is *computational* – the participants' running time is bounded by some poly(λ) where λ is the security parameter. In our model, there is a public random string but there is an additional adversarial participant that *chooses the inputs depending on the public random string*, it can be seen as a public random string that is 'fixed' in advance and therefore we called it *preset public coins* model. See Definition 2.7 for formal specification.

The two communication patterns we consider are:

Simultaneous Messages Model (SM). Alice and Bob are given x and y respectively and should compute a function $f(x, y)$ but *without communicating with each other*. Instead, each one sends a message to a third party (a referee) who calculates $f(x, y)$ given the messages from Alice and Bob.

Interactive Communication Model. Alice and Bob get their inputs x and y respectively and can communicate with each other without any limitations on the number of rounds.

In both settings, the communication complexity measure is the total length of the messages sent by Alice and Bob.

Stateful preprocessing communication. The second type of model we consider is where the communication complexity matters *only at some critical period* and the question is whether we can get very succinct protocols. The two parties can talk freely beforehand. At some point the action starts, they receive their inputs and need to decide with little communication the result.

In the SM model we consider a variation that differs by two properties:

Free talk. A protocol with *free talk* is one where Alice and Bob communicate also before getting their inputs. The messages during the free talk phase (before the inputs are chosen) do not count in the communication complexity of the protocol. Alice and Bob maintain (secret) states afterwards. However the adversary sees the whole communication and can use it when choosing the inputs.

Rushing adversary. The inputs are chosen by a computationally bounded adversary depending on the public discussion it witnesses in the preprocessing phase. A rushing adversary can choose Bob's input at the 'last moment': The adversary first chooses the input of Alice depending on the public random string and *after* Alice sends her message to the referee the adversary chooses the input of Bob depending on both the preprocessing transcript and on *and Alice's message.*

Note: while allowing "rushing" gives the adversary more power, it is not an unreasonable model, e.g. in a sketching environment, when the adversary sees how one party 'sketched' its input and may then select the inputs to the other one. We do not know whether we need this power in order to get the result that very succinct protocols imply secret key agreement.

1.1 Cryptographic Primitives

We discuss a computationally bounded world. We assume that all parties have limited resources (especially at runtime). A way to express those limits is by cryptographic primitives (see the next examples).

Necessity of Primitives. One of the aims of research in foundations of cryptography is to find out which cryptographic primitives are essential and sufficient for which tasks. Similarly, it is valuable to know whether certain primitives on their own *cannot* help us achieve a certain goal.

In this paper we prove several implications of the form that the existence of communication protocols with certain properties entails the existence of certain primitives. In other words, in order to design succinct protocols in those models we must be using somewhere in the protocol primitives of a certain kind.

1.2 Cryptographic Hash Functions

A hash function is one that maps values from a large domain to a smaller range. One of the most basic cryptographic objects is a hash function with some hardness property. For instance, a family of hash functions $\mathcal{H}$, is collision resistant if for a random $h \in_R H$ it is hard to find two inputs $x \neq y$ that collide $(h(x) = h(y))$.

For such a function, for any two inputs that were chosen by a computationally bounded adversary, we know that w.h.p., $h(x) = h(y) \implies x = y$. This means that the function *preserves* some relation between its inputs: The equality predicate is (w.h.p.) preserved also after the values were compressed by h. Moreover, since the function is collision resistant, that property holds for any (x, y) chosen by a computationally bounded adversary knowing h.

This notion can be generalized in several directions:

1. More relaxed hardness requirements can be defined. The weaker the definition the more hope we have to construct it from minimal assumptions.
2. We can extend the definitions to include random algorithms: functions that get also random bits and output correct values w.h.p.[5].
3. We can extend the definitions to hash functions that preserve more properties and not just the equality predicate.

We discuss the last two points in the section below.

[5] The probability is over the choices of the random bits.

1.3 Adversarially Robust Property-Preserving Hash Functions

Consider a predicate $P\colon U \times U \to \{0,1\}$ for a universe $U = \{0,1\}^n$. Let $x, y \in U$ and we want to compute $P(x,y)$, but we cannot have both x, y on the same machine (say, for some storage reasons). A natural approach for this issue is using *sketching*: By using sketches we get shorter strings and it is easier to get both (sketched) values on the same machine. Of course, computing P on sketched values may be impossible in terms of information, so we relax the correctness requirement: the process may fail (compute a wrong value) with at most a negligible probability[6].

Hash functions as above, that allow us to compute a predicate given the hashed values, are called *property-preserving* hash functions (hereafter PPH). We examine PPH in an *adversarial* environment, that is, the predicate should be computed correctly w.h.p. also for values chosen by an adversary. Such hash functions are called *adversarially robust* PPH. The more access to the hash function given to the adversary the more robust the PPH is.

The study of adversarially robust property-preserving hash functions was initiated by Boyle et al. [BLV18]. It can be seen as a special case of the model introduced by Mironov et al. [MNS11] who initiated the study of the *adversarial sketch model* (here the participants also get the input *online*). That notion is similar to the SM model except for the differences:

1. The allowed error probability in communication complexity is a (small) constant instead of negligible in PPHs.
2. The parties in the SM model are allowed to be randomized.
3. The PPHs model is computational.

Our model bridges some of the gaps and we will show the connection between the models. Note that the preset public coins SM model is a generalization of the PPHs model in the sense that the participants are allowed to be randomized. In this regards it is closer to the model of Mironov et al. [MNS11].

1.4 Secret Key Agreement

A secret key agreement (SKA) is a protocol where two parties with no prior common information agree on a secret key. The key has to be secret in the sense that no probabilistic polynomial time adversary given the full transcript of the communication between Alice and Bob can compute it with non-negligible probability (more accurately, distinguish it from a random string). That notion is defined formally in Definition 2.19.

We will show that certain low communication protocols imply the existence of SKA by showing a construction of SKA from those protocols.

[6] The probabilities are over the sampling of a hash function among the functions family.

1.5 Our Results

We consider preset public coins communication complexity models and prove that the lower bounds proved for the private coins model cannot be broken in our computational model without assuming the existence of distributional CRHs (dCRH is a hash function where uniformly random collisions cannot be found by a bounded adversary w.h.p., see Definition 2.16) It is known that dCRHs exist only if one-way functions exist and there is an oracle separation between them (i.e. there are no black-box constructions of dCRHs from one-way functions).

A non-trivial predicate is one with no redundant rows and columns (see Definition 2.1)

Theorem (informal, see Theorems 3.2 and 3.14). In the preset public coins Simultaneous Message model: for any non-trivial predicate, protocols with communication complexity $o(\sqrt{n})$ imply the existence of dCRHs (in the sense that a dCRH can be constructed from the protocol).

In the interactive model: The same is true for $c(n) = o(\log n)$.

Gap from upper bound. We note that such succinct protocols are achievable using a CRH. Closing the gap between CRH and dCRH is left as an open problem (see conclusions).

Consider the free talk model, where two parties communicate and may have a secret state as a result, *before* the inputs are chosen based on an eavesdropper adversary who has access to the communication but not to the secret states. If secret key agreement protocols exist, then we can get the power of the public coins model: we can construct a protocol for the equality predicate with error probability bounded by 2^{-c} where c is the communication complexity.

In the other direction, nearly optimal protocols imply a secret key agreement:

Theorem (informal, see Theorem 5.2). In the stateful free talk model, the existence of a protocol of complexity $c(n)$ for equality with failure probability bounded by $\varepsilon \leq 2^{-0.7c}$ against a rushing adversary implies the existence of a secret key agreement protocol.

Again here there are gaps between the possibility and impossibility results. For the implication we do need a low error protocol (with respect to the communication complexity) and also we do not know whether a rushing adversary is essential.

The various implications we showed are summarized in Table 1.

On the other hand, regardless of assumptions, constant communication protocols cannot exist in our model. This is in contrast to the public coins model, where there are protocols of $O(1)$ communication even in the SM model (e.g. for the equality predicate).

Theorem (informal, see Theorem 4.1). In the interactive communication model, protocols for any non trivial predicate, of communication complexity $O(\log \log n)$ bits are not secure against an adversary with running time $\text{poly}(n)$.

Table 1. Summary of Implications and Results

		Information-Theoretic Lower Bound	Comput. Bounded World: Breaking The Bound	
			Possible Using	Implies
Stateless	SM	$\Omega(\sqrt{n})$	CRH	dCRH
	Interactive	$\Omega(\log n)$	CRH	dCRH
Stateful (Rushing $\mathcal{A}$dv)	SM	$\Omega(n)$	SKA	For Equality: SKA*

* Holds only for near optimal protocols.

1.6 Related Work

Communication Complexity

The study of communication complexity was initiated by Yao [Yao79] who introduced the SM private coins model and asked what is the complexity of the equality predicate in this model. The problem was solved by Newman and Szegedy [NS96] who provided the $\Omega(\sqrt{n})$ tight lower bound. It was also solved, using different and simpler techniques, by Babai and Kimmel [BK97][7] using a combinatorial proof, and by Bottesch et al. [BGK15] using information theory[8].

