# A Full Characterization of Completeness for Two-party Randomized Function Evaluation*

Daniel Kraschewski † Hemanta K. Maji ‡ Manoj Prabhakaran § Amit Sahai ¶

**Abstract**

We settle a long standing open problem which has pursued a full characterization of completeness of (potentially randomized) finite functions for 2-party computation that is secure against active adversaries. Since the first such complete function was discovered [Kilian, FOCS 1988], the question of which finite 2-party functions are complete has been studied extensively, leading to characterization in many special cases. In this work, we completely settle this problem.

We provide a polynomial time algorithm to test whether a 2-party finite secure function evaluation (SFE) functionality (possibly randomized) is complete or not. The main tools in our solution include:

- A formal linear algebraic notion of *redundancy* in a general 2-party randomized function.
- A notion of *statistically testable games*. A kind of interactive proof in the information-theoretic setting where *both* parties are computationally unbounded but differ in their knowledge of a secret.
- An extension of the (weak) *converse of Shannon's channel coding theorem*, where an adversary can adaptively choose the channel based on it view.

We show that any function $f$, if complete, can implement any (randomized) circuit $C$ using only $O(|C| + \kappa)$ calls to $f$, where $\kappa$ is the statistical security parameter. In particular, for any two-party functionality $g$, this establishes a universal notion of its quantitative "cryptographic complexity" independent of the setup and has close connections to circuit complexity.

*This paper is the result of merger of two independent works. One of these works is available online as [Kra13].

†Institute of Cryptography and Security, Department of Informatics, Karlsruhe Institute of Technology, Germany kraschewski@kit.edu.

‡University of California, Los Angeles. Supported by NSF CI Postdoctoral Fellowship. hmaji@cs.ucla.edu.

§University of Illinois, Urbana-Champaign. Supported by NSF grant CNS 07-47027. mmp@illinois.edu.

¶University of California, Los Angeles. sahai@cs.ucla.edu.

# Contents

# 1 Introduction

Understanding the *complexity* of functions is central to theoretical computer science. While the most studied notion of complexity in this literature is that of computational complexity, there have also been other important aspects explored, most notably, *communication complexity* [Yao79]. Another aspect of complexity of a (distributed) function is its *cryptographic complexity*, which seeks to understand the cryptographic utility of a function, stemming from how it hides and reveals information. While it is only recently that the term has been explicitly used, cryptographic complexity theory has been vigorously pursued at least since Kilian introduced the notion of *completeness* of cryptographic primitives [Kil88] a quarter century ago.

Completeness has been the first and most important question of cryptographic complexity: what properties of a function let all other cryptographic tasks (in the context of secure computation) be *reduced* to it. This question has been asked and answered several times [Kil88, CK88, Kil91, Kil00, CMW04, KM11, MPR12] each time for a different class of functions, or restricted to different kinds of reductions (see Figure 1 for a summary of the state of the art). These works produced several exciting ideas and advances, and brought together concepts from different fields. For instance, [Kil00] used the Nash equilibrium in a zero-sum game defined using the function to obtain a secure protocol; earlier [CK88] identified the binary symmetric channel (noisy channel) as a complete function, paving the way to a fruitful and successful connection with information-theory literature.

However, these works left open what is arguably the hardest part of the characterization: completeness of *randomized* functions under reductions that are secure against an *active* adversary (see Figure 1). Indeed, even with a (usually simplifying) restriction that only one of the two parties receives an output from the function, it was not known which *randomized* functions are complete. In this work, we finally provide a full characterization of completeness of general[1] 2-party functions. This work brings to close this rich line of investigation, but also introduces several new ideas and notions, and poses new questions regarding cryptographic complexity.

Prior to our work, the only completeness results known for randomized functions against active adversaries were for the very restricted case of channels [CMW04], *i.e.* randomized functions that *only take input from one party*, and deliver the output to the other. Thus, in particular, before our work, no completeness characterization results against active adversaries were known for any randomized function classes that take input from both parties.

Also, along the way to our main construction, we generalize a result in another line of work, on black-box protocol constructions [IKLP06, Hai08, IPS08, CDMW09]. We give a black-box transformation from a passive-secure OT protocol *in a hybrid setting* (wherein the protocol has access to an ideal functionality) to a UC-secure OT protocol in the same hybrid setting, with access to the commitment functionality.[2] Our transformation relativizes with respect to any ideal functionality, as long as that functionality is "redundancy free" (see later). Though our focus is on information-theoretic security, we note that by considering ideal functionalities that are *not* information-theoretically complete, our transformation implies black-box equivalence of related *computational assumptions*.

[1] By a general function, we mean one without any restrictions on which parties have inputs and which parties have outputs. Earlier work on characterizing randomized functions considered only "symmetric" (both parties get same output) and "asymmetric" (only one party gets any output) functions. Beyond this, only specific examples were known, like correlated random variables considered by Beaver [Bea95].

[2] It is interesting to note that, unlike in many other settings, a black-box transformation in the plain model does not imply a transformation in a hybrid model. That is, there is no analogue of universal composition for *black-box protocol compilation*.

| | Passive-Completeness | | Active-Completeness | |
|---|---|---|---|---|
| | Example | Characterization | Example | Characterization |
| Deterministic | OT: [GV87, HM86] | Symmetric: [Kil91]<br>Asymmetric: [BMM99]<br>General: [KM11] | OT: [Kil88] | Symmetric: [Kil91]<br>Asymmetric: [Kil00]<br>General: [KM11] |
| Randomized | Rabin-OT: [Cré87] | Symmetric: [Kil00]<br>Asymmetric: [Kil00]<br>General: [MPR12] | Rabin-OT: [Cré87] | Channels: [CMW04]<br>Symmetric/Asymmetric/General: **Open**<br>*Settled in this paper* |

**Figure 1** Summary of Completeness Characterization Results.

Finally, our tools for analysis are novel in this line of work.

In particular, we introduce the notion of **statistically testable games**, which is a kind of interactive proof in the information-theoretic setting where *both* parties can be computationally unbounded, but differ in their knowledge of some secret. We discuss these in more detail in Section 1.2 and in subsequent sections.

We also formulate and prove a new converse of Shannon's Channel Coding theorem to obtain a hiding property from a "channel." This is perhaps an unusual (but in hindsight, natural) use of a converse of the channel coding theorem, which was originally used to establish the optimality of the channel coding theorem.

## 1.1 Our Results

We provide the first *algorithmic* characterization of all finite 2-party (potentially randomized) functions that are complete for secure function evaluation against active adversaries: Namely, our results provide the first explicit algorithm (see Figure 3) that can analyze any given (randomized) function $f$, and output whether or not $f$ is complete against active adversaries.

The algorithm has two steps: finding what we call the "*core*" of a given function $f$ and then checking if it is "*simple*" or not: $f$ is complete if and only if its core is not simple.

We now provide a high-level intuitive explanation of our algorithmic characterization works, by considering some easy and well-known examples. This will help in understanding our exact characterization, which is somewhat more involved since it covers general randomized functions.

The *core* of $f$ is computed by removing "redundant" parts of the function $f$. To develop some intuition for this, consider the one-sided OR function which takes two bits from Alice and Bob and outputs the logical OR of these two bits to only Bob. This function is *not* complete against active adversaries, and in fact is trivial: the reason is that a corrupt Bob can always choose his input to be "0" – and by doing so, it can always learn Alice's input, without Alice detecting this. (Thus, even a trivial protocol in which Alice sends her bit to Bob is indeed secure against active adversaries, since if Bob is corrupt, he could have learned Alice's input even in the ideal world.) Because of this, we say that Bob's input "1" is redundant from the adversary's point of view: the adversary is always better off using the input "0".

When extended to the setting of randomized functions, redundancy becomes more subtle. For instance, an input can become redundant because instead of using that input, an adversary could use a *distribution* over other inputs, without being detected. Another form of redundancy that appears for randomized functions is that of redundant outputs (for the same input). As an example, suppose in the above example, when Bob's input is 0, if Alice's input is 0 then he receives 0, but if her input is 1, he receives the output symbol $\alpha$

with probability 3/4 and the symbol $\beta$ with probability 1/4. Here, we observe that the two outcomes $\alpha$ and $\beta$ give Bob the same information about Alice's input, and could be merged into a single outcome. More generally, if two possible outputs that the adversary can obtain for the same input have identical conditional distributions for the other party's input-output pair, then the distinction between these two output values is redundant.

We provide a novel formal definition of redundancy that fully captures both these forms of redundancy: (1) it identifies inputs that are useless for the adversary; and (2) it identifies if the output can be compressed to remove aspects of the output that are useless for the adversary's goal of gaining information about the honest party's inputs. While the above intuition is useful, it is not exactly the motivation behind our formal definition. The formal definition balances the following two requirements on redundancy:

- Adding or removing redundancy does not change a function's complexity (as far as security against active corruption alone is concerned): in particular, $f$ is complete if and only if its core is complete.
- A redundancy free function removes the possibility for a party to freely deviate from its interaction with a functionality without the rest of the system (the environment and the other party) detecting any difference.

The formal definition (based on Equation 1) is linear algebraic, inspired by simulatability considerations, and seemingly more general; but as will be discussed in Section 1.2 and later, this definition coincides with exactly the above two forms of redundancies (Lemma 1 and Lemma 2). An explicit algorithm for removing redundancy and finding the "core" is given in Figure 4 in Appendix A.2.

The second phase of our algorithm determines whether the core of $f$ is *simple*, a notion defined earlier by [MPR12] generalizing Kilian's condition for passive completeness [Kil00]. Informally, a function $g$ is simple if it preserves the independence of views. To develop intuition for this, consider a common randomness function that ignores the inputs of the two parties and simply outputs a uniform independent random bit to both parties. This function is intuitively useless because, at least in the passive-security setting, this function can be trivially realized by one party sampling this bit, and sending it to the other party. The formal notion of a simple function generalizes this to arbitrary randomized functions, by ensuring that if the parties start with independent inputs, then conditioned on the "common information" present after function evaluation, the views of the two players remain independent of each other (see Section 5 for details). A natural explicit algorithm for determining whether a function is simple was already given by [MPR12], which we use here.

Beyond the basic feasibility result, we also show that secure evaluation of any finite function $g$ to a complete finite function $f$ can be carried out, asymptotically, at "constant rate." That is, $n$ copies of $g$ can be evaluated with access to $O(n+\kappa)$ copies of $f$, and in fact, only $O(n+\kappa)$ communication, overall. Here $\kappa$ is a statistical security parameter; that is, the error in security (simulation error) is negligible in $\kappa$. In fact, the total amount of communication in the protocol (including the interaction with copies of $f$) is also bounded by $O(n+\kappa)$.

This leads to our main theorem:

**Theorem 1.** *A finite 2-party function is UC-complete (or equivalently, standalone-complete) against active adversaries if and only if its core is not simple. Further, if $f$ is such a function, $n$ copies of any finite 2-party function can be securely evaluated by a protocol in $f$-hybrid with communication complexity $O(n+\kappa)$, where $\kappa$ is the security parameter.*

As an aside, we remark that the protocols we obtain for showing completeness are UC-secure *as well*

*as* passive-secure. Thus our results do not change if instead of UC-security against active adversaries, we consider a stronger notion which requires a protocol to be secure against active and passive adversaries (with simulation by active and passive simulators, respectively).

**Connections to Circuit Complexity.** An interesting measure of complexity of a function $g$ (modeled as a 2-party function) is its "OT complexity" – the number of (1 out of 2, bit) OT instances needed for securely evaluating it.[3] As sketched below, the OT complexity of a function is closely related to its circuit complexity and may provide an approach to proving explicit circuit lowerbouds. Our results show that instead of OT complexity, one could consider $f$-complexity, for any $f$ whose core is not simple. *This establishes "cryptographic complexity" as a fundamental complexity measure of (2-party) functions, independent of which complete finite 2-party function is used to securely realize it, just the same way circuit complexity is independent of which specific set of universal finite gates are used to implement it.*

Circuit complexity and OT complexity are closely related to each other as follows. By a simple protocol due to [GMW87, GV87, HM86], we know that the OT complexity of a function $g$ (defined with respect to passive security) is $O(C(g))$, where $C(g)$ stands for the circuit complexity of $g$. This means that a super-linear lowerbound for OT complexity of $g$ gives a super-linear lowerbound on $C(g)$. Of course, this only shows that it is a hard problem to lowerbound OT complexity. But interestingly, this connection does open up a new direction of approaching circuit complexity lowerbounds: the fact that most functions have exponential circuit complexity is an easy consequence of a counting argument due to Shannon; but *for OT complexity, even such an existential lowerbound is not known*. Resolving this could be an easier problem than finding explicit circuit lowerbounds, yet could lead to new insights to proving explicit OT complexity and circuit complexity lowerbounds.

The same argument applies for OT complexity defined with respect to active adversaries as well, due to the result of [IPS08]. Note that it would be easier to lowerbound OT complexity when it is defined this way, than when defined with respect to passive adversaries. The relevance of our result is that instead of OT, one can consider any 2-party function $f$ whose core is not simple. As we show that OT can be reduced to any such function at a constant rate, a super-linear lowerbound on (amortized) $f$-complexity will indeed translate to a super-linear lowerbound on circuit complexity. We discuss this more in our conclusion (Section 6) and leave it as an important direction to study.

Recently Beimel et al. [BIKK13] have shown that the OT-complexity of random functions is significantly lower than their (AND) circuit complexity, but still exponential in the input length, in the worst case.

**Related Work.** We briefly summarize the results on completeness from prior work (also refer to Figure 1). The function oblivious transfer (OT) was identified independently by Wiesner and Rabin [Rab81, Wie83]. Brassard et al. [BCR86] showed that various flavors of OT can be reduced to each other with respect to security against active adversaries. In a seminal work, Kilian identified OT as the first active-complete function [Kil88]. Prior to this Goldreich and Vainish, and independently Micali and Haber, showed that OT is passive-complete [HM86, GV87]. Crépeau and Kilian then showed that the noisy channel is also active-complete [CK88]. The first characterization of completeness appeared in [Kil91] where it was shown that among deterministic "symmetric" functions (in which both parties get the same output) a function $f$ is active-complete

[3] One may also define OT complexity to be the total amount of communication (possibly amortized) needed for securely evaluating $g$, in the OT-hybrid model.

if and only if there is an "OR minor" in the matrix representing $f$. Beimel, Malkin and Micali showed that among "asymmetric" functions (in which only one party gets the output), a function if passive-complete if and only if it is not "trivial" [BMM99]. ([BMM99] also concerned itself with the computational setting and asked cryptographic complexity questions regarding computational assumptions.) Kilian vastly generalized this by giving several completeness characterizations: active-complete deterministic asymmetric functions, passive-complete symmetric functions and passive-complete asymmetric functions [Kil00]. Kilian's result for active-completeness was extended in two different directions by subsequent work: Crépeau, Morozov and Wolf [CMW04] considered "channel functions" which are randomized asymmetric functions (only one party has output), but with the additional restriction that only one party has input; Kraschewski and Müller-Quade [KM11] considered functions in which both parties can have inputs and outputs, but restricted to deterministic functions.

Kilian's result for passive-completeness was extended to all functions in a recent work [MPR12], which also presented a unification of all the prior characterizations and posed the question of completing the characterization. The full characterization we obtain matches the unified conjecture from [MPR12].

A related, but different line of work investigated secure computability and completeness for *multi-party* computation (with more than 2 parties) (e.g., [CCD88, BGW88, RB89, KMO94, KKMO00, FM00, FGMO05]). We restrict ourselves to 2-party functions in this work. Another direction of research considers whether a short protocol for $f$ (instead of a black-box implementing $f$) is complete or not [LOZ12].

## 1.2 Technical Overview

An important ingredient of our result is a combinatorial/linear-algebraic characterization of "redundancy" in a general 2-party function. The importance of redundancy is two fold:

- Any function $f$ is "equivalent" (or *weakly isomorphic*, as defined in [MPR12]) to a "core" function $\widehat{f}$ which is redundancy free, so that $f$ is complete against active adversaries if and only if $\widehat{f}$ is. Thus it is enough to characterize completeness for redundancy free functions.
- Our various protocols rely on being given access to a redundancy free function. Redundancy makes it possible for an adversary to deviate from a prescribed interaction with a function without any chance of being detected. Thus the statistical checks used to enforce that the adversary does not deviate from its behavior crucially rely on the protocol using only redundancy free functions.

While redundancy of special classes of 2-party functions have appeared in the literature previously, it turns out that for general 2-party functions, the nature of redundancy is significantly more intricate. Recall that we discussed redundancy informally by considering an adversary that tries to learn about the other party's input-output pair: any input it can avoid, and distinction between outputs (for the same input) that provide it with identical information are both redundant. However, the role of redundancy in showing completeness is somewhat different: redundancy in a function makes it hard (if not impossible) to use it in a protocol, as it allows an active adversary to *deviate* from behavior prescribed by a protocol, with no chance of being caught. Possible deviation includes replacing its prescribed input to the function by a probabilistically chosen input, *and* probabilistically altering the output it receives from the function before using it in the protocol, *at the same time*. The goal of this deviation is to minimize detectability by the other party (and the environment). Our formal definition of redundancy uses this point of view. We define

*irredundancy* quantitatively (Definition 1) as a lowerbound on the ratio of the detection advantage to the extent of deviation ("irredundancy = detection/deviation").

The first step in our characterization is to bridge the gap between these two formulations of redundancy. While the definition of irredundancy is what allows us to use a redundancy-free function in our protocols, to find the core of a function, we rely on the formulation in terms of redundancy of individual inputs – we shall reduce redundancy one input or output at a time, until we obtain a redundancy free function. Clearly when redundancy is present, irredundancy would be 0 (i.e., can deviate without being detected); but we show that conversely, when irredundancy is 0, then one of the two forms of redundancy must be present. We stress that *a priori*, it is not at all obvious that irredundancy cannot be 0 even if there is no redundancy (i.e., detection/deviation could approach 0 by a sequence of deviations that are smaller and smaller, achieving even smaller detectability). We provide a non-trivial linear algebraic analysis of irredundancy and show that this is not the case (Lemma 3).

**Simple Function.** Following [MPR12], we define a simple function, in Section 5. First, we present a combinatorial characterization (given in Lemma 1 in [MPR12]) of a simple function, which constitutes the algorithm for determining if a function is simple or not.

Given a 2-party randomized function $f : X \times Y \to W \times Z$, consider the $|Y||Z| \times |X||W|$ matrix $\mathfrak{P}^f$, with rows indexed by $(y, z) \in Y \times Z$ and columns indexed by $(x, w) \in X \times W$, such that $\mathfrak{P}^f_{(y,z),(x,w)} = \mathfrak{p}^f[w, z|x, y]$. The function $f$ is simple if $\mathfrak{P}^f$ can be partitioned into a set of rank-1 minors such that no row or column of the matrix pass through two of these minors. Being of rank 1, each minor has all its rows (equivalently, columns) parallel to each other. (In [MPR12], this is described in terms of a bipartite-graph in which each connected component is a complete bipartite graph, with weights on the edges being proportional to the product of the weights on the two end points of the vertex.)

To better understand what being simple means, we briefly explain how it is defined (Definition 3). The *kernel* of a function $f$ is a symmetric function that provides both the parties with only the "common information" that $f$ provides them with. A simple function is one which is "isomorphic" to its kernel: i.e., given just the output from the kernel, the rest of the information from $f$ can be locally sampled by the two parties, independent of each other.

As stated in Lemma 7, the *passive*-complete functions are exactly those which are not simple. Our construction shows that *restricted to the class of redundancy free functions*, the same characterization holds for complete functions for active-security as well.

**The construction.** Our main construction shows that any redundancy free function $f$ which is not simple is also UC-complete. This construction separates into two parts:

- A protocol to UC-securely reduce the commitment functionality $\mathcal{F}_{\text{COM}}$ to $f$.
- A protocol in the $\mathcal{F}_{\text{COM}}$-hybrid model that UC-securely reduces OT to $f$, starting from a passive-secure reduction of OT to $f$ (since $f$ is passive-complete, such a protocol exists). That is, we compile (in a black-box manner) a passive-secure OT protocol using $f$, to a UC-secure OT protocol using $f$ (and $\mathcal{F}_{\text{COM}}$).

In building the commitment functionality we rely on a careful analysis of functions that are redundancy free and not simple, to show that there will exist two or more *extreme views* for one party (which cannot be

equivocated) that are *confusable* by the second party (provided it uses inputs from an "unrevealing distribution" — something that can be verified by the first party). We interpret the function invocations as a channel through which the first party transmits a message using the set of its extreme views as the alphabet. This message is encoded using an error correcting code of rate $1 - o(1)$ and $o(1)$ distance; the distance would be sufficient to prevent equivocation during opening. To argue hiding, we rely on a well-known result from information theory, namely the (weak) *converse of Shannon's Channel Coding Theorem*. We extend this theorem to the case of adaptively chosen channel characteristics, corresponding to the fact that the receiver can adaptively choose its input to the function and that determines the channel characteristics. Due to confusability, the capacity of this channel will be less than 1 (measured with the logarithm of the input alphabet size as the base). Since the rate of the code is higher than the capacity of the channel, this gives us some amount of hiding (which is then refined using an extractor).

The second part, which gives a compiler, is similar in spirit to prior protocols that established that a passive-secure OT protocol (in the plain model) can be converted to an active-secure OT protocol *in a black-box manner* [IKLP06, Hai08, CDMW09]. In particular, its high-level structure resembles that of the protocol in [CDMW09]. However, the key difference in our protocol compared to these earlier protocols (which were all in the computational setting), is that the passive-secure OT protocol that we are given is not in the plain model, but is in the $f$-hybrid model. The technical difficulty in our case is in ensuring that a cut-and-choose technique can be used to verify an adversary's claims about what inputs it sent to a 2-party function and what outputs it received, when the verifier has access to only the other end of the function. This is precisely where the statistical testability of redundancy free functions (see below) is invoked.

Also, in contrast with the above mentioned compilers, we do not use a two-step compilation to first obtain security against active corruption of the receiver and then that of the sender. Instead, we directly obtain a somewhat "noisy" OT protocol that is secure against active corruption of either player, and use techniques from [IPS08, IKO+11] to obtain the final protocol. In particular, we show how the result in [IPS08] can be extended so that it works in a noisy OT-hybrid rather than a regular OT-hybrid. (A similar extension was used in [IKO+11], to allow using a noisy channel hybrid instead of a regular OT-hybrid.) These tools help us achieve a constant rate in implementing OTs from instances of $f$.

**Statistically Testable Games.** We introduce a formal notion of statistically testable game, which is an information-theoretic analogue of interactive proofs where both players can be computationally unbounded. Note that interactive proofs are not interesting in this information-theoretic setting (or if P=PSPACE). In a statistically testable game, the statements being proven (tested) are statements regarding the private observations of the prover in a system, which provides partial observations to the verifier as well. The non-triviality of such a proof system stems not from the computational limitations of the verifier, but from the fact that the verifier cannot observe the entire system. While such proofs have been implicitly considered in several special cases in many prior works (e.g. [CK88, CMW04, IPS08, IKO+11]), the class of games we consider is much more general than those implicitly considered in these earlier instances, and the soundness of the tests we consider is not at all obvious.

