2018 IEEE 59th Annual Symposium on Foundations of Computer Science

Computational Two-Party Correlation: A Dichotomy for Key-Agreement Protocols

Iftach Haitner*, Kobbi Nissim†, Eran Omri‡, Ronen Shaltiel §, and Jad Silbak ¶
*School of Computer Science, Tel Aviv University. iftachh@cs.tau.ac.il
†Department of Computer Science, Georgetown University. kobbi.nissim@georgetown.edu
‡Department of Computer Science, Ariel University. omrier@ariel.ac.il
§Department of Computer Science, University of Haifa. ronen@cs.haifa.ac.il
¶School of Computer Science, Tel Aviv University. jadsilbak@mail.tau.ac.il

***Abstract*—Let π be an efficient two-party protocol that given security parameter κ, both parties output single bits X_κ and Y_κ, respectively. We are interested in how (X_κ, Y_κ) "appears" to an efficient adversary that only views the transcript T_κ. We make the following contributions:**

- **We develop new tools to argue about this loose notion, and show (modulo some caveats) that for every such protocol π, there exists an efficient *simulator* such that the following holds: on input T_κ, the simulator outputs a pair (X'_κ, Y'_κ) such that $(X'_\kappa, Y'_\kappa, T_\kappa)$ is (somewhat) *computationally indistinguishable* from $(X_\kappa, Y_\kappa, T_\kappa)$.**
- **We use these tools to prove the following *dichotomy theorem*: every such protocol π is:**
  - **either *uncorrelated* — it is (somewhat) indistinguishable from an efficient protocol whose parties interact to produce T_κ, but then choose their outputs *independently* from some product distribution (that is determined in poly-time from T_κ),**
  - **or, the protocol implies a key-agreement protocol (for infinitely many κ's).**

  Uncorrelated protocols are uninteresting from a cryptographic viewpoint, as the correlation between outputs is (computationally) trivial. Our dichotomy shows that every protocol is either completely uninteresting or implies key-agreement.
- **We use the above dichotomy to make progress on open problems on minimal cryptographic assumptions required for differentially private mechanisms for the XOR function.**
- **A subsequent work of Haitner et al. uses the above dichotomy to makes progress on a long-standing open question regarding the complexity of fair two-party coin-flipping protocols.**

We highlight the following ideas regarding our technique:

- **The simulator algorithm is obtained by a carefully designed "competition" between efficient algorithms attempting to forecast $(X_\kappa, Y_\kappa)|_{T_\kappa = t}$. The winner is used to simulate the outputs of the protocol.**
- **Our key-agreement protocol uses the simulation to reduce to an information theoretic setup, and is in some sense non-black box.**

***Keywords*-computational correlation; key agreement; differential privacy;**

I. Introduction

In this paper we discuss "computational correlation" of efficient single-bit output two-party protocols. We start with some notation for such protocols.

Two-party protocols with single bit output. We are interested in probabilistic polynomial-time (PPT), two-party, no-input, single-bit output protocols: the PPT parties receive a common input 1^κ (i.e., a security parameter), and each party outputs a single bit. For such protocols $\pi = (\mathsf{A}, \mathsf{B})$ we use the notation:

$$\pi(1^\kappa) = (\mathsf{A}, \mathsf{B})(1^\kappa) = (X_\kappa, Y_\kappa, T_\kappa).$$

where X_κ is the output of A, Y_κ is the output of B, and T_κ is the transcript of the protocol. Loosely speaking, we are interested in the correlation that an execution of $\pi(1^\kappa)$ generates between X_κ and Y_κ, when viewed from the point of view of a PPT algorithm that receives only the transcript T_κ as input.

Key-agreement protocols: We will be interested in "computational correlation" between the outputs of a protocol. It is instructive to consider the example of key-agreement protocols. The latter are PPT protocols with the following properties:

- *Secrecy:* $\Pr[\mathsf{E}(T_\kappa) = X_\kappa] \leq \frac{1}{2} + s(\kappa)$ for every PPT algorithm (eavesdropper) E. (Here the standard choice for $s(\kappa)$ is a negligible function, but we will also consider versions where $s(\kappa) = s$ is a constant).
- *Agreement:* $\Pr[X_\kappa = Y_\kappa] \geq \frac{1}{2} + a(\kappa)$. (Here the standard choice for $a(\kappa)$ is half minus a negligible function, but we will also consider versions where $a(\kappa) = a$ is a constant, and $a > s$).

The reader is referred to [2] for a survey on key-agreement protocols. We remark that by [2], a key-agreement protocol for constants s and a with $s < a^2/10$, implies a full-fledged key-agreement protocol (i.e., with the standard choices of agreement and secrecy).

2575-8454/18/$31.00 ©2018 IEEE
DOI 10.1109/FOCS.2018.00022

Computational correlation: Loosely speaking, from the "point of view" of a PPT algorithm E that only sees the transcript t of a key-agreement protocol, the probability space $(X_\kappa, Y_\kappa)|_{T_\kappa=t}$ "should look like" (R, R), for R being a uniform bit (unknown to E). This in contrast to the view of an *unbounded* E: since for any protocol, and every transcript t, $(X_\kappa, Y_\kappa)|_{T_\kappa=t}$ is a product distribution.

An important contribution of this paper is developing tools to formalize the vague notion of "computational correlation" in a rigorous (and as we shall explain) useful way. Specifically, we show that (modulo some caveats and technicalities that we soon explain) for every single-bit output, two-party protocol, there exists a PPT algorithm (simulator) Sim such the the following holds: on input T_κ, Sim outputs two bits (simulated outputs) (X'_κ, Y'_κ) such that the simulated experiment $(X'_\kappa, Y'_\kappa, T_\kappa)$ is computationally indistinguishable from (real) experiment $(X_\kappa, Y_\kappa, T_\kappa)$.

The simulated experiment represents the "best understanding" that a PPT can obtain on the real experiment. We find it quite surprising that such a clean notion exists. One could have expected that different PPT's have "different views" or "different understanding" of the real execution, and it is impossible to come up with a *single* simulated distribution that represents the "collective understanding" of all PPT's. Loosely speaking, the above yields that such two-party protocols can be classified as follows:

- Protocols in which the simulated distribution $(X'_\kappa, Y'_\kappa, T_\kappa)$ has the property that (X'_κ, Y'_κ) are independent, conditioned on every fixing of T_κ. We will call such protocols "uncorrelated".
- Protocols in which the simulated distribution $(X'_\kappa, Y'_\kappa, T_\kappa)$ has the property that (X'_κ, Y'_κ) are correlated given T_κ (at least for some fixings of T_κ).

Uncorrelated protocols are cryptographically uninteresting: Uncorrelated protocols are uninteresting from a cryptographic viewpoint; whenever we have such a protocol π, we can imagine that the parties use the following alternative trivial protocol $\widehat{\pi} = (\widehat{\mathsf{A}}, \widehat{\mathsf{B}})$: party $\widehat{\mathsf{A}}$ samples a transcript T_κ (on his own) and sends T_κ to $\widehat{\mathsf{B}}$. Then each party samples its output (independently) by applying the simulator for π on T_κ.

As is often the case in simulation, if a PPT adversary E is able to perform some task (that is defined in terms of the original triplet $(X_\kappa, Y_\kappa, T_\kappa)$), then it achieves roughly the same success on the simulated triplet $(X'_\kappa, Y'_\kappa, T_\kappa)$. Specifically, if π is a key-agreement protocol, then $\widehat{\pi}$ is also a key-agreement protocol. The latter, however, is obviously false. This is because given T_κ, the adversary E can use the simulator to sample X'_κ with probability that is at least as large as $\Pr[X'_\kappa = Y'_\kappa]$. This means that in $\widehat{\pi}$ secrecy is less than agreement, ruling out any meaningful form of key-agreement.

Correlated protocols yield key-agreement: In this paper we prove that (again, modulo some caveats and technicalities that we soon explain) if a protocol is correlated, then it can be transformed into a key-agreement protocol. This can be interpreted as the following dichotomy theorem:

> Every PPT single-bit output two-party protocol is either uncorrelated (and is indistinguishable from a trivial and cryptographically uninteresting protocol), or it implies a key-agreement protocol.

We find this quite surprising. Intuitively, key-agreement protocols and trivial protocols represent two extremes in the spectrum of two-party protocols, and one may expect that there are many interesting intermediate types in between the two extremes.[1]

A. Our Results

1) Every two-party single bit output protocol has a simulator and a forecaster: We show that every protocol has a PPT *simulator* that, seeing only the transcript, produces a simulated distribution simulating the (real) output distribution of the protocol.