Babai and Kimmel's result is more general and they actually proved the lower bound not only to the equality predicate but to any non-redundant predicate (see Definition 2.1). Moreover, their technique proved to be useful in more models: Ben-Sasson and Maor [BM15] applied this technique also for the interactive model and proved that for any non redundant function, any private coins protocol requires communication complexity of at least $\Omega(\log n)$ (see proof for the equality predicate in Kushilevitz and Nisan [KN96]).

Although the above mentioned results are in the information-theoretic world (can be seen as an unbounded adversary), Naor and Rothblum [NR09] introduced and studied a computational model in order to study online memory checking algorithms: The consecutive messages model where the public coins are chosen *after* the adversary chooses x (the input for Alice). They adapted this technique and showed that breaking the mentioned information-theoretic $\Omega(\sqrt{n})$ lower bound in their computational model is possible if and only if one-way functions exist. At first glance one can think that their model is very close to our preset public coins SM model. However, important details differ: For instance, the fact that x (Alice's input) does not depend on the public random string.

Public Coins vs. Private Coins In certain ways our model lies between the public and private coins ones. Therefore, it is worth pointing out the possible

[7] See in [BK97] also the similar proof of Bourgain and Wigderson.

[8] Bottesch et al. actually discuss quantum variants of the SM model and give the simpler proof for our classical case as a warm-up.

gap between them. For the interactive communication settings, Newman [New91] proved that the gap can be at most $O(\log n)$ additively. It is tight, since the equality predicate can be computed by protocols of $O(1)$ communication in the public coins model but requires $\Theta(\log n)$ bits in the private coins model.

In the SM model, as mentioned, the gap may be much larger: the equality predicate can be computed using $O(1)$ bits in the public coins model, but in the private coins model $\Omega(\sqrt{n})$ bits are required.

Settings where worst-case inputs are hard to find

One area where computationally bounded choices of inputs was considered is error correcting codes. Here assuming the channel is computationally bounded may help and better rates than those achievable by codes for worst-case errors are possible. Such works were done by Lipton [Lip94] and Micali et al. [MPSW10]. These constructions required a trusted setup with a key that should not leak. Grossman et al. [GHY20] suggested "good" uniquely decodable codes for the computationally bounded channel with transparent setup. Grossman et al. relied on strong cryptographic assumptions to construct a code better than codes for worst-case errors.

Harsha et al. [HIKNV04] studied tradeoffs between communication complexity and time complexity and described Boolean functions with a strong communication vs. runtime tradeoff.

1.7 Technical Overview

Babai and Kimmel's Characterizing Multiset. We will use the technique of Babai and Kimmel for proving connections between the communication complexity and cryptographic primitives in both models (SM and interactive). They proved that in the SM model, Alice's behavior can be characterized by a relatively small multiset of messages. Ben-Sasson and Maor expanded it for the interactive model and proved that Alice's behavior can be characterized by a multiset of *deterministic strategies*.

We use those observations and show that the adversary can use the characterizing multisets to find bad inputs for Alice and Bob. That is, we construct a function that for any x (Alice's input) generates a characterizing multiset of the behavior of Alice for this x. We claim that an adversary who can break the security of this function, can find bad inputs for the protocol. On the other direction, if such a protocol exists it implies the existence of a certain cryptographic primitive.

In more details: We show a construction of a function that for an Alice's input ($x \in X$) outputs a multiset that characterizes Alice's behaviour. We claim that a collision in such function induces bad inputs for the protocol as it implies two inputs that make Alice behaves similar. That is, the correctness of the protocol implies the security of the function. However, the construction is probabilistic and the function does not output a characterizing multiset for every inputs but only for at most every input. Hence, it may not be collision resistant but only

one-way function. Moreover, we show it also a distributional collision resistant since it outputs a characterizing for at most every input.

2 Models and Preliminaries

2.1 Model Definition

Let f be a predicate that Alice and bob would like to compute. For a predicate f to be interesting we may assume that the f has no redundancy:

Definition 2.1 (Non-Redundant Predicate). *Predicate $f\colon X \times Y \to \{0,1\}$ is non-redundant if there are no two identical rows or two identical columns in the truth matrix. In other words, $\forall x_1 \neq x_2 : \exists y$ s.t. $f(x_1, y) \neq f(x_2, y)$ and for $\forall y_1 \neq y_2$ as well.*

Also, we discuss only predicates where their non-redundancy can be 'proven' or found efficiently:

Definition 2.2 (Efficiently Separable Predicate). *Let $f\colon X \times Y \to \{0,1\}$ be a non-redundant predicate, then f is efficiently separable if there exists PPTM $\mathcal{M}$ that finds the element promised by Definition 2.1. That is $\forall x_1 \neq x_2 \in X$:*

$$\Pr_{y \leftarrow \mathcal{M}(x_1, x_2)} [f(x_1, y) \neq f(x_2, y)] = 1 - \mathrm{negl}(n)$$

and similarly for $\forall y_1 \neq y_2 \in Y$ as well.

Note 2.3. It is not clear that a Non-Redundant Predicate that is *not* Efficiently Separable implies the existence of one-way functions. In fact, it seems that hard problems in NP ("Pessiland") already implies the existence of such predicates, and hence as far as we know it is not a sufficient assumption for mounting meaningful cryptography.

The only specific predicate we discuss is the equality predicate, $EQ(x, y) = \mathbb{1}_{\{x=y\}}$. For the equality predicate it is easy to see that both Definitions 2.1 and 2.2 hold.

Now, we define the communication layouts:

Definition 2.4 (Interactive Communication Model). *Alice and Bob are given x and y respectively and should compute some function f. They may send each other messages without any limit (but the total number of bits sent is the complexity).*

Definition 2.5 (Simultaneous Messages (SM) Model). *In the simultaneous messages model, Alice and Bob are given x and y respectively and should compute some function f* without communicating with each other*. Instead, each one sends a message to a third party (a referee) who calculates $f(x, y)$ given the messages from Alice and Bob.*

Following Babai and Kimmel we assume without loss of generality that the referee is deterministic.

Fact 2.6. *In the SM model there exist protocols for the equality predicate of complexity $O(\sqrt{n})$. The protocols found independently by Ambainis, Babai and Kimmel, Naor and Newman; see [BK97] for references.*

Fact 2.6 is an example of the possible gap between probabilistic and deterministic protocols in the SM model because the equality predicate is non-redundant and because of the following well known fact:

Fact. In the SM model, the deterministic communication complexity of any non-redundant predicate is $\Omega(n)$.

Now, we are ready to define our model formally, in the above described communication layouts. Recall that our model is *computational*. That is, the participants' running time is bounded by some $\text{poly}(\lambda)$ for some security parameter $\lambda = \text{poly}(n)$, it's important especially for the adversarial participant. That is, any PPTM run time is bounded by $\text{poly}(\lambda)$.

Definition 2.7 (Preset Public Coins). *A protocol for a function f in the the preset public coins is defined by the following game: Let Alice and Bob be PPTMs with running time* $\text{poly}(\lambda)$.

1. *A public uniform random string r_{pub} is sampled*[9].
2. *The adversary sees r_{pub} and chooses $(x, y) \in X \times Y$.*
3. *Alice and Bob get (x, r_{pub}) and (y, r_{pub}) respectively.*
4. *Alice and Bob send message(s) (optionally using private coins) in order to compute some target function.*
5. *Optionally: More steps that depend on the communication settings. For instance, in the SM model the referee steps in here.*

We say that a protocol is ε-secure in this model if for every PPTM adversary $\mathcal{A}\text{dv}$ with running time $\text{poly}(\lambda)$ the probability that the computation of Alice and Bob will be correct is at least $1 - \varepsilon$*:*

$$\Pr_{\substack{r_{pub} \\ (x,y)\leftarrow\mathcal{A}\text{dv}(r_{pub}) \\ \textit{Alice and Bob private coins}}} [\textit{Protocol Fails}] \leq \varepsilon$$

Amplification. The usual requirements in communication complexity is for $\varepsilon = 1/3$ and then to argue for amplification by repetition. Here we have to be a bit more careful, since the correctness requirement is computational. However, we know that for games of a certain structure we get parallel amplification: Bellare et al. [BIN97] showed that parallel repetition of computationally sound protocols doesn't always lower the error as one may expect. However, they proved that for three-round protocols the error does go down exponentially fast[10] as in the information-theoretic case.

A protocol in our model can be evaluated in the following 3 rounds:

[9] Can be generalized to a sample from any known efficient distribution.
[10] See Canetti et al. [CHS05] for better parameters.

1. The prover chooses the public random string.
2. The adversary chooses the inputs of Alice and Bob.
3. The prover chooses Alice's and Bob's private random string.

