Secure Computability of Functions in the IT setting with Dishonest Majority and Applications to Long-Term Security

Robin Künzler* and Jörn Müller-Quade† and Dominik Raub‡

November 6, 2008 16:25

Abstract

It is well known that general secure function evaluation (SFE) with information-theoretical (IT) security is infeasible in presence of a corrupted majority in the standard model. On the other hand, there are SFE protocols (Goldreich et al. [STOC'87]) that are computationally secure (without fairness) in presence of an actively corrupted majority of the participants. Now, the issue with computational assumptions is not so much that they might be unjustified at the time of protocol execution. Rather, we are usually worried about a potential violation of the privacy of sensitive data by an attacker whose power increases over time (e.g. due to new technical developments). Therefore, we ask which functions can be computed with long-term security, where we admit computational assumptions for the duration of a computation, but require IT security (privacy) once the computation is concluded.

Toward this end we combinatorially characterize the classes of functions that can be computed IT securely in the authenticated channels model in presence of passive, semi-honest, active, and quantum adversaries (our results for quantum adversaries and in part for active adversaries are limited to the 2-party setting). In particular we obtain results on the fair computability of functions in the IT setting along the lines of the work of Gordon et al. [STOC'08] for the computational setting. Our treatment is constructive in the sense that if a function is computable in a given setting, then we exhibit a protocol.

We show that the class of functions computable with long-term security in a very practical setting where the adversary may be active and insecure channels and a public-key infrastructure are provided is precisely the class of functions computable with IT security in the authenticated channels model in presence of a semi-honest adversary.

Finally, from our results and the work of Kushilevitz [SIAM Journal on Discrete Mathematics '92] and Kraschewski and Müller-Quade we can derive a complete combinatorial classification of functions, by secure computability and completeness under passive, semi-honest, active, and quantum adversaries.

Keywords: long-term security, information-theoretic security, corrupted majority, secure function evaluation.

*Department of Computer Science, ETH Zurich, Switzerland, robink@student.ethz.ch

†IAKS/EISS, Fakultät für Informatik, Universität Karlsruhe (TH), Germany, muellerq@ira.uka.de

‡Department of Computer Science, ETH Zurich, Switzerland, raubd@inf.ethz.ch

1 Introduction

In cryptography one distinguishes *computational (CO) security* which could in principle be broken by a very powerful adversary and *information theoretical (IT) security* which withstands even an unlimited attacker. However, general IT secure protocols fail in presence of an adversary that may corrupt a majority of the participants. On the other hand, an unlimited attacker is not a realistic threat and the problem with CO assumptions is not so much that these could be unjustified right now, but that concrete CO assumptions could *eventually* be broken by an attacker whose power increases over time. With such a more realistic threat model in mind an interesting question arises: **Which cryptographic tasks can be realized with long-term (LT) security?** I.e., which tasks can be realized in presence of an attacker (potentially corrupting a majority of protocol participants) who is CO limited during the protocol execution, but becomes unlimited afterwards?

In this work we study multi-party secure function evaluation (SFE). The main result is a classification of the functions which can be computed with LT security over a network of authenticated channels. Furthermore we give a classification of all the 2-party functions which can securely be computed in presence of an adversary who is unlimited from the start. This class is strictly contained in the class of functions which can be computed with LT security and the notion of LT security hence lies strictly between CO security and IT security.

Quantum cryptography can achieve tasks, like IT secure key distribution, which cannot be achieved classically. For the task of secure function evaluation it is not known if quantum cryptography can achieve anything beyond the classically possible[1]. However, in this work we show that the class of 2-party functions which can be realized with quantum cryptography is strictly contained in the class of 2-party functions realizable with LT security. From this inclusion novel impossibility results for quantum cryptography can be derived which are no direct consequences of the results by Mayers [May97] or Kitaev [ABDR04].

All results in this paper are constructive: whenever it is claimed that a class of functions is securely computable a protocol is given. We use a stand-alone simulatability based security model with a synchronous communication network (see e.g. [Gol04]). Simulatability directly implies *privacy*[2] and *correctness*[3]. The ideal functionality that formalizes SFE in the simulation model can be chosen to additionally require *robustness*[4], *fairness*[5], or *agreement on abort*[6].

1.1 Contributions

To combinatorially characterize the class of functions which are computable with LT security we first characterize the class of *passively computable functions* $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$, i.e. the class of functions which can securely be computed by parties which are connected by authenticated channels in presence of a CO unlimited passive[7] adversary who must behave according to the protocol. Next we characterize the class of *semi-honestly computable functions* $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$ which are securely computable in the same setting as above but in presence of a stronger, *semi-honest*[8] adversary, that has to stick to the protocol, but may replace his inputs or lie about his local output.

To prove a separation between the notion of LT security and IT security we characterize the class $\mathfrak{F}_{\mathsf{2act}}$ of all 2-party functions which are securely computable in presence of an unlimited active adversary. We furthermore provide a necessary condition (which we conjecture to be also sufficient) for membership in the class of *actively computable functions* $\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$ that are securely computable in presence of an active adversary in the authenticated channels model with broadcast (BC). Next we consider the class of 2-party functions $\mathfrak{F}_{\mathsf{2qu}}$ that can securely be computed where the parties may use quantum cryptographic protocols and the attacker is an unlimited active quantum adversary. We show that the class $\mathfrak{F}_{\mathsf{2qu}}$ is strictly contained in the class $\mathfrak{F}_{\mathsf{2sh}}$ of semi-honestly computable 2-party functions which

[1] However, quantum bit commitment is impossible [May97] and hence no function implying bit commitment is computable.

[2] An adversary cannot learn more about the honest players' inputs than what is implied by inputs and outputs of the corrupted players.

[3] The protocol output equals the intended function value $f(x_1, ..., x_n)$ of the inputs or there is no output.

[4] An adversary cannot prevent the honest parties from learning their outputs.

[5] The honest parties learn their outputs if the adversary learns anything.

[6] Either all honest parties learn their outputs or none.

[7] In the literature our notion of passive is also occasionally referred to as semi-honest.