The game we consider is of 2-party function evaluation, in which the prover and the verifier interact with a (stateless) trusted third party which carries out a randomized function evaluation for them. The prover first declares a sequence of $n$ inputs it will feed the function (the verifier chooses its inputs privately and independently). After $n$ invocations of the function, the prover declares to the verifier the sequence of the $n$ outputs it received from the invocations. A statistical test is a sound and complete proof system which convinces the verifier that the input and output sequences declared by the prover has a $o(1)$ fraction

hamming distance from the actual sequences in its interaction with the trusted party. Note that the verifier can use its local observations (its input-output sequences) to carry out the verification.

A major technical ingredient of our compiler is the following theorem:

*Evaluation of a 2-party function $f$ is statistically testable if and only if $f$ is redundancy free.*

Clearly, if a function is not redundancy free, it admits is no sound statistical test. But *a priori*, it may seem possible that even if no single input has redundancy, the prover can map the entire sequence of inputs and outputs to a different sequence, with only a small statistical difference in the verifier's view, such that this difference vanishes with the length of the sequence. We show that this is not the case: if the function is redundancy-free, then there is a lowerbound on the ratio of the "detection advantage" to "extent of deviation" that does not vanish with the number of invocations.

This naturally motivates our approach of compiling a passive-secure protocol in $f$-hybrid, where $f$ is redundancy free, into one that is secure against active adversaries. We should be able to enforce honest behavior by "auditing" randomly chosen executions from a large number of executions, and the auditing would use the statistical tests. However, this idea does not work directly: the statical test models a test by an *environment*: it lets the adversary arbitrarily interact with $f$ and report back a purported output, but the purported input it sent to $f$ was fixed by the environment before the adversary obtained the output from $f$. On the other hand, in a protocol, the honest party does not get to see the input to be sent to the functionality ahead of time. It is to solve this issue that we rely on the commitment functionality: the input each party should be sending to $f$ is fixed *a priori* using commitments (and coin-tossing-in-the-well). When a session is chosen for auditing, the adversary could indeed have sent a different input to $f$ than it was supposed to, and it can lie about the output it received from $f$ as well, but it cannot choose the purported input it sent to $f$ after interacting with $f$.

## 2 Preliminaries

**Matrix Definitions.** In the following we shall refer to the following matrix norms: $\|A\|_\infty = \max_i \sum_j |a_{ij}|$ (maximum absolute row sum norm), and $\|A\|_{\text{sum}} = \sum_{i,j} |a_{ij}|$ (absolute sum norm). We shall also use the function $\max(A) = \max_{i,j} a_{ij}$ (maximum value among all entries); not that here we do not consider the absolute value of the entries in $A$. For a probability distribution $\mathfrak{p}^X$ over a space $X$ (denoted as vectors), we define $\min(\mathfrak{p}^X) = \min_{x \in X} \mathfrak{p}^X[x]$, the minimum probability it assigns to an element in $X$. The norm $\|\cdot\|_\infty$ when applied to a column vector simply equals the largest absolute value entry in the vector. We say that a matrix $P$ is a *probability matrix* if its entries are all in the range $[0, 1]$ and $\|P\|_{\text{sum}} = 1$. We say that a matrix is a *stochastic matrix* (or row-stochastic matrix) if all its entries are in the range $[0, 1]$ and every row sums up to 1. For convenience, we define the notation $\langle M \rangle_I$ for a square matrix $M$ to be the diagonal matrix derived from $M$ by replacing all non-diagonal entries by 0.

**2-Party Secure Function Evaluation.** A two-party randomized function (also called a secure function evaluation (SFE) functionality) is specified by a single randomized function denoted as $f : X \times Y \to W \times Z$. Despite the notation, the range of $f$ is, more accurately, the space of probability distributions over $W \times Z$. The functionality takes an input $x \in X$ from Alice and an input $y \in Y$ from Bob and samples $(w, z) \in W \times Z$ according to the distribution $f(x, y)$; then it delivers $w$ to Alice and $z$ to Bob. Through out,

we shall denote the probability of outputs being $(w, z)$ when Alice and Bob use inputs $x$ and $y$ respectively is represented by $\mathfrak{p}^f[w, z|x, y]$. We use the following variables for the sizes of the sets $W, X, Y, Z$:

$$|X| = m \qquad |Y| = n \qquad |W| = q \qquad |Z| = r.$$

In this paper we shall restrict to function evaluations where $m$, $n$, $q$ and $r$ are constants, i.e. as the security parameter increases the domains do not expand. (But the efficiency and security of our reductions are only polynomially dependent on $m, n, q, r$, so one could let them grow polynomially with the security parameter. We have made no attempt to optimize this dependency.) W.l.o.g., we shall assume that $X = [m]$ (i.e., the set of first $m$ positive integers), $Y = [n]$, $W = [q]$ and $Z = [r]$.

We consider standard security notions in the information-theoretic setting: UC-security, standalone-security and passive-security against computationally unbounded adversaries (and with computationally unbounded simulators). Using UC-security allows to compose our sub-protocols securely [Can05]. Error in security (simulation error) is always required to be negligible in the security parameter of the protocol, and the communication complexity of all protocols are required to be polynomial in the same parameter. However, we note that a protocol may invoke a sub-protocol with a security parameter other than its own (in particular, with a constant independent of its own security parameter).

**Complete Functionalities.** A two-party randomized function evaluation $f$ is *standalone-complete* (respectively, *UC-complete*) against information theoretic adversaries if any functionality $g$ can be standalone securely (respectively, UC securely) computed in $f$ hybrid. We shall also consider passive-complete functions where we consider security against passive (semi-honest) adversaries.

# 3 Main Tools

In this section we introduce the main tools used in our construction.

## 3.1 Characterizing Irredundancy

Redundancy in a function allows at least one party to deviate in its behavior in the ideal world and not be detected (with significant probability) by an environment. In our protocol, which are designed to detect deviation, it is important to use a function in a form in which redundancy has been removed. We define irredundancy in an explicit linear algebraic fashion, and introduce a parameter to measure the extent of irredundancy.

**Irredundancy of a System of Stochastic Matrices.** Let $P_i$, $i = 1, \ldots, m$ be a collection of $s \times q$ probability matrices (i.e., entries in the range $[0, 1]$, with $\|P_i\|_{\text{sum}} = 1$). Consider tuples of the form $(j, \{M_i, \alpha_i\}_{i=1}^m)$, where $j \in [m]$, $M_i$ are $q \times q$ stochastic matrices, and $\alpha_i \in [0, 1]$ are such that $\sum_i \alpha_i = 1$. Then we define the irredundancy of this system as

$$\mathfrak{D}(P_1, \ldots, P_m) = \inf_{(j, \{\alpha_i, M_i\}_{i=1}^m)} \frac{\|(\sum_{i=1}^m \alpha_i P_i M_i) - P_j\|_\infty}{1 - \alpha_j \|P_j \cdot \langle M_j \rangle_I\|_{\text{sum}}} \tag{1}$$

where the infimum is over tuples of the above form. (Recall that $\langle M_j \rangle_I$ refers to the diagonal matrix with the diagonal entries of $M_j$.)

Intuitively, consider the rows of $P_i$ to be probability distributions over a $q$-ary alphabet produced as the outcome of a process with the row index corresponding to a hidden part of the outcome, and the column index being an observable outcome. Then, irredundancy measures how well a $P_j$ can (or rather, cannot) be approximated by a convex combination of all the matrices $P_i$, possibly with the observable outcome transformed using a stochastic matrix (corresponding to a probabilistic mapping of the observable outcomes); the denominator normalizes the approximability by how much overall *deviation* (probability of changing the process or changing the outcome) is involved. This excludes the trivial possibility of perfectly matching $P_j$ by employing zero deviation (i.e., taking $\alpha_j = 1$ and $M_j = I$).

**Irredundancy of a 2-Party Secure Function Evaluation Function.** Recall that a 2-party SFE function $f$ with input domains, $X \times Y$ and output domain $W \times Z$ is defined by probabilities $\mathfrak{p}^f[w, z|x, y]$. We define left and right redundancy of $f$ as follows. Below, $|X| = m, |Y| = n, |W| = q, |Z| = r$.

To define left-redundancy, consider representing $f$ by the matrices $\{P^x\}_{x \in X}$ where each $P^x$ is an $nr \times q$ matrix with $P^x_{(y,z),w} = \mathfrak{p}^f[w, y, z|x]$. Here, $\mathfrak{p}^f[w, y, z|x] \triangleq \frac{1}{n}\mathfrak{p}^f[w, z|x, y]$ (where we pick $y$ independent of $x$, with uniform probability $\mathfrak{p}^f[y|x] = \frac{1}{n}$).

**Definition 1.** *For an SFE function $f : X \times Y \to W \times Z$, represented by matrices $\{P^x\}_{x \in X}$, with $P^x_{(y,z),w} = \Pr[w, y, z|x]$, we say that an input $\hat{x} \in X$ is* left-redundant *if there is a set $\{(\alpha_x, M_x)|x \in X\}$, where $0 \leq \alpha_x \leq 1$ with $\sum_x \alpha_x = 1$, and each $M_x$ is a $q \times q$ stochastic matrix such that if $\alpha_{\hat{x}} = 1$ then $M_{\hat{x}} \neq I$, and $P^{\hat{x}} = \sum_{x \in X} \alpha_x P^x M_x$.*

*We say $\hat{x}$ is* strictly left-redundant *if it is left-redundant as above, but $\alpha_{\hat{x}} = 0$. We say $\hat{x}$ is* self left-redundant *if it is left-redundant as above, but $\alpha_{\hat{x}} = 1$ (and hence $M_{\hat{x}} \neq I$).*

*We say that $f$ is* left-redundancy free *if there is no $x \in X$ that is left-redundant.*

Right-redundancy notions are defined analogously. $f$ is said to be *redundancy-free* if it is left-redundancy free and right-redundancy free.

**Lemma 1.** *For an SFE function $f$, if $\hat{x}$ is left-redundant, then it is either strictly left-redundant or self left-redundant.*

*Proof.* Suppose $\hat{x}$ is left-redundant. Then $P^{\hat{x}} = \sum_{x \in X} \alpha_x P^x M_x$ as in the definition of left-redundancy. If $\alpha_{\hat{x}} = 1$, then by definition it is self left-redundant. If $\alpha_{\hat{x}} < 1$, we shall show that it is strictly left-redundant. We can write $P^{\hat{x}}(I - \alpha_{\hat{x}} M_{\hat{x}}) = \sum_{x \neq \hat{x}} \alpha_x P^x M_x$. To rewrite $P^{\hat{x}}$ as required by strict left-redundancy we depend on the following the observation.

**Claim 1.** *If $M$ is a $q \times q$ stochastic matrix and $0 \leq \alpha < 1$, then $I - \alpha M$ is invertible and $(1-\alpha)(I-\alpha M)^{-1}$ is a stochastic matrix.*

*Proof.* Consider the series $D = I + \alpha M + \alpha^2 M^2 + \cdots$. Since $|\alpha| < 1$ and $M$ is stochastic (and in particular, $\|\alpha M\|_\infty < 1$), this series converges, and then since, $D = I + \alpha M \cdot D$, we have $(I - \alpha M)D = I$. Further, $D \cdot \mathbf{1}^T = \frac{1}{1-\alpha} \cdot \mathbf{1}^T$ (where $\mathbf{1}$ is the row matrix of all 1's), because $M^t$ is stochastic for all $t$ and $\alpha^t M^t \cdot \mathbf{1}^T = \alpha^t \cdot \mathbf{1}^T$. □

Using the above claim, let $M = (1-\alpha_{\hat{x}})(I - \alpha_{\hat{x}} M_{\hat{x}})^{-1}$ be a stochastic matrix. Then

$$\begin{aligned} P^{\hat{x}} &= \frac{1}{1-\alpha_{\hat{x}}} \cdot P^{\hat{x}} \cdot (I - \alpha_{\hat{x}} M_{\hat{x}}) \cdot M \\ &= \frac{1}{1-\alpha_{\hat{x}}} \cdot \sum_{x \neq \hat{x}} \alpha_x P^x \cdot M_x M = \sum_{x \neq \hat{x}} \alpha'_x \cdot P^x \cdot M'_x \end{aligned}$$

where $\alpha'_x = \frac{\alpha_x}{1-\alpha_{\hat{x}}}$ (except $\alpha'_{\hat{x}} = 0$) and $M'_x = M_x \cdot M$ satisfy the conditions required for strict left-redundancy. □

As we shall see later, in removing redundancy of $f$ while retaining equivalence (i.e., to find the core of $f$), we will need to identify strictly-redundant inputs and simply remove them. Note that checking if an input for $f$ is strictly-redundant can be framed as the feasibility of a linear program. However, if an input is self-redundant, we cannot simply remove this input. Instead we need to compress the corresponding output space. The following lemma gives a simple characterization of self-redundancy.

**Lemma 2.** *For an SFE function $f : X \times Y \to W \times Z$ defined by the probability matrices $\{P^x\}_{x \in X}$, if there is a self left-redundant input $\hat{x} \in X$, then $P^{\hat{x}}$ has two non-zero columns which are scalar multiples of each other. That is, if $\hat{x}$ is self-redundant, then there should be two output values $w, w' \in W$ such that Bob's input-output pair is distributed identically conditioned on Alice's view being $(x, w)$ or being $(x, w')$.*

*Similarly, if $\hat{y} \in Y$ is a self right-redundant input, then there must be two non-zero columns in $P^{\hat{y}}$ that are scalar multiples of each other.*

In fact, we shall need the following quantitative version, which shows that if no two columns of $P^{\hat{x}}$ are close to being scalar multiples of each other, then $\hat{x}$ is not close to being self left-redundant.

**Claim 2.** *Suppose $M$ is a $q \times q$ stochastic matrix such that $\max(M - I) \geq \delta > 0$. Also, suppose $P$ is an $s \times q$ matrix such that for any two columns $c_i$ and $c_j$ of $P$, $\inf_\gamma \|c_i - \gamma c_j\|_\infty \geq \epsilon > 0$. Then $\|PM - P\|_\infty \geq \delta\epsilon$.*

The proof, given in Appendix A.1 uses an inductive argument. This claim (along with Claim 1) is used in proving the next lemma, that the (left and right) irredundancy parameters of a function that is not (left or right) redundant are bounded away from 0.

**Lemma 3.** *Suppose a 2-party function $f : X \times Y \to W \times Z$ is left redundancy free. Let $\mathfrak{p}^Y$ be a probability distribution over $Y$. Let the probability matrices $\{P^x\}_{x \in X}$, be defined by $P^x_{(y,z),w} = \mathfrak{p}^f[w, z|x, y]\mathfrak{p}^Y[y]$. Then there is a constant $\epsilon_f > 0$ (depending only on $f$) such that $\mathfrak{D}(P^1, \cdots, P^m) \geq \epsilon_f \min(\mathfrak{p}^Y)$.*

The analogous statement holds for right redundancy.

## 3.2 Statistically Testable Function Evaluation

In this section we consider the notion of a statistically testable function evaluation game. (The notion is more general and could be extended to reactive systems, or multi-player settings; for simplicity we define it only for the relevant setting of 2-party functions.) We informally defined a statistical test in Section 1.2. As mentioned there, we shall show that *evaluation of a 2-party function is statistically testable if and only*

*if the function is redundancy free*. For simplicity, we define a particular test and show that it is sound and complete for redundancy free functions (without formally defining statistical tests in general). (It is easy to see that functions with redundancy cannot have a sound and complete test. Since this is not relevant to our proof, we omit the details.)

Let $f$ be redundancy free. Consider the following statistical test, formulated as a game between an honest challenger (verifier) and an adversary (prover) in the $f$-hybrid.

Left-Statistical-Test$(f, \mathfrak{p}^Y; N)$:

1. The adversary picks $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_N) \in X^N$, and for each $i \in [N]$ the challenger (secretly) picks uniform i.i.d $y_i \in Y$, according to the distribution $\mathfrak{p}^Y$.
2. For each $i \in [N]$, the parties invoke $f$ with inputs $x_i$ and $y_i$ respectively; the adversary receives $w_i$ and the challenger receives $z_i$, where $(w_i, z_i) \xleftarrow{\$} f(x_i, y_i)$.
3. The adversary then outputs $\tilde{\mathbf{w}} = (\tilde{w}_1, \ldots, \tilde{w}_N) \in W^N$.

The adversary wins this game (breaks the soundness) if the following conditions hold:

1. Consistency: Let $\mu_{\tilde{w},\tilde{x},y,z}$ be the number of indices $i \in [N]$ such that $\tilde{w}_i = \tilde{w}, \tilde{x}_i = \tilde{x}$, $y_i = y$ and $z_i = z$. Also, let $\mu_{\tilde{x},y}$ be the number of indices $i \in [N]$ such that $\tilde{x}_i = \tilde{x}$ and $y_i = y$. The consistency condition requires that $\forall (w, x, y, z) \in W \times X \times Y \times Z$,
$$\mu_{\tilde{w},\tilde{x},y,z} = \mu_{\tilde{x},y} \times \mathfrak{p}^f[\tilde{w}, z | \tilde{x}, y] \pm N^{2/3}.$$
2. Separation: Let vectors $\mathbf{A}, \tilde{\mathbf{A}} \in (W \times X)^N$ be defined by $A_i := (w_i, x_i)$ and $\tilde{A}_i = (\tilde{w}_i, \tilde{x}_i)$. The separation condition requires that the hamming distance between the vectors $\mathbf{A}$ and $\tilde{\mathbf{A}}$ is $\Delta(\mathbf{A}, \tilde{\mathbf{A}}) \geq N^{7/8}$.

The *Right-Statistical-Test*$(f, \mathfrak{p}^X; N)$ is defined analogously. The experiment *Statistical-Test*$(f, \mathfrak{p}^X, \mathfrak{p}^Y; N)$ consists of Left-Statistical-Test$(f, \mathfrak{p}^Y; N)$ and Right-Statistical-Test$(f, \mathfrak{p}^X; N)$, and the adversary wins if it wins in either experiment.

Before proceeding, we note that the above statistical test is indeed "complete": if the prover plays "honestly" and uses $\tilde{\mathbf{x}} = \mathbf{x}$ and $\tilde{\mathbf{w}} = \mathbf{w}$, then the consistency condition will be satisfied with all but negligible probability (for any choice of $\mathbf{x}$).

**Lemma 4.** *If $f$ is redundancy free, and $\mathfrak{p}^X$ and $\mathfrak{p}^Y$ are constant distribution which have full support over $X$ and $Y$ respectively, then the probability that any adversary wins in Statistical-Test$(f, \mathfrak{p}^Y, \mathfrak{p}^X; N)$ is* $\mathrm{negl}(N)$.[4]

*Proof.* We prove this in Appendix B. Here we sketch the outline of that proof, omitting the calculations.

We shall only argue that if $f$ is left-redundancy free, then the probability of any adversary winning the Left-Statistical-Test$(f, \mathfrak{p}^Y; N)$ is negligible in $N$. The argument for the Right-Statistical-Test is similar. Then the result follows by union bound.

[4] The distributions $\mathfrak{p}^X$ and $\mathfrak{p}^Y$ are constant while $N$ is a growing parameter.

The experiment involves the adversary adaptively choosing $x_i$. To facilitate the analysis, instead we shall analyze *all* choices of $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w}, \tilde{\mathbf{w}})$, but restricted to $\mathbf{w}$ being "typical" for a randomly chosen $\mathbf{y}$ (for the given vector $\mathbf{x}$). Since this would hold except with negligible probability (over random choice of $\mathbf{y}$ and the randomness of $f$), this restriction will not affect the conclusion. Then, assuming that the adversary satisfies the sufficient-distance condition, we analyze the probability of the consistency condition holding. We shall argue that this probability is negligible if $f$ is redundancy free.

We shall consider the expectation of the quantity $\mu_{\tilde{w},\tilde{x},y,z} - \mathfrak{p}^f[\tilde{w}, z|\tilde{x}, y]\mu_{\tilde{x},y}$ and argue that for some value of $x, \tilde{y}, \tilde{z}$, the absolute value of this expectation should be large, say, $\Omega(N^{7/8})$. Note that, once we fix $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w}, \tilde{\mathbf{w}})$, then for any quadruple $(\tilde{x}, x, w, \tilde{w})$, $\mu_{\tilde{w},\tilde{x},y,z}$ and $\mu_{\tilde{x},y}$ can both be written as the sum of i.i.d indicator random variables. This is because the random experiment we consider consists only of picking $y_i, z_i$, for each $i$ independently: if $x_i = x$ and $w_i = w$, then $\Pr[y_i = y, z_i = z] = \mathfrak{p}^{f,Y}[y, z|x, w] := \frac{\mathfrak{p}^Y[y]\cdot\mathfrak{p}^f[w,z|x,y]}{\sum_{z',y'}\mathfrak{p}^Y[y']\cdot\mathfrak{p}^f[w,z'|x,y']}$. Then by Chernoff bounds, we obtain that except with negligible probability, the consistency condition will be violated.

We shall define the set Good of "good" $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w})$ in which, for each $\tilde{x}, x, w$, the number of positions $i$ with $w_i = w$ among the positions $i$ with $\tilde{x}_i = \tilde{x}, x_i = x$ is as expected (over uniformly random i.i.d $y_i$ and randomness of $f$) up to an additive error of $N^{2/3}$. (Note that this assumption is non-trivial only when there are at least $N^{2/3}$ positions with $\tilde{x}_i = \tilde{x}, x_i = x$.) The analysis below would be for every tuple $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w}) \in$ Good. W.l.o.g we assume that for each $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w})$ the adversary chooses $\tilde{\mathbf{w}}$ deterministically.

Fix $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w}) \in$ Good and an arbitrary $\tilde{\mathbf{w}}$. By the separation condition of the test, we know that there is some value $\hat{x} \in X$ such that in at least $\frac{1}{m}N^{7/8}$ indices $i$ where $\tilde{x}_i = \hat{x}$, the adversary deviates: either $x_i \neq \tilde{x}_i$ or $w_i \neq \tilde{w}_i$. In the rest of the analysis we restrict our attention to the set of indices with $\tilde{x}_i = \hat{x}$. We write $\tilde{I}_{\tilde{x}}$ to denote this set of indices, and $\tilde{J}_{\hat{x}} \subseteq \tilde{I}_{\hat{x}}$ to denote the subset of indices where there is a deviation.

The probabilities in the expressions below are conditioned on $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w})$, where the random choices made are of $\mathbf{y}$ and $(\mathbf{w}, \mathbf{z})$. (We do not assume any distribution over $\tilde{\mathbf{x}}$ and $\mathbf{x}$ which are chosen by the adversary.)