Note 2.8. Alice's and Bob's algorithms are public and since they don't have any secret state or secret input they can be simulated (also by the adversarial participant). It is a core fact when we are using the technique of Babai and Kimmel in computational settings in the proofs of Theorems 3.2 and 3.14 and Theorem 4.1.

2.2 Free Talk Model

We consider a variation to the SM model where Alice and Bob are allowed to communicate freely in a preprocessing phase, before the inputs are chosen:

Free talk. Free talk is a 'free' communication that Alice and Bob can have before the inputs are chosen by the adversary. Alice and Bob can generate states (possibly secret) in the free talk phase. Those states can be used afterward to reduce the communication complexity.

However, the adversary is also stronger, in two ways:

Free Talk Eavesdropping. The transcript of the free talk phase is known to the adversary and it may choose the inputs depending also on it.

Rushing. Rushing adversary decides Bob's input at the 'last moment': Rushing adversary chooses the input of Bob *after* Alice produces its message. That is, first Alice's input is chosen and Alice sends its message, and afterwards, Bob's input is chosen depending on Alice's message and Bob sends its message.

Definition 2.9 (SM Preset Public Coins With Stateful Free Talk and Rushing Adversary). *A protocol for a function f in the the SM Preset Public Coins With stateful free talk and rushing adversary is defined by the following game: Let Alice and Bob be PPTMs with running time* $\mathrm{poly}(\lambda)$.

1. *Alice and Bob toss coins and communicate in order to generate their (possibly secret) states τ_A and τ_B respectively.*
2. *The adversary sees their full* communication *(but not their internal states τ_A and τ_B) and sets Alice's input $x \in X$.*
3. *Alice (that has τ_A as her internal state) gets x and sends a message m_A to the referee.*
4. *The adversary sees m_A and chooses Bob's input $y \in Y$, optionally depending on m_A and the free talk's transcript.*
5. *Bob (that has τ_B as his internal state) gets y and sends a message m_B to the referee.*
6. *The referee, as a function of m_A and m_B, computes the target predicate.*

To simplify the description we assume, without loss of generality, that Alice and Bob are probabilistic only in the first step. (This is wlog since they can toss coins in the first step and save them in their private states for later use.)

In the stateful model we consider a scenario with a 'patient' adversary: there are multiple sessions between Alice and Bob and the (rushing) adversary can choose one session to attack among them, after seeing the message Alice choose.

Definition 2.10 (Stateful Free Talk ε-Secure Protocol). *We say that a protocol is ε-secure in the stateful model if for every PPTM adversary $\mathcal{A}$dv with running time* $\mathrm{poly}(\lambda)$ *who involved in* $\mathrm{poly}(\lambda)$ *sessions and chooses among them one session as the defining one (after seeing Alice's message) the probability that the computation of Alice and Bob will be correct for the chosen session is at least* $1-\varepsilon$.

2.3 Notation

Messages Space. Denote by M_A and M_B Alice's and Bob's messages spaces. In the interactive model we consider Alice's and Bob's deterministic strategies, every strategy is represented by a rooted binary tree of depth c (the total communication): The protocol begins in the root, each vertex is owned by one party who chooses one of the children and informs the other party by sending a bit. Finally, the leaves represent the protocol's result. We denote the set of deterministic strategies of Alice and Bob by S_A and S_B respectively.

Private random string. Denote by $r_A \in R_A$ the private random string of Alice.

Public random string. Denote by $r_{\mathrm{pub}} \in R_{\mathrm{pub}}$ the public random string in the protocol (it is given also to the adversary).

Secret State. When Alice and Bob have secret states we denote them by τ_A and τ_B respectively.

Participant. In the SM model, for a public random string r_{pub} denote the strategy of Alice by $A_{r_{\mathrm{pub}}} : X \times R_A \to M_A$ and Bob by $B_{r_{\mathrm{pub}}} : X \times R_B \to M_B$. When the public random string r_{pub} is clear from the context we may omit the subscript. (When Alice and Bob have a secret state we denote Alice and Bob as a function that gets a secret state τ instead of private random string).

Referee. In the SM model denote the referee by a function $\rho_{r_{\mathrm{pub}}} : M_A \times M_B \to \{0,1\}$ for a public random string r_{pub} or ρ when r_{pub} is clear from the context.

Communication Complexity. Denote the length of the total communication by $c = c(n, \lambda)$.

Protocol. We denote the protocol by π, and $\pi(x, y)$ denotes running the protocol on inputs x and y.

2.4 Probability

To measure distance between two distributions we use the *total variation distance*:

Definition 2.11 (Statistical Distance). *Let D_1 and D_2 be two distributions and $D(E)$ be the probability of event E under the distribution D.*

$$\Delta(D_1, D_2) = \max_{Event\ E} |D_1(E) - D_2(E)|$$

We use in our proofs the following concentration lemma:

Lemma 2.12. *Let $X_1, X_2, \ldots, X_t$ be mutually independent random variables where $\mathbf{E}[X_i] = 0$ and $|X_i| \leq 1$. Let $S = \frac{1}{t}\sum_{i=1}^{t} X_i$ then*

$$\mathbf{Pr}[S > \delta] < e^{-\delta^2 t/2}$$

which is a rephrasing of the following Chernoff bound:

Theorem 2.13 (Chernoff Bound [AS08, Theorem A.1.16]). *Let $X_1, \ldots, X_t$ be mutually independent random variables where $\mathbf{E}[X_i] = 0$ and $|X_i| \leq 1$ and let $S = \sum_{i=1}^{t} X_i$. Then*

$$\mathbf{Pr}[S > a] < e^{-a^2/2t}$$

2.5 Collision Resistant Hash Functions

A collision resistant hash function (CRH) is a function that any efficient algorithm has at most a negligible probability of a collision:

Definition 2.14 (CRH). *Let a functions family $\mathcal{H}$ be a family of functions that (1) compress (2) are computable in polynomial time. $\mathcal{H}$ is a family of CRHs if for every polynomial $p(\cdot)$ for every PPTM $\mathcal{A}$dv and large enough λ,*

$$\mathop{\mathbf{Pr}}_{\substack{h \in \mathcal{H} \\ (x,y) \leftarrow \mathcal{A}\mathrm{dv}(h)}} [x \neq y \wedge h(x) = h(y)] < \frac{1}{p(\lambda)}$$

Note that, the output of the function cannot be too small with respect to the security parameter. Otherwise, collisions can be found easily by trying sufficient inputs.

Simon [Sim98] showed that a CRH cannot be built from black-box one-way functions. Since one-way functions are existential equivalent to a lot of basic cryptographic primitives, we know that also they cannot be black-box used to construct CRHs. For an example, see Wee [Wee07] who ruled out constructions for statistically hiding commitments with low round complexity that are based only on black-box one-way functions.

Distributional Collision Resistant Hash Functions Distributional collision resistant hash functions (dCRH) are functions where it is hard for any adversary to generate collisions that *are close to random collisions*. We first have to define an ideal collision finder:

Definition 2.15 (Ideal Collision Finder $\mathcal{COL}$). *The random function $\mathcal{COL}$ gets a description of a hash function h and outputs (x, x') s.t. x is uniformly random and x' is uniformly random from $h^{-1}(x)$. Note that:*

1. *The marginal distribution of x and x' is the same: x and x' are uniformly random (but not independent).*
2. *It is possible that $x = x'$.*

That notion of distributional collision resistance hash functions is due to Dubrov and Ishai [DI06]. However, Bitansky et al. [BHKY19] deviated from this definition and used a stronger definition[11]. Since our results hold also for the stronger definition we will use it:

Definition 2.16 (dCRH). *Let a functions family $\mathcal{H}$ be a family of functions that (1) compress (2) are computable in polynomial time. $\mathcal{H}$ is a family of distributional CRHs if there exists some polynomial $p(\cdot)$ s.t. for every PPTM $\mathcal{A}\text{dv}$, and large enough λ,*

$$\Delta(\mathcal{COL}(h), \mathcal{A}\text{dv}(h)) \geq \frac{1}{p(\lambda)}$$

where $h \leftarrow \mathcal{H}$.

This definition is a generalization of distributional one-way functions[12] and hence implies it. Furthermore, Bitansky et al. showed that dCRHs can be used for applications that one-way functions aren't known to achieve (and are black-box separated) [BHKY19].

Although the notion of dCRH is much weaker than CRH, as noted by Dubrov and Ishai [DI06], the black-box separation result of Simon [Sim98] applied also for dCRH: Its collision finder is the same as $\mathcal{COL}$ in our definition (Definition 2.15). Simon proved that relative to $\mathcal{COL}$ one-way functions exist (although (d)CRHs do not).