[8] In the literature our notion of semi-honest is also sometimes referred to as *weakly* semi-honest or *weakly* passive.

gives rise to novel impossibility results beyond those of Mayers [May97] or Kitaev [ABDR04].

To obtain the desired result on LT security we prove that the class of semi-honestly computable functions $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$ equals the class $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$ of functions which can LT securely be computed given an authenticated BC channel. Furthermore, we show that the class of LT securely computable functions remains unchanged if we replace the authenticated BC channel by a network of authenticated channels or by the realistic communication infrastructure of a network of insecure channels with a given public-key infrastructure (PKI). Hence our classification applies to a very practical internet-like setting.

Unlike our IT secure protocols the LT secure protocols given in this work do not achieve robustness or fairness. We show that this is optimal in the sense that generally functions implementable with LT security cannot be implemented with fairness. However, we present protocols which guarantee that only a specific designated party can abort the computation after learning the output. I.e. the fairness property can only be violated by this designated party. Interestingly these protocols make use of CO secure oblivious transfer (OT) protocols even though OT itself cannot be achieved with LT security.

Summarizing our results and importing the treatment of complete two party functions from [KMQ08] (i.e. functions which are cryptographically as powerful as oblivious transfer) we arrive at a complete classification of two party functions. Interestingly, there is a class of functions which cannot securely be computed but still are not complete. This shows that for non-boolean functions there is no zero-one law for privacy [CK89].

1.2 Related Work

Secure computability of functions was first discussed by [CK89]. They characterize the symmetric boolean functions (all parties receive the same output $y \in \{0,1\}$) that can be computed with IT security in presence of passive adversaries in the private channels model. In this scenario functions are either computable or complete (zero-one law for privacy).

Kushilevitz [Kus92] presented the first result for non-boolean functions describing the symmetric 2-party functions which can be computed with perfect security in presence of an unbounded passive adversary. Our protocols and proof techniques draw heavily upon [Kus92]. Also in the 2-party setting, [MQ05] sketches a generalization of [Kus92] to the asymmetric, IT case, connections to LT security and discusses quantum aspects, though without proper formalization or proofs. Our work goes beyond the results of [CK89, Kus92, MQ05] in that we consider IT secure computability of asymmetric, non-boolean functions, in presence of passive, semi-honest, active, and quantum adversaries, for the most part in the multi-party setting.

Gordon et al. [GHKL08] characterize the boolean functions computable with CO fairness in the 2-party setting in presence of active adversaries. Our protocols for active adversaries are robust (and hence fair) and being applicable to asymmetric, non-boolean functions, pertain to a larger class of functions than those of [GHKL08], but in the IT scenario instead of the CO setting.

Other works that deal with the computability of 2-party functions in the perfect or IT setting are [Kil91, Kil00, BMM99, KMQ08]. However, these papers focus mostly on reducibility and completeness, while we are more interested in computability in the authenticated channels model and implications for LT security. Computability of a few interesting special functions in presence of dishonest majorities is discussed in [BT07].

Our impossibility result in the quantum case makes use of a result of Kitaev showing the impossibility of quantum coin flipping which is published in [ABDR04].

Everlasting security from temporary assumptions has been investigated in cryptographic research for some time. It was shown that a bound on the memory available to the adversary allows key exchange and OT protocols [CM97, CCM02] which remain secure even if the memory bound holds only during the execution of the protocol. This idea has been pursued further to achieve everlasting security from a network of distributed servers providing randomness [Rab03]. In [DM04] it was shown that using a CO secure key exchange in the bounded storage model need not yield everlasting security. For some time general quantum cryptographic protocols were sought which obtain everlasting security from a temporary assumption. Such protocols are now generally accepted to be impossible [BCMS99]. Additional assumptions, like a temporary bound on the quantum memory can again provide everlasting security for secure computations [DFSS05].

In this paper we investigate the power of temporary CO assumptions in the standard model. This is along the lines of [MQU07]. However, in [MQU07] strong composability requirements are imposed under which little is possible without additional setup assumptions, like the temporary availability of secure hardware.

2 Security Definitions and Notation

In *secure function evaluation* (SFE) the goal is to compute a function $f : \mathcal{X}_1 \times \ldots \times \mathcal{X}_n \to \mathcal{Y}_1 \times \ldots \times \mathcal{Y}_n$ securely among n parties $\mathfrak{P} = \{P_1, \ldots, P_n\}$.[9] Each party $P_i \in \mathfrak{P}$ ($i \in [n] := \{1, \ldots, n\}$) holds an input $x_i \in \mathcal{X}_i$ from a finite set $\mathcal{X}_i$ and is supposed to receive output $f_i(x_1, \ldots, x_n) := y_i \in \mathcal{Y}_i$, where $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$. We extend this notation to subsets $M = \{P_{m_1}, \ldots, P_{m_{|M|}}\} \subseteq \mathfrak{P}$ and write $f_M(x_1, \ldots, x_n) := y_M := (y_{m_1}, \ldots, y_{m_{|M|}})$ and $x_M := (x_{m_1}, \ldots, x_{m_{|M|}})$. We call the set of all n-party functions $\mathfrak{F}_n$ and the set of multi-party functions $\mathfrak{F} := \bigcup_{n \geq 1} \mathfrak{F}_n$.

In order to compute the function f the parties may execute a protocol π, utilizing a set of resources[10] (communication primitives) R. We designate by $\mathfrak{H} \subset \mathfrak{P}$ the set of honest parties, that execute their protocol machine π_i as specified by protocol π, and by $\mathfrak{E} := \mathfrak{P} \setminus \mathfrak{H}$ the set of corrupted parties that may deviate from the protocol. We generally make the worst case assumption that corrupted parties are controlled by a central adversary E. The adversary (if present, i.e. if at least one party is corrupted) acts for the corrupted parties, sees messages sent over authenticated channels, and can manipulate messages sent over insecure channels. If no party is corrupted, we assume that no adversary is present. External adversaries that can listen on authenticated channels and manipulate insecure channels even when no party is corrupted are easily modelled by adding an additional party, that has constant function output and whose input is ignored. *Securely* computing f then means that privacy of the inputs and correctness of the result should be guaranteed for the honest parties.