Theorem I.1 (Existence of PPT simulators (informal))**.** *Let $\pi = (\mathsf{A}, \mathsf{B})$ be a* PPT *no-input, single-bit output two-party protocol. For every $\rho > 0$ there exists a* PPT Sim *such that when given $(1^\kappa, t)$, $\mathsf{Sim}(1^\kappa, t)$ outputs two bits, (x', y') such that the following holds: Let* $\text{REAL} = \{\text{REAL}_\kappa\}_{\kappa\in\mathbb{N}}$ *and* $\text{SML} = \{\text{SML}_\kappa\}_{\kappa\in\mathbb{N}}$ *be ensembles defined as follows:* $\text{REAL}_\kappa = \pi(1^k) = (X_\kappa, Y_\kappa, T_\kappa)$ *and let* $\text{SML}_\kappa = (X'_\kappa, Y'_\kappa, T_\kappa)$ *for* $(X'_\kappa, Y'_\kappa) = \mathsf{Sim}(1^\kappa, T_\kappa)$*. For infinitely many $\kappa \in \mathbb{N}$,* REAL *cannot be distinguished from* SML *with advantage ρ by* PPT *algorithms.*

(A precise formal definition of computational indistinguishability with advantage ρ is given in Definition II.1. Theorem I.1 is formally stated in Section III in a more general form.)

Theorem I.1 comes with two caveats:

[1] One illuminating "intermediate setup" is "defective key-agreement protocols" in which the agreement and secrecy properties above hold, but with $a < s$ (namely, agreement is smaller than secrecy, and this is not a cryptographically meaningful key-agreement). Such protocols can be uncorrelated (and trivial), but they can also be correlated, and thus, by our result, imply key-agreement. As we shall explain, this approach yields several new results, as in some cases it was previously unknown whether key-agreement protocols are implied, but it is possible to show that the protocol is not uncorrelated.

- The simulated ensemble SML is only guaranteed to resemble the real ensemble REAL on some infinite subset $\mathcal{I}$ of $\kappa \in \mathbb{N}$.
- For $\kappa \in \mathcal{I}$, REAL and SML are only weakly indistinguishable as ρ is not negligible.

We do not know whether the theorem can be proven without these caveats. We mention that most the machinery that we develop (with one notable exception) can be used towards proving a version without the caveats. As we will demonstrate, in some cases, the caveats do not affect applications, and we can prove clean results using the theorem.

Remark I.2 (Auxiliary input simulators, and the leakage simulation lemma)**.** *Theorem I.1 is similar in spirit to the so called "leakage simulation lemma" [3, 4, 5, 6, 7, 8].*

In the leakage simulation lemma one considers a pair (T, Z) of random variables, and a finite class $\mathcal{C}$ of "distinguisher functions" (which is typically the class of circuits of some size s, and so we will assume this for this discussion). The lemma states that there is a "simulator function" Sim *of circuit complexity s', which on input T produces a string Z' such that no distinguisher D from $\mathcal{C}$ can distinguish (T, Z) from (T, Z') with advantage greater than some parameter $\rho > 0$. The complexity s' is some polynomial in $s, \ell, \frac{1}{\rho}$ (where ℓ is the bit length of Z). The reader is referred to [8] for a discussion of works in this framework.*

There are two differences between the leakage simulation lemma and Theorem I.1:

- *The class of distinguishers $\mathcal{C}$ that we consider are randomized polynomial time machines, and we show the existence of a simulator* Sim *that* belongs *to this class. This is crucial in our applications. In contrast, in the leakage simulation lemma the simulator is a circuit of size $s' > s$ and* does not *belong to the class $\mathcal{C}$. Moreover, there are negative results [3, 8] showing limitations on proving the leakage simulation lemma with $s' \leq s$.*
- *In Theorem I.1 we can only achieve $\rho > 0$ that is constant, whereas the leakage simulation lemma can achieve much smaller ρ (and this is crucial in some of its applications).*

Forecasters. In applications, it will be useful to assume that the simulators work in the following specific fashion: there is a "forecaster algorithm" F which on input t, generates a description of the probability space $(X'_\kappa, Y'_\kappa)|_{T_\kappa = t}$. For technical reasons, it is helpful to think of the forecaster F as a deterministic poly-time algorithm that receives its random coin r, as an additional input. Given input $(1^\kappa, t, r)$ the forecaster outputs three numbers:

- p_{A} which is a "forecast" for $\Pr[X_\kappa = 1 \mid T_\kappa = t]$.
- $p_{\mathsf{B}|0}$ which is a "forecast" for $\Pr[Y_\kappa = 1 \mid T_\kappa = t, X_\kappa = 0]$.
- $p_{\mathsf{B}|1}$ which is a "forecast" for $\Pr[Y_\kappa = 1 \mid T_\kappa = t, X_\kappa = 1]$.

All that is left for the simulator is to sample according to this forecast. For $p \in [0, 1]$, we will use the notation U_p to denote the distribution of a biased coin that is one with probability p. We can now restate Theorem I.1 in the following more general form:

Theorem I.3 (Existence of PPT forecasters, informal)**.** *Let $\pi = (\mathsf{A}, \mathsf{B})$ be a* PPT *no-input, single-bit output two-party protocol. For every $\rho > 0$ there exists a deterministic poly-time machine F that on input $(1^\kappa, t, r)$ outputs three numbers $p_{\mathsf{A}}, p_{\mathsf{B}|0}, p_{\mathsf{B}|1} \in [0, 1]$ such that the following holds: let R_κ be a uniform polynomially long string (intuitively R serves as the random coins of F), and let* $\text{REAL} = \left\{\text{REAL}_\kappa = (\pi(1^k), R_\kappa) = (X_\kappa, Y_\kappa, T_\kappa, R_\kappa)\right\}$ *and* $\text{SML} = \{\text{SML}_\kappa = (X'_\kappa, Y'_\kappa, T_\kappa, R_\kappa)\}$ *be the distribution ensembles obtained by:*

- $(p_{\mathsf{A}}, p_{\mathsf{B}|0}, p_{\mathsf{B}|1}) = \mathsf{F}(1^\kappa, T_\kappa, R_\kappa)$.
- $X'_\kappa \leftarrow U_{p_{\mathsf{A}}}$ *and* $Y'_\kappa \leftarrow U_{p_{\mathsf{B}|X'_\kappa}}$.

Then for infinitely many $\kappa \in \mathbb{N}$, REAL *cannot be distinguished from* SML *with advantage ρ by* PPT *algorithms.*

(Theorem I.3 is formally stated in Section III.)

Theorems I.1 and I.3 may be of independent interest, and we believe that they will find more applications. This is because the simulator induces a *single* distribution that is computationally indistinguishable (albeit only with advantage $\rho = o(1)$) from the real output distribution of the protocol. Moreover, in the simulated distribution $(X'_\kappa, Y'_\kappa, T_\kappa)$ (sampled using the forecaster) the variables (X'_κ, Y'_κ) have *information theoretic uncertainty* conditioned on $\{T_\kappa = t\}$. This enables us to use tools and techniques from information theory on the simulated distribution, and obtain results about the computational security of the original protocol (and protocols that we construct from it). Indeed, we use this approach in our applications.

We believe that a helpful analogy is the notion of *computational entropy*: which in some cases, given a distribution X assigns a distribution X' that is computationally indistinguishable from X and has *information theoretic uncertainty*.

2) A Dichotomy of Single-bit Output Two-Party Protocols: We now give a precise definition of uncorrelated protocols. For that purpose we introduce the

following notion of a "decorrelator". Loosely speaking, a decorrelator is a forecaster that forecasts that (X_κ, Y_κ) are independent conditioned on T. Once again, for technical reasons, it is helpful to think of a decorrelator as a deterministic poly-time algorithm that receives its random coin r, as an additional input.