2.6 Adversarially Robust Property-Preserving Hash Functions

Here we define the notion of adversarially robust property-preserving hash functions. We follow Boyle et al.'s notion of direct-access robust property-preserving hash functions:

Definition 2.17 (Direct-Access Robust PPHs). *Let a functions family $\mathcal{H}$ be a family of functions that (1) compress (2) are computable in polynomial time and let Eval be a deterministic polynomial time algorithm. $\mathcal{H}$ (with Eval) is a*

[11] By switching the order of quantifiers, they require one polynomial for any adversary and not that for any adversary there exists a polynomial. See the comparison in [BHKY19].

[12] Functions where it is hard to sample uniformly from $h^{-1}(h(x))$ for random x. Such functions are known to exist if and only if one-way functions exist [IL89]. (in contrast to dCRH).

family of Direct-Access Robust PPHs for a predicate $P\colon X \times X \to \{0,1,*\}$ [13] *if for every polynomial* $p(\cdot)$ *for every PPTM* $\mathcal{A}\mathrm{dv}$ *and large enough* λ,

$$\Pr_{\substack{h\in\mathcal{H}\\(x,y)\leftarrow\mathcal{A}\mathrm{dv}(h)}}[P(x,y)\neq *\wedge P(x,y)\neq \mathit{Eval}(h,h(x),h(y))\neq P(x,y)]<\frac{1}{p(\lambda)}$$

2.7 Secret Key Agreement and its Amplification

In a secret key agreement protocol two participants who do not have a common secret, but each one has its own source of randomness, both output a value (the secret). The participants' output has to satisfy two properties: it should be the same value for the two participants (agreement), and it has to be unknown to any efficient observer (secrecy). We follow the definition of Holenstein [Hol05]:

Definition 2.18 ((α,β)-Secret Bit Agreement (SBA)). *An efficient two party protocol without input (aside from the security parameter* λ*), with one bit output for each participant* b *and* b' *respectively where* $b,b'\in\{0,1\}$ *is an* (α,β)*-secret bit agreement if*

$$\mathbf{Pr}[b=b']\geq\frac{1+\alpha}{2}$$

and for every PPTM $\mathcal{A}\mathrm{dv}$ *with running time bounded by* $\mathrm{poly}(\lambda)$

$$\mathbf{Pr}[\mathcal{A}\mathrm{dv}(\tau)=b\mid b=b']\leq 1-\frac{\beta}{2}$$

where τ *is the complete transcript of the protocol.*

The previous definition is a weaker notion of the usually desirable stronger notion:

Definition 2.19 (Secret Key Agreement). *(α,β)-secret bit agreement is a secret key agreement protocol if* $\alpha=1-\mathrm{negl}(\lambda)$ *and* $\beta=1-\mathrm{negl}(\lambda)$.

Holenstein [Hol06] proved when an (α,β)-secret bit agreement can be amplified efficiently to a secret key agreement:

Theorem 2.20 ([Hol06, Corollary 7.5]). *Let efficiently computable functions* $\alpha(\lambda),\beta(\lambda)$*, be given such that*

$$\frac{1-\alpha}{1+\alpha}<\beta$$

Let $\varphi=\max(2,\frac{8}{\log\left(\frac{\beta(1+\alpha)}{1-\alpha}\right)})$ *and* $\gamma=\frac{1}{\log(1+((1-\alpha)/(1+\alpha))^{\varphi})}$*, and assume that* $\frac{\varphi\cdot 2^{4\gamma}}{\alpha}\in\mathrm{poly}(\lambda)$*. If there exists an* (α,β)*-secret bit agreement protocol for all but finitely many* k*, then there exists a computationally secure key agreement.*

Note that a secret key agreement is unlikely to be based (only) on one-way permutations and collision resistant hash functions in a black-box manner: It is known that any secret key agreement protocol in the random oracle model[14] can be broken using an $O(n^2)$ queries attack [IR89; BM09] and this is tight.

[13] $*$ is a don't care symbol, see [BLV18] for a comparison of "total vs. partial predicates".
[14] CRHs exist in this model.

3 Collision Resistance and the Preset Public Coins model

3.1 CRHs imply succinct protocols

We start by noting that the lower bounds shown in Section 3.2 (about the necessity of dCRHs) are almost tight, since using a CRH one can break those bounds ($\Omega(\sqrt{n})$ in the SM model (Theorem 3.2) and $\Omega(\log n)$ in the interactive model (Theorem 3.14)). See the full version for details.

Theorem 3.1. *If CRHs exist, then given a family of CRHs* $\{h\colon \{0,1\}^n \to \{0,1\}^\lambda\}$,

In the preset public coins SM model: *There exist protocols of communication complexity* $O(\sqrt{\lambda})$ *for the Equality predicate.*

In the preset public coins interactive model: *There exist protocols of communication complexity* $O(\log \lambda)$ *for the Equality predicate.*

3.2 Succinct Protocols Imply dCRHs

Theorem 3.2. *Let* $c(n) \leq o(\sqrt{n})$. *Given a protocol for an efficiently separable predicate (Definition 2.2) of complexity* $c(n)$ *in the preset public coins SM model, then distributional CRH functions exist and it is possible to construct them from the protocol.*

Proof. Our intuition is that after fixing the public random string r_{pub}, the model is similar to the private coins SM model where the adversary is faced with a problem defined by the random string. We therefore appeal to Babai and Kimmel's definitions and techniques. Furthermore, in Lemma 3.5 we will also repeat the proof of [BK97, Lemma 2.3] with a different constant and make it constructive.

For each multiset of Alice's messages and one message from Bob we consider the probability of acceptance by the referee:

Definition 3.3 (Referee's Expected Value for a Multiset). *For any* r_{pub}, *for a multiset* T *of members from* M_A *and* $m_B \in M_B$, *let*

$$Q(T, m_B) = \mathop{\mathbf{E}}_{i\in[t]}[\rho_{r_{pub}}(T[i], m_B)] = \frac{1}{t}\sum_{i\in[t]} \rho_{r_{pub}}(T[i], m_B)$$

where $t = |T|$.

Now, we show that for every input of Alice $x \in X$, there exists a multiset *characterizing* the behavior of Alice on x. In other words, instead of running Alice, we can approximate the protocol's result (referee's output) by a uniform sample from the multiset. Furthermore, we prove that such a multiset can be found (w.h.p.) by some (relatively few) independent samples from the distribution defined by Alice (given x and r_{pub}).

Definition 3.4 (Characterizing Multiset). *For any r_{pub}, a multiset T of elements from M_A* characterizes *Alice for $x \in X$ if $\forall m_B \in M_B$,*

$$\left| Q(T, m_B) - \mathop{\mathbf{Pr}}_{r_A}\left[\rho_{r_{pub}}(A_{r_{pub}}(x, r_A), m_B) = 1\right] \right| \leq 0.1$$

where $Q(T_x, m_B)$ is the referee's expected value for the multiset T_x and Bob's possible message $m_B \in M_B$ (Definition 3.3).

Lemma 3.5 (Sample a Characterizing Multiset). *For any r_{pub}, for $x \in X$, let $r' = (r_A^1, ..., r_A^t)$ be t independent uniform samples from R_A where $t = 2 \cdot 200 \cdot \ln(2|M_B|)$. Then, for the multiset $T_x = \{A_{r_{pub}}(x, r_A^i) : i \in [t]\}$ it holds that $\forall m_B \in M_B$,*

$$\mathop{\mathbf{Pr}}_{r'}\left[\left| Q(T_x, m_B) - \mathop{\mathbf{Pr}}_{r_A}\left[\rho_{r_{pub}}(A_{r_{pub}}(x, r_A), m_B) = 1\right] \right| \leq 0.1\right] \geq 1 - \frac{1}{2|M_B|}$$

(i.e., T_x characterizes Alice for x)

Proof. Let T_x be as defined. $\forall i \in [t], m_B \in M_B$,

$$\mathbf{E}\left[\rho_{r_{\text{pub}}}(T_x[i], m_B)\right] = \mathop{\mathbf{Pr}}_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x, r_A), m_B) = 1\right]$$

$$\implies \mathbf{E}\left[\rho_{r_{\text{pub}}}(T_x[i], m_B) - \mathop{\mathbf{Pr}}_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x, r_A), m_B) = 1\right]\right] = 0$$

where the probability is over the random choice $T_x[i] \leftarrow A_{r_{\text{pub}}}(x)$.