We define security using a simulation based stand-alone[11] model (see e.g. [Gol04]) with synchronous message passing. The security of a protocol (the real model) is defined with respect to an ideal model, where f is evaluated by a *trusted third party* or *ideal functionality* I. A protocol π then achieves security according to the simulation paradigm if whatever an adversary E controlling a subset $\mathfrak{E} \subseteq \mathfrak{P}$ of parties can do in the real model, a simulator (or ideal adversary) S (connected to the interfaces of the corrupted parties to the ideal functionality I) could replicate in the ideal model.

This is formalized by means of a distinguisher D. D provides inputs x_i ($P_i \in \mathfrak{H}$) for the honest parties and x_E for the adversary E. In the ideal setting the x_i ($P_i \in \mathfrak{H}$) are input to I, while x_E is passed to the simulator S (which in turn computes inputs $x'_\mathfrak{E}$ to I for the corrupted parties). In the real setting the protocol machines π_i are run on input x_i ($P_i \in \mathfrak{H}$) with the adversary E on input x_E and with resources R. Finally the outputs $y_\mathfrak{H}$ of the honest parties and y_E of the adversary or simulator are passed to D, which then has to output a decision bit $d \in \{0, 1\}$, that can be regarded as the distinguisher's guess if it is connected to the real system $\mathsf{E} \circ \pi_\mathfrak{H} \circ \mathsf{R}$ or the ideal system $\mathsf{S}(\mathsf{E}) \circ I$. To facilitate a unified treatment of different corruption models, we will generally wlog assume that $x_\mathsf{E} = (x_\mathfrak{E}, x'_\mathsf{E})$ and $y_\mathsf{E} = (y_\mathfrak{E}, y'_\mathsf{E})$ where $x_\mathfrak{E} \in \mathcal{X}_\mathfrak{E}$ and $y_\mathfrak{E} \in \mathcal{Y}_\mathfrak{E}$ are function inputs and outputs respectively, x'_E is an auxiliary input and y'_E is the protocol transcript observed by the adversary.[12]

Denoting statistical distance[13] by $\Delta(\cdot, \cdot)$, we can then define the advantage of a class $\mathcal{D}$ of distinguishers in distinguishing systems S and S' as

$$\Delta^\mathcal{D}(S, S') := \max_{\mathsf{D} \in \mathcal{D}} \Delta_\mathsf{D}(S, S') := \max_{\mathsf{D} \in \mathcal{D}} \Delta(\mathsf{D}(S), \mathsf{D}(S')) = \max_{\mathsf{D} \in \mathcal{D}} |\Pr(\mathsf{D}(S) = 1) - \Pr(\mathsf{D}(S') = 1)|,$$

where $\mathsf{D}(S)$ denotes the output of distinguisher D when interacting with system S. A protocol is now secure if for any adversary (corrupting a certain set of parties) there is a simulator which renders ideal and real scenario indistinguishable for any distinguisher.

[9] In the 2-party setting we will occasionally use A and B instead of P_1 and P_2.

[10] In this work these are most often a complete network of authenticated channels or an authenticated broadcast (BC) channel.

[11] as opposed to a universally composable model

[12] Note that arbitrary inputs can be passed to the adversary via x'_E and whatever the adversary might compute from its observations can also be computed from the protocol transcript y'_E directly.

[13] The statistical L1 distance of random variables $X_A, X_B : \Omega \to \mathcal{X}$ with $\mathcal{X}$ finite is $\Delta(X_A, X_B) = \frac{1}{2} \sum_{x \in \mathcal{X}} |\Pr_{X_A}(x) - \Pr_{X_B}(x)|$.

Definition 2.1 (Stand-Alone Security)**.** Given a class of distinguishers $\mathcal{D}$, a class of adversaries $\mathcal{E}$, a class of simulators $\mathcal{S}$, and an advantage function[14] $\epsilon(\kappa)$ (where κ is the security parameter) a protocol π securely implements an ideal functionality I if for every adversary $\mathsf{E} \in \mathcal{E}$ corrupting a set $\mathfrak{C}$ of parties there is a simulator $\mathsf{S}(\mathsf{E}) \in \mathcal{S}$ such that $\Delta^{\mathcal{D}}(\mathsf{S}(\mathsf{E}) \circ I, \mathsf{E} \circ \pi_{\mathfrak{F}} \circ \mathsf{R}) \leq \epsilon(\kappa)$. The type of security is dependent on the choice of $\mathcal{D}$, $\mathcal{E}$, $\mathcal{S}$, and $\epsilon(\kappa)$, and on the ideal functionality I. We will discuss several resulting security paradigms below.

We consider both computational (CO) security, where distinguishers, adversaries and simulators must be efficient[15] (Poly) and information-theoretic (IT) security where distinguishers, adversaries and simulators are arbitrary unbounded algorithms (Algo). Furthermore, we investigate long-term (LT) security where adversaries and simulators must be efficient but distinguishers may be unbounded, formalizing that we expect CO assumptions to hold only for the duration of the protocol. In all three cases the advantage function $\epsilon(\kappa)$ is chosen to be negligible in the security parameter κ. If in the IT case we fix an advantage of $\epsilon = 0$ we arrive at perfect (PF) security. We may vary the IT and PF cases by demanding efficient simulators (ITE, PFE).