Definition I.4 (ρ-decorrelator, and ρ-uncorrelated protocols, informal)**.** *A deterministic poly-time algorithm* $\mathsf{Decor}(t, r)$ *is a* ρ-*decorrelator for protocol* $\pi = (\mathsf{A}, \mathsf{B})$ *if the following holds: let* $\text{REAL} = \{\text{REAL}_\kappa\}_{\kappa\in\mathbb{N}}$ *and* $\text{UCR} = \{\text{UCR}_\kappa\}_{\kappa\in\mathbb{N}}$ *be ensembles defined as follows:* $\text{REAL}_\kappa = (\pi(1^k); R_\kappa) = (X_\kappa, Y_\kappa, T_\kappa, R_\kappa)$ *where* R_κ *is a uniformly chosen independent polynomially long string (that intuitively serves as the random coins of* Decor*). Let* $\text{UCR}_\kappa = (X'_\kappa, Y'_\kappa, T_\kappa, R_\kappa)$ *where* $(p_\mathsf{A}, p_\mathsf{B}) = \mathsf{Decor}(T_\kappa, R_\kappa)$*, and (independently sampled)* $X'_\kappa \leftarrow U_{p_\mathsf{A}}$ *and* $Y'_\kappa \leftarrow U_{p_\mathsf{B}}$*. It is required that for infinitely many* $\kappa \in \mathbb{N}$*,* REAL *cannot be distinguished from* UCR *with advantage* ρ *by* PPT *algorithms.*
A protocol π *is* ρ-*uncorrelated if it has a* ρ*-decorrelator.*

(Definition I.4 is formally stated in Section III.)

Loosely speaking, the fact that the randomness R_κ appears in the two experiments, prevents the decorrelator from using R_κ to correlate between X'_κ and Y'_κ. In the definition the latter should appear independent, even after seeing R_κ.

We observe that ρ-uncorrelated protocols are uninteresting from a cryptographic viewpoint in the following sense (that is made precise in Section III):

- A ρ-uncorrelated protocol cannot be a key-agreement protocol for $s < a + 2\rho$.
- If a "black-box construction" that makes ℓ invocations to a ρ-uncorrelated protocol, yields a key-agreement protocol with $s < a + 3 \cdot \ell \cdot \rho$, then the black-box construction itself can be used to give a key-agreement (with the standard choices of secrecy and agreement) that does not use the original protocol. This means that a ρ-uncorrelated protocol cannot be converted into an "interesting" protocol by a black-box construction that invokes it few times.

Loosely speaking, both properties follow because an uncorrelated protocol is somewhat indistinguishable from one in which one party samples (T_κ, R_κ) on his own, sends them to the other party, and each of the parties runs $\mathsf{Decor}(T_\kappa, R_\kappa)$ and samples its output independently (party A samples $X \leftarrow U_{p_\mathsf{A}}$, and party B samples $Y \leftarrow U_{p_\mathsf{B}}$). The latter protocol can be easily attacked, and by indistinguishability, this attack also succeeds on the original protocol.

We prove the following classification theorem:

Theorem I.5 (Dichotomy theorem, informal)**.** *Let* $\pi = (\mathsf{A}, \mathsf{B})$ *be a* PPT *no-input, single-bit output two-party protocol. Then at least one of the following hold:*

- π *can be transformed into a key-agreement protocol (for infinitely many* $\kappa \in \mathbb{N}$*).*
- *For every constant* $\rho > 0$*,* π *is* ρ*-uncorrelated (for infinitely many* $\kappa \in \mathbb{N}$*).*

(Theorem I.5 is formally stated in Section III.)

The fact that we have statements on "infinitely many κ's" is unavoidable: it could be the case that on even κ, the protocol is a key agreement, and on odd κ, the protocol is trivial and performs no interaction.[2]

Once again, a caveat is the fact that we only get the result for $\rho = o(1)$ and not for negligible ρ (as is the standard in computational indistinguishability). It is an interesting open problem to extend our results to small ρ.

We demonstrate the usefulness of Theorem I.5 below. It is important to emphasize that the caveats in Theorem I.5 (and specifically, the limitation on ρ) do not matter for some of our suggested applications.

3) Perspective: Comparison to Impagliazzo and Luby Dichotomy Theorem: A celebrated result of Impagliazzo and Luby [9] is that distributional one-way functions imply one-way functions. This can be loosely stated this way:

Theorem I.6 (Impagliazzo and Luby [9], informal)**.** *Let* f *be a poly-time computable function, then at least one of the following holds:*

- f *can be transformed into a* one-way function*.*
- f *has a* PPT inverter *(for infinitely many* $\kappa \in \mathbb{N}$*). Namely, for every constant* c*, there exists a* PPT Inv *such that for infinitely many* $\kappa \in \mathbb{N}$ *the following holds: let* $X_\kappa \leftarrow U_\kappa$ *and* $T_\kappa = f(X_\kappa)$*. It holds that* (X_κ, T_κ) *is* $(\rho = \kappa^{-c})$*-close to* (X'_κ, T_κ)*, for* $X'_\kappa = \mathsf{Inv}(T_\kappa)$*.*

This theorem is celebrated for (at least) two reasons: first, it gives a dichotomy of poly-time functions (ruling out intermediate cases). Second, it gives a methodology to show that cryptographic primitives imply one-way functions: it is sufficient to show that the primitive has a component that cannot be inverted.

[2]However, the fact that we have "for infinitely many κ" in the two items, and not just in one, is an artifact of our proof technique, and it is natural to ask whether the result can be improved to have such a statement in only one of the items (as in the case of the Theorem of Impagliazzo and Luby [9] that we mention in the next section).

Our Theorem I.5 can be viewed as an analogous theorem for *two-party protocols*: either a protocol π implies *key-agreement* or it has a PPT *decorrelator*. Indeed, Theorem I.5 gives a dichotomy of two-party protocols, and in order to show that a protocol implies key-agreement, it is now sufficient to show that it is not uncorrelated. We will present applications of this methodology in Section I-B.

We remark that many of the applications of the Impagliazzo and Luby [9] classification do not require that ρ is small, and would have worked just the same for constant ρ.[3] Analogously, the fact that ρ is not very small in our theorem is sometimes unimportant in applications.

B. Consequences of our Dichotomy Theorem

We demonstrate the usefulness of our result by showing that it can be used to answer some open problems regarding differentially private protocols and coin flipping protocols. We now elaborate on these results.

1) Application to Differentially Private XOR: In a symmetric differentially private computation, the parties wish to compute a joint function of their inputs while keeping their inputs somewhat private. This is somewhat different from the classical client-server setting that is commonly addressed in the differentially privacy literature, where the the server, holding the data, answers the client's question while keeping the the data somewhat private.

A natural question is what assumptions are needed for such (symmetric) differentially private computation achieving certain level of accuracy. A sequence of work showed that for certain tasks, achieving high accuracy requires one-way functions [10, 11, 12, 13]; some cannot even be instantiated in the random oracle model [14]; and some cannot be black-box reduced to key agreement [15]. See Section I-D for more details on these results. Recently, see more details below, [16] have shown that a protocol for computing the XOR of *optimal* accuracy (i.e., that matches the client server accuracy for XOR) implies the existence of oblivious transfer protocols (that are also sufficient for this task).

We show that the existence of a symmetric differential private protocol for computing Boolean XOR that achieves *non-trivial accuracy* (i.e., better that what can be achieved when the eavesdropper is unbounded), implies the existence of a key-agreement protocol.

To prove the above result we consider protocols in which the two parties receive inputs $x, y \in \{0,1\}$ and each outputs a bit. A two-party protocol $\pi = (\mathsf{A}, \mathsf{B})$ for computing the XOR functionally is α-*correct*, if

$$\Pr\left[(\mathsf{A}(X), \mathsf{B}(Y)) = (X \oplus Y, X \oplus Y)\right] \geq \frac{1}{2} + \alpha$$

Such a protocol is (computationally) ε-*differentially private*, if for every x and efficient distinguisher D

$$\frac{\Pr\left[\mathsf{D}(\mathrm{view}^{\mathsf{A}}_{\pi}(x,0)) = 1\right]}{\Pr\left[\mathsf{D}(\mathrm{view}^{\mathsf{A}}_{\pi}(x,1)) = 1\right]} \in e^{\pm\varepsilon}$$

letting $\mathrm{view}^{\mathsf{A}}_{\pi}(x,y)$ being A's view in a random execution of $(\mathsf{A}(x), \mathsf{B}(y))$;[4] namely, the input of B remains somewhat private from the point of view of A. And the same should hold for the privacy of A.

The protocol has perfect *agreement*, if the parties' output is always the same (though might be different from the XOR). The results below are all stated with respect to such perfect agreement protocols, though the lower bound (including ours) allows disagreement in the magnitude of the differential privacy parameter ε.