Now, for $i \in [i]$, define random variables

$$\eta(i) = \rho_{r_{\text{pub}}}(T_x[i], m_B) - \mathop{\mathbf{Pr}}_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x, r_A), m_B) = 1\right].$$

Since the members of T_x are independent random variables, we have that all $\{\eta(i) : i \in [t]\}$ are independent random variables with expectation 0. Hence, we can use a Chernoff bound to bound the probability that, for a fixed $m_B \in M_B$,

$$\left| \sum_{i \in [t]} \eta(T_x[i]) \right| > 0.1 \cdot t.$$

In other words, the probability that

$$\left| Q(T_x, m_B) - \mathop{\mathbf{Pr}}_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x, r_A), m_B) = 1\right] \right| > 0.1$$

is bounded by

$$\begin{aligned}
\Pr_{r'}[|\sum_{i\in[t]} \rho_{r_{\text{pub}}}(T_x[i], m_B) - \mathbf{E}[\sum_{i\in[t]} \rho_{r_{\text{pub}}}(T_x[i], m_B)]| > 0.1] &< 2e^{-\frac{(0.1)^2 \cdot t}{2}} && \text{(Lemma 2.12)} \\
&= 2e^{-\frac{t}{200}} \\
&= 2e^{-2\ln(2|M_B|)} \\
&= \frac{1}{2|M_B|^2}.
\end{aligned}$$

By the union bound (over all $m_B \in M_B$),

$$\Pr_{r'}\left[\exists m_B \text{ s.t. } \left|Q(T,m_B) - \Pr_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x,r_A),m_B)=1\right]\right| > 0.1\right] < \frac{|M_B|}{2|M_B|^2} = \frac{1}{2|M_B|}$$

□

We define a hash function by following the process of Lemma 3.5 (running Alice t times independently):

Construction 3.6 Characterizing Multiset Function

Definition: The function is defined by the public random string r_{pub} and t Alice's random tapes $r_A^1, ..., r_A^t \in R_A$.

Output: For $x \in X$, the value of the function is the multiset as in Lemma 3.5:

$$h(x) = \text{The multiset } \{A_{r_{\text{pub}}}(x, r_A^i) : i \in [t]\}$$

where the multiset is encoded as a sequence $A_{r_{\text{pub}}}(x, r_A^1), \ldots, A_{r_{\text{pub}}}(x, r_A^t)$, note that every Alice's message can be encoded using $\log |M_A| = c$ bits.

Observation 3.7. *For all $x \in X$, the function from Construction 3.6 outputs a multiset that characterizes x w.p. $1 - \frac{1}{2|M_B|}$ where the probability is over the uniform random choice of $r_A^1, ..., r_A^t \in R_A$.*

Observation 3.8. *The function from Construction 3.6 is compressing: The domain of the function is of size 2^n, but the range is of size at most*

$$(2^c)^t = 2^{400c\cdot(c+1)\cdot\ln 2} = 2^{\Theta(c^2)} = 2^{o(n)}$$

Next, we prove that any x and x' which share a characterizing multiset, induce bad inputs for the protocol (since Alice's behavior on x and x' is similar).

Proposition 3.9. *Let $x, x' \in X$ and $y \in Y$ that separates them (Definition 2.1), if there is a multiset T that is characterizing for both x and x' then, the sum of the failure probability of $\pi(x,y)$ and $\pi(x',y)$ is at least 0.8. In other words, at least one of them fails.*

Proof. Since T is a characterizing multiset (Definition 3.4) of both x and x', then $\forall m_B \in M_B$

$$\left|Q(T,m_B) - \Pr_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x,r_A),m_B)=1\right]\right| \leq 0.1$$

and the same for x'. This means

$$\Pr_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x,r_A),m_B)=1\right]\in[Q(T,m_B)\pm 0.1]$$

and

$$\Pr_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x',r_A),m_B)=1\right]\in[Q(T,m_B)\pm 0.1]\,.$$

Putting it together we get that:

$$\left|\Pr_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x,r_A),m_B)=1\right]-\Pr_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x',r_A),m_B)=1\right]\right|\leq 0.2. \tag{1}$$

Assume without loss of generality that $f(x,y)=0$ and $f(x',y)=1$

$$\begin{aligned}
\mathbf{Pr}\left[\pi \text{ fails on } (x,y)\right] &= \Pr_{r_A,r_B}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x,r_A),B_{r_{\text{pub}}}(y,r_B))=1\right]\\
&= \mathop{\mathbf{E}}_{r_A,r_B}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x,r_A),B_{r_{\text{pub}}}(y,r_B))\right]\\
&= \mathop{\mathbf{E}}_{r_B}\left[\mathop{\mathbf{E}}_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x,r_A),B_{r_{\text{pub}}}(y,r_B))\right]\right]\\
&\geq \mathop{\mathbf{E}}_{r_B}\left[\mathop{\mathbf{E}}_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x',r_A),B_{r_{\text{pub}}}(y,r_B))\right]-0.2\right] \qquad \text{(Equation (1))}\\
&= \mathop{\mathbf{E}}_{r_B}\left[\mathop{\mathbf{E}}_{r_A}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x',r_A),B_{r_{\text{pub}}}(y,r_B))\right]\right]-0.2\\
&= \Pr_{r_A,r_B}\left[\rho_{r_{\text{pub}}}(A_{r_{\text{pub}}}(x',r_A),B_{r_{\text{pub}}}(y,r_B))=1\right]-0.2\\
&= \mathbf{Pr}\left[\pi \text{ succeeds on } (x',y)\right]-0.2\\
&= 1-\mathbf{Pr}\left[\pi \text{ fails on } (x',y)\right]-0.2\\
&= 0.8-\mathbf{Pr}\left[\pi \text{ fails on } (x',y)\right]
\end{aligned}$$

Hence, the sum of the failure probability of the protocol on (x,y) and the failure probability of the protocol on (x',y) is

$$\mathbf{Pr}\left[\pi(x,y) \text{ fails}\right]+\mathbf{Pr}\left[\pi(x',y) \text{ fails}\right]\geq 0.8$$

□

However, now we deal with the fact that there exist x's s.t. the multiset $h(x)$ does not characterize x (Observation 3.7).

Lemma 3.10. *Let π be an SM protocol of complexity $c(n)=o(\sqrt{n})$ and $h(x)$ be as in Construction 3.6. If we have an efficient adversary $\mathcal{A}\text{dv}_{collision}$ that breaks the security of h as a distributional CRH for some $p\in\text{poly}(\lambda)$:*

$$\Delta\left(\mathcal{A}\text{dv}_{collision}(h),\mathcal{COL}(h)\right)\leq\frac{1}{p(\lambda)}$$

Then, we can construct an adversary $\mathcal{A}\text{dv}_\pi$ *with running time of the same order as* $\mathcal{A}\text{dv}_{collision}$ *s.t.*

$$\mathbf{Pr}\left[\pi \textit{ fails on inputs from } \mathcal{A}\text{dv}_\pi\right] \geq 0.4\left(1 - \frac{1}{p(\lambda)}\right) - \text{negl}(\lambda)$$

Proof. $\mathcal{A}\text{dv}_\pi$'s algorithm is:

Algorithm 3.11 Near Ideal Collision Finder for h to Bad Inputs for Protocol π

1. Construct $h(x)$ using the public random string of π and as in Construction 3.6.
2. $x, x' \leftarrow \mathcal{A}\text{dv}_{\text{collision}}(h)$.
3. Find $y \in Y$ which separates x and x' (promised to be efficient by Definition 2.2).
4. Pass to Alice and Bob (x, y) w.p. $1/2$ or (x', y) w.p. $1/2$.

First, we consider $\mathcal{COL}$'s distribution: A pair (x, x') that was sampled from $\mathcal{COL}$ (the ideal collisions finder, Definition 2.15) will not be usable for Algorithm 3.11 if any of the following conditions hold:

1. $x = x'$.
2. $h(x) = h(x')$ is not characterizing x or x'.

We call a pair (x, x') a *colliding* pair if neither of the above two conditions hold. In the following claims we bound the probability for those bad events.

Proposition 3.12. *The probability of sampling a pair* (x, x) *from* $\mathcal{COL}$ *(i.e.,* $x = x'$*) is negligible. That is,*

$$\mathbf{Pr}_{(x,x')\leftarrow\mathcal{COL}}[x = x'] = \text{negl}(n)$$

Proof. First, consider the number of pairs (x, x') s.t. $x \neq x'$ but $h(x) = h(x')$. By the pigeonhole principle there exists a set of x's of size at least 2^{n-c^2} with the same image. Hence, there are at least $\binom{2^{n-c^2}}{2} = \Theta((2^{n-c^2})^2)$ many pairs (x, x') s.t. $x \neq x'$ but $h(x) = h(x')$. On the other hand, the number of pairs (x, x) is 2^n. Hence,

$$\mathbf{Pr}_{(x,x')\leftarrow\mathcal{COL}}[x = x'] = O\left(\frac{2^n}{2^n + (2^{n-o(n)})^2}\right) = \text{negl}(\lambda) \qquad (c^2 = o(n))$$

□

Proposition 3.13. *For a random* h*, the probability of sampling from* $\mathcal{COL}$ *a pair* (x, x') *s.t. the multiset* $h(x)$ *does not characterize* x *or* x' *is negligible.*

Proof. Let $(x, x') \leftarrow \mathcal{COL}$, recall that the distribution of each element from $\mathcal{COL}$ (x and x') is uniform (Definition 2.15). For each element in the pair, the probability that the multiset $h(x)$ does not characterize it is at most 2^{-c} (Observation 3.7) and by the union bound the claim follows. □