Paradigm	**Short**	$\mathcal{D} =$	$\mathcal{E} =$	$\mathcal{S} =$	$\epsilon(\kappa)$	**Notation**
Perfect security	PF	Algo	Algo	Algo	$\epsilon(\kappa) = 0$	$\pi \succcurlyeq^{\mathsf{PF}} I$
Information-theoretical security	IT	Algo	Algo	Algo	$\epsilon(\kappa) < \mathsf{negl}$	$\pi \succcurlyeq^{\mathsf{IT}} I$
PF security with efficient simulator	PFE	Algo	Algo	Poly	$\epsilon(\kappa) = 0$	$\pi \succcurlyeq^{\mathsf{PFE}} I$
IT security with efficient simulator	ITE	Algo	Algo	Poly	$\epsilon(\kappa) < \mathsf{negl}$	$\pi \succcurlyeq^{\mathsf{ITE}} I$
Computational security	CO	Poly	Poly	Poly	$\epsilon(\kappa) < \mathsf{negl}$	$\pi \succcurlyeq^{\mathsf{CO}} I$
Long-term security	LT	Algo	Poly	Poly	$\epsilon(\kappa) < \mathsf{negl}$	$\pi \succcurlyeq^{\mathsf{LT}} I$

Table 1: Basic Security Paradigms

We refine these security paradigms further by defining adversarial models, i.e. restrictions that we can impose on the adversaries and simulators for any of the above paradigms. We discuss active (act) adversaries, where adversaries and simulators in the classes $\mathcal{E}_{\mathsf{act}}$, $\mathcal{S}_{\mathsf{act}}$ are not restricted further; semi-honest (sh) adversaries[8], where adversaries in the class $\mathcal{E}_{\mathsf{sh}}$ are restricted to generate messages according to the prescribed protocol π with the inputs $x_{\mathfrak{C}}$ provided by the distinguisher D, and simulators in the class $\mathcal{S}_{\mathsf{sh}}$ are not restricted further; and passive (pas) adversaries[7], where adversaries are in the class $\mathcal{E}_{\mathsf{pas}} = \mathcal{E}_{\mathsf{sh}}$ and simulators in the class $\mathcal{S}_{\mathsf{pas}}$ are restricted to forward the inputs $x_{\mathfrak{C}}$ provided by the distinguisher D to the ideal functionality I.

We briefly motivate our definition of sh adversaries. When CO tools are applied to force active adversaries to behave passively, they can still substitute inputs. The sh setting is intended to model this scenario, where, in contrast to the pas setting, the adversaries and simulators can substitute the inputs provided by the distinguisher for different ones. However, for simplicity, the definition above only allows for simulators (and not adversaries) to substitute inputs, as this is actually equivalent under the distinguisher classes $\mathcal{D}$ we consider: For any $\mathsf{D} \in \mathcal{D}$ we can find a distinguisher $\mathsf{D}' = \mathsf{D} \circ \sigma \in \mathcal{D}$ that incorporates the input substitution of the adversary $\mathsf{E} = \mathsf{E}' \circ \sigma$. So we can find a passive adversary $\mathsf{E}' \in \mathcal{E}_{\mathsf{sh}} = \mathcal{E}_{\mathsf{pas}}$ and a distinguisher D' that yield the same advantage as E and D.

Security paradigms and adversarial models as defined above are combined by intersecting their defining sets, i.e. sh IT security is described by $\mathcal{D}^{\mathsf{IT}}_{\mathsf{sh}} = \mathcal{D}^{\mathsf{IT}}$, $\mathcal{S}^{\mathsf{IT}}_{\mathsf{sh}} = \mathcal{S}^{\mathsf{IT}} \cap \mathcal{S}_{\mathsf{sh}}$, $\mathcal{E}^{\mathsf{IT}}_{\mathsf{sh}} = \mathcal{E}^{\mathsf{IT}} \cap \mathcal{E}_{\mathsf{sh}}$, $\epsilon(\kappa) < \mathsf{negl}$ and denoted $\pi \succcurlyeq^{\mathsf{IT}}_{\mathsf{sh}} I$. By definition we have the following implications among security paradigms and adversarial models respectively:

$$\begin{array}{ccccccc} \mathsf{PFE} & \Longrightarrow & \mathsf{ITE} & \Longrightarrow & \mathsf{LT} & \Longrightarrow & \mathsf{CO} \\ \Downarrow & & \Downarrow & & & & \\ \mathsf{PF} & \Longrightarrow & \mathsf{IT} & & & & \end{array} \qquad \mathsf{act} \Longrightarrow \mathsf{sh} \Longleftarrow \mathsf{pas}$$

We can now formalize the computation of a function f with a specific set of security properties under each of the definitions above by providing an appropriate ideal functionality. Let $f \in \mathfrak{F}_n$ be a concrete function[16] and let $\mathfrak{C} \subset \mathfrak{P}$ be a set of corrupted players.

[14] usually a negligible function in the security parameter κ

[15] By efficient we mean polynomially bounded in the security parameter κ.

[16] In this work we take the function f to be independent of the security parameter κ. As such the efficiency of protocols is always discussed

Demanding *privacy, correctness and agreement on abort* only for the computation of f is captured by the ideal functionality I_f^{ab}, which operates as follows: I_f^{ab} accepts an input x_i from each party P_i. If a party P_i provides no input, a default input x_i^{def} is used. I_f^{ab} then computes the outputs $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$ and outputs $y_{\mathfrak{E}}$ to the adversary (simulator). If $|\mathfrak{E}| > 0$, the adversary may decide whether the other parties also receive the output (output flag $o = 1$) or not (output flag $o = 0$). Finally, I_f^{ab} sends either the outputs y_i or the empty value $\perp$ to the honest parties, depending on the output flag received from the adversary.[17]

The ideal functionality I_f^{fair} specifying *privacy, correctness and fairness* (which implies agreement on abort) works like I_f^{ab} but takes an output flag *before* making output to the adversary. Then for output flag 1 the functionality I_f^{fair} sends the result y to *all* parties and for output flag 0 it sends $\perp$ to *all* parties.

Computing function f with *full security*, which implies all the security notions mentioned above, is specified by means of the ideal functionality I_f. The functionality I_f operates like I_f^{fair} but takes no output flag and instead directly delivers the output y to *all* parties.

Computing function f with a *designated aborter* (DA) is a slighlty weaker notion of security than fairness in that only the *designated party* P_1 can abort the protocol after receiving output. The corresponding ideal functionality I_f^{des} operates as follows: I_f^{des} accepts an input x_i from each party P_i. If a party P_i provides no input, a default input x_i^{def} is used. I_f^{des} then computes the outputs $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$. If $P_1 \in \mathfrak{E}$ the functionality I_f^{des} outputs $y_{\mathfrak{E}}$ to the adversary (simulator). If $|\mathfrak{E}| > 0$, the adversary may decide whether the other parties also receive the output (output flag $o = 1$) or not (output flag $o = 0$). Finally, I_f^{des} either delivers the remaining outputs y_i or the empty value $\perp$, depending on the output flag received from the adversary.