We show (see Appendix B for the calculations) the following:

$$\mathrm{E}\left[\mu_{\tilde{w},\hat{x},y,z} - \mathfrak{p}^f[\tilde{w}, z|\hat{x}, y]\cdot\mu_{\hat{x},y}\right] = |\tilde{I}_{\hat{x}}|\left((\sum_x \alpha^x P^x \cdot B^x) - P^{\hat{x}}\right)_{(y,z),\tilde{w}} \pm O(N^{2/3})$$

where $P^x$ is an $nr \times q$ matrix with $P^x_{(y,z),w} = \mathfrak{p}^{f,Y}[w, y, z|x]$, $B^x$ is a $q \times q$ stochastic matrix for each $x$, and $\alpha^x \geq 0$ with $\sum_x \alpha^x = 1$. Further, we can rewrite $|\tilde{I}_{\hat{x}}|$ in terms of $|\tilde{J}_{\hat{x}}|$ using

$$|\tilde{J}_{\hat{x}}| = |\tilde{I}_{\hat{x}}|\left(1 - \alpha^{\hat{x}}\|P^{\hat{x}} \cdot \langle B^{\hat{x}}\rangle_I\|_{\mathrm{sum}}\right) \pm N^{2/3}$$

Putting these together (and using the fact that $|\tilde{J}_{\hat{x}}| = \Omega(N^{7/8})$), we can show that the expected difference, maximized over all $(\tilde{w}, y, z)$ is

$$\max_{(\tilde{w},y,z)} |\mathrm{E}\left[\mu_{\tilde{w},\hat{x},y,z} - \mathfrak{p}^f[\tilde{w}, z|\hat{x}, y]\cdot\mu_{\hat{x},y}\right]| \geq \frac{|\tilde{J}_{\hat{x}}|}{q}\mathfrak{D}(P^1, \ldots, P^m) \pm o(N^{7/8}).$$

Finally, by Lemma 3, since $f$ is redundancy free, $\mathfrak{D}(P^1, \ldots, P^m) \geq \epsilon_f \cdot \min(\mathfrak{p}^Y)$, where $\epsilon_f > 0$ is a constant. Since $\mathfrak{p}^Y$ has full support (and is independent of $N$), $\min(\mathfrak{p}^Y) > 0$ is also a constant. Thus, the above quantity is $\Omega(N^{7/8})$. To complete the proof we use Chernoff bounds to argue that with all but negligible probability, for $(\tilde{w}, y, z)$ which maximizes the above expectation, $|\mu_{\tilde{w},\hat{x},y,z} - \mathfrak{p}^f[\tilde{w}, z|\hat{x}, y]\cdot\mu_{\hat{x},y}| > N^{2/3}$ (when $N$ is sufficiently large). □

### 3.3 A Converse of The Channel Coding Theorem

A converse of the channel coding theorem states that message transmission is not possible over a noisy channel at a rate above its capacity, except with a non-vanishing rate of errors (see, for e.g., [CT91]). We give a generalization of the (weak) converse of channel coding theorem where the receiver can adaptively choose the channel based on its current view. We show that if in at least a $\mu$ fraction of the transmissions, the receiver chooses channels which are noisy (i.e., has capacity less than that of a noiseless channel over the same input alphabet), then we can lower bound its probability of error in predicting the input codeword as a function of $\mu$, an upper bound on the noisy channel capacities, and the rate of the code.

**Lemma 5** (Weak Converse of Channel Coding Theorem, Generalization)**.** *Let $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_K\}$ be a set of $K$ channels which take as input alphabets from a set $\Lambda$, with $|\Lambda| = 2^\lambda$. Let $\mathcal{G} \subseteq [K]$ be such that for all $i \in \mathcal{G}$, the capacity of the channel $\mathcal{F}_i$ is at most $\lambda - c$, for a constant $c > 0$.*

*Let $\mathcal{C} \subseteq \Lambda^N$ be a rate $R \in [0, 1]$ code. Consider the following experiment: a random codeword $c_1 \ldots c_N \equiv \mathbf{c} \xleftarrow{\$} \mathcal{C}$ is drawn and each symbol $c_1 \ldots c_N$ is transmitted sequentially; the channel used for transmitting each symbol is chosen (possibly adaptively) from the set $\mathcal{F}$ by the receiver.*

*Conditioned on the receiver choosing a channel in $\mathcal{G}$ for $\mu$ or more transmissions, the probability of error of the receiver in predicting $\mathbf{c}$ is*

$$P_e \geq 1 - \frac{1}{NR\lambda} - \frac{1 - c\mu/\lambda}{R}.$$

We prove this in Appendix C. This result is used in our commitment protocol (Section 4.1).

## 4 Main Construction

In this section we prove the following theorem, which forms the main ingredient for the proof of Theorem 1.

**Theorem 2.** *If $f$ is a redundancy free 2-party function and $f$ is passive-complete, then there is a constant rate UC-secure protocol for $\mathcal{F}_{\text{OT}}$ in the $f$-hybrid model.*

Since $f$ is passive-complete we know that OT does reduce to $f$ against passive adversaries. We shall take such a passive-secure OT protocol in the $f$-hybrid, and convert it into a UC-secure protocol. For this we need two ingredients: first a UC-secure commitment protocol in the $f$-hybrid model, and secondly a compiler to turn the passive secure OT protocol in the $f$-hybrid model to a UC-secure protocol in the commitment-hybrid model. In building the UC-secure commitment protocol, we rely on the irredundancy of $f$ as well as the combinatorial characterization that passive-complete functions are exactly those that are not simple (see Section 1.2).

### 4.1 A UC Secure Commitment Protocol

In this section we present the outline of a UC-secure commitment protocol in the $f$-hybrid model, for any 2-party randomized function $f$ that is redundancy free (Definition 1) and is not simple (see Section 1.2). The full details of the construction are given in Appendix D.

The high-level structure of the protocol is as follows. The underlined terms will be explained below.

1. Commitment phase:
   (a) The sender plays the role of (say) Alice in $f$, and the receiver plays the role of Bob in $f$, and invoke $f$ several times, with random inputs $x \in X$. The receiver will be required to pick its inputs from an *unrevealing distribution* $\mathfrak{p}^Y$.
   (b) The sender checks if the frequencies of all the input-output pairs $(x, w)$ it sees are consistent with the receiver using $\mathfrak{p}^Y$.
   (c) The sender announces a subset of indices for which in the corresponding invocations, it obtained an *extreme* input-output pair.
   (d) The sender picks a random codeword from an appropriate code, and masks this codeword with the sequence of input-output pairs from the previous step, and sends it to the receiver.
   (e) The sender also sends the bit to be committed masked by a bit extracted from the codeword in the previous step.
2. Reveal phase: The sender sends its view from the commitment phase. The receiver checks that this is consistent with its view and the protocol (in particular, the purported codeword indeed belongs to the code, and for each possible value $(x, w)$ of the sender's input-output pair to $f$, the frequency of input-output pairs $(y, z)$ on its side are consistent with the function). If so, it accepts the purported committed bit.

In the above protocol, security will be obtained as follows:

• **Binding:** *extreme* input-output pairs are such that the sender cannot significantly equivocate during the reveal phase without being detected (even when the receiver is using the prescribed distribution $\mathfrak{p}^Y$). The error-correcting code ensures that a small number of equivocations cannot yield an explanation consistent with the protocol.

• **Hiding:** the *unrevealing distribution* $\mathfrak{p}^Y$ is such that there will be non-zero "confusion" for the receiver about the sender's input-output pair (even when restricted to extreme input-output pairs for the sender, and even if the receiver somewhat deviates from $\mathfrak{p}^Y$). We can interpret $f$ as a collection of channels, with the receiver adaptively choosing which channel to use for each symbol, with the restriction that a significant fraction of the time a "noisy" channel (i.e., one with capacity less than that of a noiseless channel over the same alphabet) is chosen. The error-correcting code will have a high rate (in fact, $1 - o(1)$) and by (an appropriate extension of) the weak converse of Shannon's channel coding theorem, we can show that that receiver has significant uncertainty about the codeword being sent. Then the extraction step (following [DORS08]) ensures that the committed bit is well-hidden.

The delicate part of this construction is to show that there will indeed be a set of extreme input-output pairs and an unrevealing distribution as required above. We point out that *we cannot use our results on statistical testability of the function evaluation game from Section 3.2 directly* to argue that binding would hold for all input-output pairs. This is because the game there requires the adversary to declare the input part of its purported view *before* invoking the function. Indeed, once we have a commitment functionality at our disposal, we can exploit the binding nature of this game; but to construct our commitment protocol this is not helpful.

Before pointing to our results on their existence, we sketch the definitions extreme input-output pairs and unrevealing distribution. We consider the matrix $\mathfrak{P}^f$ whose rows are indexed by Bob views $(y, z) \in Y \times Z$ and columns are indexed by Alice views $(x, w) \in X \times W$. The entry in this matrix indexed by Bob's view $(y, z)$ and Alice's view $(x, w)$ is $\mathfrak{p}^f[w, z|x, y]$.

**Extreme views and confusability.** An Alice view $(x, w)$ is an *extreme view* if the column indexed by $(x, w)$ cannot be written as convex linear combination of other columns in $\mathfrak{P}^f$.[5] This prevents Alice from claiming that in a set of invocations she obtained a certain extreme view, unless most of those invocations actually had this view.

The set of all extreme views of Alice is represented by $\mathfrak{a}^f$. We say $\mathfrak{a}^f$ is *confusable* if there exists a Bob view $(y, z)$ such that there are two different views $(x, w)$ and $(x', w')$ in $\mathfrak{a}^f$ such that the two entries in $\mathfrak{P}^f$ indexed by $((y, z), (x, w))$ and $((y, z), (x', w'))$ are both positive.

It is not necessary that $\mathfrak{a}^f$ be confusable. However, we show that if $f$ is not simple then $\mathfrak{a}^f$ *or* $\mathfrak{b}^f$ is confusable (Lemma 11), where $\mathfrak{b}^f$ is defined analogous to $\mathfrak{a}^f$. Below, we shall assume that $\mathfrak{a}^f$ is confusable (see Remark below).

**Unrevealing Distribution.** An unrevealing distribution $\mathfrak{p}^Y$ has three competing requirements: (a) it should not allow Bob to learn Alice's view exactly, even when it is given that Alice obtained a view in $\mathfrak{a}^f$; (b) Alice should be able to use her view to ensure that Bob is choosing his inputs from this distribution – or at least that he is not using a completely "revealing" input distribution; (c) extreme views for Alice should remain extreme (and hence unequivocable) even when Bob is choosing his inputs from this distribution.

The last condition is easily satisfied by picking $\mathfrak{p}^Y$ to have all of $Y$ as its support (with at least a constant weight on each $y \in Y$). To ensure the other conditions, consider the set $Y_0 \subseteq Y$ such that restricted to $y \in Y_0$, no two extreme views of Alice are confusable. Since $\mathfrak{a}^f$ is confusable, we know that there is at least one $y^* \in Y \setminus Y_0$. Alice needs to ensure that Bob is not choosing his inputs (almost) exclusively from $Y_0$. We say that $y^* \in Y \setminus Y_0$ is *mimiced* by $Y_0$, if there exists a probability distribution over $Y_0$ such that Alice's view when Bob is choosing his inputs from this distribution is indistinguishable from her view when Bob is using $y^*$. We show that if $y^* \in Y \setminus Y_0$ can be mimiced by $Y_0$ then $y^*$ is strictly redundant (Lemma 12).

The requisite unrevealing distribution $\mathfrak{p}^Y$ (as defined in Equation 6) can be obtained as an appropriate convex combination of the uniform distribution over $Y$ (so that the support of $\mathfrak{p}^Y$ is all of $Y$) and a distribution that puts all its weight on an unmimicable $y^* \in Y$ (Lemma 13).

**Remark.** The commitment protocol above could require that $f$ be used in a specific direction only (i.e., sender in the commitment could be required to play, say, the role of Alice in $f$) since Lemma 11 ensures only that either $\mathfrak{a}^f$ or $\mathfrak{b}^f$ is confusable. Since in our final OT protocol, we shall require both parties to make commitments to each other, this would require $f$ to be used in both directions. However, we can obtain a sharper result which requires $f$ to be used in only one direction, by obtaining commitment protocols that can use $f$ in either direction. In the final version, we describe how commitment in one direction can be leveraged to build a commitment protocol in the other direction in the $f$-hybrid model.

## 4.2 Passive-to-Active Security Compiler

For any redundancy free SFE $f$, we describe a "compiler" $\Pi_f$ which takes a 2-party semi-honest OT protocol $\pi_{\text{SH-OT}}$ in the $f$-hybrid and produces another 2-party protocol $\Pi_f(\pi_{\text{SH-OT}})$ in the commitment-hybrid such that if $\pi_{\text{SH-OT}}$ is a semi-honest $\binom{2}{1}$-OT protocol, then $\Pi_f(\pi_{\text{SH-OT}})$ is a UC secure $\binom{2}{1}$-OT protocol. For

[5] For simplicity, here we ignore the possibility that different columns could be parallel to each other. In Appendix D, we use maps $\phi_A$ (resp. $\phi_B$) to identify parallel columns (resp. rows) with each other.

convenience, we shall place a requirement on $\pi_{\text{SH-OT}}$ that it uses $f$ with uniformly independent inputs chosen independently by the two parties.[6]

We present the compiled protocol in two steps. In the first step, we build a protocol $\rho_{\widetilde{\text{OT}}}$ that UC-securely realizes the following functionality $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$, both of which are shown in Figure 2.

Then we shall implement an OT protocol in the $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$ -hybrid using techniques from [IPS08, IKO+11]. This involves invoking the protocol $\rho_{\widetilde{\text{OT}}}$ with *a large enough constant* security parameter, and using a (new) extended form of the IPS-compiler [IPS08]. The resulting protocol is a *constant rate* protocol in the $f$-hybrid model. This step is detailed in Appendix G. In the rest of this section, we focus on how to implement $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$ in the commitment hybrid model.

We prove the security of the compiled protocol in Appendix E. This gives us the following result.

**Lemma 6.** *Suppose $f$ is a 2-party randomized SFE which is redundancy free and passive-complete, and $\pi$ is a passive-secure protocol for $\mathcal{F}_{\text{OT}}$ in the $f$-hybrid model. Then the protocol $\rho_{\widetilde{\text{OT}}}$ in the $(f, \mathcal{F}_{\text{COM}})$-hybrid UC-securely realizes $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$, for $\delta(\kappa) = \kappa^{15/16}$ (where $\kappa$ is the security parameter).*

## 5 Full Characterization of Completeness

In this section we prove Theorem 1 from the constructions so far, and other observations regarding redundancy free functions. This derivation is summarized in Figure 3.

First, we introduce some definitions, following [MPR12]. In a *local protocol for $f$ which uses $g$ as a setup*, each party probabilistically maps her $f$-input to a $g$-input, calls $g$ once with that input and, based on her local view (i.e. her given $f$-input, the output of $g$, and possibly local randomness), computes her final output, without any further communication between the parties.

**Definition 2** ((Weak) Isomorphism [MPR12])**.** *We say that $f$ and $g$ are* isomorphic *to each other if there exist two local protocols $\pi_1$ and $\pi_2$ such that:*

1. *$\pi_1^g$ UC-securely realizes $f$ and $\pi_2^f$ UC-securely realizes $g$;*
2. *$\pi_1^g$ passive-securely realizes $f$ and $\pi_2^f$ passive-securely realizes $g$.*

*$f$ and $g$ are said to be* weakly isomorphic *to each other if condition 1 is satisfied.*

Note that isomorphism and weak isomorphism are equivalence relations. Also note that if two functions are weakly isomorphic to each other then one is UC-complete if and only if the other is. Further, this holds for standalone-completeness as well, since a *local protocol* that is standalone-secure must be UC-secure as well.

A *core* of a 2-party function $f$ is a redundancy free function $\widehat{f}$ which is weakly isomorphic to $f$. From Lemma 10, it follows that every finite 2-party function $f$ has a core. By the above observation about weak isomorphism, to characterize standalone or UC completeness of finite 2-party functions, it is enough to characterize it for redundnacy free functions. Note that Appendix A.2 gives an explicit procedure for finding a core of a given function. While the core is not unique, all the cores of a function are weakly isomorphic with each other.

[6] This suffices for our main result, since we shall invoke this compiler with a protocol $\pi_{\text{SH-OT}}$ that satisfies this requirement. However, we remark that a more tedious analysis could be used to remove this restriction.

**Functionality $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$.** Parametrized by a function $\delta(\kappa)$.

- Set $b = 1$ with probability $p = \delta(\kappa)$; otherwise $b = 0$.
- Provide the parties access to a (2-choose-1 bit) OT functionality. But if $b = 1$, let the adversary control this functionality.

**Protocol $\rho_{\widetilde{\text{OT}}}$.** Alice's inputs are a pair of bit $(x_0, x_1)$ and Bob's input is a choice bit $b$. In the following protocol, they will invoke several instances of $\pi_{\text{SH-OT}}$ with security parameter $\kappa_{\pi_{\text{SH-OT}}} = \kappa^c$ for some constant $c > 0$; $c$ chosen to be sufficiently small so that the total number of $f$ invocations (in either direction) in one session of $\pi_{\text{SH-OT}}$ is upperbounded by $\kappa^{1/8}$.

PHASE I: **Coin tossing in the well.** Alice and Bob commit to $2\kappa$ strings each (of poly$(\kappa)$ length, corresponding to the length of the random tape and input (two bits) required in $\pi_{\text{SH-OT}}$ with security parameter $\kappa_{\pi_{\text{SH-OT}}}$). Let Alice's strings be $\{\rho_i^A\}_{i=1}^{2\kappa}$, and Bob's strings be be $\{\rho_i^B\}_{i=1}^{2\kappa}$. Then Alice sends $2\kappa$ strings $\{\sigma_i^A\}_{i=1}^{2\kappa}$ to Bob and Bob sends $\{\sigma_i^B\}_{i=1}^{2\kappa}$ to Alice. Alice defines input/random-tapes $\{\tau_i^A\}_{i=1}^{2\kappa}$ where $\tau_i^A = \rho_i^A \oplus \sigma_i^B$. Similarly Bob defines input/random-tapes $\{\tau_i^B\}_{i=1}^{2\kappa}$ where $\tau_i^B = \rho_i^B \oplus \sigma_i^A$.

PHASE II: **Execution.** Alice and Bob engage in $2\kappa$ executions of protocol $\pi_{\text{SH-OT}}$ in the $f$-hybrid model. The security parameter of these executions is set to $\kappa_{\pi_{\text{SH-OT}}} = \kappa^c$ for a sufficiently small constant $c > 0$ so that the total number of $f$ invocations (in either direction) in one session of $\pi_{\text{SH-OT}}$ is upperbounded by $\kappa^{1/8}$.

In the $i^{\text{th}}$ instance, Alice and Bob use $\tau_i^A$ and $\tau_i^B$ respectively as their input/random tape.

PHASE III: **Cut and Choose.** Alice and Bob use a protocol in the $\mathcal{F}_{\text{COM}}$-hybrid to UC-securely generate random coins to randomly choose a subset $L \subseteq [2\kappa]$ with $|L| = \kappa$. For each $i \in L$, Alice and Bob must "open" their views in the $i^{\text{th}}$ execution of $\pi_{\text{SH-OT}}$: that is, Alice and Bob should reveal $\{\rho_i^A\}_{i \in L}$ and $\{\rho_i^B\}_{i \in L}$ respectively. Further each party should also report the outputs it received from $f$ in each invocation of $f$.

Then each party checks (a) if the messages received in the protocol are consistent with (i.e., has non-zero probability) the views opened/reported by the other party, and (b) if, for the sequence of invocations of $f$ over all the executions of $\pi_{\text{SH-OT}}$ that are opened, the consistency condition in the Statistical-Test (see Section 3.2) holds. If either of the checks fail, then the party should abort.

PHASE IV: **Random Selection.** Alice and Bob use a coin-tossing protocol in the $\mathcal{F}_{\text{COM}}$-hybrid, to randomly select an index in $\overline{L} = [2\kappa] \setminus L$.

PHASE V: **Finalizing.** Alice and Bob now perform a standard procedure for carrying out a fresh OT given the pre-computed OT instance selected in the previous step. (If $(s_0, s_1)$ and $u$ denote the inputs for Alice and Bob in the selected instance of $\pi_{\text{SH-OT}}$ and $v$ denotes the output that Bob received from it, then Bob sends $c := b \oplus u$ to Alice and Alice responds with $(r_0, r_1) := (x_0 \oplus s_c, x_1 \oplus s_{1-c})$ to Bob. Bob outputs $r_b \oplus v$.)

**Figure 2** The functionality $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$ and the protocol $\rho_{\widetilde{\text{OT}}}$

The *kernel* of a 2-party function $f$ is a function which outputs to the two parties only the "common information" that $f$ makes available to them. To formalize this, we define a weighted bipartite graph $\mathrm{G}(f)$ with partite sets are $X \times W$ and $Y \times Z$, and for every $(x, w) \in X \times W$ and $(y, z) \in Y \times Z$, the edge joining these two vertices is assigned weight $\mathtt{wt}\Big((x, w), (y, z)\Big) := \frac{\mathtt{p}^f[w,z|x,y]}{|X \times Y|}$. The kernel of $f$ is a randomized function which takes inputs $x \in X$ and $y \in Y$ from the parties, samples $(w, z) \xleftarrow{\$} f(x, y)$, and outputs to both parties the connected component of $\mathrm{G}(f)$ which contains the edge $\Big((x, w), (y, z)\Big)$.

**Definition 3** (Simple Function [MPR12])**.** *A (possibly randomized) 2-party function $f$ is said to be* simple *if it is isomorphic to its kernel.*

The combinatorial characterization of simple functions mentioned in Section 1.2 was shown in [MPR12]. Also, relying on the construction of [Kil00], the following was shown in [MPR12].

**Lemma 7.** *[MPR12] A finite 2-party function is passive-complete if and only if it is not simple.*

Now we can state our characterization as follows:

**Theorem 3.** *Suppose $f$ is a (possibly randomized) finite 2-party function. Then the following are equivalent.*

1. *$f$ is UC-complete.*
2. *$f$ is standalone-complete.*
3. *every core of $f$ is passive-complete.*
4. *$f$ has a core that is passive-complete.*

*Proof.* (1) $\Rightarrow$ (2) because UC-completeness implies standalone-completeness. If $f$ is standalone complete, every core of $f$ is also standalone complete (by weak isomorphism), and by Lemma 16, it is passive-complete. So (2) $\Rightarrow$ (3), which in turn implies (4). Our main work is in showing (4) $\Rightarrow$ (1): i.e., if $f$ has a core that is passive-complete then $f$ is UC-complete. Since $f$ is weakly isomorphic to its core, it is enough to show that any redundancy free function that is passive-complete is also UC-complete. To show that such a function is UC-complete, it is enough to show that OT can be UC-securely reduced to $f$, since OT is known to be UC-complete [Kil88, IPS08]. This is precisely what Theorem 2 proves. □

Finally, Theorem 1 follows easily from these results. The first part of Theorem 1 is a corollary to Theorem 3, since a function is passive-complete if and only if it is not simple (Lemma 7). The second part, regarding constant-rate reduction, follows from the fact that it holds for the case when $f$ is OT and $g$ is arbitrary (by a result in [IPS08]) and it holds for the case when $f$ is an arbitrary complete function and $g$ is OT (by Theorem 2).

## 5.1 A Special Case

In particular, for the class for asymmetric function evaluations, i.e. where only one of the parties receives outputs, we obtain the following dichotomy:

**Theorem 4** (Special Case: Dichotomy for Asymmetric 2-party SFE)**.** *Any asymmetric 2-party SFE is either standalone/UC-trivial or standalone/UC-complete.*

**Input:** A 2-party randomized SFE $f$, given as a matrix $\mathfrak{P}^f$ of conditional probabilities $\mathfrak{p}^f[w,z|x,y]$.
**Output:** A UC-secure protocol for $\mathcal{F}_{\text{OT}}$ in $f$-hybrid.