By Propositions 3.12 and 3.13, a sample from $\mathcal{COL}$ is colliding w.p. $1 - \text{negl}(\lambda)$. However, the distribution of $\mathcal{A}\text{dv}_{\text{collision}}$ is not exactly the same as $\mathcal{COL}$, but

$$\begin{aligned}\frac{1}{p(\lambda)} &\geq \Delta(\mathcal{COL}, \mathcal{A}\text{dv}_{\text{collision}}) \\ &\geq \left| \Pr_{(x,x') \leftarrow \mathcal{A}\text{dv}_{\text{collision}}} [(x, x') \text{ not colliding}] - \Pr_{(x,x') \leftarrow \mathcal{COL}} [(x, x') \text{ not colliding}] \right|\end{aligned}$$

and we can conclude that the probability that Algorithm 3.11 does not get a colliding pair (x, x') in step 2 is bounded by,

$$\Pr_{(x,x') \leftarrow \mathcal{A}\text{dv}_{\text{collision}}} [(x, x') \text{ isn't colliding}] \leq \frac{1}{p(\lambda)} + \text{negl}(\lambda)$$

To conclude: In cases that a colliding pair (x, x') was found by the adversary. The adversary chooses at random a pair from (x, y) and (x', y) (where y separates x and x', and can be found efficiently by Definition 2.2). By Proposition 3.9,

$$\Pr[\pi(x, y) \text{ fails}] + \Pr[\pi(x', y) \text{ fails}] \geq 0.8$$

and hence the failure probability over the random choice of the pair is at least

$$\Pr_{\substack{(z,y) \leftarrow \mathcal{A}\text{dv}_\pi \\ \pi}} [\pi(z, y) \text{ fails}] \geq 0.4$$

Now, put it together with the probability of finding a colliding pair (for h) and we get the probability that the protocol π fails on inputs from the adversary:

$$\begin{aligned}&\Pr[\mathcal{A}\text{dv}_\pi \text{ finds a colliding } (x, x')] \cdot \Pr_{\substack{(z,y) \leftarrow \mathcal{A}\text{dv}_\pi \\ \pi}} [\pi(z, y) \text{ fails}] \\ &\geq \left(1 - \frac{1}{p(\lambda)} - \text{negl}(\lambda)\right) \cdot \frac{4}{10}\end{aligned}$$

□

We get that given an adversary for the distributional CRH we can find bad inputs for the protocol as required for the proof.

□

Interactive Protocols

For general (interactive) protocols we can also prove a similar implication as in Theorem 3.2 for a logarithmic bound by the technique adaptation of Ben-Sasson and Maor [BM15]:

Theorem 3.14. *Let $c(n) < \delta_\varepsilon \log n$, where $\delta_\varepsilon < 1/2$ is a constant that depends only on ε. For an efficiently separable predicate (satisfying Definition 2.2), given a protocol of complexity $c(n)$ in the preset public coins interactive model, a distributional CRH can be constructed.*

Ben-Sasson and Maor studied protocols in the general communication settings and instead of using a characterizing multiset of *messages* they used a characterizing multiset of *deterministic strategies*. They have a variation of [BK97, Lemma 2.3] that says that there exists a strategies multiset of size $2^{O(c)}$ that characterizes the behavior of Alice for $x \in X$.

For a detailed proof see the full version.

A Corollary for PPHs

Corollary 3.15. *Without assuming the existence of distributional CRHs one cannot get better than $\sqrt{\cdot}$ compression for a direct-access robust equality PPH, even when extending the definitions for randomized hash functions.*

Proof. Observe that any PPH can be used to solve the same problem in the preset public coins SM model. Hence, this corollary is simply rephrasing Theorem 3.2 in the terms of adversarially robust property-preserving hash functions. □

Note that the other direction is true as well: Every protocol in the preset public coins SM model for $f(x, y)$ of $c(n)$ bits 'induces' a PPH for f of $c(n) \cdot n^\varepsilon$ bits for some $\varepsilon > 0$. That is, we start with any preset public coins SM protocol and repeat it n^ε times to make the error probability negligible. This protocol defines a family of PPHs and its random coins are fixed when sampling a function from the family.

4 No Ultra Short Interactive Communication

The power of the preset public coins model power lies between the public and the private coins models. As noted, the public random coins model is strictly more powerful than the private one: there are protocols of $O(1)$ bits only in this model. We show (unconditionally) that in our model there are *no* functions with $o(\log \log n)$ communication complexity:

Theorem 4.1. *Let $c(n) \colon \mathcal{N} \mapsto \mathcal{N}$ be s.t. $2^{3c(n)} = O(\log n)$ and let $f \colon X \times Y \to \{0, 1\}$ be an efficiently separable predicate (satisfying Definition 2.2, i.e., non redundant s.t. can be proven efficiently). In the preset public coins interactive communication model, if the adversary has a running time of $\mathrm{poly}(\lambda)$ (where λ is the security parameter) then, there are no protocols of complexity $O(c(n))$.*

Proof. Assume there is such a protocol in the preset public coins interactive model for some non-redundant function f of complexity $c(n)$.

In the proof of Theorem 3.14 we adapted Construction 3.6 for interactive protocols. The constructed hash function has the following properties:

– Random collisions in the function induce (w.h.p.) bad inputs in the protocol (Lemma 3.10).
– The range of the function is of size

$$|S_A|^t = |S_A|^{2^{2\cdot 200\ln(2|S_B|)}}$$

Those properties are the key points of the adversary described by Algorithm 4.2 that searches for random collisions in a brute force manner.

Algorithm 4.2 Finding Bad Inputs in Ulta-Succinct Protocols

1. Construct a characterizing function $h(\cdot)$ (Construction 3.6).
2. Repeat at most $3 \cdot 2^{2^{3c}} = \text{poly}(\lambda)$ times:
   (a) Choose a pair $x \neq x' \in X$ uniformly at random.
   (b) If $h(x) = h(x')$:
      i. Find $y \in Y$ that separates x and x' (can be done efficiently, as promised by Definition 2.2).
      ii. Output (x, y) w.p. $1/2$ or (x', y) w.p. $1/2$
      iii. Halt

Let h be a characterizing function of the protocol (Construction 3.6). The proof relies on the following two claims:

Proposition 4.3. *There must be a collision in h.*

Proof. The range of the characterizing function $h(x)$ is of size (number of possible characterizing sets):

$$|S_A|^{2^{2\cdot 200\ln(2|S_B|)}} = 2^{2^c \cdot 2\cdot 200\ln\left(2^{2^c}+1\right)} < 2^{2^{3c}}$$

Hence, since $2^{3c} = O(\log n) = o(n)$ there must be a collision in the function. □

Proposition 4.4. *The adversary described in Algorithm 4.2 finds a collision w.h.p.*

Proof. Since the range is small (same order as the running time of the adversary $2^{2^{3c}} = \text{poly}(\lambda)$), the adversary can find random collisions easily. The probability for a random pair to collide is at least $\frac{1}{2^{2^{3c}}}$ and hence, after $3 \cdot 2^{2^{3c}}$ trials, the probability that a collision was not found is at most:

$$\Pr_{x,x'}\left[h(x) \neq h(x')\right]^{3\cdot 2^{2^{3c}}} \leq \left(\left(1 - \frac{1}{2^{2^{3c}}}\right)^{2^{2^{3c}}}\right)^3 \to e^{-3}$$

$$\implies \Pr_{x,x'}\left[h(x) \neq h(x')\right]^{3\cdot 2^{2^{3c}}} < 0.05$$

□

We get that w.h.p. the adversary finds a collision in the function h. However, not every collision implies bad inputs for the protocol: the construction of the characterizing function implies that there exist also bad collisions: x and x' s.t. $h(x) = h(x')$ but $h(x)$ doesn't characterizes x or x' (recall Observation 3.7). However, in almost all collisions it is not the case and $h(x)$ characterizes x and x' (recall Proposition 3.13). Now, since the collision that Algorithm 4.2 finds is completely random we can conclude,

$$\mathbf{Pr}\left[\text{the adversary finds a colliding pair}\right] \geq 1 - 0.05 - \frac{1}{|S_B|}$$

and by Proposition 3.9

$$\mathbf{Pr}\left[\text{the protocol will fail}\right] \geq \frac{1}{2} \cdot \frac{8}{10}\left(1 - 0.05 - \frac{1}{|S_B|}\right) > \frac{1}{3}.$$

□

5 Secret Key Agreement from Efficient SM Protocols

5.1 Optimal Protocols from SKA

Our first observation is that it is possible to obtain an optimal protocol (in terms of the error as a function of the communication) for the equality predicate once given a secret key agreement protocol, following relatively simple principles. The error is 2^{-c} (where c is the communication complexity after the free talk) plus a negligible factor reflecting the probability of breaking the secret-key exchange. For details see the full version.