Note that in the 2-party setting (but not for $n > 2$ parties) robustness and fairness amount to the same. Given I_f^{fair} we can implement I_f by having P_i output $f_i(x_i, x_{2-i}^{\mathsf{def}})$ when it receives $\perp$. Conversely, given I_f, we directly use it as implementation of I_f^{fair}. The simulators are straightforward.

Lemma 2.2. *In the 2-party setting, I_f^{fair} and I_f are efficiently and PFE securely locally mutually reducible, even in presence of active adversaries, i.e. robustness and fairness amount to the same.*

Finally we show that computability by public discussion (authenticated BC only as resources R) and in the authenticated channels model (complete network of authenticated channels as resources R) lead to identical results for semi-honest or passive adversaries. In the authenticated channels model we can securely (against sh and pas adversaries) implement BC by simply sending messages to all other parties. Conversely in the authenticated BC model, authenticated channels can be implemented by broadcasting messages and instructing parties other than the intended recipient not to listen. By the last argument computability by public discussion and in the authenticated channels model *with BC* lead to identical results for active adversaries also.

Lemma 2.3. *In presence of semi-honest or passive adversaries, a function $f \in \mathfrak{F}$ is securely computable in the authenticated channels model if and only if it is computable by public discussion (authenticated BC only). In presence of active adversaries, a function $f \in \mathfrak{F}$ is securely computable in the authenticated channels model* with BC *if and only if it is computable by public discussion.*

3 The Class $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$ of Passively Computable Functions

We subsequently characterize the class $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$ of n-party functions $f \in \mathfrak{F}_n$ that are computable IT securely in the authenticated channels model in presence of a passive adversary.

Definition 3.1 ($\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$: Passively Computable Functions)**.** The class of *passively computable* functions $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$ consists of the functions $f \in \mathfrak{F}$ for which an efficient protocol $\pi \in \mathsf{Poly}$ exists that implements I_f with IT security in presence of a passive adversary in the authenticated channels model.

for a fixed function in terms of the security parameter κ. This is the most relevant case for applications, however, our proofs still hold for a family of functions f_κ, where the input domain grows at most polynomially fast in the security parameter κ.

[17] We could relax the definition further by allowing the adversary to send one output flag for each party, dropping agreement on abort. However, all our protocols will achieve agreement on abort.

Note that by Lem. 2.3 we have $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}} = \mathfrak{F}_{\mathsf{pas}}^{\mathsf{bc}}$, where $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{bc}}$ denotes the functions computable by public discussion in the setting above. Hence we may, for the sake of simplicity, assume an authenticated BC channel instead of authenticated channels as the sole underlying resource in the following discussion.

An important subset $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$ is the set $\mathfrak{F}_{\mathsf{loc}}$ of locally computable n-party functions.

Definition 3.2 ($\mathfrak{F}_{\mathsf{loc}}$: Locally Computable Functions)**.** A function $f \in \mathfrak{F}$ is called *locally computable* ($f \in \mathfrak{F}_{\mathsf{loc}}$) if each party P_i can compute its function value $y_i = f_i(x_1, \ldots, x_n)$ locally, without interacting with a resource or another party.

Obviously, for f to be locally computable, f_i cannot depend on the inputs of parties other then P_i:

Lemma 3.3 (Characterization of $\mathfrak{F}_{\mathsf{loc}}$)**.** *A function $f \in \mathfrak{F}$ is locally computable ($f \in \mathfrak{F}_{\mathsf{loc}}$) if and only if for every $i \in [n]$ and every $x_i \in \mathcal{X}_i$ the restriction $f_i \mid_{\mathcal{X}_1 \times \ldots \times \mathcal{X}_{i-1} \times \{x_i\} \times \mathcal{X}_{i+1} \times \ldots \times \mathcal{X}_n}$ of f is constant.*

Towards a characterization of $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$, we give a combinatorial definition of a set $\mathfrak{F}'_{\mathsf{pas}}$ of functions that we call *passively decomposable*. Passive decomposability captures the fact that a party can send a message about its input such that no adversary can learn anything that is not implied by its own input and function output.

Definition 3.4 ($\mathfrak{F}'_{\mathsf{pas}}$: Passively Decomposable Functions)**.** A function $f \in \mathfrak{F}_n$ is called *passively decomposable*, denoted $f \in \mathfrak{F}'_{\mathsf{pas}}$, if for any restriction $f \mid_{\widetilde{\mathcal{X}}_1 \times \ldots \times \widetilde{\mathcal{X}}_n}$ of f to subsets $\widetilde{\mathcal{X}}_j \subseteq \mathcal{X}_j$ ($j \in [n]$) we have:

1. $f \mid_{\widetilde{\mathcal{X}}_1 \times \ldots \times \widetilde{\mathcal{X}}_n}$ is locally computable ($f \mid_{\widetilde{\mathcal{X}}_1 \times \ldots \times \widetilde{\mathcal{X}}_n} \in \mathfrak{F}_{\mathsf{loc}}$) or
2. there is an $i \in [n]$ and a partition (K-Cut) of $\widetilde{\mathcal{X}}_i$ into non-empty sets $\mathcal{X}'_i \dot{\cup} \mathcal{X}''_i = \widetilde{\mathcal{X}}_i$ such that for all $P_e \in \mathfrak{P} \setminus \{P_i\}$ and all $x_e \in \widetilde{\mathcal{X}}_e$: $f_e(x_e, \widetilde{\mathcal{X}}_{\mathfrak{H}'}, \mathcal{X}'_i) \cap f_e(x_e, \widetilde{\mathcal{X}}_{\mathfrak{H}'}, \mathcal{X}''_i) = \emptyset$ ($\mathfrak{E} := \{P_e\}$, $\mathfrak{H}' := \mathfrak{H} \setminus \{P_i\}$).