1. Compute a core $\widehat{f}$ of $f$ using the algorithm in Figure 4 inAppendix A.2.
2. Check if $\widehat{f}$ is simple or not (using combinatorial characterization in [MPR12], given in Section 1.2).
3. If $\widehat{f}$ is simple, declare that $f$ is not complete. (By Lemma 16 and Lemma 7.)
4. Else (i.e., $\widehat{f}$ is not simple):
   (a) Construct a passive-secure protocol $\pi_{\text{SH-OT}}$ for $\mathcal{F}_{\text{OT}}$ in $\widehat{f}$-hybrid (using the construction in [MPR12]).
   (b) Construct a UC-secure protocol $\pi_{\text{COM}}$ for $\mathcal{F}_{\text{COM}}$ in $\widehat{f}$-hybrid (using protocol in Section 4.1).
   (c) Compile $\pi_{\text{SH-OT}}$ into a UC-secure protocol $\rho_{\widetilde{\text{OT}}}$ for $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$ in ($\widehat{f}$,$\mathcal{F}_{\text{COM}}$)-hybrid (applying Lemma 6 to $\widehat{f}$). Compose $\rho_{\widetilde{\text{OT}}}$ with $\pi_{\text{COM}}$ to obtain a UC-secure protocol for $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$ in $\widehat{f}$-hybrid.
   (d) Instantiate $\pi_{\text{COM}}$ with a large enough constant security parameter; using statistical to perfect lemma of [IKO+11], interpret this as a (perfectly) UC-secure constant rate protocol $\pi_{c\text{-OT}}$ for $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ in $\widehat{f}$-hybrid (using Lemma 18).
   (e) Construct a UC-secure constant rate protocol $\pi_{\text{STRING-OT}}$ for $\mathcal{F}_{\text{STRING-OT}[\ell]}$ in $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ -hyrbid (using protocol in [IKO+11]).
   (f) Construct a UC-secure constant rate protocol $\pi_{\text{OT}}$ for $\mathcal{F}_{\text{OT}}$ in $(\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}, \mathcal{F}_{\text{STRING-OT}[\ell]})$-hybrid (using the extension of the IPS-compiler given in Lemma 21).
   (g) Compose $\pi_{\text{OT}}$ with $\pi_{c\text{-OT}}$ and $\pi_{\text{STRING-OT}}$ to obtain the final OT protocol in $\widehat{f}$-hybrid. Finally, compose with a UC-secure protocol for $\widehat{f}$ in $f$-hybrid[a] to obtain a UC-secure constant rate OT protocol in $f$-hybrid.

[a] Such a protocol exists as $\widehat{f}$ is weakly isomorphic to $f$. In fact, the natural protocol for this is not only UC-secure, but passive-secure as well. This ensures that our final protocol for OT in $f$-hybrid model also enjoys both levels of security.

**Figure 3** This algorithm summarizes our results. It tests whether a function $f$ is UC-complete or not. If $f$ is complete then it constructs $\mathcal{F}_{\text{OT}}$ in $\mathcal{F}_f$-hybrid at constant rate.

This theorem was proven for the deterministic case in [BMM99]. Note that in the randomized case, if there exists an input for the receiver such that it can determine the sender's input with certainty, then the SFE is standalone-/UC-trivial; because the protocol where the sender sends her input to the receiver is a standalone-/UC-secure protocol. On the other hand, if all receiver inputs are such that the receiver input cannot be predicted with certainty then our construction provides a standalone-/UC-secure construction of OT from this SFE.

# 6 Conclusion

While we have closed a line of work on cryptographic complexity theory that has sought to characterize complete functionalities for 2-party SFE, we conclude with a new line of investigation this points to. The focus so far has mostly been on *qualitative* aspects of cryptographic complexity. However, as mentioned in the introduction, *quantitative* questions of cryptographic complexity are closely related to questions of circuit complexity of functions. Our constant-rate protocols establish a unified measure of cryptographic complexity (up to constant factors): given a (2-party) function $g$, asymptotically, how many copies of *any*

*complete finite 2-party function* are needed for securely[7] evaluating $g$, per instance of $g$? While one could have defined this with respect to a specific complete function like OT, the fact that it is invariant (up to constants) of the choice of this function makes it a more fundamental measure of complexity.

As mentioned in the introduction, finding explicit functions with super-linear lowerbounds for cryptographic complexity is no easier than doing the same for circuit complexity. However, even the presumably easier problem of showing super-linear lowerbounds for *random* functions remains open for cryptographic complexity. We leave it a wide-open problem to study cryptographic complexity lower-bounds, and possibly discover techniques that are applicable to other complexity notions as well.

**Acknowledgments.** We thank Vinod Prabhakaran for helpful discussions on the converse of the Channel Coding Theorem.

[7] One could use passive or active security in this definition, but clearly the latter admits the possibility of stronger lower-bound techniques.

## References


[BCR86] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In Andrew M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 234–238. Springer, 1986. 4

[Bea95] Donald Beaver. Precomputing oblivious transfer. In Don Coppersmith, editor, *CRYPTO*, volume 963 of *Lecture Notes in Computer Science*, pages 97–109. Springer, 1995. 1

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In Janos Simon, editor, *STOC*, pages 1–10. ACM, 1988. 5

[BIKK13] Amos Beimel, Yuval Ishai, Ranjit Kumaresan, and Eyal Kushilevitz. On the cryptographic complexity of the worst functions. http://www.cs.umd.edu/~ranjit/BIKK.pdf. Retrieved Oct 16, 2013, 2013. 4

[BMM99] Amos Beimel, Tal Malkin, and Silvio Micali. The all-or-nothing nature of two-party secure computation. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 80–97. Springer, 1999. 2, 5, 20

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Electronic Colloquium on Computational Complexity (ECCC) TR01-016, 2001. Previous version "A unified framework for analyzing security of protocols" available at the ECCC archive TR01-016. Extended abstract in FOCS 2001. 22

[Can05] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067, 2005. Revised version of [Can01]. 9

[Can08] Ran Canetti, editor. *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008*, volume 4948 of *Lecture Notes in Computer Science*. Springer, 2008. 23

[CC06] Hao Chen and Ronald Cramer. Algebraic geometric secret sharing schemes and secure multi-party computations over small fields. In Dwork [Dwo06], pages 521–536. 48

[CCD88] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols. In Janos Simon, editor, *STOC*, pages 11–19. ACM, 1988. 5

[CDMW09] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Simple, black-box constructions of adaptively secure protocols. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 387–402. Springer, 2009. 1, 7

[CK88] Claude Crépeau and Joe Kilian. Achieving oblivious transfer using weakened security assumptions (extended abstract). In *FOCS*, pages 42–52. IEEE, 1988. 1, 4, 7

[CMW04] Claude Crépeau, Kirill Morozov, and Stefan Wolf. Efficient unconditional oblivious transfer from almost any noisy channel. In Carlo Blundo and Stelvio Cimato, editors, *SCN*, volume 3352 of *Lecture Notes in Computer Science*, pages 47–59. Springer, 2004. 1, 2, 5, 7

[Cré87] Claude Crépeau. Equivalence between two flavours of oblivious transfers. In Pomerance [Pom88], pages 350–354. 2

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991. 14

[DI06] Ivan Damgård and Yuval Ishai. Scalable secure multiparty computation. In Dwork [Dwo06], pages 501–520. 48

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008. 15, 40

[Dwo06] Cynthia Dwork, editor. *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*. Springer, 2006. 22, 23

[FGMO05] Matthias Fitzi, Juan A. Garay, Ueli M. Maurer, and Rafail Ostrovsky. Minimal complete primitives for secure multi-party computation. *J. Cryptology*, 18(1):37–61, 2005. 5

[FM00] Matthias Fitzi and Ueli M. Maurer. From partial consistency to global broadcast. In F. Frances Yao and Eugene M. Luks, editors, *STOC*, pages 494–503. ACM, 2000. 5

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play ANY mental game. In Alfred V. Aho, editor, *STOC*, pages 218–229. ACM, 1987. See [Gol04, Chap. 7] for more details. 4, 48

[Gol04] Oded Goldreich. *Foundations of Cryptography: Basic Applications*. Cambridge University Press, 2004. 23

[GV87] Oded Goldreich and Ronen Vainish. How to solve any protocol problem - an efficiency improvement. In Pomerance [Pom88], pages 73–86. 2, 4

[Hai08] Iftach Haitner. Semi-honest to malicious oblivious transfer - the black-box way. In Canetti [Can08], pages 412–426. 1, 7

[HIKN08] Danny Harnik, Yuval Ishai, Eyal Kushilevitz, and Jesper Buus Nielsen. OT-combiners via secure computation. In Canetti [Can08], pages 393–411. 48

[HM86] Stuart Haber and Silvio Micali. Unpublished manuscript, 1986. 2, 4

[IKLP06] Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-box constructions for secure computation. In *STOC*, pages 99–108. ACM, 2006. 1, 7

[IKO+11] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, Amit Sahai, and Jürg Wullschleger. Constant-rate oblivious transfer from noisy channels. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 667–684. Springer, 2011. 7, 17, 20, 45, 46

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 572–591. Springer, 2008. 1, 4, 7, 17, 19, 45, 47

[Ish11] Yuval Ishai, editor. *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, volume 6597 of *Lecture Notes in Computer Science*. Springer, 2011. 24

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In Janos Simon, editor, *STOC*, pages 20–31. ACM, 1988. 1, 2, 4, 19

[Kil91] Joe Kilian. A general completeness theorem for two-party games. In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *STOC*, pages 553–560. ACM, 1991. 1, 2, 4

[Kil00] Joe Kilian. More general completeness theorems for secure two-party computation. In F. Frances Yao and Eugene M. Luks, editors, *STOC*, pages 316–324. ACM, 2000. 1, 2, 3, 5, 19

[KKMO00] Joe Kilian, Eyal Kushilevitz, Silvio Micali, and Rafail Ostrovsky. Reducibility and completeness in private computations. *SIAM J. Comput.*, 29(4):1189–1208, 2000. 5

[KM11] Daniel Kraschewski and Jörn Müller-Quade. Completeness theorems with constructive proofs for finite deterministic 2-party functions. In Ishai [Ish11], pages 364–381. 1, 2, 5

[KMO94] Eyal Kushilevitz, Silvio Micali, and Rafail Ostrovsky. Reducibility and completeness in multi-party private computations. In *FOCS*, pages 478–489. IEEE Computer Society, 1994. 5

[Kra13] Daniel Kraschewski. Completeness theorems for all finite stateless 2-party primitives. Cryptology ePrint Archive, Report 2013/161, 2013. http://eprint.iacr.org/. 1

[LOZ12] Yehuda Lindell, Eran Omri, and Hila Zarosim. Completeness for symmetric two-party functionalities - revisited. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 116–133. Springer, 2012. 5

[MPR12] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. A unified characterization of completeness in secure function evaluation. To appear at INDOCRYPT, 2012. 1, 2, 3, 5, 6, 17, 19, 20

[Pom88] Carl Pomerance, editor. *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, volume 293 of *Lecture Notes in Computer Science*. Springer, 1988. 23

[Rab81] M. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981. 4

[RB89] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In David S. Johnson, editor, *STOC*, pages 73–85. ACM, 1989. 5

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15:78–88, January 1983. 4

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *STOC*, pages 209–213. ACM, 1979. 1

# A Results on Redundancy

## A.1 Strict and Self Redundancy

**Claim 2** (Restated.) *Suppose $M$ is a $q \times q$ stochastic matrix such that $\max(M - I) \geq \delta > 0$. Also, suppose $P$ is an $s \times q$ matrix such that for any two columns $c_i$ and $c_j$ of $P$, $\inf_\gamma \|c_i - \gamma c_j\|_\infty \geq \epsilon > 0$. Then $\|PM - P\|_\infty \geq \delta\epsilon$.*

*Proof.* Firstly, since we require that $\inf_\gamma \|c_i - \gamma c_j\|_\infty \geq \epsilon$, $\|c_i\|_\infty \geq \epsilon$, for any column $c_i$ in $P$. Also note that $M \neq I$ since $\max(M - I) > 0$. We need to establish a lowerbound on $\|PN\|_\infty$, where $N = M - I$.

For this, we prove the following by induction, for all integers $t \geq 1$. Suppose $N$ is a $q \times q$ matrix with $t$ non-zero rows, such that all diagonal entries of $N$ are at most 0 and all non-diagonal entries of $N$ are at least 0; also every row of $N$ sums up to 0. Then $\|PN\|_\infty \geq \delta\epsilon$ (where $P$ is as given).

Base case: $t = 1$. Consider $(i, j)$ such that $N_{ij} = \max(N)$. Since there is only one non-zero row, the $j^{\text{th}}$ column in $N$ has this as the only non-zero entry. Hence the $j^{\text{th}}$ column in $PN$ is $\max(N) \cdot c_i$. Since $\max(N) \geq \delta$, $\|\max(N) \cdot c_i\|_\infty \geq \delta\|c_i\|_\infty \geq \delta\epsilon$. Hence, $\|PN\|_\infty \geq \delta\epsilon$.

Base case: $t = 2$. Again, consider $(i, j)$ such that $N_{ij} = \max(N)$. If this is the only non-zero entry in the $j^{\text{th}}$ column in $N$, then the same analysis as before holds. Otherwise, there is one more non-zero entry, say $N_{i'j}$ in that column. Then the $j^{\text{th}}$ column in $PN$ equals $N_{ij}c_i + N_{i'j}c_j = N_{ij}(c_i + \gamma c_j)$ where $\gamma = N_{i'j}/N_{ij}$. Hence, $\|PN\|_\infty \geq N_{ij}\|c_i + \gamma c_j\|_\infty \geq \delta\epsilon$.

Induction step. Suppose $N$ has $t \geq 3$ non-zero rows. We shall construct $N'$ with non-negative non-dagonal entries, with each row summing up to 0, with $t' < t$ non-zero rows and with $\max(N') \geq \max(N)$, such that $\|PN\|_\infty \geq \|PN'\|_\infty$. Then by the induction hypothesis, it follows that $\|PN'\|_\infty \geq \delta\epsilon$.

To construct $N'$ from $N$, consider $(i, j)$ such that $N_{ij} = \max(N)$. Let $k$ be a non-zero row in $N$ such that $k \neq i$ and $k \neq j$. (This is possible since $t' > 2$.) For $j' \neq k$ and all $i'$, we set $N'_{i'j'} = N_{i'j'} - \frac{N_{kj'}}{N_{kk}} N_{i'k}$. Also we set $N'_{i'k} = 0$ for all $i'$. This zeroes out the $k^{\text{th}}$ row and $k^{\text{th}}$ column of $N'$. Note that $N_{kk} < 0$ and for $i' \neq k$ and $j' \neq k$ we have $N_{kj'}, N_{i'k} \geq 0$; so $N'_{i'j'} \geq N_{i'j'}$ for all elements except those in the $k^{\text{th}}$ row or column (which are 0 in $N'$). In particular, $N'_{ij} \geq N_{ij}$. So $\max(N') \geq \max(N)$.

We can write $N' = NT$, where $T$ is a $q \times q$ matrix defined as follows:

$$T_{i'j'} = \begin{cases} 1 & \text{if } i' = j' \neq k. \\ -\frac{N_{kj'}}{N_{kk}} & \text{if } i' = k \neq j'. \\ 0 & \text{otherwise.} \end{cases}$$

Hence $\|PN'\|_\infty = \|PNT\|_\infty \leq \|PN\|_\infty \|T\|_\infty$ by the sub-multiplicativity of the $\|\cdot\|_\infty$ norm. But $\|T\|_\infty = 1$ (since the $k^{\text{th}}$ row has positive numbers that sum up to 1, and the other rows have a single non-zero entry equal to 1). Thus, $\|PN'\|_\infty \leq \|PN\|_\infty$ as required. □

**Lemma 3** (Restated.) *Suppose a 2-party function $f : X \times Y \to W \times Z$ is left redundancy free. Let $\mathfrak{p}^Y$ be a probability distribution over $Y$. Let the probability matrices $\{P^x\}_{x \in X}$, be defined by $P^x_{(y,z),w} = \mathfrak{p}^f[w, z|x, y]\mathfrak{p}^Y[y]$. Then there is a constant $\epsilon_f > 0$ (depending only on $f$) such that $\mathfrak{D}(P^1, \cdots, P^m) \geq \epsilon_f \min(\mathfrak{p}^Y)$.*

*Proof.* Consider any tuple $(\hat{x}, \{M_i, \alpha_i\}_{i=1}^n)$ as in the definition of irredundancy such that it is not the case that $\alpha_{\hat{x}} = 1$ and $M_{\hat{x}} = I$ (so that the denominator in the irredundancy parameter is non-zero). Let the denominator (i.e., the probability of deviation) be $1 - \alpha_{\hat{x}}\|P^{\hat{x}} \cdot \langle M_{\hat{x}} \rangle_I\|_{\text{sum}} = \delta_0 > 0$. We need to show that such tuples cannot achieve arbitrarily low values of the irredundancy parameter.

We consider two cases (in terms of a constant $\frac{1}{2} \geq \epsilon_0 > 0$ to be specified):

- if $\alpha_{\hat{x}} \geq 1 - \epsilon_0\delta_0$, we derive the lower-bound using the fact that $\hat{x}$ is not self-redundant, and
- if $\alpha_{\hat{x}} < 1 - \epsilon_0\delta_0$, we derive the lower-bound using the fact that $\hat{x}$ is not strictly redundant.

Below, we define the matrices $Q^x$ to be similar to $P^x$ but with using the uniform distribution over $y$ rather than the distribution $\mathfrak{p}^Y$. That is, $Q^x_{(y,z),w} = \mathfrak{p}^f[w, z|x, y] \cdot \frac{1}{n}$.

Case $\alpha_{\hat{x}} \geq 1 - \epsilon_0\delta_0$: Firstly note that there must be $w \in W$ such that $\alpha_{\hat{x}}(M_{\hat{x}})_{ww} \leq (1 - \delta_0)$, because otherwise $\alpha_{\hat{x}}\|P^{\hat{x}} \cdot \langle M_{\hat{x}} \rangle_I\|_{\text{sum}}$ will be strictly greater than $1 - \delta_0$. Then, $(M_{\hat{x}})_{ww} \leq \frac{1-\delta_0}{\alpha_{\hat{x}}} \leq \frac{1-\delta_0}{1-\epsilon_0\delta_0}$. Since $M_{\hat{x}}$ is a stochastic matrix, there must be $w' \neq w$ such that $(M_{\hat{x}})_{ww'} \geq \frac{1}{q}(1 - \frac{1-\delta_0}{1-\epsilon_0\delta_0}) = \frac{\delta_0}{q}(\frac{1-\epsilon_0}{1-\epsilon_0\delta_0}) \geq \frac{\delta_0(1-\epsilon_0)}{q}$. Since we have required $\epsilon_0 \leq \frac{1}{2}$, this implies $(M_{\hat{x}})_{ww'} \geq \frac{\delta_0}{2q}$. Thus $\max(M_{\hat{x}} - I) \geq \frac{\delta_0}{2q}$.

Now, since $\hat{x}$ is not a self left-redundant input, for any two non-zero columns $c_i$ and $c_j$ of $Q^{\hat{x}}$, it is not the case that $c_i$ is proportional to $c_j$. That is, the point $c_i \in \mathbb{R}^q$ lies outside the line through origin and $c_j$. Note that the matrix $Q^{\hat{x}}$ depends only on $f$ (and one of a finite number of possibilities for $\hat{x}$); so the infimum, $\inf_{i,j,\gamma} \|c_i - \gamma c_j\|_\infty$ is lowerbounded by some constant $\epsilon > 0$ that depends only on $f$. Then by considering $P$ in Claim 2 to be $Q^{\hat{x}}$ without the zero columns, we have $\|Q^{\hat{x}}M - Q^{\hat{x}}\|_\infty \geq \frac{\delta_0}{2q}\epsilon$. Since each row of $P^{\hat{x}}$ is obtained by multiplying a row of $Q^{\hat{x}}$ by $n\mathfrak{p}^Y[y]$ for some $y$, $\|P^{\hat{x}}M - P^{\hat{x}}\|_\infty \geq n \cdot \min(\mathfrak{p}^Y) \cdot \frac{\delta_0}{2q}\epsilon$. We use this in the last step below.

$$
\begin{aligned}
\frac{\|(\sum_x \alpha_x P^x M_x) - P^{\hat{x}}\|_\infty}{1 - \alpha_{\hat{x}}\|P^{\hat{x}} \cdot \langle M_{\hat{x}} \rangle_I\|_{\text{sum}}} &= \frac{\|\alpha_{\hat{x}}P^{\hat{x}}M_{\hat{x}} - P^{\hat{x}} + \sum_{x \neq \hat{x}} \alpha_x P^x M_x\|_\infty}{\delta_0} \\
&\geq \frac{\|\alpha_{\hat{x}}P^{\hat{x}}M_{\hat{x}} - P^{\hat{x}}\|_\infty - (1 - \alpha_{\hat{x}})}{\delta_0} \\
&= \frac{\|(P^{\hat{x}}M - P^{\hat{x}}) + (1 - \alpha_{\hat{x}})P^{\hat{x}}M\|_\infty - (1 - \alpha_{\hat{x}})}{\delta_0} \\
&\geq \frac{\|(P^{\hat{x}}M - P^{\hat{x}})\|_\infty - 2(1 - \alpha_{\hat{x}})}{\delta_0} \geq \frac{\|(P^{\hat{x}}M - P^{\hat{x}})\|_\infty - 2\delta_0\epsilon_0}{\delta_0} \\
&\geq \frac{\epsilon}{2q} \cdot n \cdot \min(\mathfrak{p}^Y) - 2\epsilon_0.
\end{aligned}
$$

We set $\epsilon_0 = \frac{\epsilon}{8q} \cdot n \cdot \min(\mathfrak{p}^Y)$. Then, $\frac{\|(\sum_x \alpha_x P^x M_x) - P^{\hat{x}}\|_\infty}{1 - \alpha_{\hat{x}}\|P^{\hat{x}} \cdot \langle M_{\hat{x}} \rangle_I\|_{\text{sum}}} \geq 2\epsilon_0$.

Case $\alpha_{\hat{x}} < 1 - \delta_0\epsilon_0$. Let $K := \sum_x \alpha_x P^x M_x - P^{\hat{x}} = (\sum_{x \neq \hat{x}} \alpha_x P^x M_x) - P^x(I - \alpha_{\hat{x}}M_{\hat{x}})$. Using Claim 1, we can write $(I - \alpha_{\hat{x}}M_{\hat{x}})^{-1} = \frac{1}{1-\alpha_{\hat{x}}}M$ for a stochastic matrix $M$. Then $K(I - \alpha_{\hat{x}}M_{\hat{x}})^{-1} = (\sum_{x \neq \hat{x}} \alpha'_x P^x M'_x) - P^{\hat{x}}$, where $\sum_{x \neq \hat{x}} \alpha'_x = 1$ and $M'_x$ are stochastic. Now, we note that the set of points $\mathcal{R}_{\hat{x}} := \{ (\sum_{x \neq \hat{x}} \alpha'_x P^x M'_x) \mid \alpha'_x \geq 0, \sum_{x \neq \hat{x}} \alpha'_x = 1 \text{ and } M'_x \text{ stochastic } \}$ is a *closed* region (with $nr \times q$ matrices treated as points in $\mathbb{R}^{nrq}$). Since $\hat{x}$ is not a redundant input, $P^{\hat{x}}$ is not in $\mathcal{R}_{\hat{x}}$. Then, there is a positive distance between $P^{\hat{x}}$ and $\mathcal{R}_{\hat{x}}$ (under various norms, including the $\|\cdot\|_\infty$ norm we use in the numerator

of the irredundancy parameter). In fact, we can analogously define $\mathcal{R}_x$ for all $x \in [m]$ and for all such $x$, $P^x \notin \mathcal{R}_x$. So we can define a positive constant $\epsilon_1 := \min_{x\in[m]} \|\mathcal{R}_x - P^x\|_\infty$, where $\epsilon_1 > 0$ depends only on $f$.