Theorem 5.1. *In the stateful preset public coins SM with free talk model: Given a secret key agreement protocol there is, for any $c(n)$, a protocol for the equality predicate of complexity $c(n)$, where any adversary can cause an incorrect answer with probability at most $2^{-c} + \mathrm{negl}(n)$, satisfying Definition 2.10.*

5.2 SKA from Near Optimal Protocols

Theorem 5.2. *An SM protocol with stateful free talk for the equality predicate of complexity $c(n) = O(\log \log n)$ for $c(n)$ larger from some constant, that is ε-secure (Definition 2.10) with $\varepsilon \leq 2^{-0.7c(n)}$, implies the existence of secret key-agreement protocols.*

Proof. Assume we have such a protocol π for the equality predicate $EQ\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. We will use π for constructing a secret key-agreement protocol. The idea is to construct a weak secret bit agreement (Definition 2.18) that can be amplified into a full secret key agreement (α and β according to Theorem 2.20). The construction is based on the following: (α, β)-SBA protocol:

Algorithm 5.3 Weak Bit Agreement

1. Alice and Bob communicate and toss coins according to the free talk of protocol π to generate their secret states τ_A and τ_B respectively.
2. Alice selects at random a bit $b \in \{0,1\}$ and uniformly random inputs $x_0, x_1 \in \{0,1\}^n$.
3. Alice evaluates $m_A = A(x_b, \tau_A)$ (that is, a message of the protocol π for $EQ(\cdot,\cdot)$).
4. Alice sends to Bob (m_A, x_1).
5. Bob evaluates $m_B = B(x_1, \tau_B)$.
6. Alice outputs b and Bob outputs $b' = \rho(m_A, m_B)$.

Lemma 5.4. *Algorithm 5.3 is a* $\left(\alpha = 1 - 2^{-c/2-3}, \beta = 2^{-c/2+1}\right)$*-SBA protocol.*

Proof. Let $c = c(n)$. We have to show its agreement and secrecy properties:

Agreement. By the properties of protocol π, specifically that the error $\varepsilon \leq 2^{-0.7c(n)}$:

$$\mathbf{Pr}\left[b = b'\right] \geq 1 - \left(\frac{1}{2}\right)^{0.7c} \geq 1 - \left(\frac{1}{2}\right)^{0.5c-2} = \frac{1 + (1 - 2^{-c/2-3})}{2} = \frac{1+\alpha}{2}.$$

Secrecy. We should show that for every PPTM adversary $\mathcal{A}\mathrm{dv}_{\mathrm{sba}}$

$$\mathbf{Pr}\left[\mathcal{A}\mathrm{dv}_{\mathrm{sba}}(m_A, x_1) = b \mid b = b'\right] \leq \frac{2-\beta}{2} = \frac{2 - 2^{-c/2+1}}{2} = \frac{2^{c/2} - 1}{2^{c/2}}.$$

Assume towards contradiction that $\mathbf{Pr}\left[\mathcal{A}\mathrm{dv}_{\mathrm{sba}}(m_A, x_1) = b \mid b = b'\right] > \frac{2^{c/2}-1}{2^{c/2}}$. We show that given $\mathcal{A}\mathrm{dv}_{\mathrm{sba}}$, we can construct $\mathcal{A}\mathrm{dv}_{\mathrm{eq}}$ that finds bad inputs for the protocol π (with probability higher than ε):

Lemma 5.5. *Given an adversary* $\mathcal{A}\mathrm{dv}_{sba}$ *with success probability (guessing* b *when it is equal to* b'*) at least* $\frac{2^{c/2}-1}{2^{c/2}}$*, we can construct an adversary* $\mathcal{A}\mathrm{dv}_{eq}$ *with running time* $O(6 \cdot 2^{c+1})$ *s.t.*

$$\mathbf{Pr}\left[\pi \textit{ fails on inputs from } \mathcal{A}\mathrm{dv}_{eq}\right] > 2^{-0.7c} \geq \varepsilon.$$

Proof. The strategy of the adversary $\mathcal{A}\mathrm{dv}_{\mathrm{eq}}$ to find bad inputs is:

Algorithm 5.6 $\mathcal{A}\mathrm{dv}_{\mathrm{eq}}$ – Find Bad Inputs Using $\mathcal{A}\mathrm{dv}_{\mathrm{sba}}$

1. Repeat at most $6 \cdot 2^{c+1}$ times:
   (a) Select uniformly at random $x \in \{0,1\}^n$ and set it as Alice's input.
   (b) Let Alice's message (output) be $m_A \in M_A$.
   (c) Select uniformly at random $x' \in \{0,1\}^n$.
   (d) If $\mathcal{A}\mathrm{dv}_{\mathrm{sba}}(x, m_A) = 1$ and $\mathcal{A}\mathrm{dv}_{\mathrm{sba}}(x', m_A) = 1$:
      i. Pass the message m_A to the referee and set Bob's input to be either $y = x$ w.p. $1/2$ or $y = x'$ w.p. $1/2$.
      ii. Halt.
   (e) Otherwise, continue to the next session.

Recall that the private states of Alice and Bob are τ_A and τ_B (unknown to the adversary). The success of the adversary $\mathcal{A}\mathrm{dv}_{\mathrm{eq}}$ relies on choosing a *colliding* x' (i.e., x' s.t. $A(x, \tau_B) = A(x', \tau_B)$).

For $x \in \{0,1\}^n$ denote by p_x^0 and p_x^1 the probability of a random $x' \in \{0,1\}^n$ to be bad in the following sense:

1. Let p_x^0 be the probability that for a random x': x and x' collide yet are not identified by $\mathcal{A}\mathrm{dv}_{\mathrm{sba}}$ as such. I.e.,

$$p_x^0 = \mathop{\mathbf{Pr}}_{x', \mathcal{A}\mathrm{dv}} \left[A(x) = A(x') \wedge \mathcal{A}\mathrm{dv}_{\mathrm{sba}}(x', A(x, \tau_A)) = 0 \right].$$

2. Let p_x^1 be the probability that for a random x': x and x' do not collide yet are identified by $\mathcal{A}\mathrm{dv}_{\mathrm{sba}}$ as such. I.e.,

$$p_x^1 = \mathop{\mathbf{Pr}}_{x', \mathcal{A}\mathrm{dv}} \left[A(x) \neq A(x') \wedge \mathcal{A}\mathrm{dv}_{\mathrm{sba}}(x', A(x, \tau_A)) = 1 \right].$$

Let $p_x = \max(p_x^0, p_x^1)$. For a random free talk session and over the random choice of $x \in \{0,1\}^n$, we know that $\mathbf{E}_x [p_x] = \sum_{p_x} p_x \mathbf{Pr} [p_x] \leq 2^{-c/2+1}$.

We will analyze the probability of success of each session and then argue that at least one session succeeds in outputting a value w.h.p. We need to show the probability of outputting a pair of strings that agree on the message Alice sends ("Correct") is not much smaller than outputting a pair that does not agree ("Wrong"). To compare the probability of outputting a wrong value to the probability of outputting the correct value, note that the probability of outputting a wrong value is at most $\sum_{\text{all } p_x} \Pr[p_x] \cdot p_x \leq 2^{-c/2}$. On the other hand we will argue that the probability of outputting a correct value (with identification) is roughly (at least) 2^{-c}. The ratio between them is roughly $2^{-0.5c}$ and we get an attack of the equality protocol that is better than its purported security.

For the rest of the proof see the full version. □

Lemma 5.5 implies the secrecy of Algorithm 5.3 (otherwise, we get a contradiction for the security of protocol π).

This means that Algorithm 5.3 is an $(1 - 2^{-c/2-3}, 2^{-c/2+1})$-SBA and the proof of Lemma 5.4. □

Finally, we have to show that Theorem 2.20 can be used to amplify the secret bit agreement:

Proposition 5.7. *For the functions $\alpha(c) = 1 - 2^{-c/2-3}$ and $\beta(c) = 2^{-c/2+1}$ the conditions in Theorem 2.20 hold.*

For proof see the full version.

We conclude, by Proposition 5.7 that the SBA of Algorithm 5.3 (Lemma 5.4) can be amplified efficiently into a full-fledged secret key agreement protocol. □

6 Conclusions

Role of private randomness. In this paper we introduced a computational model for communication complexity. However, it can also be seen as a generalization of (deterministic) property preserving hash functions to probabilistic algorithms. We studied some relations between the power of private randomness and cryptographic primitives such as collision resistance. The main open problem left from this point is whether CRHs are equivalent to preset public coins SM protocols of complexity $o(\sqrt{n})$ and whether we can break that bound using a primitive weaker than CRHs. Another direction could be to show how to use $o(\sqrt{n})$ equality protocols in order to get low communication string commitment.