The above definition only discusses adversary sets $\mathfrak{E}$ of cardinality $|\mathfrak{E}| = 1$. As the following lemma shows that this is actually equivalent to quantifying over all sets $\mathfrak{E} \subseteq \mathfrak{P}$.

Lemma 3.5 (An Equivalent Characterization of $\mathfrak{F}'_{\mathsf{pas}}$)**.** *A function $f \in \mathfrak{F}_n$ is passively decomposable if and only if for any restriction $f \mid_{\widetilde{\mathcal{X}}_1 \times \ldots \times \widetilde{\mathcal{X}}_n}$ of f to subsets $\widetilde{\mathcal{X}}_j \subseteq \mathcal{X}_j$ ($j \in [n]$) we have:*

1. *$f \mid_{\widetilde{\mathcal{X}}_1 \times \ldots \times \widetilde{\mathcal{X}}_n}$ is locally computable ($f \mid_{\widetilde{\mathcal{X}}_1 \times \ldots \times \widetilde{\mathcal{X}}_n} \in \mathfrak{F}_{\mathsf{loc}}$) or*
2. *there is an $i \in [n]$ and a partition (K-Cut) of $\widetilde{\mathcal{X}}_i$ into non-empty sets $\mathcal{X}'_i \dot{\cup} \mathcal{X}''_i = \widetilde{\mathcal{X}}_i$ such that for all $\emptyset \neq \mathfrak{E} \subseteq \mathfrak{P} \setminus \{P_i\}$ and all $x_{\mathfrak{E}} \in \widetilde{\mathcal{X}}_{\mathfrak{E}}$: $f_{\mathfrak{E}}(x_{\mathfrak{E}}, \widetilde{\mathcal{X}}_{\mathfrak{H}'}, \mathcal{X}'_i) \cap f_{\mathfrak{E}}(x_{\mathfrak{E}}, \widetilde{\mathcal{X}}_{\mathfrak{H}'}, \mathcal{X}''_i) = \emptyset$ ($\mathfrak{H}' := \mathfrak{H} \setminus \{P_i\}$).*

The proof of Lemma 3.5 is by induction over the size of the adversary set $\mathfrak{E}$ and can be found in Appendix B. There we also provide a graphical illustration of passive decomposability (Fig. 2).

We now show that passive decomposability as defined above indeed characterizes the passively computable n-party functions:

Theorem 3.6. *A function $f \in \mathfrak{F}$ is passively computable if and only if it is passively decomposable. In short $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}} = \mathfrak{F}'_{\mathsf{pas}}$. Furthermore, any function $f \in \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$ can efficiently[18] be computed with PFE security.*

The proof of Thm. 3.6 can be found in App. C. $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}} \subseteq \mathfrak{F}'_{\mathsf{pas}}$ is shown by demonstrating that in absence of a K-cut no protocol participant can send a message that bears any information about his input without losing security. The proof of $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}} \supseteq \mathfrak{F}'_{\mathsf{pas}}$ is constructive in the sense that it inductively describes an efficient passively PFE secure protocol π_f to compute a function $f \in \mathfrak{F}'_{\mathsf{pas}}$. The protocol π_f generalizes the approach of [Kus92] to asymmetric n-party functions:

Wlog assume that there is a partition of $\mathcal{X}_i = \mathcal{X}_i^{(1)} \dot{\cup} \mathcal{X}_i^{(2)}$ as described in Definition 3.4. The protocol π_f then proceeds as follows: The party P_i determines the message $m_1 \in \{0, 1\}$ such that for the input $x_i \in \mathcal{X}_i$ of P_i we have $x_i \in \mathcal{X}_i^{(m_1)}$ and broadcasts m_1. The parties $\mathfrak{P}$ then restrict the function f to $f|_{\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_i^{(m_1)} \times \ldots \times \mathcal{X}_n}$ and proceed with a partition for the restricted function in the same fashion. The process is iterated until the parties arrive at a locally computable restriction of f, at which point they can determine the output locally.

We conjecture that the above protocol achieves the optimal round complexity if it is refined to use the finest possible decomposition (according to [Kus92]) of the input domains in every round.

[18] in the security parameter κ

4 The Class $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$ of Semi-Honestly Computable Functions

Next we characterize the class $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$ of n-party functions that are IT securely computable in the authenticated channels model in presence of a semi-honest adversary. Here, in order to obtain extra information, the corrupted parties are allowed to exchange their inputs for different ones, but must still behave according to the prescribed protocol. The results in this chapter will later help us to characterize LT secure functions in a very practical setting.

Definition 4.1 ($\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$: Semi-Honestly Computable Functions)**.** The class of *semi-honestly computable* functions $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$ consists of the functions $f \in \mathfrak{F}$ for which an efficient protocol $\pi \in \mathsf{Poly}$ exists that implements I_f with IT security in presence of a semi-honest adversary in the authenticated channels model.

Note that by Lem. 2.3 we have $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} = \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}}$, where $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}}$ denotes the functions computable by public discussion in the setting above. Hence we may, for the sake of simplicity, assume an authenticated BC channel instead of authenticated channels as the sole underlying resource in the following discussion.

We intend to characterize the class $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$ combinatorially. To this end we introduce the concept of redundancy-freeness for n-party functions, generalizing the 2-party definitions of [KMQ08]. For a party P_i, two of its possible inputs x_i and x_i' to f may be completely indistinguishable to the other parties (by their output from f), while the input x_i may yield a more informative output from f for P_i than x_i'. We then say the input x_i yielding more information dominates the input x_i' giving less information. As semi-honest (and active) adversaries can select their inputs, generally with the goal to obtain as much information as possible, the dominated input x_i' giving less information is not useful to a corrupted P_i. Along the same lines an ideal adversary (simulator) can always use the dominating input x_i instead x_i' of for simulation. As such the input x_i' is redundant, irrelevant in terms of security, and we can eliminate it from the function f under consideration. This procedure yields a *redundancy-free version* $\hat{f}$ of f, with new, smaller, dominating input sets.