Then $\|K(I - \alpha_{\hat{x}} M_{\hat{x}})^{-1}\|_\infty \geq \epsilon_1$. But $\|(I - \alpha_{\hat{x}} M_{\hat{x}})^{-1}\|_\infty \leq \frac{1}{1-\alpha_{\hat{x}}}$ and by the sub-multiplicativity of the norm, $\|K\|_\infty \geq \epsilon_1(1 - \alpha_{\hat{x}}) \geq \epsilon_1\delta_0\epsilon_0$. Thus, we have $\frac{\|(\sum_x \alpha_x P^x M) - P^{\hat{x}}\|_\infty}{1-\alpha_{\hat{x}}\|P^{\hat{x}} \cdot \langle M_{\hat{x}} \rangle_I\|_{\text{sum}}} = \frac{\|K\|_\infty}{\delta_0} \geq \epsilon_1\epsilon_0$.

Thus in either case, the expression in the irredundancy parameter is lowerbounded by a positive constant that depends only on $f$, irrespective of the choice of the tuple $(\hat{x}, \{M_i, \alpha_i\}_{i=1}^n)$. □

## A.2 An Algorithm to Find a Core

In this section we show that every function has a core and we give an explicit algorithm to find one. We begin by proving two results.

**Lemma 8.** *Suppose $x^* \in X$ is a strictly left-redundant input of a function $f : X \times Y \to W \times Z$. Let $g$ be the function obtained by restricting $f$ to the domain $(X \setminus \{x^*\}) \times Y$. Then, $f$ and $g$ are weakly isomorphic.*

*Proof.* Since $x^*$ is strictly redundant, there exists $\{(\alpha_x, P^x, M_x) | x \in X\}$ and $x^*$ such that $P^{x^*} = \sum_{x\in X} \alpha_x P^x M_x$, $\sum_{x\in X} \alpha_x = 1$, $\alpha_x \geq 0$ (for all $x \in X$) and $\alpha_{x^*} = 0$.

First, we show that there exists standalone/UC secure local protocol for $f$ in the $g$-hybrid. Bob always feeds his input $y$ to $g$. If Alice's input is $x \neq x^*$, simply feed $x$ to $g$ and both parties obtain the correct output distribution. If Alice's input is $x = x^*$, then sample sample $x'$ from $X \setminus \{x^*\}$ according the the probability distribution $\{\alpha_x | x \in X \setminus \{x\}\}$. Alice invokes $g$ with input $x'$. It receives outcome $w'$ from the function. Sample an output $w$ according to the distribution in $M_{x'}$ corresponding to output $w'$ (i.e. the distribution represented by the row corresponding to output symbol $w'$). By definition of strict row redundancy, the protocol is correct. Simulation is trivial for both malicious Alice and Bob cases (the simulators just forward the input provided to the $g$-hybrid to the external ideal functionality and forward the output back to the party).

For the other direction, i.e. a secure protocol for $g$ in $f$ hybrid, the protocol is trivial. Both parties invoke $f$ with their respective inputs and report their outputs. The simulator for malicious Bob is trivial (simply forward the input to $f$-hybrid to the external ideal functionality and report back the received outcome). The simulator for malicious Alice is as follows. If the $f$-hybrid is invoked with $x \neq x^*$, then simply forward that input to the ideal functionality $g$ and report back the received output. If the $f$-hybrid is invoked with $x = x^*$, then sample $x'$ according to the distribution $\{\alpha_x | x \in X \setminus \{x\}\}$. Invoke the ideal functionality $g$ with input $x'$ and receive the outcome $w'$. Translate $w'$ into $w$ by sampling according to the distribution in the row of $M_{x'}$ corresponding to the output symbol $w'$. The simulation is perfect due to strict left redundancy. □

**Lemma 9.** *Suppose $x \in X$ is a self left redundant input for $f$, and the two columns corresponding to $w$ and $w'$ in $P^x$ are scalar multiples of each other. Suppose $g$ is a function obtained by transferring all probability mass of $w'$-th column of $P^x$ to $w$-th column: i.e., for all $y \in Y, z \in Z$, $\mathfrak{p}^g[w, z|x, y] = \mathfrak{p}^f[w, z|x, y] + \mathfrak{p}^f[w', z|x, y]$ and $\mathfrak{p}^g[w', z|z, y] = 0$. Then, $f$ and $g$ are weakly isomorphic.*

*Proof.* Protocol for $f$ in $g$-hybrid is constructed as follows. Alice and Bob forwards their inputs to $g$. Alice, on input $x$, if she receives $w$ as the output translates it into $w'$ with probability $\mu/(1 + \mu)$, where the column corresponding to $w'$ was $\mu$ times the column corresponding to $w$. Correctness is trivial. Simulator

for malicious Bob simply forwards the input to $g$ hybrid to the external ideal functionality and forwards the received output. Simulator for malicious Alice forwards the input $g$ hybrid to the external ideal functionality. If the input was $x$ and the received outcome was $w'$ then it forwards $w$ to the adversary; otherwise it simply forwards the received outcome.

Protocol for $g$ in the $f$ hybrid is constructed as follows. Both parties forwards their inputs to $f$. If Alice receives output $w'$ then it outputs $w$; otherwise she honestly reports the received output. Simulation for malicious Bob is trivial. Simulation for malicious Alice does the following: It forwards the input for $f$ hybrid to the external ideal functionality and receives the output. If the input was $x$ and the output was $w$, then it reports $w'$ with probability $\mu/(\mu+1)$; otherwise it honestly reports $w$. □

Similar results also hold for strict right redundancy and self right redundancy. Using these we obtain an algorithm that given a function $f$ finds a core $\widehat{f}$. The algorithm is shown in Figure 4. If $f$ is redundancy free, then it is a core of itself. Otherwise, by Lemma 1, $f$ is either strictly redundant or self redundant. In the former case, obtain $g$ as in Lemma 8, and in the latter case obtain $g$ as in Lemma 9. Note that in either case $g$ is guaranteed to be well-defined (and in particular does not have an empty input or output domain). Then recursively apply this algorithm to $g$. Note that at every step the number of pairs $(x, w) \in X \times W$ or the number of pairs $(y, z) \in Y \times Z$ such that $\mathfrak{p}^f[w, z|x, y] > 0$ strictly reduces. Since we will never reach a situation where one of these sets become empty, the procedure must terminate with a well-defined function $\widehat{f}$ that is redundancy free. Since the function we chose at every step is weakly isomorphic to the previous function, $\widehat{f}$ is weakly isomorphic to $f$. Thus it is a core of $f$.

**Input:** A 2-party randomized SFE $f$.
**Output:** A core of $f$, $\widehat{f}$.

1. Repeat:
   (a) Comment: $f$ is redundant if and only if $f$ is either strictly redundant or self redundant (By Lemma 1).
   (b) For each $x \in X$, let $P^x$ be the $nr \times q$ matrix with $P^x_{(y,z),w} = \frac{1}{n}\mathfrak{p}^f[w, z|x, y]$
   (c) Check whether Alice has a strictly redundant input or not and if so update $f$ as follows:
      i. For every input of Alice $x^* \in X$:
         A. Use linear programming to find if there exists a distribution $\{\alpha_x\}_{x\in X}$ with $\alpha_{x^*} = 0$, and a set of $q \times q$ stochastic matrices $\{M_x\}_{x\in X}$ such that: $\sum_{x\in X} \alpha_x P^x M_x = P^{x^*}$.
         B. If a feasible solution exists, then update $f$ by removing $x^*$ from the domain $X$.
   (d) Check whether Alice has a self redundant input or not and if so update $f$ as follows:
      i. Check whether there exists $x \in X$ such that for outputs $w, w' \in W$ the columns corresponding to $w$ and $w'$ are scalar multiples of each other in $P^x$.
      ii. If such $(x, w, w')$ exists then:
         A. Update $f$ so that whenever $w'$ used to be output to Alice when her input is $x$, now $w$ is output.
   (e) Analogous to the above two steps, check whether Bob has a strictly redundant or a self redundant input, and in either case carry out the analogous updates.
2. Until $f$ has not been modified in an iteration
3. Output current $f$ as $\widehat{f}$.

**Figure 4** Algorithm to find the core of a 2-party SFE $f$.

In particular, we have the following:

**Lemma 10.** *Every finite 2-party function has a core.*

# B Statistically Testable Function Evaluation

**Lemma 4** (Restated.) *If $f$ is redundancy free, and $\mathfrak{p}^X$ and $\mathfrak{p}^Y$ are constant distribution which have full support over $X$ and $Y$ respectively, then the probability that any adversary wins in Statistical-Test$(f, \mathfrak{p}^Y, \mathfrak{p}^X; N)$ is* $\mathrm{negl}(N)$.[8]

*Proof.* We shall only argue that if $f$ is left-redundancy free, then the probability of any adversary winning the Left-Statistical-Test$(f, \mathfrak{p}^Y; N)$ is negligible in $N$. The argument for the Right-Statistical-Test is similar. Then the result follows by union bound.

The experiment involves the adversary adaptively choosing $x_i$. To facilitate the analysis, instead we shall analyze *all* choices of $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w}, \tilde{\mathbf{w}})$, but restricted to $\mathbf{w}$ being "typical" for a randomly chosen $\mathbf{y}$ (for the given vector $\mathbf{x}$). Since this would hold except with negligible probability (over random choice of $\mathbf{y}$ and the randomness of $f$), this restriction will not affect the conclusion. Then, assuming that the adversary satisfies the sufficient-distance condition, we analyze the probability of the consistency condition holding. We shall argue that this probability is negligible if $f$ is redundancy free.

We shall consider the expectation of the quantity $\mu_{\tilde{w},\tilde{x},y,z} - \mathfrak{p}^f[\tilde{w}, z|\tilde{x}, y]\mu_{\tilde{x},y}$ and argue that for some value of $x, \tilde{y}, \tilde{z}$, the absolute value of this expectation should be large, say, $\Omega(N^{7/8})$. Note that, once we fix $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w}, \tilde{\mathbf{w}})$, then for any quadruple $(\tilde{x}, x, w, \tilde{w})$, $\mu_{\tilde{w},\tilde{x},y,z}$ and $\mu_{\tilde{x},y}$ can both be written as the sum of i.i.d indicator random variables. This is because the random experiment we consider consists only of picking $y_i, z_i$, for each $i$ independently: if $x_i = x$ and $w_i = w$, then $\Pr[y_i = y, z_i = z] = \mathfrak{p}^{f,Y}[y, z|x, w] := \frac{\mathfrak{p}^Y[y]\cdot\mathfrak{p}^f[w,z|x,y]}{\sum_{z',y'}\mathfrak{p}^Y[y']\cdot\mathfrak{p}^f[w,z'|x,y']}$. Then by Chernoff bounds, we obtain that except with negligible probability, the consistency condition will be violated.

We shall define the set Good of "good" $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w})$ in which, for each $\tilde{x}, x, w$, the number of positions $i$ with $w_i = w$ among the positions $i$ with $\tilde{x}_i = \tilde{x}, x_i = x$ is as expected (over uniformly random i.i.d $y_i$ and randomness of $f$) up to an additive error of $N^{2/3}$. (Note that this assumption is non-trivial only when there are at least $N^{2/3}$ positions with $\tilde{x}_i = \tilde{x}, x_i = x$.) The analysis below would be for every tuple $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w}) \in$ Good. W.l.o.g we assume that for each $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w})$ the adversary chooses $\tilde{\mathbf{w}}$ deterministically.

Fix $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w}) \in$ Good and an arbitrary $\tilde{\mathbf{w}}$. Let $\tilde{I}_{\tilde{x}\tilde{w}}$ denote the subset of indices $i \in [N]$ such that $(\tilde{x}_i, \tilde{w}_i) = (\tilde{x}, \tilde{w})$, and $I_{y,z}$ denote the set of $i$ such that $(y_i, z_i) = (y, z)$. We also write $\tilde{I}_{\tilde{x}}$ to denote the set of all indices $i$ with $\tilde{x}_i = \tilde{x}$.

Let $\tilde{J}_{\tilde{x}} = \tilde{I}_{\tilde{x}} \setminus \cup_{w\in W}(\tilde{I}_{\tilde{x},w} \cap I_{\tilde{x},w})$. That is, $\tilde{J}_{\tilde{x}}$ is the set of indices $i$ such that $\tilde{x}_i = \tilde{x}$ and there is some "deviation": either $x_i \neq \tilde{x}_i$ or $w_i \neq \tilde{w}_i$. By the separation condition of the test, we know that there is some value $\hat{x} \in X$ such that $|\tilde{J}_{\hat{x}}| \geq \frac{1}{m}N^{7/8}$. Henceforth, we restrict our attention to $\tilde{I}_{\hat{x}}$.

The probabilities in the expressions below are conditioned on $(\tilde{\mathbf{x}}, \mathbf{x}, \mathbf{w})$, where the random choices made are of $\mathbf{y}$ and $(\mathbf{w}, \mathbf{z})$. (We do not assume any distribution over $\tilde{\mathbf{x}}$ and $\mathbf{x}$ which are chosen by the adversary.)

[8] The distributions $\mathfrak{p}^X$ and $\mathfrak{p}^Y$ are constant while $N$ is a growing parameter.

For any $y \in Y$, we have:

$$\begin{aligned}\mathrm{E}\left[\mu_{\tilde{w},\hat{x},y,z}\right] &= \mathrm{E}\left[|\tilde{I}_{\hat{x},\tilde{w}} \cap I_{y,z}|\right] = \sum_{\substack{x\in X, w\in W \\ \mathfrak{p}^{f,Y}[w|x]>0}} \mathrm{E}\left[|\tilde{I}_{\hat{x}\tilde{w}} \cap I_{w,x,y,z}|\right] \\ &= \sum_{x,w} |\tilde{I}_{\hat{x}\tilde{w}} \cap I_{xw}| \cdot \mathfrak{p}^{f,Y}[y,z|x,w] = \sum_{x,w} |\tilde{I}_{\hat{x}\tilde{w}} \cap I_{xw}| \cdot \frac{\mathfrak{p}^{f,Y}[w,y,z|x]}{\mathfrak{p}^{f,Y}[w|x]}\end{aligned}$$

Here, $\mathfrak{p}^{f,Y}[w,y,z|x] \triangleq \mathfrak{p}^{Y}[y]\mathfrak{p}^{f}[w,z|x,y]$ (since we pick $y$ independent of $x$, with probability $\mathfrak{p}^{Y}[y|x] = \mathfrak{p}^{Y}[y]$) and $\mathfrak{p}^{f,Y}[w|x] = \sum_{y,z} \mathfrak{p}^{f,Y}[w,y,z|x]$. Also, we define $\beta^{x}_{w\tilde{w}}$ to be the fraction among the indices $i$ (within $\tilde{I}_{\hat{x}}$) in which the adversary sent $x_i = x$ to $f$ and obtained $w_i = w$, for which it reported $\tilde{w}_i = \tilde{w}$.[9]

$$\beta^{x}_{w\tilde{w}} = \begin{cases} \frac{|\tilde{I}_{\hat{x}\tilde{w}} \cap I_{xw}|}{|\tilde{I}_{\hat{x}} \cap I_{xw}|} & \text{if } |\tilde{I}_{\hat{x}} \cap I_{xw}| \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

$$|\tilde{I}_{\hat{x}\tilde{w}} \cap I_{xw}| = |\tilde{I}_{\hat{x}} \cap I_{xw}| \cdot \beta^{x}_{w\tilde{w}} \qquad \text{by definition of } \beta^{x}_{w\tilde{w}} \tag{2}$$

$$= \left(|\tilde{I}_{\hat{x}} \cap I_{x}| \cdot \mathfrak{p}^{f,Y}[w|x] \pm N^{2/3}\right) \cdot \beta^{x}_{w\tilde{w}} \qquad \text{since } (\tilde{\mathbf{x}},\mathbf{x},\mathbf{w}) \in \mathsf{Good}. \tag{3}$$

We substitute this into the above expression for $\mathrm{E}\left[\mu_{\tilde{w},\hat{x},y,z}\right]$. Note that $\mathfrak{p}^{f,Y}[w|x] > 0$ implies that it is lower-bounded by a positive constant (depending on $f$, independent of $N$), and so $\frac{N^{2/3}}{\mathfrak{p}^{f,Y}[w|x]} = O(N^{2/3})$. Thus,

$$\begin{aligned}\mathrm{E}\left[\mu_{\tilde{w},\hat{x},y,z}\right] &= \sum_{x,w} |\tilde{I}_{\hat{x}\tilde{w}} \cap I_{xw}| \cdot \mathfrak{p}^{f,Y}[w,y,z|x] \cdot \beta^{x}_{w\tilde{w}} \pm O(N^{2/3}) \\ &= |\tilde{I}_{\hat{x}}| \cdot \sum_{x} \alpha^{x} \left(P^{x} \cdot B^{x}\right)_{(y,z),\tilde{w}} \pm O(N^{2/3})\end{aligned}$$

where $\alpha^{x} = \frac{|\tilde{I}_{\hat{x}} \cap I_{x}|}{|\tilde{I}_{\hat{x}}|}$, $P^{x}$ is an $nr \times q$ matrix with $P^{x}_{(y,z),w} = \mathfrak{p}^{f,Y}[w,y,z|x]$ and $B^{x}$ is a $q \times q$ matrix with $B^{x}_{w\tilde{w}} = \beta^{x}_{w\tilde{w}}$. Note that the sum of all the entries in $P^{x}$ is 1; also, $\sum_{x} \alpha^{x} = 1$ and for each $x$, $B^{x}$ is a stochastic matrix.

Next we consider the following:

$$\begin{aligned}\mathrm{E}\left[\mu_{\hat{x},y}\right] &= \sum_{x,w} \mathfrak{p}^{f,Y}[y|x,w]|\tilde{I}_{\hat{x}} \cap I_{xw}| \\ &= \sum_{x,w} \mathfrak{p}^{f,Y}[y|x,w]\mathfrak{p}^{f,Y}[w|x]|\tilde{I}_{\hat{x}} \cap I_{x}| \pm O(N^{2/3}) && \text{since } (\tilde{\mathbf{x}},\mathbf{x},\mathbf{w}) \in \mathsf{Good} \\ &= |\tilde{I}_{\hat{x}}| \sum_{x,w} \alpha^{x} \mathfrak{p}^{f,Y}[w,y|x] \pm O(N^{2/3}) \\ &= |\tilde{I}_{\hat{x}}| \mathfrak{p}^{Y}[y] \sum_{x} \alpha^{x} \pm O(N^{2/3}) && \text{since } \mathfrak{p}^{f,Y}[y|x] = \mathfrak{p}^{Y}[y] \\ &= |\tilde{I}_{\hat{x}}| \mathfrak{p}^{Y}[y] \pm O(N^{2/3}).\end{aligned}$$

[9] Note that we omit $\hat{x}$ from the notation of $\beta^{x}_{w\tilde{w}}$ (and below, $\alpha^{x}$), since we are restricting our attention to $\tilde{I}_{\hat{x}}$.

So, $\mathfrak{p}^f[\tilde{w}, z|\hat{x}, y] \cdot \mathrm{E}\left[\mu_{\hat{x},y}\right] = |\tilde{I}_{\hat{x}}| P^{\hat{x}}_{(y,z),\tilde{w}} \pm O(N^{2/3})$ since $P^{\hat{x}}_{(y,z),\tilde{w}} = \mathfrak{p}^{f,Y}[\tilde{w}, y, z|\hat{x}] = \mathfrak{p}^f[\tilde{w}, z|\hat{x}, y]\mathfrak{p}^{f,Y}[y|\hat{x}] = \mathfrak{p}^f[\tilde{w}, z|\hat{x}, y]\mathfrak{p}^Y[y]$. Thus,

$$\mathrm{E}\left[\mu_{\tilde{w},\hat{x},y,z} - \mathfrak{p}^f[\tilde{w}, z|\hat{x}, y] \cdot \mu_{\hat{x},y}\right] = |\tilde{I}_{\hat{x}}| \left( (\sum_x \alpha^x P^x \cdot B^x) - P^{\hat{x}} \right)_{(y,z),\tilde{w}} \pm O(N^{2/3})$$

Finally, we can rewrite $|\tilde{I}_{\hat{x}}|$ in terms of $|\tilde{J}_{\hat{x}}|$ as follows:

$$\begin{aligned}
|\tilde{J}_{\hat{x}}| &= |\tilde{I}_{\hat{x}}| - \sum_w |\tilde{I}_{\hat{x}w} \cap I_{\hat{x}w}| \\
&= |\tilde{I}_{\hat{x}}| - \left( |\tilde{I}_{\hat{x}} \cap I_{\hat{x}}| \sum_w \mathfrak{p}^{f,Y}[w|\hat{x}] \cdot \beta^{\hat{x}}_{ww} \right) \pm N^{2/3} && \text{by Equation 3} \\
&= |\tilde{I}_{\hat{x}}| \left( 1 - \alpha^{\hat{x}} \cdot \sum_{w,y,z} \mathfrak{p}^{f,Y}[w, y, z|\hat{x}] \cdot \beta^{\hat{x}}_{ww} \right) \pm N^{2/3}
\end{aligned}$$

Since $|\tilde{J}_{\hat{x}}| = \Omega(N^{7/8})$ and $|\tilde{I}_{\hat{x}}| \leq N$, this implies $\left(1 - \alpha^{\hat{x}} \cdot \sum_{w,y,z} \mathfrak{p}^{f,Y}[w, y, z|\hat{x}] \cdot \beta^{\hat{x}}_{ww}\right) = \Omega(N^{-1/8})$. Then,

$$\begin{aligned}
\mathrm{E}\left[\mu_{\tilde{w},\hat{x},y,z} - \mathfrak{p}^f[\tilde{w}, z|\hat{x}, y] \cdot \mu_{\hat{x},y}\right] &= \left(|\tilde{J}_{\hat{x}}| \pm O(N^{2/3})\right) \left( \frac{\left((\sum_x \alpha^x P^x \cdot B^x) - P^{\hat{x}}\right)_{(y,z),\tilde{w}}}{1 - \alpha^{\hat{x}} \cdot \sum_{w,y,z} \mathfrak{p}^{f,Y}[w, y, z|\hat{x}] \cdot \beta^{\hat{x}}_{ww}} \right) \pm O(N^{2/3}) \\
&= |\tilde{J}_{\hat{x}}| \left( \frac{\left((\sum_x \alpha^x P^x \cdot B^x) - P^{\hat{x}}\right)_{(y,z),\tilde{w}}}{1 - \alpha^{\hat{x}} \cdot \sum_{w,y,z} \mathfrak{p}^{f,Y}[w, y, z|\hat{x}] \cdot \beta^{\hat{x}}_{ww}} \right) \pm o(N^{7/8})
\end{aligned}$$

where in the last step we used that fact that $1/\left(1 - \alpha^{\hat{x}} \cdot \sum_{w,y,z} \mathfrak{p}^{f,Y}[w, y, z|\hat{x}] \cdot \beta^{\hat{x}}_{ww}\right) = O(N^{1/8})$, and $N^{2/3} \cdot N^{1/8} = o(N^{7/8})$.