Theorem I.7. *[Differentially private XOR to key agreement, informal] For every $\varepsilon > 0$, the existence of $21\varepsilon^2$-correct ε-differentially private protocol for computing XOR, implies the existence of an infinitely-often secure key-agreement protocol.*

The above dependency between ε and α is tight since a $\Theta(\varepsilon^2)$-correct, ε-differential private, protocol for computing XOR can be constructed (with information theoretic security) using the so-called *randomized response* approach Warner [18]. It improves, in the (ε, α) dependency aspect, upon Goyal et al. [16] who showed that, for some constant $c > 0$, a $c\varepsilon$-correct ε-differentially private XOR implies oblivious transfer, and upon Goyal et al. [13] who showed that $c\varepsilon^2$-correct ε-differentially XOR implies one-way functions.

Theorem I.7 extends for a weaker notion of privacy in which differential privacy is only guaranteed to hold against an *external* observer (assuming that the protocol's transcript explicitly states the parties common output). For such protocols, key agreement is a sufficient assumption.[5] Finally, we mention that since we use

[3] Loosely speaking, this happens whenever we have a cryptographic primitive where security can be amplified. For such protocols, a weaker version of [9] yields that either the primitive implies one-way functions or it has a PPT ρ-inverter for some constant $\rho > 0$. Then, using security amplification we obtain a more secure target primitive, such that an adversary that breaks the target primitive with small success $\rho' = \kappa^{-c}$ can be transformed into one that breaks the original protocol with large success $\rho > 0$.

[4] A more general definition allows also an additive error term. We address this definition in the full version [17].

[5] One party sends its *encrypted* input to the other party, who in turn computes the XOR of both inputs and publishes a noisy version (e.g., flipped with probability $\frac{1}{2} - \varepsilon$) of the outcome.

Theorem I.5, the reduction we use to prove Theorem I.7 is non black box in the adversary.

2) Application to Fair Coin Flipping: In a follow-up work, Haitner, Makriyannis, and Omri [1] used Theorem I.5 to prove that key-agreement is a necessary assumption for *two-party* r-round coin-flipping protocol of bias smaller than $1/\sqrt{r}$ (as long as r is independent of the security parameter). This partially answers a long-standing open question asking whether the existence of such two-party fair-coin flipping implies public-key cryptography. Previous to Haitner et al. [1] result, it was not even known that such protocols cannot be constructed in the random oracle model [19, 20].

In a very high level, [1] took the following approach. Assume key-agreement protocols do not exists, then the main result of this paper (Theorem I.5) yields that any protocol, and in particular an r-round coin-flipping protocol, has a decorrelator. Haitner et al. [1] showed how to use this decorrelator to mount an efficient variant of the Cleve and Impagliazzo [21] attack to bias the outcome of one of the parties by $1/\sqrt{r}$. (The bound of [1] only holds for constant-round protocols, since for the attack to go through the decorrelator's error has to be smaller than $1/\sqrt{r}$, which can only be achieved, at least using Theorem I.5, for constant r.)

C. Our Technique

1) A Competition of Forecasters: In this section we explain the high level idea behind the proof of Theorem I.3. Our goal is to understand "how X_κ and Y_κ are distributed from the point of view of a PPT algorithm that receives T_κ as input". For this purpose, we set up a competition between all PPT forecasters. We will use the winner in this competition as our forecaster.

Given a transcript t, a participant forecaster is required to output three numbers $p_{\mathsf{A}}, p_{\mathsf{B}|0}, p_{\mathsf{B}|1} \in [0,1]$. For every forecaster F and every $\kappa \in \mathbb{N}$, we associate a *price* $\mathsf{price}_\kappa(\mathsf{F})$. The minimal price is obtained by a forecaster that outputs $p_{\mathsf{A}} = \Pr[X_\kappa = 1 \mid T_\kappa = t]$ and $p_{\mathsf{B}|b} = \Pr[Y_\kappa = 1 \mid T_\kappa = t, X_\kappa = b]$. Note however, that a PPT forecaster might not be able to compute these quantities.

Existence of optimal forecasters. We will not give a precise definition of the price function in this overview. At this point, we observe that for every choice of price function where prices are in $[0,1]$, this competition has winners, in the following sense: we say that F is μ-optimal, if there exists an infinite subset $\mathcal{I} \subseteq \mathbb{N}$ such that $\mathsf{price}_\kappa(\mathsf{F}) \leq \mathsf{price}_\kappa(\mathsf{F}') + \mu$ for every other PPT F' and sufficiently large $\kappa \in \mathcal{I}$. This intuitively says that F cannot be significantly improved on the subset $\mathcal{I}$. We claim that for every constant $\mu > 0$ there exists a μ-optimal forecaster.

This follows as we can imagine the following iterative process: we start with some forecaster F and $\mathcal{I} = \mathbb{N}$. At each step, either F cannot be improved by μ, on infinitely many $\kappa \in \mathcal{I}$ (which means that F is μ-optimal), or else, there exists an infinite $\mathcal{I}' \subseteq \mathcal{I}$, and a forecaster F' that improves F by μ in $\mathcal{I}'$. In that case we set $\mathcal{I} = \mathcal{I}'$, $\mathsf{F} = \mathsf{F}'$ and continue. It is clear that at every iteration we improve the price by μ, and this can happen only $1/\mu$ times, this process shows the existence of a μ-optimal forecaster.[6]

Indistinguishability for optimal forecasters: Let F be a μ-optimal forecaster, we can use F to produce a forecasted distribution (as in Theorem I.3). Namely, given $t \leftarrow T_\kappa$, we apply $\mathsf{F}(t)$ to compute $p_{\mathsf{A}}(t), p_{\mathsf{B}|0}(t), p_{\mathsf{B}|1}(t)$, and use these forecasts to produce a distribution (X'_κ, Y'_κ) by sampling $X'_\kappa \leftarrow U_{p_{\mathsf{A}}(t)}$ and $Y'_\kappa \leftarrow U_{p_{\mathsf{B}|X'_\kappa}(t)}$. This can indeed be done in poly-time (and in this informal discussion we omit the additional random input r).

We show that if a PPT D distinguishes $(X_\kappa, Y_\kappa, T_\kappa)$ from $(X'_\kappa, Y'_\kappa, T_\kappa)$, then D can be used to construct an improved PPT F' whose $\mathsf{price}_\kappa(\mathsf{F}')$ is smaller than $\mathsf{price}_\kappa(\mathsf{F})$ by some function of the distinguishing advantage ρ.[7] This is a contradiction to the μ-optimality of F if ρ is sufficiently large.

At the risk of getting too technical, let us try to explain how this argument works. The reader can skip to Section I-C2 that does not depend on the next paragraph.

It is helpful to note that $(X'_\kappa, Y'_\kappa, T_\kappa)$ can be seen as $(X'_\kappa, g(X'_\kappa, T_\kappa), T_\kappa)$ where g is a probabilistic function. It is helpful to consider the hybrid distribution $H = (X_\kappa, g(X_\kappa, T_\kappa), T_\kappa)$. Using a hybrid argument, we have that one of the following happens:

- D distinguishes $(X'_\kappa, g(X'_\kappa, T_\kappa), T_\kappa)$ from $H = (X_\kappa, g(X_\kappa, T_\kappa), T_\kappa)$. This induces a D' that distinguishes $(X'_\kappa, T_\kappa) = (U_{p_{\mathsf{A}}(T_\kappa)}, T_\kappa)$ from (X_κ, T_κ)
- D distinguishes $(X_\kappa, Y_\kappa, T_\kappa)$ from $H = (X_\kappa, g(X_\kappa, T_\kappa), T_\kappa)$. This gives that there exists $b \in \{0,1\}$, and a D' such

[6] A drawback of the argument above is that it only works for constant $\mu > 0$. The distinguishing parameter ρ, will be selected to be say $\mu^{1/10}$, and this is why we only get the result in Theorem I.1, Theorem I.3 and Theorem I.5 for constant $\rho > 0$. Consequently, if we could guarantee the existence of an optimal forecaster for smaller μ, we will immediately improve our results. Another drawback is that this argument only works on some infinite subset $\mathcal{I} \subseteq \mathbb{N}$ and this is the reason we get "for infinitely many κ" in our theorems. The remainder of our machinery does not require these caveats.