Definition 4.2 (Domination and Redundancy-Freeness)**.** Given an n-party function $f \in \mathfrak{F}_n$ we say $x_i \in \mathcal{X}_i$ *dominates* $x_i' \in \mathcal{X}_i$ if and only if for all $x_{\mathfrak{P}'} \in \mathcal{X}_{\mathfrak{P}'}$: $f_{\mathfrak{P}'}(x_i, x_{\mathfrak{P}'}) = f_{\mathfrak{P}'}(x_i', x_{\mathfrak{P}'})$ and for all $x_{\mathfrak{P}'}, x'_{\mathfrak{P}'} \in \mathcal{X}_{\mathfrak{P}'}$: $f_i(x_i', x_{\mathfrak{P}'}) \neq f_i(x_i', x'_{\mathfrak{P}'}) \implies f_i(x_i, x_{\mathfrak{P}'}) \neq f_i(x_i, x'_{\mathfrak{P}'})$, where $\mathfrak{P}' = \mathfrak{P} \setminus \{P_i\}$.

We proceed to define sets of dominating inputs $\widetilde{\mathcal{X}}_j := \{\mathcal{X} \subseteq \mathcal{X}_j \mid \forall x' \in \mathcal{X}_j \exists x \in \mathcal{X} : x \text{ dominates } x'\}$ ($j \in [n]$). We define the *dominating set* $\hat{\mathcal{X}}_j$ as (some) element of minimal cardinality in $\widetilde{\mathcal{X}}_j$. We then call $\hat{f} := f|_{\hat{\mathcal{X}}_1 \times \ldots \times \hat{\mathcal{X}}_n}$ the *redundancy-free version* of f. Furthermore, for $x_j \in \mathcal{X}_j$ let $\hat{x}_j \in \hat{\mathcal{X}}_j$ be the (unique) element that dominates x_j.

The redundancy-free version $\hat{f}$ of f is uniquely defined up to a renaming of input and output values (also see App. G or Sec. 8). Domination is a reflexive and transitive relation. Furthermore it is antisymmetric up to renaming of input and output symbols. Hence two different dominating sets $\hat{\mathcal{X}}_i$ and $\hat{\mathcal{X}}_i'$ are sets of maximal elements under the domination relation and equal up to renaming of input and output values.

Since corrupted parties can cooperate to choose their inputs to obtain as much information as possible, it is important to note that the above Def. 4.2 generalizes to the combined input of the corrupted parties $\mathfrak{E}$ as stated in Lem. 4.3 below. So if each corrupted party P_{e_j} chooses an input x_{e_j} dominating input x'_{e_j}, then the combined adversarial input $x_{\mathfrak{E}}$ actually dominates $x'_{\mathfrak{E}}$.

Lemma 4.3. *Let $x_{\mathfrak{E}} = (x_{e_1}, \ldots, x_{e_{|\mathfrak{E}|}})$, $x'_{\mathfrak{E}} = (x'_{e_1}, \ldots, x'_{e_{|\mathfrak{E}|}})$ such that each x_{e_j} dominates x'_{e_j} ($j \in [|\mathfrak{E}|]$). Then we have for all $x_{\mathfrak{H}} \in \mathcal{X}_{\mathfrak{H}}$: $f_{\mathfrak{H}}(x_{\mathfrak{E}}, x_{\mathfrak{H}}) = f_{\mathfrak{H}}(x'_{\mathfrak{E}}, x_{\mathfrak{H}})$ and for all $x_{\mathfrak{H}}, x'_{\mathfrak{H}} \in \mathcal{X}_{\mathfrak{H}}$: $f_{\mathfrak{E}}(x'_{\mathfrak{E}}, x_{\mathfrak{H}}) \neq f_{\mathfrak{E}}(x'_{\mathfrak{E}}, x'_{\mathfrak{H}}) \implies f_{\mathfrak{E}}(x_{\mathfrak{E}}, x_{\mathfrak{H}}) \neq f_{\mathfrak{E}}(x_{\mathfrak{E}}, x'_{\mathfrak{H}})$. Again we say that $x_{\mathfrak{E}}$ dominates $x'_{\mathfrak{E}}$.*

The proof of Lem. 4.3 is by induction on $|\mathfrak{E}|$ and can be found in App. D.

The following lemma states that the functions f and $\hat{f}$ are locally[19] and efficiently mutually reducible. This means that it does not matter in terms of security which of the two functions is used and redundant inputs can safely be eliminated.

Lemma 4.4. *The functions f and $\hat{f}$ are efficiently and PFE securely locally mutually reducible, even in presence of active adversaries.*

[19] without using any communication resources

The proof of Lem. 4.4 is fairly straightforward, by showing how to implement $I_{\hat{f}}$ when I_f is given and vice versa. It can be found in App. E. The protocol essentially replaces inputs x_i with dominating inputs $\hat{x}_i$.

As PFE security in presence of active adversaries implies IT security in presence of semi-honest adversaries, we can derive the following simple corollary:

Corollary 4.5. *For any function $f \in \mathfrak{F}$ we have: $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} \iff \hat{f} \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$.*

An n-party function f is then sh computable if and only if its redundancy-free version $\hat{f}$ is pas computable.

Theorem 4.6. *For a function $f \in \mathfrak{F}$ we have: $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} \iff \hat{f} \in \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$.*

The full proof of Thm. 4.6 can be found in App. F, we only give a sketch here. By Cor. 4.5 we know that $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} \iff \hat{f} \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$. Therefore it suffices to show for redundancy-free functions f where $f = \hat{f}$ that we have $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} \iff f \in \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$. The implication $f \in \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}} \implies f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$ is then clear by definition. The implication $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} \implies f \in \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$ is shown along the lines of the proof of Thm. 3.6, demonstrating that $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} \implies f \in \mathfrak{F}'_{\mathsf{pas}}$. The proof exploits the redundancy-freeness of f due to which a (working) simulator in the sh setting cannot actually substitute inputs.

The functions $f^{(5)}$ and $f^{(6)}$ in Fig. 1 are examples of *not* sh computable functions taken from [Kus92]. The function $f^{(6)}$ is of particular interest as it is of strictly less cryptographic strength than oblivious transfer. Function $f^{(9)}$ is sh computable: After eliminating the redundant input x_3, the function is pas computable (as indicated by the horizontal and vertical lines).