Finally, by Lemma 3, since $f$ is redundancy free, $\mathfrak{D}(P^1, \ldots, P^m) \geq \epsilon_f \cdot \min(\mathfrak{p}^Y)$, where $\epsilon_f > 0$ is a constant. Since $\mathfrak{p}^Y$ has full support (and is independent of $N$), $\min(\mathfrak{p}^Y) > 0$ is also a constant. Thus,

$$\begin{aligned}
\max_{(\tilde{w},y,z)} \left| \mathrm{E}\left[\mu_{\tilde{w},\hat{x},y,z} - \mathfrak{p}^f[\tilde{w}, z|\hat{x}, y] \cdot \mu_{\hat{x},y}\right] \right| &\geq |\tilde{J}_{\hat{x}}| \left( \frac{\|(\sum_x \alpha^x P^x \cdot B^x) - P^{\hat{x}}\|_{\max}}{1 - \alpha^{\hat{x}} \cdot \sum_{w,y,z} \mathfrak{p}^{f,Y}[w, y, z|\hat{x}] \cdot \beta^{\hat{x}}_{ww}} \right) \pm o(N^{7/8}) \\
&\geq \frac{|\tilde{J}_{\hat{x}}|}{q} \left( \frac{\|(\sum_x \alpha^x P^x \cdot B^x) - P^{\hat{x}}\|_{\infty}}{1 - \alpha^{\hat{x}} \cdot \sum_{w,y,z} \mathfrak{p}^{f,Y}[w, y, z|\hat{x}] \cdot \beta^{\hat{x}}_{ww}} \right) \pm o(N^{7/8}) \\
&\geq \frac{|\tilde{J}_{\hat{x}}|}{q} \mathfrak{D}(P^1, \ldots, P^m) \pm o(N^{7/8}) = \Omega(N^{7/8}).
\end{aligned}$$

To complete the proof we use Chernoff bounds to argue that with all but negligible probability, for $(\tilde{w}, y, z)$ which maximizes the above expectation, $|\mu_{\tilde{w},\hat{x},y,z} - \mathfrak{p}^f[\tilde{w}, z|\hat{x}, y] \cdot \mu_{\hat{x},y}| > N^{2/3}$ (when $N$ is sufficiently large). □

# C Converse of The Channel Coding Theorem: Proof

**Lemma 5** (Restated.) [Weak Converse of Channel Coding Theorem, Generalization] *Let $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_K\}$ be a set of $K$ channels which take as input alphabets from a set $\Lambda$, with $|\Lambda| = 2^\lambda$. Let $\mathcal{G} \subseteq [K]$ be such that for all $i \in \mathcal{G}$, the capacity of the channel $\mathcal{F}_i$ is at most $\lambda - c$, for a constant $c > 0$.*

*Let $\mathcal{C} \subseteq \Lambda^N$ be a rate $R \in [0, 1]$ code. Consider the following experiment: a random codeword $c_1 \ldots c_N \equiv \mathbf{c} \xleftarrow{\$} \mathcal{C}$ is drawn and each symbol $c_1 \ldots c_N$ is transmitted sequentially; the channel used for transmitting each symbol is chosen (possibly adaptively) from the set $\mathcal{F}$ by the receiver.*

*Let $S$ denote the set of indices $j \in [N]$ for which the receiver chose a channel in $\mathcal{G}$ for receiving $c_j$. If the receiver always chooses the channels such that $|S|/N \geq \mu$, then the probability of error of the receiver in predicting $\mathbf{c}$ is*

$$P_e \geq 1 - \frac{1}{NR\lambda} - \frac{1 - c\mu/\lambda}{R}.$$

*Proof.* Let the codeword $\mathbf{c}$ be chosen uniformly from the code; and $\mathbf{d} = (d_1, \ldots, d_N)$ represent the symbols received by the receiver, and $\mathbf{y} = (y_1, \ldots, y_N)$ represent the sequence of channels chosen adaptively by the receiver. Note that each $y_j \in [K]$ and it can depend on $(d_1, \ldots, d_{j-1})$ and $(y_1, \ldots, y_{j-1})$ First note that:

$$\begin{aligned} NR\lambda = H(\mathbf{c}) &= H(\mathbf{c}|\mathbf{y}, \mathbf{d}) + I(\mathbf{c}; \mathbf{y}, \mathbf{d}) \\ &\leq 1 + P_e NR\lambda + I(\mathbf{c}; \mathbf{y}, \mathbf{d}) \qquad \text{By Fano's Inequality} \end{aligned}$$

Now, we shall upper bound the mutual information $I(\mathbf{c}; \mathbf{y}, \mathbf{d})$. We use $\mathbf{c}^{(j)}$ to denote $(c_1, \ldots, c_j)$; and similarly define $\mathbf{y}^{(j)}$ and $\mathbf{d}^{(j)}$. We can write:

$$\begin{aligned} I(\mathbf{c}; \mathbf{y}, \mathbf{d}) &= \sum_{j \in [N]} I(\mathbf{c}; y_j, d_j | \mathbf{y}^{(j-1)}, \mathbf{d}^{(j-1)}) \\ &= \sum_{j \in [N]} I(\mathbf{c}; y_j | \mathbf{y}^{(j-1)}, \mathbf{d}^{(j-1)}) + I(\mathbf{c}; d_j | \mathbf{y}^{(j)}, \mathbf{d}^{(j-1)}) \end{aligned}$$

Note that $\mathbf{c} \to (\mathbf{y}^{(j-1)}, \mathbf{d}^{(j-1)}) \to y_j$, so we have $I(\mathbf{c}; y_j | \mathbf{y}^{(j-1)}, \mathbf{d}^{(j-1)}) = 0$. Therefore, we get:

$$\begin{aligned} I(\mathbf{c}; \mathbf{y}, \mathbf{d}) &= \sum_{j \in [N]} I(\mathbf{c}; d_j | \mathbf{y}^{(j)}, \mathbf{d}^{(j-1)}) \\ &= \sum_{j \in [N]} H(d_j | \mathbf{y}^{(j)}, \mathbf{d}^{(j-1)}) - H(d_j | \mathbf{y}^{(j)}, \mathbf{d}^{(j-1)}, \mathbf{c}) \\ &\leq \sum_{j \in [N]} H(d_j | y_j) - H(d_j | \mathbf{y}^{(j)}, \mathbf{d}^{(j-1)}, \mathbf{c}) \\ &= \sum_{j \in [N]} H(d_j | y_j) - H(d_j | y_j, c_j) \\ &= \sum_{j \in [N]} I(d_j; c_j | y_j) \end{aligned}$$

Let $E_j$ be the indicator variable for the event that the $j$-th index $i_j \in \mathcal{G}$, i.e. the adversary chooses a channel with capacity $\leq \lambda - c$. Define $p_j$ as the probability of $E_j = 1$. We know that $\sum_{j\in[N]} p_j \geq \mu N$. Now, we know that $I(d_j; c_j|y_j) \leq p_j(\lambda - c) + (1 - p_j)\lambda = \lambda - cp_j$ Therefore, $I(\mathbf{c}; \mathbf{y}, \mathbf{d}) \leq N(\lambda - c\mu)$.

Combining this with the previous result, we get:

$$NR\lambda \leq 1 + P_e NR\lambda + N(1 - c\mu/\lambda)\lambda$$
$$\Rightarrow P_e \geq 1 - \frac{1}{NR\lambda} - \frac{1 - c\mu/\lambda}{R}$$

This completes the proof of the lemma. □

# D A UC Secure Commitment Protocol

In this section we present the details of the UC-secure commitment protocol that was outlined in Section 4.1. The protocol is in the $f$-hybrid model, for any 2-party randomized function $f$ that is redundancy free (Definition 1) and is not simple (see Section 1.2).

Before presenting the protocol, we define some terminology associated with a function $f$. We define the $nr \times mq$ matrix $\mathfrak{P}^f$, with rows indexed by $(y, z) \in Y \times Z$ and columns indexed by $(x, w) \in X \times W$, such that $\mathfrak{P}^f_{(y,z),(x,w)} = \mathfrak{p}^f[w, z|x, y]$.

**The Maps $\phi_A$ and $\phi_B$.** For each $(x, w) \in X \times W$ let the vector $\mathbf{d}^B_{(x,w)} \in \mathbb{R}^{nr}$ be the column indexed by $(x, w)$ in the matrix $\mathfrak{P}^f$. Let $\phi_A : [m] \times [q] \to [\ell]$ (for a sufficiently large $\ell \leq mq$) be such that $\phi_A(x, w) = \phi_A(x', w')$ iff $\mathbf{d}^B_{(x,w)} = c\mathbf{d}^B_{(x',w')}$ for some positive scalar $c$.

Similarly define $\mathbf{d}^A_{y,z} \in \mathbb{R}^{mq}$ to be the row of $\mathfrak{P}^f$ indexed $(y, z)$; then, let $\phi_B : [n] \times [r] \to [\ell]$ be such that $\phi_B(y, z) = \phi_B(y', z')$ iff $\mathbf{d}^A_{y,z} = c\mathbf{d}^A_{y',z'}$ for $c > 0$.

**Extreme Views.** We say that $(x, w) \in X \times W$ is an *extreme view* if the point $\mathbf{d}^B_{(x,w)}$ is not in the linear span of $\{\mathbf{d}^B_{(x',w')} \mid x' \in X, w' \in W, \phi_A(x', w') \neq \phi_A(x, w)\}$. We denote the set of extreme views in $X \times W$ by $\mathfrak{a}^f$. Note that for any $(x, w) \in X \times W$, we can write

$$\mathbf{d}^B_{(x,w)} = \sum_{(\hat{x},\hat{w})\in\mathfrak{a}^f} \gamma^{(x,w)}_{(\hat{x},\hat{w})} \mathbf{d}^B_{(\hat{x},\hat{w})}$$

If we consider an experiment in which $x \in X$ is picked uniformly at random,[10] we can write that for each $(y, z) \in Y \times Z$,

$$p(xwz|y) = \sum_{(\hat{x},\hat{w})\in\mathfrak{a}^f} \gamma^{(x,w)}_{(\hat{x},\hat{w})} p(\hat{x}\hat{w}z|y). \tag{4}$$

[10] We could use any distribution over $X$ with full support, by appropriately scaling the quantities $\gamma^{(x,w)}_{(\hat{x},\hat{w})}$. In the following, it will suffice to consider the uniform distribution.

Similarly, we say that $(y,z) \in y \times z$ is an *extreme view* if the point $\mathbf{d}^A_{y,z}$ is not in the linear span of $\{\mathbf{d}^A_{y',z'} \mid \phi_B(y',z') \neq \phi_B(y,z)\}$. We denote the extreme views in $Y \times Z$ by $\mathfrak{b}^f$.

We shall restrict to extreme views for the sender, for use in the commitment protocol. The motivation for this is that the extreme views cannot be equivocated.

Also, for each $(y,z)$ we define the set $\mathfrak{a}^f|_{(y,z)}$ to be the set of extreme views $(x,w)$ such that the row in $\mathfrak{P}^f$ indexed by $(y,z)$ has a positive entry in the column indexed by $(x,w)$. $\mathfrak{b}^f|_{(x,w)}$ is defined analogously. That is,

$$\mathfrak{a}^f|_{(y,z)} = \{(x,w) \in \mathfrak{a}^f \mid \mathfrak{p}^f[w,z|x,y] > 0\}$$
$$\mathfrak{b}^f|_{(x,w)} = \{(y,z) \in \mathfrak{b}^f \mid \mathfrak{p}^f[w,z|x,y] > 0\}.$$

**Confusable Views.** We say that $\mathfrak{a}^f$ is confusable if there exists $(y,z) \in Y \times Z$ and two elements $(x,w),(x',w') \in \mathfrak{a}^f|_{(y,z)}$ such that $\phi_A(x,w) \neq \phi_A(x',w')$.

Similarly, we say that $\mathfrak{b}^f$ is confusable if there exists $(x,w) \in X \times W$ and two elements $(y,z),(y',z') \in \mathfrak{b}^f|_{(x,w)}$ such that $\phi_B(y,z) \neq \phi_B(y',z')$.

**Lemma 11.** *If $\mathfrak{a}^f$ is not confusable and $\mathfrak{b}^f$ is not confusable, then $f$ is simple.*

*Proof.* For convenience, below we shall say that a row and a column of $\mathfrak{P}^f$ *intersect* each other if the common entry in the row and the column is non-zero.

We shall show that if $\mathfrak{a}^f$ is not confusable, then every non-zero row of $\mathfrak{P}^f$ is in $\mathfrak{b}^f$. Then, since $\mathfrak{b}^f$ is not confusable, two non-zero rows of $\mathfrak{P}^f$ intersect the same column only if they are parallel to each other. Thus if we partition the non-zero rows of $\mathfrak{P}^f$ into equivalence classes of parallel rows, then the sets of columns which intersect the rows in each equivalence class form a partition of the columns of $\mathfrak{P}^f$. Thus each equivalence class of parallel rows defines a minor (consisting of those rows and the columns they intersect), such that (a) every non-zero entry of $\mathfrak{P}^f$ falls into one such minor, (b) in each minor the rows are parallel to each other (i.e., rank 1) and (c) no two minors share the same row or column. This corresponds to a simple function.

It remains to show that if $\mathfrak{a}^f$ is not confusable, then every non-zero row of $\mathfrak{P}^f$ is in $\mathfrak{b}^f$.

For each $(x,w) \in \mathfrak{a}^f$, let $B_{x,w} = \{(y,z) \mid (x,w) \in \mathfrak{a}^f|_{(y,z)}\}$. That is, for $(x,w) \in \mathfrak{a}^f$, $B_{x,w}$ consists of all rows of $\mathfrak{P}^f$ indexed by $(y,z)$ such that $\mathfrak{p}^f[w,z|x,y] > 0$. Note that if $\phi_A(x,w) = \phi_A(x',w')$ then $B_{x,w} = B_{x',w'}$. Also, if $\phi_A(x,w) \neq \phi_A(x',w')$ then, since $\mathfrak{a}^f$ is not confusable, $B_{x,w} \cap B_{x',w'} = \emptyset$. Another consequence of $\mathfrak{a}^f$ being not confusable, combined with the fact that any column of $\mathfrak{P}^f$ can be written as a linear combination of the columns corresponding to $\mathfrak{a}^f$, is that, if for some $(x,w) \in \mathfrak{a}^f$, we have $(y,z) \in B_{x,w}$ and $(y',z') \in B_{x,w}$, then the rows corresponding to $(y,z)$ and $(y',z')$ must be parallel: i.e., there must be a positive constant $c$ such that $\mathbf{d}^B_{(y,z)} = c\mathbf{d}^B_{(y',z')}$.

Now, at least one row in $B_{x,w}$ should be in $\mathfrak{b}^f$. This is because, if all the rows in $\mathfrak{b}^f$ have the $(x,w)$-th coordinate 0, then as every row in $\mathfrak{P}^f$ is in the linear span of the rows in $\mathfrak{b}^f$, all the rows will have the $(x,w)$-th coordinate 0, and $B_{x,w}$ will be empty. Since we argued that all rows in $B_{x,w}$ are parallel to each other, this implies that all rows in $B_{x,w}$ are in $\mathfrak{b}^f$.

Further, every $(y,z) \in Y \times Z$ belongs to some $B_{x,w}$, unless the row $\mathbf{d}^B_{(y,z)} = 0$. Hence, every non-zero row of $\mathfrak{P}^f$ is in $\mathfrak{b}^f$, as was required. □

**Extremity Revealing Input.** We say that $y \in Y$ is an *extremity revealing input* if for all $z \in Z$, and all $(x, w), (x', w') \in \mathfrak{a}^f|_{(y,z)}$, $\phi_A(x, w) = \phi_A(x', w')$.

Extremity revealing inputs in $X$ are defined symmetrically.

**Mimicing an Input.** We say that $y^* \in Y$ can be *mimiced* by a set of inputs $Y_0 \subseteq Y$ if there exists a probability distribution $\eta$ over $Y_0$ such that for all $(x, w) \in X \times W$, it holds that

$$\sum_{z \in Z} \mathfrak{p}^f[z, w|x, y^*] = \sum_{z \in Z} \sum_{y \in Y_0} \eta(y) \mathfrak{p}^f[z, w|x, y]. \tag{5}$$

Note that if $y^* \notin Y_0$ can be mimiced by $Y_0$, it does not necessarily mean that $y^*$ is redundant, because for redundancy there must exist a probabilistic mapping from $Y_0 \times Z$ to $\{y^*\} \times Z$. However the following lemma shows that if each element of $Y_0$ is extremity revealing, then $y^*$ would be redundant if it can be mimiced by $Y_0$.

**Lemma 12.** *Suppose $Y_0 \subseteq Y$ is a set of extremity revealing inputs. If $y^* \in Y \backslash Y_0$ can be mimiced by $Y_0$ then $y^*$ is a strictly redundant input.*

*Proof.* To show that $y^*$ is a strictly redundant input, we consider two experiments: in the first experiment Bob chooses an input $\hat{y} \in Y_0$ with probability $\widehat{p}[\hat{y}]$ and then on obtaining an output $\hat{z}$ from $f$ (Alice picks an input uniformly at random), maps it to an output $z$ with probability $\widehat{p}[z|\hat{z}\hat{y}]$ and reports $z$ to Alice. In the other experiment, Bob picks his input to be $y^*$ and obtains an output $z$ from $f$ which he reports to Alice. (We denote the probabilities in the first experiment using $\widehat{p}$ and in the second experiment using $p$.) To show that $y^*$ is strictly redundant, we show that the views of Alice in the two experiments (given by the probabilities $\widehat{p}[xwz]$ and $p[xwz|y^*]$) are identical.

$$\begin{aligned}
\widehat{p}[xwz] &= \sum_{\hat{y}} \widehat{p}[wx\hat{y}z] = \sum_{\hat{y}} \widehat{p}[\hat{y}] \cdot \widehat{p}[xwz|\hat{y}] \\
&= \sum_{\hat{y},\hat{z}} \widehat{p}[\hat{y}] \cdot \widehat{p}[xwz\hat{z}|\hat{y}] \\
&= \sum_{\hat{y},\hat{z}} \widehat{p}[\hat{y}] \cdot \widehat{p}[xw\hat{z}|\hat{y}] \cdot \widehat{p}[z|\hat{z}\hat{y}] && \text{because } \widehat{p}[z|xw\hat{z}\hat{y}] = \widehat{p}[z|\hat{z}\hat{y}] \\
&= \sum_{\hat{y},\hat{z}} \widehat{p}[\hat{y}] \cdot p[xw\hat{z}|\hat{y}] \cdot \widehat{p}[z|\hat{z}\hat{y}] && \text{because } \widehat{p}[xw\hat{z}|\hat{y}] = p[xw\hat{z}|\hat{y}]
\end{aligned}$$

$$
\begin{aligned}
p[xwz|y^*] &= \sum_{(\hat{x},\hat{w})\in\mathfrak{a}^f} \gamma^{(x,w)}_{(\hat{x},\hat{w})} p[\hat{x}\hat{w}z|y^*] && \text{by Equation 4}\\
&= \sum_{(\hat{x},\hat{w})\in\mathfrak{a}^f} \gamma^{(x,w)}_{(\hat{x},\hat{w})} \cdot p[z|\hat{x}\hat{w}y^*] \cdot p[\hat{x}\hat{w}|y^*]\\
&= \sum_{(\hat{x},\hat{w})\in\mathfrak{a}^f} \gamma^{(x,w)}_{(\hat{x},\hat{w})} \cdot p[z|\hat{x}\hat{w}y^*] \cdot \sum_{z\in Z} p[\hat{x}\hat{w}z|y^*]\\
&= \sum_{(\hat{x},\hat{w})\in\mathfrak{a}^f} \gamma^{(x,w)}_{(\hat{x},\hat{w})} \cdot p[z|\hat{x}\hat{w}y^*] \sum_{\substack{(\hat{y},\hat{z}):\\ \hat{y}\in Y_0}} \eta(\hat{y}) p[\hat{x}\hat{w}\hat{z}|\hat{y}] && \text{by Equation 5}\\
&= \sum_{\substack{(\hat{y},\hat{z}):\\ \hat{y}\in Y_0}} \sum_{(\hat{x},\hat{w})\in\mathfrak{a}^f} \gamma^{(x,w)}_{(\hat{x},\hat{w})} \cdot \eta(\hat{y}) \cdot p[z|\hat{x}\hat{w}y^*] \cdot p[\hat{x}\hat{w}\hat{z}|\hat{y}]
\end{aligned}
$$

Note that for $\hat{y} \in Y_0$ and any $\hat{z} \in Z$, and $(\hat{x},\hat{w}) \in \mathfrak{a}^f|_{(\hat{y},\hat{z})}$, the quantity $p[z|\hat{x}\hat{w}y^*]$ depends only on $(\hat{y},\hat{z})$; this is because $\hat{y} \in Y_0$ is an extremity revealing input, and $p[z|\hat{x}\hat{w}y^*]$ is identical for all $(\hat{x},\hat{w}) \in \mathfrak{a}^f|_{(\hat{y},\hat{z})}$. So, for $\hat{y} \in Y_0$, $\hat{z} \in Z$, and $(\hat{x},\hat{w}) \in \mathfrak{a}^f|_{(\hat{y},\hat{z})}$, we define $p^*_{\hat{y},\hat{z}} = p[z|\hat{x}\hat{w}y^*]$ as a function of $(\hat{y},\hat{z})$ alone. Now,

$$
\begin{aligned}
p[xwz|y^*] &= \sum_{\substack{(\hat{y},\hat{z}):\\ \hat{y}\in Y_0}} \eta(\hat{y}) \cdot p^*_{\hat{y},\hat{z}} \cdot \sum_{(\hat{x},\hat{w})\in\mathfrak{a}^f} \gamma^{(x,w)}_{(\hat{x},\hat{w})} p[\hat{x}\hat{w}\hat{z}|\hat{y}]\\
&= \sum_{\substack{(\hat{y},\hat{z}):\\ \hat{y}\in Y_0}} \eta(\hat{y}) \cdot p^*_{\hat{y},\hat{z}} \cdot p[xw\hat{z}|\hat{y}] && \text{by Equation 4}
\end{aligned}
$$

These two expressions can be made equal by setting $\widehat{p}[\hat{y}] = \eta(\hat{y})$ for $\hat{y} \in Y_0$ (and 0 outside $Y_0$), and $\widehat{p}[z|\hat{z}\hat{y}] = p^*_{\hat{y},\hat{z}}$. □

**Unrevealing Distribution.** An *unrevealing distribution* over $Y$ is a distribution with its support being all of $Y$, such that the resulting distribution over $X \times W$ is outside the convex hull of the distributions resulting from extremity revealing inputs in $Y$. That is, if $Y_0$ is the set of extremity revealing inputs in $Y$, then $\mathfrak{p}^Y$ is an unrevealing distribution if the support of $\mathfrak{p}^Y$ is $Y$ and there is a constant $\epsilon$ such that for all distributions $\eta$ over $Y_0$, for some $(x,w)$ it holds that

$$
\Big|\sum_{z\in Z}\sum_{y\in Y} \mathfrak{p}^Y[y] \cdot \mathfrak{p}^f[z,w|x,y] - \sum_{z\in Z}\sum_{y\in Y_0} \eta(y) \cdot \mathfrak{p}^f[z,w|x,y]\Big| > \epsilon. \tag{6}
$$

An unrevealing distribution is defined as one that can be used by the receiver in the commitment protocol. This is why we have included the requirement that it has all of $Y$ in its support: this ensures that the columns of $\mathfrak{P}^f$ corresponding to $\mathfrak{a}^f$ still remain extreme views, even if the rows of $\mathfrak{P}^f$ corresponding to $y \in Y$ are scaled by $\mathfrak{p}^Y[y]$.