[7] This overall approach is also taken by some proofs of the "leakage simulation lemma" that was mentioned in remark I.2.

that D' distinguishes $(Y_\kappa, T_\kappa)|_{X_\kappa=b}$ from $(Y'_\kappa, T_\kappa)|_{X_\kappa=b} = (U_{p_{\mathsf{B}|b}(T_\kappa)}, T_\kappa)|_{X_\kappa=b}$.

We have made progress, in that in both cases we have reduced the number of variables from three to two, while obtaining a distinguisher D' that distinguishes between a "real distribution" and a "forecasted distribution". Let's assume without loss of generality that the first case happens. Note that D' obtains no distinguishing advantage on t if $\mathsf{D}'(t,0) = \mathsf{D}'(t,1)$.

Assume without loss of generality that D' is more likely to answer one on the real distribution than on the forecasted distribution. This intuitively means that on average, given a $t \leftarrow T_\kappa$, by trying out $\mathsf{D}'(t,0)$ and $\mathsf{D}'(t,1)$ we can figure out what "D' thinks" is more likely to be the bit of the forecasted distribution, and improve the forecast of F. Specifically,

- If $\mathsf{D}'(t,0) = \mathsf{D}'(t,1)$ then D does not gain on t, and we won't modify the forecast of F on t.
- If $\mathsf{D}'(t,1) = 1$ and $\mathsf{D}'(t,0) = 0$ then "D' thinks" that F's forecast for $\Pr[X_\kappa = 1 \mid T_\kappa = t]$ was too low, and it makes sense to increase it.
- If $\mathsf{D}'(t,0) = 1$ and $\mathsf{D}'(t,1) = 0$ then "D' thinks" that F's forecast for $\Pr[X_\kappa = 1 \mid T_\kappa = t]$ was too high, and it makes sense to decrease it.

By using this rationale, we can guarantee that the modified forecast (which can be computed in poly-time) improves upon F's forecast (at least on average $t \leftarrow T_\kappa$). We choose the price function carefully, so that this translates to a significant reduction in price, contradicting F's μ-optimality.

2) Using the Forecaster to Prove the Dichotomy: In this section we explain how to prove Theorem I.5 given Theorem I.3. Given a protocol π, we consider the optimal forecaster F from Theorem I.3 (which is F from the previous section). We will once again oversimplify and ignore the random coin string r. Recall that on input $t \leftarrow T_\kappa$, F computes three numbers $p_{\mathsf{A}}, p_{\mathsf{B}|0}, p_{\mathsf{B}|1}$, and induces a forecasted distribution $(X'_\kappa, Y'_\kappa, T_\kappa)$ that is ρ-indistinguishable from $\pi(1^\kappa) = (X_\kappa, Y_\kappa, T_\kappa)$, and furthermore, that $\Pr[X'_\kappa = 1 \mid T_\kappa = t] = p_{\mathsf{A}}$, and $\Pr[Y'_\kappa = 1 \mid T_\kappa = t, X'_\kappa = b] = p_{\mathsf{B}|b}$.

Note that if for every possible transcript T_κ it holds that $\mathsf{F}(T_\kappa)$ produces $p_{\mathsf{B}|0} = p_{\mathsf{B}|1}$, then by setting $\mathsf{Decor}(T_\kappa) = (p_{\mathsf{A}}, p_{\mathsf{B}|0})$ we obtain a ρ-decorrelator. Increasing ρ slightly, this also extends to the case where with high probability over $t \leftarrow T_\kappa$, $p_{\mathsf{B}|0}$ is "not far" from $p_{\mathsf{B}|1}$. If the condition above does not hold, we will want to use F to convert π into a key-agreement π'. We can use the forecaster as follows (and in fact this methodology seems quite general):

- When using π as a component in π', we can imagine that the output distribution of π is the forecasted distribution. More precisely, we are allowed to work in the following "information theoretic setting": party A receives X'_κ, party B receives Y'_κ and the adversary receives T_κ. Note that X'_κ and Y'_κ have *information theoretic uncertainty* given T_κ, and so we can now apply techniques and protocols from the information theoretic world. Information theoretic security in the latter setup translates into computational security in the original setup (with an additive loss of ρ).
- Consequently, we can use information theoretic methods to construct key-agreement to construct π' from the "simulation of" π. This then translates into computational security (with a constant loss ρ in security). By using security amplification for key agreement [2], we can amplify this security to give key-agreement with standard choices of secrecy and agreement. (This demonstrates that the fact that ρ cannot be made negligible, is not a problem, and we can get computational security with respect to negligible functions).[8]
- Moreover, when we work in the information theoretic setup, the honest parties are allowed to see T, and run the forecaster (that runs in polynomial time). This is in some sense "non-black-box" as the parties gain access to specific properties of the probability space $(X'_\kappa, Y'_\kappa, T_\kappa)$ by applying the forecaster on T_κ and can use its outputs $p_{\mathsf{A}}, p_{\mathsf{B}|0}, p_{\mathsf{B}|1}$ when constructing information theoretic key-agreement.

The one-sided von-Neumann protocol. The information theoretic setup described above can be thought of as follows: whenever the two parties invoke the protocol π, we can imagine that A receives variable X'_κ, B receives variable Y'_κ and the eavesdropper receives T_κ. Moreover, A and B can use F to compute all probabilities in the probability space $(X'_\kappa, Y'_\kappa)|_{T_\kappa=t}$. We now explain how to construct a key-agreement protocol.

- The two parties receive X'_κ and Y'_κ by running π, they also receive the transcript T_κ.

[8]Continuing the analogy to computational entropy, this approach can be thought of as analogous to the constructions of Håstad et al. [22] and following work [23, 24] of pseudorandom generators from one-way functions. Indeed, a key idea in these works is that of "computational entropy" which given a distribution X (with low real entropy) presents an indistinguishable distribution X' (with a lot of entropy). This allows the construction to apply "information theoretic tools" (e.g., randomness extractors) on X and argue that the result is pseudorandom, by imagining that the information theoretic tools are applied on X'. Continuing this analogy, it is often the case that "pulling the result back" to the computational realm, suffers a significant loss in security, and computational amplification of security is performed to obtain stronger final results.

- The two parties use F to compute $\mathsf{F}(T_\kappa) = (p_\mathsf{A}, p_{\mathsf{B}|0}, p_{\mathsf{B}|1})$. Party A samples an independent random variable $X''_\kappa \leftarrow U_{p_\mathsf{A}}$ (that is, an independent variable that is distributed like X'_κ).
- The two parties can use the von-Neumann trick [25] to obtain a shared random coin as follows: A informs B whether $X'_\kappa = X''_\kappa$.
  - If $X'_\kappa = X''_\kappa$, the parties output independent uniform bits.
  - If $X'_\kappa \neq X''_\kappa$, party A outputs X'_κ and party B outputs Y'_κ.

For every $t \in \mathrm{Supp}(T_\kappa)$,
$\Pr[X'_\kappa = 1, X''_\kappa = 0 \mid T_\kappa = t] =$
$\Pr[X'_\kappa = 0, X''_\kappa = 1 \mid T_\kappa = t]$, and consequently:

$$\Pr[X'_\kappa = 1 \mid T_\kappa = t, X'_\kappa \neq X''_\kappa] = \tfrac{1}{2}.$$

This means that this information theoretic key-agreement protocol has perfect secrecy. We now consider the agreement property. Recall that we are assuming that X'_κ and Y'_κ are correlated conditioned on some fixings of $t \leftarrow T_\kappa$. This can be used to show that the output bits of our protocol are correlated. (In the actual proof, we need a slightly more complicated protocol which also relies on $p_{\mathsf{B}|0}, p_{\mathsf{B}|1}$ to guarantee agreement, rather than just correlation).

Thus, this protocol is an information theoretic key agreement with secrecy $s = 0$ and agreement $a > 0$. By controlling the parameters, the gap between agreement and secrecy can be made significantly larger than ρ so that we can implement our overall plan.

D. Related Work

We now discuss some related work that was not yet mentioned in the previous sections.

Characterization of two-party computations: The most relevant result is the classification of two-party protocols in the random oracle model (ROM) given in Haitner, Omri, and Zarosim [14]. In this model, the parties and the adversary are given an oracle access to a common random function, that they can query a limited number of times. The ROM is typically used to analyze the security of cryptographic protocols in an idealistic model, and to prove impossibility results for such protocols. In particular, an impossibility result in the ROM yields that the security of protocol in consideration cannot be based in a black-box way on one-way functions or collision resistant hash functions.