Boyle et al.'s lower bounds. Boyle et al. [BLV18] proved two general lower bounds for property preserving hash functions using communication complexity[15]:

1. A lower bound for *reconstructing* predicates: Boyle et al. proved that for predicates that can be used for reconstructing the original string there cannot exist (compressing) property preserving hash functions. This lower bound is also true for our preset public coins SM model. However, we didn't necessarily consider reconstructing predicates (for instance, the equality predicate is not a reconstructing predicate).
2. General lower bound from one-way communication: Boyle et al. proved that any property preserving hash function cannot compress better than the one-way communication complexity[16]. This lower bound is also true in our model, but it is too loose in our context since in our model the inputs and the public random string may be dependent (e.g., the equality predicate complexity is $O(1)$ in the one-way communication complexity model).

Multi CRHs (MCRH). For $k \geq 3$, A familty of hash function is k-multi-collision resistant if finding a collision of size k is hard: no PPTM can succeed in finding $x_1, \ldots, x_k$ s.t. $h(x_1) = \ldots = h(x_k)$ with non-negligible probability (for $k = 2$ it is the regular notion of collision resistance); see [KNY18; KY18; RV22] for the relationship between MCRHs, dCRHS and CRHs. One question is whether MCRHs can be constructed from succinct protocols in a black-box manner.

Secret key agreement. We showed a tight relationship between secret key agreement protocols and succinct protocols for the equality predicate in the SM preset public coins stateful free talk model. On the one hand, SKA can be used for constructing an equality protocol in this model, and on the other hand, equality protocols with good error in this model can be used for constructing SKA protocols. The open questions are (i) whether the existence of protocols with much worse error probability (e.g., constant error probability for c which $O(\log \log \lambda)$)

[15] See also Hardt and Woodruff [HW13] who proved robustness limitations for *linear* functions.

[16] See Fleischhacker and Simkin [FS21] and Fleischhacker et al. [FLS22] for more such lower bounds.

also imply SKA and (ii) whether the fact that we allowed the adversary $\mathcal{A}\mathrm{dv}_{\mathrm{eq}}$ to be *rushing* was essential.

Acknowledgments

We thank Shahar Dobzinski, Ilan Komargodski, Guy Rothlbum and Eylon Yogev for useful discussions and suggestions and the Crypto 2022 referees for the helpful comments and questions.

Bibliography

[AS08] Noga Alon and Joel H. Spencer. *The Probabilistic Method, Third Edition*. Wiley, 2008.

[BGK15] Ralph Bottesch, Dmitry Gavinsky, and Hartmut Klauck. "Equality, revisited". In: *International Symposium on Mathematical Foundations of Computer Science*. Springer. 2015, pp. 127–138.

[BHKY19] Nir Bitansky, Iftach Haitner, Ilan Komargodski, and Eylon Yogev. "Distributional collision resistance beyond one-way functions". In: *EUROCRYPT 2019*. Springer. 2019, pp. 667–695.

[BIN97] Mihir Bellare, Russell Impagliazzo, and Moni Naor. "Does parallel repetition lower the error in computationally sound protocols?" In: *Proceedings 38th Symposium on Foundations of Computer Science*. IEEE. 1997, pp. 374–383.

[BK97] László Babai and Peter G Kimmel. "Randomized simultaneous messages: Solution of a problem of Yao in communication complexity". In: *Proceedings of Computational Complexity. Twelfth IEEE Conference*. IEEE. 1997, pp. 239–246.

[BLV18] Elette Boyle, Rio LaVigne, and Vinod Vaikuntanathan. "Adversarially Robust Property-Preserving Hash Functions". In: *ITCS 2019*. 2018.

[BM09] Boaz Barak and Mohammad Mahmoody-Ghidary. "Merkle puzzles are optimal—an $O(n^2)$-query attack on any key exchange from a random oracle". In: *CRYPTO 2009*. Springer. 2009, pp. 374–390.

[BM15] Eli Ben-Sasson and Gal Maor. "Lower bound for communication complexity with no public randomness". In: *Electron. Colloquium Comput. Complex.* 22 (2015), p. 139.

[CHS05] Ran Canetti, Shai Halevi, and Michael Steiner. "Hardness amplification of weakly verifiable puzzles". In: *Theory of Cryptography Conference*. Springer. 2005, pp. 17–33.

[DI06] Bella Dubrov and Yuval Ishai. "On the randomness complexity of efficient sampling". In: *Proceedings of the thirty-eighth ACM symposium on Theory of computing*. 2006, pp. 711–720.

[FLS22] Nils Fleischhacker, Kasper Green Larsen, and Mark Simkin. "Property-Preserving Hash Functions for Hamming Distance from Standard Assumptions". In: *Advances in Cryptology - EUROCRYPT 2022 Proceedings, Part II*. Vol. 13276. LNCS. Springer, 2022, pp. 764–781.

[FS21] Nils Fleischhacker and Mark Simkin. "Robust Property-Preserving Hash Functions for Hamming Distance and More". In: *EUROCRYPT 2021*. LNCS. Springer, 2021.

[GHY20] Ofer Grossman, Justin Holmgren, and Eylon Yogev. "Transparent Error Correcting in a Computationally Bounded World". In: *TCC 2020*. Springer, 2020.

[HIKNV04] Prahladh Harsha, Yuval Ishai, Joe Kilian, Kobbi Nissim, and Srinivasan Venkatesh. "Communication versus computation". In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2004, pp. 745–756.

[Hol05] Thomas Holenstein. "Key agreement from weak bit agreement". In: *Proceedings of the 37th ACM symposium on Theory of computing*. 2005, pp. 664–673.

[Hol06] Thomas Holenstein. "Strengthening key agreement using hard-core sets". PhD thesis. ETH Zurich, 2006.

[HW13] Moritz Hardt and David Woodruff. "How robust are linear sketches to adaptive inputs?" In: *Proceedings of the forty-fifth ACM symposium on Theory of computing*. 2013, pp. 121–130.

[IL89] Russell Impagliazzo and Michael Luby. "One-way functions are essential for complexity based cryptography". In: *30th IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society. 1989, pp. 230–235.

[IR89] Russell Impagliazzo and Steven Rudich. "Limits on the provable consequences of one-way permutations". In: *Proceedings of the 21st ACM symposium on Theory of Computing*. 1989, pp. 44–61.

[KN96] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1996.

[KNY18] Ilan Komargodski, Moni Naor, and Eylon Yogev. "Collision Resistant Hashing for Paranoids: Dealing with Multiple Collisions". In: *Advances in Cryptology - EUROCRYPT 2018 Proceedings, Part II*. Springer, 2018, pp. 162–194.

[KY18] Ilan Komargodski and Eylon Yogev. "On Distributional Collision Resistant Hashing". In: *Advances in Cryptology - CRYPTO 2018*. Vol. 10992. LNCS. Springer, 2018, pp. 303–327.

[Lip94] Richard J. Lipton. "A New Approach To Information Theory". In: *STACS 94, Proceedings of the 11th Symposium on Theoretical Aspects of Computer Science, February 24-26, 1994*. Vol. 775. LNCS. Springer, 1994, pp. 699–708.

[MNS11] Ilya Mironov, Moni Naor, and Gil Segev. “Sketching in adversarial environments”. In: *SIAM Journal on Computing* 40.6 (2011), pp. 1845–1870.

[MPSW10] Silvio Micali, Chris Peikert, Madhu Sudan, and David A. Wilson. “Optimal Error Correction for Computationally Bounded Noise”. In: *IEEE Trans. Inf. Theory* 56.11 (2010), pp. 5673–5680.

[New91] Ilan Newman. “Private vs. Common Random Bits in Communication Complexity”. In: *Inf. Process. Lett.* 39.2 (1991), pp. 67–71.

[NR09] Moni Naor and Guy N. Rothblum. “The complexity of online memory checking”. In: *Journal of the ACM* 56.1 (2009), pp. 1–46.

[NS96] Ilan Newman and Mario Szegedy. “Public vs. private coin flips in one round communication games”. In: *Proc. of the twenty-eighth ACM symposium on Theory of computing*. 1996, pp. 561–570.

[RV22] Ron D. Rothblum and Prashant N. Vasudevan. “Collision-Resistance from Multi-Collision-Resistance”. In: *Electron. Colloquium Comput. Complex.* (2022), p. 17.

[RY20] Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.

[Sim98] Daniel R. Simon. “Finding Collisions on a One-Way Street: Can Secure Hash Functions Be Based on General Assumptions?” In: *Advances in Cryptology - EUROCRYPT ’98*. Springer. 1998, pp. 334–345.

[Wee07] Hoeteck Wee. “One-Way Permutations, Interactive Hashing and Statistically Hiding Commitments”. In: *TCC 2007*. Ed. by Salil P. Vadhan. Vol. 4392. LNCS. Springer, 2007, pp. 419–433.

[Yao79] Andrew Chi-Chih Yao. “Some Complexity Questions Related to Distributive Computing”. In: *Proc. of the 11th ACM symposium on Theory of computing*. STOC ’79. 1979, pp. 209–213.