**Lemma 13.** *Suppose $f$ is redundancy free. If $\mathfrak{a}^f$ is confusable, then there exists an unrevealing distribution over $Y$.*

*Proof.* By definition of confusability, $Y_0 \subsetneq Y$, where $Y_0$ is the set of extremity revealing inputs. So there exists an input $y^* \in Y \setminus Y_0$. Then, since $f$ is redundancy free, by Lemma 12, $y^*$ is not mimicable by $Y_0$. Hence there must be an unrevealing distribution over $Y$ (which puts sufficient amount of probability mass on $y^*$, and puts, say uniform weight on all other elements in $Y$). □

**The Commitment Protocol.** Suppose $f$ is not simple. Then by Lemma 11, either $\mathfrak{a}^f$ is confusable or $\mathfrak{b}^f$ is confusable. W.l.o.g we assume that $\mathfrak{a}^f$ is confusable. Further, suppose $f$ is redundancy free; then by Lemma 13 there is an unrevealing distribution $\mathfrak{p}^Y$ over $Y$. In Figure 5 we present our commitment protocol assuming $\mathfrak{a}^f$ is confusable and that there is an unrevealing distribution over $Y$. (Instead if $\mathfrak{b}^f$ is confusable, then the function $f$ will be used in the reverse direction.)

We sketch the proof of security for this commitment protocol (with say $N_0 = M = \kappa$). Since we are in the information-theoretic setting, with computationally unbounded adversaries and simulators, we focus on showing the statistical hiding property and statistical binding property separately. These can be easily turned into a simulation argument.[11]

**Binding.** Binding relies on the fact that the protocol requires the sender to use extreme views and on the distance of the code used. Consider an opening made by the sender. For any block $\mathbf{c}_i$, for each $\varphi \in \Lambda$, consider the positions where the sender claimed its view to be $(x, w) \in \mathfrak{a}^f$ such that $\phi_A(x, w) = \varphi$. Consider the fraction of positions where the actual view of the sender was $(x', w')$ such that $\phi_A(x', w') \neq \varphi$. In this case, the expected view of Bob in those positions is given by a linear combination of the columns $\mathbf{d}^A_{x',w'}$ (with co-ordinates for each $y$ scaled by $\mathfrak{p}^Y[y]$). If this linear combination is not close to the vector $\mathbf{d}^A_{x,w}$ (scaled by $\mathfrak{p}^Y$) then with all but negligible probability, the opening will not be accepted by the receiver. On the other hand, if the linear combination is close to $\mathbf{d}^A_{x,w}$, since $\mathbf{d}^A_{x,w}$ is outside the linear span of other $\mathbf{d}^A_{x',w'}$ with $\phi_A x', w' \neq \phi_A x, w$, only at a small number (sub-linear fraction) of places can the sender open to $(x, w)$ but have had an actual view $(x', w')$ such that $\phi_A(x', w') \neq \phi_A(x, w)$. By using a code such that the distance of the code (while still sublinear) is much larger than the number of positions where the sender can cheat as above, we guarantee binding.

Note that the simulator, which sees all $(x_{i,j}, w_{i,j})$ and $r_{i,j}$, can find $c_{i,j} = \phi_A(x_{i,j}, w_{i,j}) \oplus r_{i,j}$ for all $(i, j)$. Then, for each $i$ it can find the closest codeword $\mathbf{c}_i$ to $(c_{i,1}, \ldots, c_{i,N})$ and use these $\mathbf{c}_i$ to extract a bit $b$ which it sends to the commitment functionality. If for any of the blocks transmitted, if the closest codeword is not unique, then the simulator commits to an arbitrary bit. In this case, the distance between the transmitted word and the codeword is large (linear in $N$), and so in a real execution, any decommitment will be rejected with all but negligible probability and thus the simulation can be finished without opening the bit the simulator committed to.

**Hiding.** We outline the proof of statistical hiding below. The detailed argument is in Appendix D.1.

To see the hiding property, consider the use of the function $f$ as a "channel," which accepts $c_{i,j}$ from Alice, $y_{i,j}$ from Bob and samples $(x_{i,j}, w_{i,j}, z_{i,j})$ and outputs $w_{i,j}$ to Alice and $a_{i,j} + c_{i,j}$ to Bob, where $a_{i,j} = \phi_A(x_{i,j}, w_{i,j})$. The hiding property relies on the fact that Bob is forced to use $f$ as channel with

[11] If the code used admits efficient equivocation (sampling a codeword within a small radius) and efficient extraction (decoding within a smaller radius), we obtain a slightly stronger security property in which the simulator is efficient except for the black-box invocations of the adversary. Since, in our final protocol, we will be invoking this construction with the security parameter set to a constant, this will not be relevant to us.

**Bit-Commitment**$(b, f, M, N_0)$:
Suppose $f$ is such that $\mathfrak{a}^f$ is confusable and there is an unrevealing distribution $\mathfrak{p}^Y$ over $Y$.
Let $\alpha_{x,w} = \sum_{y \in Y} \mathfrak{p}^Y[y]\mathfrak{p}^f[x,w|y]$ (this is the probability of $(x,w)$ being the Alice's view in an invocation of $f$ when $x$ is chosen uniformly at random and $y$ is chosen according to $\mathfrak{p}^Y$). Let $N = \frac{1}{2} \cdot N_0 \cdot \sum_{(x,w) \in \mathfrak{a}^f} \alpha_{x,w}$.
The protocol is presented in terms of a code $\mathcal{C}$ over the alphabet $\Lambda = \phi_A(\mathfrak{a}^f) \subseteq [\ell]$ (i.e., the image of $\mathfrak{a}^f$ under the map $\phi_A$) with block-length $N$, rate 1 (or more precisely, rate $1 - o(1)$ as a function of $N$), and distance $\omega(N^{7/8})$. (An explicit code is not necessary: the receiver can pick at random $\omega(N^{7/8})$ "parity checks" to construct the code and announce it to the sender.)

1. Commit Phase:

   (a) The sender picks $\{x_{i,j}\}_{i \in [M], j \in [N_0]} \xleftarrow{\$} X^{MN_0}$. The receiver picks $\{y_{i,j}\}_{i \in [M], j \in [N_0]} \in Y^{MN_0}$, where each $y_{i,j}$ is i.i.d., according to the unrevealing distribution $\mathfrak{p}^Y$. Both parties invoke $f$ with respective inputs $x_{i,j}$ and $y_{i,j}$; the function computes $(w_{i,j}, z_{i,j}) \xleftarrow{\$} f(x_{i,j}, y_{i,j})$, and sends $w_{i,j}$ to the sender and $z_{i,j}$ to the receiver.

   (b) For each $i \in [M]$, the sender carries out a consistency check on $\{(x_{i,j}, w_{i,j})\}_{j=1}^{N_0}$: it checks that for each value of $(x,w) \in X \times W$ the number of indices $j$ with $(x_{i,j}, w_{i,j}) = (x,w)$ is $N_0 \cdot \alpha_{x,w} \pm N_0^{2/3}$. If not, it aborts the protocol.

   (c) Next, for each $i \in [M]$ the sender announces a subset of $N$ (arbitrarily chosen) indices $j$, such that $(x_{i,j}, w_{i,j}) \in \mathfrak{a}^f$. (If no such set exists, then the sender would have aborted the protocol in the previous step.) In the following, we renumber the indices so that for each $i$, these $N$ indices are considered $j = 1, \ldots, j = N$.

   (d) For each $i \in [M]$, the sender picks a codeword $\mathbf{c}_i = (c_{i,1}, \ldots, c_{i,N}) \xleftarrow{\$} \mathcal{C} \subseteq \Lambda^N$. Let $r_{i,j} = c_{i,j} + \phi_A(x_{i,j}, w_{i,j})$. The sender sends $\{r_{i,j}\}_{i \in [M], j \in [N]}$ to the receiver.

   (e) The sender picks $h \leftarrow \mathcal{H}$, a universal hash function family mapping $\Lambda^{MN}$ to $\{0,1\}$ and sends $(h, \zeta)$ to the receiver, where $\zeta = b \oplus h(c_{1,1} \ldots c_{MN})$.

2. Reveal Phase: Sender sends $\{(x_{i,j}, w_{i,j})\}_{i \in [M], j \in [N_0]}$ to the receiver, who proceeds as follows:

   Recover $\mathbf{c}_i = (c_{i,1}, \ldots, c_{i,N})$, for each $i$ where as $c_{i,j} = r_{i,j} - \phi_A(x_{i,j}, w_{i,j})$. Then accept $b = \zeta \oplus h(c_{1,1} \ldots c_{MN})$ as the opening if and only if the following assertions hold:

   (i) $(x_{i,j}, y_{i,j}) \in \mathfrak{a}^f$ when $j \leq N$; (ii) $\mathbf{c}_i \in \mathcal{C}$ for each $i \in [M]$; (iii) $(x_{1,1}, w_{1,1}), \ldots (x_{M,N_0}, w_{M,N_0})$ and $(y_{1,1}, z_{1,1}), \ldots (y_{M,N_0}, z_{M,N_0})$ satisfy the consistency checks in the Left-Statistical-Test.[a]

[a] It would be enough for the sender to reveal $(x_{i,j}, w_{i,j})$ only for $j \in \{1, \ldots, N\}$ and for the receiver to do consistency tests restricted to these indices; but for convenience we describe the tests in terms of the Left-Statistical-Test.

**Figure 5** Commitment protocol in $f$-hybrid model, assuming $\mathfrak{a}^f$ is not confusable and there is an unrevealing distribution $\mathfrak{p}^Y$ over $Y$. If $f$ is redundancy free and not simple, then this assumption holds, possibly with Alice's and Bob's roles (i.e., $(X, W)$ and $(Y, Z)$) reversed.

capacity strictly less than $\log nq$: as we shall see below, this is enforced by the sender's check in step (b). Then we appeal to our extension of the weak converse of Shannon's Channel Coding Theorem (Lemma 5) to

argue that since the code has rate 1, some information about the codeword remains hidden from the receiver. We need an extension of the (weak) converse of the channel coding theorem to handle that facts that (a) the receiver can adaptively choose the channel characteristic, by picking $y_{i,j}$ adaptively, and (b) some of the channel characteristics that can be chosen include a noiseless channel, but the number of times such a characteristic can be used cannot be large (except with negligible probability). The reason this restriction can be enforced is because the $\mathfrak{p}^Y$is an unrevealing distribution. The check carried out by the sender is simple and cannot completelt *bind* the receiver to using $\mathfrak{p}^Y$, but it ensures that the receiver cannot (almost) always use only extremity revealing $y$s.

Finally, the commitment is made hiding by masking the bit to be committed by a bit extracted from the codewords $\mathbf{c}_i$.

## D.1 Hiding of the Commitment Protocol

In this section, we show that the commitment protocol in Figure 5 is statistically hiding. For this we use the converse of the channel coding theorem and the irredundancy of the function $f$.

**First Game: Error in predicting each codeword.** Note that if $\lambda = \Theta(1)$, $N = \omega(1)$, $R = 1 - o(1)$, $c = \Theta(1)$ and $\mu = \Theta(1)$ in Lemma 5, then $P_e = \Theta(1)$. As a direct application of this result, we get the following result:

Let $\Lambda = \{\phi_A(x, w) \mid (x, w) \in \mathfrak{a}^f\}$. Define $\lambda = \log |\Lambda|$; and note that $|\Lambda|$ is at least 2. Consider the following game between an honest challenger and an adversary:

1. The challenger picks a codeword $c_1 \dots c_N \equiv \mathbf{c} \xleftarrow{\$} \mathcal{C}$, where $\mathcal{C} \subseteq \Lambda^N$, $\log |\mathcal{C}| = NR\lambda$ and $R = 1-o(1)$.
2. For each $i \in [N]$, sequentially repeat these steps:
   (a) The challenger picks $x_i \xleftarrow{\$} X$, for $i \in [N]$ and feeds into $f$.
   (b) The adversary feeds $y_i \in Y$.
   (c) The output $(w_i, z_i) \xleftarrow{\$} f(x_i, y_i)$ is computed; and $w_i$ is given to the challenger and $z_i$ is given to the adversary.
   (d) If $(x_i, w_i) \notin \mathfrak{a}^f$ then repeat this step.
   (e) The challenger sends $r_i = \phi_A(x_i, w_i) + c_i$ to the adversary.
3. The adversary outputs $\tilde{\mathbf{c}} \in \Lambda^N$.

The adversary wins the game if $\tilde{\mathbf{c}} = \mathbf{c}$, i.e. it is able to correctly guess the codeword. We shall show that the probability that the adversary loses this game is at least a constant, if the adversary feeds $y_i \in Y$ which are not completely revealing for at least $\mu N$ rounds, where $\mu$ is a constant.

To directly apply Lemma 5 consider the following alternate, but equivalent, game:

1. The challenger picks a codeword $c_1 \dots c_N \equiv \mathbf{c} \xleftarrow{\$} \mathcal{C}$, where $\mathcal{C} \subseteq \Lambda^N$, $\log |\mathcal{C}| = NR\lambda$ and $R = 1-o(1)$.
2. For each $i \in [N]$, sequentially repeat these steps:

(a) The adversary feeds $y_i \in Y$ to the channel.

(b) The challenger sends $c_i \in \Lambda$ to the channel.

(c) The channel samples $(x_i, w_i, z_i)$ according to the following distribution $\mathcal{D}_{y_i}$ and sends $r_i = c_i + \phi_A(x_i, w_i)$ and $z_i$ to Bob: Consider a matrix $M_{y_i}$ with rows indexed by $Z$ and columns indexed by $X \times W$. If $(x, w) \notin \mathfrak{a}^f$ then $M_{y_i}(\ z,\ (x, w)\ ) = 0$ for all $z \in Z$. Otherwise $M_{y_i}(\ z,\ (x, w)\ ) = \mathfrak{p}^f[w, z|x, y_i]$ for every $z \in Z$. The probability of $(x, w, z)$ according to the distribution $\mathcal{D}_{y_i}$ is proportional to the entry $M_{y_i}(\ z,\ (x, w)\ )$.

3. The adversary outputs $\tilde{\mathbf{c}} \in \Lambda^N$.

The adversary wins the game if $\tilde{\mathbf{c}} = \mathbf{c}$.

Recall $Y_0 \subseteq Y$ is the set of inputs which are extremity revealing. We know that for every $y \in Y \setminus Y_0$, there exists $z \in Z$, $(x_0, w_0), (x_1, w_1) \in X \times W$ such that: $\mathfrak{p}^f[w_0, z|x_0, y] > 0$, $\mathfrak{p}^f[w_1, z|x_1, y] > 0$ but $\phi_A(x_0, w_0) \neq \phi_A(x_1, w_1)$. Further, $|Y \setminus Y_0| \geq 1$ if $f$ is redundancy free. If the challenger uses some $y \in Y \setminus Y_0$ with $\Theta(1)$ probability, then at least $\mu = \Theta(1)$ fraction of the channels are not fully revealing.

Now, this is formulated as a game where the adversary can pick the channel, at least $\mu = \Theta(1)$ fraction of whom are not fully revealing. Applying Lemma 5, we directly get that the adversary loses the game with probability $P_e = \Theta(1)$, if $N = \omega(1)$, $R = 1 - o(1)$ and $\mu = \Theta(1)$.

**Second Game: Negligible advantage in predicting the bit.** Consider the following game between an honest challenger and an adversary:

1. The challenger chooses $\mathbf{c}_1 \dots \mathbf{c}_M \xleftarrow{\$} \mathcal{C}^M$.

2. The challenger and adversary perform $M$ copies of the previous game and in the $k$-th game, the challenger uses $\mathbf{c_k}$.

3. Interpret $\mathbf{c}_1 \dots \mathbf{c}_M \equiv (u_{1,1} \dots u_{1,N}) \dots (u_{M,1} \dots u_{M,N})$, where $u_{i,j} \in \Lambda$ for all $(i, j) \in [M] \times [N]$. The challenger draws $h \xleftarrow{\$} \mathcal{H}$, where $\mathcal{H}$ is a family of universal hash functions mapping $\Lambda^{MN}$ to $\{0, 1\}$. The challenger computes $b = h(u_{1,1} \dots u_{M,N})$ and sends $h$ to the adversary.

4. The adversary output $\tilde{b} \in \{0, 1\}$.

The adversary wins the game if $b = \tilde{b}$.

The following analysis is conditioned on the fact that among the inputs $\{y_{k,1}, \dots, y_{k,N}\}$ used by the adversary, at least $\mu = \Theta(1)$ fraction of them are not fully revealing, for every $k \in [M]$. We shall be using the notion of average min-entropy (denoted by $\tilde{H}_\infty$) as introduced by [DORS08]. Let us denote the complete view of the adversary by $V$ and $\mathbf{u}$ denote the random variable $u_{1,1} \dots u_{M,N}$. Note that for each $k \in [M]$, the codeword $\mathbf{c_k}$ is incorrectly predicted by the adversary with probability at least $P_e = \Theta(1)$. Therefore, $\tilde{H}_\infty(\mathbf{u}|V) \geq \Theta(M)$. Finally, using the result that universal hash functions are good strong extractors for sources with high average min-entropy [DORS08], we get that $b$ is statistically hidden from the adversary, if $M = \omega(1)$. Formally, let $U$ be the uniform bit. Then $\mathsf{SD}\left((b, V), (U, V)\right) \leq \frac{1}{\sqrt{2}} 2^{-\tilde{H}_\infty(\mathbf{u}|V)/2} = 2^{-\Theta(M)}$.

Combining these two results, we get the following lemma:

**Lemma 14.** *Let $N = \omega(1)$, $R = 1 - o(1)$ and $\mathcal{C} \subseteq \Lambda^N$ be a rate $R$ code. If the adversary uses $\Theta(N)$ inputs among $\{y_{k,1}, \ldots, y_{k,N}\}$ which are not fully revealing, for every $k \in [M]$, then the advantage of the adversary in the following game is at most $2^{-\Theta(M)}$.*

Hiding-Game$(N, M, \mathcal{C})$*:*

1. *For $k \in [M]$ Repeat the following steps:*
   (a) *The challenge picks a codeword $c_{k,1} \ldots c_{k,N} \xleftarrow{\$} \mathcal{C} \subseteq \Lambda^N$.*
   (b) *For $i \in [N]$ repeat the following steps:*
      i. *The challenger picks $x_{k,i} \xleftarrow{\$} X$.*
      ii. *The adversary picks $y_{k,i} \in Y$.*
      iii. *They invoke $f$ with these inputs and receive respective outcomes $w_{k,i}$ and $z_{k,i}$ from the functionality.*
      iv. *If $(x_{k,i}, w_{k,i}) \notin \mathfrak{a}^f$ then repeat the above steps again.*
      v. *Otherwise, the challenge sends $r_{k,i} = c_{k,i} + \phi_A(x_{k,i}, w_{k,i})$ to the adversary.*
2. *The challenger samples $h \leftarrow \mathcal{H}$ and sends $h$ to the adversary. Define $b = h(c_{1,1} \ldots c_{M,N})$.*
3. *The adversary finally outputs $\tilde{b}$.*

*The adversary wins the game if $b = \tilde{b}$.*

**Final Argument.** Finally, for a non-redundant $f$, we need to show that any (malicious) receiver for the protocol in Figure 5 uses $y \in Y \setminus Y_0$ with constant probability if the commitment-phase of the bit commitment protocol succeeds.

Let $V(y)$ be the distribution over $\mathfrak{a}^f \subseteq X \times W$ conditioned on the event that the receiver uses input $y$ and the view of Alice lies in the set $\mathfrak{a}^f$. Define $\mathcal{P}$ as the convex hull of the points $\{V(y) \mid y \in Y_0\}$. Let $\mathcal{Q}$ be the convex hull of the points $\{V(y) \mid y \in Y\}$. Since $f$ is not redundant, Lemma 12 implies that there exists $y^* \in Y \setminus Y_0$ such that $V(y^*)$ is outside $\mathcal{P}$. Thus, there exists a point in $\mathcal{Q}$ such that its statistical distance from every point in $\mathcal{P}$ is at least a constant and has full support over $Y$.

Formally, there exists a distribution $\mathfrak{p}^Y$ over $Y$ such that the following conditions hold: Define $V(\mathfrak{p}^Y) = \sum_{y \in Y} \mathfrak{p}^Y[y] \cdot V(y)$.

1. $\mathsf{SD}\left(V(\mathfrak{p}^Y), \mathcal{P}\right) \geq \tau = \Theta(1)$, and
2. $\mathfrak{p}^Y[y] \geq \gamma = \Theta(1) > 0$, for all $y \in Y$.

First consider the case that the sender only wants to send one codeword $\mathbf{c_1} \in \mathcal{C}$ to the receiver. Fix a view of the receiver. For this view, let $\widehat{p}(y)$ represent the fraction of indices of $[N]$ where the receiver uses $y$ as input. Conditioned on this fixed view of the receiver, we shall consider the probabilistic event that the sender completes the commitment-phase of the protocol in Figure 5. There are two cases to consider:

1. Suppose the receiver uses inputs from $Y \setminus Y_0$ in at least $(\tau/2) \times N$ invocations of $f$, i.e. $\widehat{p}(Y \setminus Y_0) \geq \tau/2$. In this case we are done, because $(\tau/2) = \Theta(1)$; and feeding any input from $Y \setminus Y_0$ results in a hiding channel.

2. If the receiver uses inputs from $Y_0$ in at least $(1-\tau/2)\times N$ invocations of $f$, i.e. $\widehat{p}(Y\setminus Y_0) \leq \tau/2$, then $\mathsf{SD}\left(V(\widehat{p}), V(\mathfrak{p}^Y)\right) \geq \tau - \tau/2 = \tau/2$.[12] In this case, the sender will detect that the distribution over $\mathfrak{a}^f$ is not close to the honest distribution $V(\mathfrak{p}^Y)$ with $(1-\mathrm{negl})$ probability (by Chernoff bound). So, the sender will not complete the commitment-phase of the protocol, except with negligible probability.

Averaging over the views of the receiver, if the commitment-phase of the protocol completes with $(1-\mathrm{negl})$ probability, then $\widehat{p}(Y \setminus Y_0) \geq \tau/2$ with $(1-\mathrm{negl})$ probability.

Finally, using union bound, the receiver uses $\widehat{p}(Y \setminus Y_0) \geq \tau/2$ while receiving the $k$-th codeword $\mathbf{c_k}$, for every $k \in [M]$. This complete the argument that the bit commitment protocol in Figure 5 statistically hides the bit $b$.

# E Passive-to-Active Security Compiler: Proof

Here we prove that the protocol $\rho_{\widetilde{\mathrm{OT}}}$ in Section 4.2 UC-securely realizes $\mathcal{F}_{\widetilde{\mathrm{OT}}}^{(\delta)}$ with parameter $\delta(\kappa) = \kappa^{-1/16}$ (or more precisely, $\delta(\kappa) = \lceil \kappa^{15/16} \rceil/\kappa$, so that $\kappa \cdot \delta(\kappa)$ is an integer).

Firstly, we note that if both parties are honest, then by the completeness of the statistical test, the probability that the protocol is aborted is negligible in $\kappa$. Combined with the correctness of $\pi_{\text{SH-OT}}$, this ensures that a trivial simulation is good for this case.

When at least one party is corrupt, we need to build a simulator interacting with the ideal functionality $\mathcal{F}_{\widetilde{\mathrm{OT}}}^{(\delta)}$ playing the role of the corrupt party. It simulates to the adversary an interaction of the protocol $\rho_{\widetilde{\mathrm{OT}}}$ in the $f$-hybrid as follows. Till Phase III it plays the part of the honest party faithfully. Note that the inputs to the protocol are not used until Phase V, so this can be carried out faithfully. If the simulated honest party aborts its execution before entering Phase IV, the simulator completes the simulation. Otherwise it proceeds as follows. (Below we abbreviate $\delta(\kappa)$ as $\delta$.)