In their seminal work Impagliazzo and Rudich [26] proved that a key-agreement protocols cannot be constructed in the ROM. That is, they show that for any query efficient protocol (i.e., polynomial query complexity) in the ROM, there exists a query efficient eavesdropper that finds the common key. Haitner et al. [14], using techniques developed by Barak and Mahmoody [27], showed that for any no-input two-party random oracle protocol there exists a query efficient mapping into a *no oracle* protocol such that the distribution of the transcript and parties output are essentially the same. Since in the non-input setting the parties output are always uncorrelated (as far as no input protocol are concerned), the existence of such efficient mapping also tell us that interesting correlation cannot exits in the ROM. Our main result capturing the minimal assumption for (output) correlation in actual protocol (rather than the hypothetical random oracle, model) is in a sense the non black-box version of the above characterization.

Other relevant results are amplifications of weak primitives into a full-fledge ones, and in particular that of key-agreement [2] and obvious transfer [28, 29, 30]. Such results aims to classify the different functionalities into groups of equivalent expression power, and many of them are achieved via the study of information-theoretic two-party correlation (also known as, channels): each party, including the observer, is given random variable from a predetermined distribution, and their goal is to use them to achieve a cryptographic task (i.e., key agreement). Our result demonstrates that going solely through the above information theoretic paradigm, is sometimes a too limited approach.

Minimal assumptions for differentially private symmetric computation: An accuracy parameter α is *trivial* with respect to a given functionality f and differential privacy parameter ε, if a protocol computing f with such accuracy and privacy exists information theoretically (i.e., with no computational assumptions). The accuracy parameter is called *optimal*, if it matches the bound achieved in the client-server model. Gaps between the trivial and optimal accuracy parameters have been shown in the multiparty case for count queries [10, 11] and in the two-pary case for inner product and hamming distance functionalities [12]. [14] showed that the same holds also when a random oracle is available to the parties, implying that non-trivial protocols (achieving non-trivial accuracy) for computing these functionalities cannot be black-box reduced to one-way functions. [13] initiated the study of Boolean functions, showing a gap between the optimal and trivial accuracy for the XOR or the AND functionalities, and that non-trivial protocols imply one-way functions. [15] have shown that optimal protocols for computing the

XOR or AND, cannot be black-box reduced to key agreement. Recently, [16] have shown that optimal protocols for computing the XOR imply oblivious transfer.

Paper Organization

Standard notions and definitions are given in Section II. In Section III we formally define simulators, forecasters, decorrelators, and uncorrelated protocols, and state there our main results. This version is an extended abstract. Due to space limitations it only contains a high level description of our results and techniques. The reader is referred to the full version [17] for precise details and full proofs.

II. Preliminaries

Distributions and random variables: For $0 \leq p < 1$, let U_p denote the distribution of a biased coin which is one with probability p. Given jointly distributed random variables X, Y and $x \in \mathcal{X}$, let $Y|_{X=x}$ denote the distribution of Y induced by the conditioning $X = x$ (set arbitrarily if $\Pr[X = x] = 0$). The *statistical distance* between two random variables X and Y over a finite set $\mathcal{U}$, denoted $\mathrm{SD}(X, Y)$, is defined as $\frac{1}{2} \cdot \sum_{u \in \mathcal{U}} |\Pr[X = u] - \Pr[Y = u]|$.

Computational indistinguishability (and infinitely-often variant): We first need the following variance of computational indistinguishability where the distinguishing advantage ρ is a parameter. We also discuss infinitely often indistinguishability.

Definition II.1 (Computational indistinguishability with a parameter ρ)**.** *For a function $\rho : \mathbb{N} \to \mathbb{R}$, two distribution ensembles $X = \{X_\kappa\}_{\kappa \in \mathbb{N}}$, $Y = \{Y_\kappa\}_{\kappa \in \mathbb{N}}$ are ρ-*indistinguishable*, denoted $X \overset{\mathrm{C}}{\approx}_\rho Y$, if for every* PPTM D*, for every sufficiently large $\kappa \in \mathbb{N}$,*

$$|\Pr[\mathsf{D}(1^\kappa, X_\kappa) = 1] - \Pr[\mathsf{D}(1^\kappa, Y_\kappa) = 1]| \leq \rho(\kappa)$$

We omit 1^κ when the security parameter κ is clear from the context.

*For an infinite set $\mathcal{I} \subseteq \mathbb{N}$, the two ensembles X and Y are ρ-*indistinguishable in $\mathcal{I}$*, denoted $X \overset{\mathrm{C}}{\approx}_{\rho,\mathcal{I}} Y$, if the condition above holds when replacing the condition "for every sufficiently large $\kappa \in \mathbb{N}$" with "for every sufficiently large $\kappa \in \mathcal{I}$". We say that X and Y are* io-ρ-indistinguishable*, if there exists an infinite set $\mathcal{I} \subseteq \mathbb{N}$ such that X and Y are ρ-indistinguishable in $\mathcal{I}$.*

A. Protocols

We will mainly focus on no-input two-party protocol single-bit PPT output protocol: the two PPT parties only input is the common security parameter, given in unary, and at the end of the protocol each party output a single bit. Throughout, we assume without loss of generality that the transcript contains 1^κ as the first message.

Let $\pi = (\mathsf{A}, \mathsf{B})$ be such two-party single-bit protocol. For $\kappa \in \mathbb{N}$, let π_κ be protocol π with the common security parameter fixed (i.e., hardwired) to 1^κ. Protocol π has transcript length $m(\cdot)$, if the transcript of π_κ is of length at most $m(\kappa)$. We will assume without loss of generality that the protocol of consideration have fixed transcript length per security parameter. For $\kappa \in \mathbb{N}$, let $(X^\pi_\kappa, Y^\pi_\kappa, T^\pi_\kappa)$ denote the A and B outputs respectively, and the execution transcript, in a random execution of π_κ. We sometimes denote this triplet of random variables by $\pi(1^\kappa)$.

1) Key-Agreement Protocols (and Infinitely Often Variant): We focus on single bit key agreement protocols.

Definition II.2 (Key-agreement protocols)**.** *A* PPT *single-bit output two-party protocol $\pi = (\mathsf{A}, \mathsf{B})$ is a secure* key-agreement *with respect to a set $\mathcal{I} \subseteq \mathbb{N}$, if the following hold for κ's in $\mathcal{I}$.*

- Agreement: $\Pr[X^\pi_\kappa = Y^\pi_\kappa] \geq 1 - \mathrm{neg}(\kappa)$.
- Secrecy: *For every* PPT E, $\Pr[\mathsf{E}(T^\pi_\kappa) = X^\pi_\kappa] \leq 1/2 + \mathrm{neg}(\kappa)$.

Let $s, a : \mathbb{N} \mapsto \mathbb{R}$ be functions. A PPT *single-bit output two-party protocol $\pi = (\mathsf{A}, \mathsf{B})$ is an (s, a)-*key agreement *if the following two conditions hold.*

- Agreement: $\Pr[X^\pi_\kappa = Y^\pi_\kappa] \geq 1/2 + a(\kappa)$ *for sufficiently large $\kappa \in \mathbb{N}$.*
- Secrecy: *For every* PPTM E, $\Pr[\mathsf{E}(T^\pi_\kappa) = X^\pi_\kappa] \leq 1/2 + s(\kappa)$ *for sufficiently large $\kappa \in \mathbb{N}$.*

If we omit (s, a) then we mean that the key-agreement has standard choices for secrecy and agreement, namely it is a $(\mathrm{neg}(\kappa), 1/2 - \mathrm{neg}(\kappa))$-key agreement.

*Protocol π is an (s, a)-*key agreement in an infinite set *$\mathcal{I} \subseteq \mathbb{N}$, if the security and agreement conditions hold when replacing $\mathbb{N}$ above with $\mathcal{I}$. The protocol is an* io-(s, a)-key agreement *if there exits an infinite set $\mathcal{I} \subseteq \mathbb{N}$ for which the protocol is an (s, a)-key agreement in $\mathcal{I}$.*

III. Classification of Boolean Two-Party Protocols

In this section we formally define simulator, forecasters, decorelators and uncorrelated protocols discussed in Section I, and formally state the main results of this paper. Throughout this section we focus on no-input, single-bit output, two-party protocols.

A. Simulators and Forecasters

The results of this section holds for *any* no-input, single-bit output two-party protocols, even inefficient ones.