- If the simulated honest party does not abort its execution, but the adversary has deviated from the execution it has been committed to in more than $\delta \cdot \kappa$ of the executions of $\pi_{\text{SH-OT}}$ indexed by $\overline{L}$, the simulator bails out. We shall use the binding property of $f$ to argue that this happens with negligible probability.
- Else, let $C \subseteq \overline{L}$ be a set with $|C| = \delta \cdot \kappa$ such that indices of all the executions of $\pi_{\text{SH-OT}}$ in which the adversariary deviated are included in $C$.

  The simulator checks if $\mathcal{F}_{\widetilde{\mathrm{OT}}}^{(\delta)}$ yields control to it. If so, it simulates the random selection in Phase IV to pick a random index $i^* \in C$. Else, it picks a random index $i^* \notin C$.

  In the former case, it will carry out Phase V execution faithfully using the correct input of the honest player, and (if Bob is the honest player) takes its output from that execution and makes $\mathcal{F}_{\widetilde{\mathrm{OT}}}^{(\delta)}$ provide that output to the honest player.

  In the latter case (when $\mathcal{F}_{\widetilde{\mathrm{OT}}}^{(\delta)}$ does not yield control), the simulator extracts the adversary's input(s), uses an arbitrary bit for the (part of) honest player's input that it does not obtain from the functionality,

[12] The lower bound follows from the following result: Let $S$ be the space of vectors in $[-1,1]^N$ such that the components of the vectors sum up to 0. For vectors $\mathbf{a},\ \mathbf{b} \in S$, $\mathsf{SD}(\mathbf{a}, \mathbf{0}) \geq \tau$ implies $\mathsf{SD}(\mathbf{a}, \rho\mathbf{b}) \geq (\tau - \rho)$.

and completes the simulation of Phase V. That is, if Alice is the honest player, then the simulator will first extract the choice bit $b$ from the adversary's last message and the output it received in the execution of $\pi_{\text{SH-OT}}$ indexed by $i^*$ (in which the adversary did not deviate); then it will send this bit to the functionality and receive $x_b$. It will complete the simulation of Phase V by sending a random bit instead of $r_{1-b} = x_{1-b} \oplus s_{1-u}$ as Alice's last message. On the other hand, if Bob is the honest player, the simulator will extract $(x_0, x_1)$ from the adversary and send them to the functionality; it will complete the simulation using a random bit instead of $c = b \oplus u$ as Bob's last message.

We point out that the simulator does not ever employ the simulation for $\pi_{\text{SH-OT}}$, but rather runs the protocol $\pi_{\text{SH-OT}}$ itself. The security guarantee for $\pi_{\text{SH-OT}}$ is only used in arguing that the simulation is good. This allows us to not rely on *adaptive* security for $\pi_{\text{SH-OT}}$.

We argue that this is a good simulation with only a negligible statistical difference with the real execution. Note that we can couple the real and ideal executions upto the end of Phase III. To prove that the entire simulation is good, we show:

(a) probability of the event bail-out is negligible in the coupled execution, and

(b) conditioned on the event bail-out not occuring in the coupled execution, the two executions have negligible statistical difference.

The first part follows from Lemma 4. Suppose the adversary deviates in $t_0$ instances of $\pi_{\text{SH-OT}}$, and $t_1$ of those instances were indexed in $L$ during the cut-and-choose phase. With high probability $t_1$ is close to $t_0/2$. For the honest party to not abort, in all the $t_1$ instances in $L$, the adversary should pass parts (a) and (b) of the checks. Note that the only part not determined by the protocol, given the view of the honest party and the committed values, are the views of the adversary from $f$ invocations: so for a deviation to be not caught by part (a) of the check, either the deviation should be that the adversary actually fed a different value as input to an instance of $f$ than it was supposed to, or it altered the output it received from $f$ and continued the execution faithfully with this altered output (and reported the altered output). Thus there are at least $t_1$ executions of $f$ from the $t_1$ executions of $\pi_{\text{SH-OT}}$ in which the adversary deviated as above. Of these at least $t_1/2$ have $f$ invoked in the same direction (with the adversary playing the role of the first party (with input domain $X$) or of the second party: w.l.o.g., assume that the adversary plays the role of the first party in $t_1/2$ instances of $f$ in which it deviated. Let $N$ denote the total number of instances of $f$ invoked in this direction out of all the $\kappa$ instances of $\pi_{\text{SH-OT}}$ indexed by $\overline{L}$. Recall that $\pi_{\text{SH-OT}}$ is invoked with a security parameter $\kappa_{\pi_{\text{SH-OT}}} = \kappa^c$ for a small enough constant $c > 0$ so that the number of invocations of $f$ in each instance of $\pi_{\text{SH-OT}}$ is at most $\kappa^{1/8}$; then $N \leq \kappa^{9/8}$. By the binding lemma, we know that if the consistency check is cleared then $t_1 \leq N^{7/8} \leq \kappa^{54/64} < 2t(\kappa)$ with all but negligible probability (since $t(\kappa) = \kappa^{15/16}$). Thus the probability of $t_0 \geq t(\kappa)$, which is the probability of the event bail-out, is negligible.

To prove the second part we shall show that if an environment can distinguish between the two executions, then we can break the statistical (semi-honest) security of $\pi_{\text{SH-OT}}$.

Firstly, note that the random selection is perfectly simulated, conditioned on bail-out not occurring: indices in $C$ are chosen with probability exactly $\delta = |C|/\kappa$. Indeed, we can define the set $C$ in the real execution as well, and couple the executions till end of Phase IV, so that if the functionality yields control to the adversary, $i^*$ is chosen to be an index in $C$ (in both executions). Also, conditioned on the functionality yielding control to the simulator, the simulation is perfect. So consider conditioning on the event (in simulation) that the functionality does not yield control to the simulator, and an index $i^* \notin C$ is chosen, corresponding to an execution of $\pi_{\text{SH-OT}}$ in which the adversary does not deviate. To argue that the simulation is good in this case we consider the advantage of the adversary in the following experiment: a fail-stop adversary takes

part in $\kappa$ executions of $\pi_{\text{SH-OT}}$ with randomly chosen inputs (for both players). The adversary follows the protocol honestly but it can adaptively choose to abort any number of these executions, and whenever it aborts an execution, it will be given the state of the honest party in that execution. When all the executions finish, for each execution that was not aborted, define the "hidden bit" to be (the part of) the input of the honest party that is not revealed to the adversary by the ideal OT functionality (for the inputs). Then the adversary is given either the actual hidden bits in all the unaborted executions, or independently randomly chosen bits. The adversary's advantage is the difference in its probability of outputting 1 in these two cases. By a hybrid argument it is enough to consider a single execution. Then firstly, the adversary can be assumed to not abort the execution (its advantage remains the same by not aborting and instead making a random guess); that is, we can consider only semi-honest adversaries in this experiment. By the security guarantee of $\pi_{\text{SH-OT}}$, the advantage of semi-honest adversary in distinguishing the actual hidden bit from a random bit is negligible (in $\kappa_{\pi_{\text{SH-OT}}}$ and hence in $\kappa$).

# F Active-Completeness Implies Passive-Completeness

**Lemma 15.** *Let $f$ be a redundancy free 2-party function. If $f$ has a standalone (or UC) secure protocol in $g$-hybrid, then $f$ also has a passive-secure protocol in $g$-hybrid.*

*Proof.* We will show that the same protocol that is a standalone secure realization of $f$ in $g$-hybrid is also a passive-secure protocol for $f$ in $g$-hybrid.

Consider the case when Alice is corrupt. We are given that there exists a simulator for corrupt Alice in the standalone or UC setting. We need to show that, conditioned on the existence of such a simulator, we get a semi-honest simulator for $f$. In fact, we shall leverage the left-redundancy of $f$ to show this result.

For any input $x$, let $N_x$ be the event that the simulator invokes the ideal functionality on inputs other than $x$ or malicious Alice gets an output which was not the output sent by the ideal functionality to the simulator. If probability of $N_x$ is negligible, then we consider a *semi-honest* simulator which faithfully simulates the standalone/UC simulator. If the input sent to the ideal functionality is different from $x$ or the output obtained by malicious Alice is different from the output obtained from the ideal functionality then it aborts. For an external environment, interactions with these two simulator are statistically indistinguishable because the semi-honest simulation is statistically close to the original simulation. Hence, we can conclude that there exists a semi-honest simulator.

If the probability of the event $N_x$ is non-negligible for some $x \in X$, then there exists an infinite set of security parameters $\kappa$ where probability of $N_x$ (represented by $p_x(\kappa)$) is significant, i.e. $1/\text{poly}(\kappa)$, but the statistical distance between the real and simulated view of the environment is $\delta_x(\kappa) = \text{negl}(\kappa)$ close to its real view. Now consider the set $V$ of simulator views such that, on input $x$, the event $N_x$ takes place. Define the following adversarial algorithm $A$: Randomly pick a view from $V$ and follow its simulation strategy. Consider interaction of $A$ in the Left-Statistical-Test. The separation condition is trivially satisfied, because the input fed to the simulator or the output received from the simulator does not match the input or the output given to the external environment.

Note that $p_x(\kappa)$ is significant, while the probability $\delta_x(\kappa)$ is negligible. Thus, restricted to the views $V$, the statistical distance between environment views can be at most $\delta_x(\kappa)/p_x(\kappa) = \text{negl}(\kappa)$. This ensures that consistency check is also satisfied. So, we arrive at a contradiction (because for left redundancy free

functionalities, it is impossible to win the binding experiment, except with negligible probability); thus, it is not possible that there exists $x \in X$ such that $N_x$ is non-negligible.

Note that the whole argument is independent of the hybrid $g$ being used. Further, considering the simulator for Bob and leveraging that $f$ is right redundancy free, we can similarly conclude that there exists a semi-honest simulator for Bob. This concludes the proof. □

We can use this result to claim the following:

**Lemma 16.** *If a 2-party function $g$ is standalone-complete (or UC-complete) then it is also passive-complete.*

*Proof.* Suppose $g$ is standalone-complete (or UC-complete). Then there is a standalone-secure protocol for OT in $g$-hybrid. Since OT is redundancy free, by Lemma 15, this protocol is passive-secure as well. Since OT is passive-complete and passive-security admits secure composition, we conclude that $g$ is passive-complete as well. □

# G Constant Rate Reduction of $\mathcal{F}_{\text{OT}}$ to $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$

Let $f : X \times Y \mapsto W \times Z$ be a 2-party function evaluation such that one of its cores is passive complete. We represent the secure function evaluation functionality of $f$ by $\mathcal{F}_f$. For brevity, we shall use $\mathcal{F}$ instead.

In this section we show how to realize the Oblivious Transfer (OT) functionality at constant rate in the $\mathcal{F}$-hybrid, where $f$ has a core which is passive-complete. The constant rate achieved by our protocols is in amortized sense, i.e. we shall show how to securely implement $\kappa$ independent copies of (2-choose-1 bit) OTs by performing at most $\Theta(\kappa)$ calls to $\mathcal{F}$ in the $\mathcal{F}$-hybrid.

This section crucially relies on the techniques introduced in [IKO+11, IPS08].

## Part One: Getting $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ from $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$ at Constant Rate

Our starting point is the protocol for the functionality $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$ in $\mathcal{F}$-hybrid. Recall the following definition of the functionality $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$ :

**Functionality $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$.** Parametrized by a function $\delta(\kappa)$.

- Set $b = 1$ with probability $p = \delta(\kappa)$; otherwise $b = 0$.
- Provide the parties access to a (2-choose-1 bit) OT functionality. If $b = 1$, let the adversary control the functionality.

We know that there exists a function $\delta$ such that $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$ can be realized in the $\mathcal{F}$-hybrid with statistically small simulation error, say $\epsilon(\kappa)$ (Lemma 6).

Now, we shall use the "Statistical to Perfect Lemma" introduced by [IKO+11]:

**Lemma 17** (Statistical to Perfect Lemma [IKO+11])**.** *Let $f : X \times Y \mapsto W \times Z$ be a two-party function evaluation, and $\mathcal{F}_f$ the secure function evaluation functionality for $f$. The functionality $\mathcal{F}_{\widetilde{f}}^{(\delta(\kappa))}$ is the*

*functionality which implements $\mathcal{F}_f$ but yields the control to the adversary with probability $\delta(\kappa)$. Suppose $\mathcal{G}$ is a 2-party functionality and $\pi$ is a $D$-round protocol such that $\pi$ UC (resp., standalone) securely realizes $\mathcal{F}_{\widetilde{f}}^{(q(\kappa))}$ in the $\mathcal{G}$-hybrid model, with statistical security error of $\epsilon(\kappa)$. Then $\pi$ UC (resp., standalone) securely realizes $\mathcal{F}_{\widetilde{f}}^{(p(\kappa))}$ in the $\mathcal{G}$-hybrid with perfect security, where $p(\kappa) = D|X||Y| \cdot (q(\kappa) + \epsilon(\kappa))$.*

The lemma as stated is slightly different from the statement of this theorem in [IKO+11]. The statement of the theorem in [IKO+11] has $q(\kappa) = 0$. But this generalization is immediate from the observation that a protocol which securely implements $\mathcal{F}_{\widetilde{f}}^{(q(\kappa))}$ with $\epsilon(\kappa)$ simulation error is also a secure implementation of $\mathcal{F}_f$ with simulation error $q(\kappa) + \epsilon(\kappa)$.

As a direct application of this result on the protocol which securely implements $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta)}$ in the $\mathcal{F}$-hybrid with $\epsilon(\kappa) = \text{negl}(\kappa)$ simulation error, we get the following result:

**Lemma 18.** *There exists $\delta'(\kappa) = \Theta(\delta(\kappa) + \epsilon(\kappa)) = o(1)$ such that $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta')}$ has a perfectly secure protocol in the $\mathcal{F}$-hybrid.*

*In fact, for every constant $c^* > 0$, there exists a constant $c \leq c^*$ and a perfectly secure protocol $\pi_c$ for $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ in the $\mathcal{F}$-hybrid with constant communication complexity. In particular, $\pi_c$ performs only a constant number of calls to $\mathcal{F}$.*

The second part of the result follows from the following argument: Since $\delta'(\kappa) = o(1)$, pick the smallest $\kappa_c$ such that $\delta'(\kappa_c) \leq c^*$. Set $c = \delta'(\kappa_c)$ and define $\pi_c$ as the perfectly secure protocol for $\mathcal{F}_{\widetilde{\text{OT}}}^{(\delta')}$ with security parameter fixed to $\kappa = \kappa_c$.

## Part Two: Getting $\mathcal{F}_{\text{OT}}$ from $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ at Constant Rate

In this section we shall show the following result:

**Lemma 19.** *There exists a constant $c^* > 0$ such that, for every $c \leq c^*$, $\mathcal{F}_{\text{OT}}$ UC-securely reduces to $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ at constant rate.*

It is easy to see that this result along with Lemma 18 yields our main result Theorem 1.

First, we reduce $\mathcal{F}_{\text{STRING-OT}[\ell]}$ to $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$, for sufficiently small constant $c$, at constant rate. In $\mathcal{F}_{\text{STRING-OT}[\ell]}$, the sender sends two $\ell$ bit strings and the receiver, oblivious the the sender, chooses to receive one of the strings. The constant rate in this scenario refers to the fact that $\mathcal{F}_{\text{STRING-OT}[\ell]}$ can be securely realized in $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ -hybrid by performing at most $\Theta(\ell)$ calls to $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$. Such a reduction was explicitly provided by Ishai et al. [IKO+11].

**Lemma 20** (Reduction of $\mathcal{F}_{\text{STRING-OT}[\ell]}$ to $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ at constant rate [IKO+11])**.** . *There exists a constant $c_1^* > 0$ such that, for all $c < c_1^*$, there exists a UC-secure constant rate reduction of $\mathcal{F}_{\text{STRING-OT}[\ell]}$ to $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ -hybrid.*

Henceforth, we shall assume that $c^* < c_1^*$. To complete the proof, we extend the IPS compiler to $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ -hybrid.

**Extension of IPS compiler [IPS08] from $\mathcal{F}_{\text{OT}}$-hybrid to ($\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ ,$\mathcal{F}_{\text{STRING-OT}[\ell]}$)-hybrid.** We shall show the following result:

**Lemma 21** (Generalization of IPS)**.** *Suppose $\Pi$ is a protocol among $n = \Theta(\kappa)$ servers and 2 clients, for a 2-party functionality $\mathcal{F}^*$ between clients with UC-security against adaptive, active corruption of $t = \Omega(n)$ servers and adaptive, active corruption of (any number of) clients. Suppose $\rho^{\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}}$ is a 2-party protocol in the $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ -hybrid model, that semi-honest securely realizes the functionality of each server in the protocol $\Pi$, with error tolerance. Then there is a 2-party (compiled) protocol for the functionality $\mathcal{F}^*$ in the $(\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}, \mathcal{F}_{\text{STRING-OT}[\ell]})$-hybrid model, with UC-security against adaptive, active adversaries. Further, if the (insecure) protocol $\tilde{\Pi}$ obtained by directly implementing each servers of $\Pi$ using $\rho^{\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}}$ has constant rate, then the compiled protocol has constant rate too.*

**Definition of Constant Rate.** The term *constant rate* needs some explanation. The *overall complexity* of a protocol is defined as the sum of total communication complexity and total randomness complexity of the protocol. Suppose $\mathcal{F}^*$ implements $\alpha$ independent instances of a functionality. For example, say $\mathcal{F}^* \equiv \mathcal{F}_{\text{OT}^\alpha}$, i.e. $\mathcal{F}^*$ computes $\alpha$ independent instances of $\mathcal{F}_{\text{OT}}$. If the overall complexity of a protocol is $\Theta(\alpha)$, then it is said to be constant rate.

Most of the IPS compiler analysis remains identical. We only highlight the main differences in this section. There are two cases to take care of:

1. A modification of the watchlist infrastructure setup, and
2. Modifications of the consistency checks performed by parties for server communications which are on its watchlist.

**Watchlist Initialization Modification.** In original IPS compiler parties *choose* the set of $\Theta(t)$ servers to put on their watchlist. In the 2 party setting, we can allow parties to have random $\kappa$ servers on their watchlist. Suppose the overall complexity of the $j$-th server, for $j \in [n]$, is $\sigma_j$ (we assume, without loss of generality, that $\sigma_j = \Omega(\kappa)$). To establish a watchlist for this server, we need $\Theta(n/t) = \Theta(1)$ instances of $\mathcal{F}_{\text{STRING-OT}[\ell = \sigma_j]}$.

**Consistency Checks.** Suppose Alice has the $j$-th server on her watchlist which is being simulated by the $j$-th session of the inner protocol (represented by $\rho_j^{\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}}$). Then she, first, gets to see the outcome of the "coin-tossing-in-the-well" phase of $\rho_j^{\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}}$. So, she knows the exact random tape to be used by Bob in the execution of $\rho_j^{\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}}$. Next, all messages which are sent over the communication channel are checked for consistency when revealed over the watchlist. Finally, calls to OT instances are also checked for consistency. Suppose $\rho_j^{\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}}$ performs $\mu_j$ calls to $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ . If the number of inconsistencies $\geq 2c\mu_j$ then Alice declares that Bob is cheating.

*Semi-honest setting.* Note that when both parties are honest, $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ yields control to the adversary with probability $c$. Thus, it is possible that there are inconsistencies in the reported Bob views; but the number

of such inconsistencies is $< 2c\mu_j$, except with probability negligible in $\mu_j$. The simulation for an semi-honest party (say, Bob) is simple. The simulator internally simulates a $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ instance. If it yields control to the adversary, then the simulator corrupts the external $\mathcal{F}_{\text{OT}}$ instance and grants adversary control to it. The maximum number of external corruptions performed is $< 2c\mu_j$, except with probability negligible in $\mu_j$.

*One-party malicious setting.* When Bob is malicious, $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ yields control to the adversary (i.e. Bob) at $< 2c\mu_j$ instances and in these instances Bob could lie without being detected. For every other instance where malicious Bob lies, it is caught with probability $1/2$, because OT instances are always invoked with random inputs. So, if Bob lies in $\geq 6c\mu_j$ instances, then Alice catches $\geq 2c\mu_j$ inconsistencies. So, Bob could lie in $< 8c\mu_j$ instances without getting caught.

The simulation in this case proceeds as follows: The simulator honestly simulates a run of $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ internally. If $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ yields control to the adversary, then the simulator corrupts the external $\mathcal{F}_{\text{OT}}$ instance and gives the adversary control to that $\mathcal{F}_{\text{OT}}$ instance. If $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ implements a secure $\mathcal{F}_{\text{OT}}$ instance then the simulator does the following: The simulator simply forwards messages between the external $\mathcal{F}_{\text{OT}}$ instance and malicious Bob. If Bob lies in the watchlist, then the simulator corrupts the external $\mathcal{F}_{\text{OT}}$ instance and performs the consistency check with respect to the external $\mathcal{F}_{\text{OT}}$ view.

Rest of the analysis remains similar to the original IPS analysis.

*Choice of the parameter $c$.* Suppose the error-tolerant nature of $\rho^{\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}}$ ensures that if $< c_2^*\mu_j$ instances of $\mathcal{F}_{\text{OT}}$ are (semi-honest) corrupted then $\rho^{\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}}$ remains secure. If $c < c_2^*/8$, then we can ensure that corruption of $8c\mu_j$ external $\mathcal{F}_{\text{OT}}$ instances would not violate the security of $\rho_j^{\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}}$ protocol.

**Constant Rate.** Assume that the overall complexity of the protocol $\tilde{\Pi}$ is $\Theta(\alpha)$. Note that the total number of calls to $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ performed in the compiled protocol is: $\sum_{j\in[n]} \mu_j \leq \sum_{j\in[n]} \sigma_j = \Theta(\alpha)$. Recall that, if $c$ is sufficiently small, then $\mathcal{F}_{\text{STRING-OT}[\ell]}$ reduces to $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ at constant rate. So, to implement the watchlist infrastructure, we need $\Theta(\sigma_j)$ instance of $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ for the $j$-th server's watchlist. Thus, we need a total of: $\sum_{j\in[n]} \Theta(\sigma_j) = \Theta(\alpha)$ instances of $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ for watchlist infrastructure setup. This shows that the IPS compiler is constant rate if $\tilde{\Pi}$ is constant rate.

**Particular Instantiation.** For the inner protocol we use: GMW [GMW87] semi-honest secure protocol in $\mathcal{F}_{\text{OT}}$-hybrid. And the $\mathcal{F}_{\text{OT}}$ instances are in turn obtained by using the constant-rate semi-honest OT combiner of Harnik et al. [HIKN08].

For the outer protocol we use the optimized version of Dämgard-Ishai [DI06, CC06] protocol.

We set $\mathcal{F}^* \equiv \mathcal{F}_{\text{OT}^\alpha}$. It is easy to verify that for such a choice of protocols, the overall complexity of $\tilde{\Pi}$ is $\Theta(\alpha) + \text{poly}(\kappa)$. By using $\alpha$ as a sufficiently large polynomial in $\kappa$, we get a constant rate protocol for $\mathcal{F}_{\text{OT}}$ in $\mathcal{F}_{\widetilde{\text{OT}}}^{(c)}$ -hybrid.