1) Simulators: Recall that a simulator seeing the protocol transcript, outputs a pair of bits that looks indistinguishable from the parties' real outputs, from the point of view of an efficient distinguisher that sees only the protocol's transcript. We now define this concept precisely, and state our results.

Definition III.1 (Simulator)**.** *A* simulator *is a* PPT *algorithm that on input in* $(1^\kappa, t) \in 1^* \times \{0,1\}^*$ *outputs two bits.*

We associate the following two distribution ensembles with a two-party protocol and a simulator.

Definition III.2 (Real and simulated distributions)**.** *Let* $\pi = (\mathsf{A}, \mathsf{B})$ *be a single-bit output two-party protocol, and let* Sim *be a simulator. We define the* real and simulated distribution ensembles REAL^π *and* $\text{SML}^{\pi,\mathsf{Sim}}$ *as follows. For* $\kappa \in \mathbb{N}$*, let* X_κ*,* Y_κ *and* T_κ *be the parties' outputs and protocol transcript in a random execution of* π_κ*. Then*

- Real: $\text{REAL}^\pi_\kappa = (X_\kappa, Y_\kappa, T_\kappa)$.
- Simulated: $\text{SML}^{\pi,\mathsf{Sim}}_\kappa = (\mathsf{Sim}_\kappa(T_\kappa), T_\kappa)$.

(Recall that $\mathsf{Sim}_\kappa(t)$ denotes the output of Sim on input $(1^\kappa, t)$.)

The following theorem states that every single-bit output two-party protocol (even inefficient one) has a simulator.

Theorem III.3 (Existence of simulators)**.** *For every single-bit output, two-party protocol* π*,* $\rho > 0$ *and infinite set* $\mathcal{I} \subseteq \mathbb{N}$*, there exist a simulator* Sim *and an infinite set* $\mathcal{I}' \subseteq \mathcal{I}$ *such that*

$$\text{REAL}^\pi \overset{\text{C}}{\approx}_{\rho,\mathcal{I}'} \text{SML}^{\pi,\mathsf{Sim}}.$$

Theorem III.3 is an immediate corollary of the existence of forecasters theorem given below.

2) Forecasters: A forecaster seeing the protocol transcript, outputs a *description* of a two bits distribution, that looks indistinguishable from the parties' real outputs, from the point of view of an an efficient distinguisher that sees only the protocol's transcript. Thus, a forecaster is a specific method for constructing simulators: the resulting simulator outputs the two bits according the the distribution described by the forecaster.

Definition III.4 (Forecasters)**.** *A* forecaster F *is a* PPTM *that on input* $(1^\kappa, t) \in 1^* \times \{0,1\}^*$*, outputs a triplet in* $[0,1]^3$*. We use* $\mathsf{F}(1^\kappa, t; r)$ *to denote the instantiation of* $\mathsf{F}(1^\kappa, t)$ *when using the string* r *as random coins.*[9]

We associate the following two distribution ensembles with a two-party protocol and a forecasters. To define these distributions, we associate triplets in $[0,1]^3$ with distribution over $\{0,1\}^2$ in the following way.

Notation III.5. *For* $p = (p_\mathsf{A}, p_{\mathsf{B}|0}, p_{\mathsf{B}|1}) \in [0,1]^3$*, let* U_p *denote the random variable over* $\{0,1\}^2$ *defined by* $\Pr[U_p = (x,y)] = \Pr[U_{p_\mathsf{A}} = x] \cdot \Pr\left[U_{p_{\mathsf{B}|x}} = y\right]$*. For* $p = (p_\mathsf{A}, p_\mathsf{B}) \in [0,1]^2$*, let* U_p *denote the random variable* $U_{(p_\mathsf{A}, p_\mathsf{B}, p_\mathsf{B})}$.

With this notation, the variable $U_p = (X', Y')$ is composed of two random variables such that $\Pr[X' = 1] = p_\mathsf{A}$ and for $b \in \{0,1\}$, $\Pr[Y' = 1 | X' = b] = p_{\mathsf{B}|b}$. In particular, if $p_{\mathsf{B}|0} = p_{\mathsf{B}|1}$ then (X', Y') are independent.

Definition III.6 (Real and forecasted distributions)**.** *Let* $\pi = (\mathsf{A}, \mathsf{B})$ *be a single-bit output two-party protocol and let* F *be a forecaster. We define the* real and forecasted distribution ensembles $\text{REAL}^{\pi,\mathsf{F}}$ *and* $\text{FST}^{\pi,\mathsf{F}}$ *as follows. For* $\kappa \in \mathbb{N}$*, let* X_κ*,* Y_κ *and* T_κ *be the parties' outputs and protocol transcript in a random execution of* π_κ*, and let* R_κ *be a uniform and independent string whose length is the (maximal) number of coins used by* F_κ*. Then,*

- Real: $\text{REAL}^{\pi,\mathsf{F}}_\kappa = (X_\kappa, Y_\kappa, T_\kappa, R_\kappa)$.
- Forecasted: $\text{FST}^{\pi,\mathsf{F}}_\kappa = (U_p, T_\kappa, R_\kappa)$ *for* $p = \mathsf{F}_\kappa(T_\kappa; R_\kappa) = (p_\mathsf{A}, p_{\mathsf{B}|0}, p_{\mathsf{B}|1})$.

(Recall that $\mathsf{F}_\kappa(t; r)$ denotes the output of F on input $(1^\kappa, t)$ when using randomness r.)

The computational distance between the real and forecasted distribution measures how well the forecaster realizes the real distribution, in the eyes of a computationally bounded distinguisher.

Definition III.7 (Forecaster indistinguishability)**.** *A forecaster* F *is* $(\rho, \mathcal{I})$-indistinguishable*, for* $\rho > 0$ *and infinite subset* $\mathcal{I} \subseteq \mathbb{N}$*, with respect to protocol* π*, if*

$$\text{REAL}^{\pi,\mathsf{F}} \overset{\text{C}}{\approx}_{\rho,\mathcal{I}} \text{FST}^{\pi,\mathsf{F}}.$$

That is, for sufficiently large $\kappa \in \mathcal{I}$, the forecasted and real distributions are ρ indistinguishable for polytime distinguishers.

The following theorem states that every single-bit output two-party protocol (even inefficient one) has a forecaster.

[9] Since we only care about PPT algorithms, we will implicitly assume that the number of coins used by them on a given security parameter is efficiently computable.

Theorem III.8 (Existence of forecasters)**.** *For every single-bit output two-party protocol π, $\rho > 0$ and infinite set $\mathcal{I} \subseteq \mathbb{N}$, there exist a forecaster* F *and an infinite set $\mathcal{I}' \subseteq \mathcal{I}$, such that* F *is* $(\rho, \mathcal{I}')$-indistinguishable *with respect to π.*

B. Decorrelators and the Dichotomy Theorem

In the introduction we explained the concept of decorrelators and uncorrelated protocols, in informal Definition I.4. We now repeat the definition using more precise language.

Definition III.9 (Decorrelators)**.** *A* decorrelator Decor *is a* PPTM *that on input $(1^\kappa, t) \in 1^* \times \{0,1\}^*$, outputs two numbers in $[0,1]$. We use $\mathsf{Decor}(1^\kappa, t; r)$ to denote the instantiation of $\mathsf{Decor}(1^\kappa, t)$ when using the string r as random coins.*

We associate the following two distribution ensembles with a two-party protocol and a decorrelator.

Definition III.10 (Real and uncorrelated distributions)**.** *Let $\pi = (\mathsf{A}, \mathsf{B})$ be a single-bit output two-party protocol, and let* Decor *be a decorrelator. We define the real and uncorrelated distribution ensembles $\text{REAL}^{\pi,\mathsf{Decor}}$ and $\text{UCR}^{\pi,\mathsf{Decor}}$ as follows. For $\kappa \in \mathbb{N}$, let X_κ, Y_κ and T_κ be the parties' outputs and protocol transcript in a random execution of π_κ, and let R_κ be a uniform and independent string whose length is the (maximal) number of coins used by Decor_κ (see Footnote 9). Then,*

- Real: $\text{REAL}^{\pi,\mathsf{Decor}}_\kappa = (X_\kappa, Y_\kappa, T_\kappa, R_\kappa)$.
- Uncorrelated: $\text{UCR}^{\pi,\mathsf{Decor}}_\kappa = (U_{p_\mathsf{A}}, U_{p_\mathsf{B}}, T_\kappa, R_\kappa)_{(p_\mathsf{A},p_\mathsf{B})=\mathsf{Decor}_\kappa(T_\kappa;R_\kappa)}$.

(Recall that $\mathsf{Decor}_\kappa(t;r)$ denotes the output of Decor on input $(1^\kappa, t)$ when using randomness r.) *Uncorrelated protocols*, are those protocols for which the above distributions are computational close.

Definition III.11 (Uncorrelated protocols)**.** *Let $\pi = (\mathsf{A}, \mathsf{B})$ be a single-bit output two-party protocol, let $\rho > 0$ and $\mathcal{I} \subseteq \mathbb{N}$. Decorrelator* Decor *is a* $(\rho, \mathcal{I})$-decorrelator for π, if

$$\text{REAL}^{\pi,\mathsf{Decor}} \overset{\text{C}}{\approx}_{\rho,\mathcal{I}} \text{UCR}^{\pi,\mathsf{Decor}}.$$

Protocol π is $(\rho, \mathcal{I})$-uncorrelated, if it has a $(\rho, \mathcal{I})$-decorrelator. Protocol π is io-ρ-uncorrelated, if there exists an infinite set $\mathcal{I} \subseteq \mathbb{N}$ such that π is $(\rho, \mathcal{I})$-uncorrelated.

This is the formal statement of our main theorem (that restates Theorem I.5 from Section I).

Theorem III.12 (Dichotomy of two-party protocols)**.** *For every* PPT *single-bit output two-party protocol, one of the following holds:*

- *For every constant $\rho > 0$ and every infinite $\mathcal{I} \subseteq \mathbb{N}$, there exists an infinite set $\mathcal{I}' \subseteq \mathcal{I}$ such that the protocol is ρ-uncorrelated in $\mathcal{I}'$.*
- *Exists a two-party io key-agreement protocol.*

IV. Conclusion

V. Conclusion and open problems

In this paper we prove a dichotomy theorem (Theorem III.12) for PPT two-party protocols with no inputs and single bit outputs: every such protocol is either ρ-uncorrelated (for every $\rho > 0$, on infinitely many κ) or it implies key agreement (on infinitely many κ). The theorem comes with caveats: it has "infinitely many κ" in both statements (rather than just in one), and it only achieves constant $\rho > 0$. A natural open problem is to remove these caveats from Theorem III.12 (it is natural to try to first try to remove the caveats from Theorem III.3 and Theorem III.8).

Other interesting open problems are related to our applications. What is the minimal assumption needed for differentially private computation of the XOR (and other natural) functions? (This question can be asked for various ranges of accuracy and differential privacy parameters). Can the coin tossing result of [1] be extended to hold for a number of rounds that depends on the security parameter?

Acknowledgment

Iftach Haitner was supported by an ERC starting grant 638121; and is a member of the Israeli Center of Research Excellence in Algorithms (ICORE) and the Check Point Institute for Information Security. Kobbi Nissim was supported by NSF grant CNS-1565387. Eran Omri was supported by ISF grants 544/13 and 152/17. Ronen Shaltiel was supported by ISF grant 1628/17. Jad Silbak was supported by an ERC starting grant 638121. Most of this research was done while Jad Silbak was a student at the University of Haifa, and supported by ISF grant 1628/17.

We are very grateful to Omer Reingold and Guy Rothblum for very useful discussions.

References

[1] I. Haitner, N. Makriyannis, and E. Omri, "On the complexity of fair coin flipping," www.cs.tau.ac.il/~iftachh/papers/CFtoKA/TwoPartyCoinFlipToKA.pdf, 2018, manuscript.

[2] T. Holenstein, "Strengthening key agreement using hard-core sets - PhD thesis," 2006.

[3] L. Trevisan, M. Tulsiani, and S. Vadhan, "Regularity, boosting, and efficiently simulating every high-entropy distribution," in *Computational Complexity, 2009. CCC'09. 24th Annual IEEE Conference on*. IEEE, 2009, pp. 126–136.

[4] D. Jetchev and K. Pietrzak, "How to fake auxiliary input," in *TCC*. Springer, 2014, pp. 566–590.

[5] S. Vadhan and C. J. Zheng, "A uniform min-max theorem with applications in cryptography," in *CRYPTO 2013*. Springer, 2013, pp. 93–110.

[6] M. Skorski, "Simulating auxiliary inputs, revisited," in *TCC*. Springer, 2016, pp. 159–179.

[7] M. Skórski, "A subgradient algorithm for computational distances and applications to cryptography." *IACR ePrint Archive*, vol. 2016, p. 158, 2016.

[8] Y.-H. Chen, K.-M. Chung, and J.-J. Liao, "On the complexity of simulating auxiliary input," in *Eurocrypt*. Springer, 2018, pp. 371–390.

[9] R. Impagliazzo and M. Luby, "One-way functions are essential for complexity based cryptography," in *FOCS*, 1989, pp. 230–235.

[10] A. Beimel, K. Nissim, and E. Omri, "Distributed private data analysis: Simultaneously solving how and what," in *CRYPTO*, 2008, pp. 451–468.

[11] T. H. Chan, E. Shi, and D. Song, "Optimal lower bound for differentially private multi-party aggregation," in *ESA*, 2012, pp. 277–288.

[12] A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. P. Vadhan, "The limits of two-party differential privacy," *ECCC*, p. 106, 2011, preliminary version in *FOCS'10*.

[13] V. Goyal, I. Mironov, O. Pandey, and A. Sahai, "Accuracy-privacy tradeoffs for two-party differentially private protocols," in *CRYPTO*, 2013, pp. 298–315.

[14] I. Haitner, E. Omri, and H. Zarosim, "Limits on the usefulness of random oracles," *Journal of Cryptology*, vol. 29, no. 2, pp. 283–335, 2016.

[15] D. Khurana, H. K. Maji, and A. Sahai, "Black-box separations for differentially private protocols," in *ASIACRYPT*, 2014, pp. 386–405.

[16] V. Goyal, D. Khurana, I. Mironov, O. Pandey, and A. Sahai, "Do distributed differentially-private protocols require oblivious transfer?" in *LIPIcs*, vol. 55. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[17] I. Haitner, K. Nissim, E. Omri, R. Shaltiel, and J. Silbak, "Computational two-party correlation." Technical Report TR18-071, ECCC, 2018.

[18] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[19] D. Dachman-Soled, Y. Lindell, M. Mahmoody, and T. Malkin, "On the black-box complexity of optimally-fair coin tossing," in *TCC*, vol. 6597, 2011, pp. 450–467.

[20] D. Dachman-Soled, M. Mahmoody, and T. Malkin, "Can optimally-fair coin tossing be based on one-way functions?" in *TCC*, ser. Lecture Notes in Computer Science, Y. Lindell, Ed., vol. 8349. Springer, 2014, pp. 217–239.

[21] R. Cleve and R. Impagliazzo, "Martingales, collective coin flipping and discrete control processes (extended abstract)," http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.1797, 1993.

[22] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, "A pseudorandom generator from any one-way function," *SIAM Journal on Computing*, vol. 28, no. 4, pp. 1364–1396, 1999, preliminary versions in *STOC'89* and *STOC'90*.

[23] I. Haitner, O. Reingold, and S. Vadhan, "Efficiency improvements in constructing pseudorandom generators from one-way functions," *SIAM Journal on Computing*, vol. 42, no. 3, pp. 1405–1430, 2013, special Issue on *STOC '10*.

[24] S. Vadhan and C. J. Zheng, "Characterizing pseudoentropy and simplifying pseudorandom generator constructions," in *STOC*, 2012, pp. 817–836.

[25] J. von Neumann, "Various techniques used in connection with random digits," *Applied Math Series*, vol. 12, pp. 36–38, 1951.

[26] R. Impagliazzo and S. Rudich, "Limits on the provable consequences of one-way permutations," in *STOC*. ACM Press, 1989, pp. 44–61.

[27] B. Barak and M. Mahmoody, "Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle," in *CRYPTO*, 2009, pp. 374–390.

[28] I. Haitner, "Implementing oblivious transfer using collection of dense trapdoor permutations," in *TCC*, 2004, pp. 394–409.

[29] J. Wullschleger, "Oblivious-transfer amplification," in *EUROCRYPT*, 2007, pp. 555–572.

[30] Crepeau and Kilian, "Weakening security assumptions and oblivious transfer," in *CRYPTO*, 1988.