5 The Class $\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$ of Actively Computable Functions

In contrast to previous sections some of the results in this section are limited to the 2-party case. We characterize the class $\mathfrak{F}_{\mathsf{2act}}$ of 2-party functions which can securely be computed in presence of an unlimited active adversary. We give some generalizations to the n-party scenario, but our 2-party results are sufficient to see that $\mathfrak{F}_{\mathsf{2act}}$ is strictly contained in $\mathfrak{F}_{\mathsf{2sh}}$ and hence the notion of LT security lies strictly between IT security and CO security.

Definition 5.1 ($\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$: Actively Computable Functions)**.** The class of *actively computable* functions $\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$ consists of the functions $f \in \mathfrak{F}$ for which an efficient protocol $\pi \in \mathsf{Poly}$ exists that implements I_f with IT security in presence of an active adversary in the authenticated channels model with broadcast.

Note that by Lem. 2.3 we have $\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}} = \mathfrak{F}_{\mathsf{act}}^{\mathsf{bc}}$, where $\mathfrak{F}_{\mathsf{act}}^{\mathsf{bc}}$ denotes the functions computable by public discussion in the setting above. Hence we may, for the sake of simplicity, assume an authenticated BC channel as the sole underlying resource in the following discussion.

Interestingly there are some useful functions in the class $\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$, e.g. $f^{(7)}$ in Fig. 1 which is a formalization of a Dutch flower auction, where the price is lowered in every round until a party decides to buy.

We next give a combinatorial characterization of actively computable functions, which essentially states that a party P_i must be able to send a message about its input such that the corrupted parties $\mathfrak{C}$ reacting to this new information by changing their input from $x'_{\mathfrak{C}}$ to $x''_{\mathfrak{C}}$ could have achieved the same effect on the output by selecting a third input $x_{\mathfrak{C}}$ a priori:

Definition 5.2 ($\mathfrak{F}'_{\mathsf{act}}$: Actively Decomposable Functions)**.** A function $f \in \mathfrak{F}$ is called *actively decomposable*, denoted $f \in \mathfrak{F}'_{\mathsf{act}}$, if and only if $\hat{f} \in \widehat{\mathfrak{F}_{\mathsf{act}}}$. We have $f \in \widehat{\mathfrak{F}_{\mathsf{act}}}$ if one of the following holds:

1. f is locally computable ($f \in \mathfrak{F}_{\mathsf{loc}}$);
2. there is an $i \in [n]$ and a partition (T-Cut) of $\mathcal{X}_i$ into non-empty sets $\mathcal{X}'_i \dot{\cup} \mathcal{X}''_i = \mathcal{X}_i$ such that
   (i) $f|_{\mathcal{X}_1 \times \ldots \times \mathcal{X}'_i \times \ldots \times \mathcal{X}_n}, f|_{\mathcal{X}_1 \times \ldots \times \mathcal{X}''_i \times \ldots \times \mathcal{X}_n} \in \widehat{\mathfrak{F}_{\mathsf{act}}}$ and

(ii) for all $\mathfrak{E} \subseteq \mathfrak{P} \setminus \{P_i\}$ and $\mathfrak{H}' := \mathfrak{H} \setminus \{P_i\}$ we have

$$\forall \bar{x}_{\mathfrak{E}} \in \mathcal{X}_{\mathfrak{E}} : f_{\mathfrak{E}}(\bar{x}_{\mathfrak{E}}, \mathcal{X}_{\mathfrak{H}'}, \mathcal{X}'_i) \cap f_{\mathfrak{E}}(\bar{x}_{\mathfrak{E}}, \mathcal{X}_{\mathfrak{H}'}, \mathcal{X}''_i) = \emptyset \qquad \text{(K-cut)} \qquad \text{and}$$
$$\forall x'_{\mathfrak{E}}, x''_{\mathfrak{E}} \in \mathcal{X}_{\mathfrak{E}}\ \exists x_{\mathfrak{E}} \in \mathcal{X}_{\mathfrak{E}}\ \forall x_{\mathfrak{H}'} \in \mathcal{X}_{\mathfrak{H}'}\ (\quad \forall x'_i \in \mathcal{X}'_i : f_{\mathfrak{H}}(x'_{\mathfrak{E}}, x_{\mathfrak{H}'}, x'_i) = f_{\mathfrak{H}}(x_{\mathfrak{E}}, x_{\mathfrak{H}'}, x'_i)$$
$$\wedge\ \forall x''_i \in \mathcal{X}''_i : f_{\mathfrak{H}}(x''_{\mathfrak{E}}, x_{\mathfrak{H}'}, x''_i) = f_{\mathfrak{H}}(x_{\mathfrak{E}}, x_{\mathfrak{H}'}, x''_i)\ )$$

Active decomposability indeed characterizes the actively computable functions.

Theorem 5.3. *A function $f \in \mathfrak{F}$ is actively computable if it is actively decomposable. In short $\mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}} \supseteq \mathfrak{F}'_{\mathsf{act}}$. In the 2-party case*[20] *we even have $\mathfrak{F}_{2\mathsf{act}} \subseteq \mathfrak{F}'_{2\mathsf{act}}$, i.e. $\mathfrak{F}_{2\mathsf{act}} = \mathfrak{F}'_{2\mathsf{act}}$. Furthermore, any function $f \in \mathfrak{F}'_{\mathsf{act}}$ can be computed efficiently with PFE security.*

Furthermore, we conjecture:

Conjecture 5.4. *A function $f \in \mathfrak{F}$ is actively computable if and only if it is actively decomposable. In short $\mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}} = \mathfrak{F}'_{\mathsf{act}}$. Furthermore, any function $f \in \mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}}$ can be computed efficiently with PFE security.*

The proof of Thm. 5.3 can be found in App. H. The implication $f \in \mathfrak{F}'_{\mathsf{act}} \implies f \in \mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}}$ is proven by showing the protocol for the semi-honest scenario secure against active adversaries, when applied to the T-cuts of a function $f \in \mathfrak{F}'_{\mathsf{act}}$ instead of the K-cuts of a function in $\mathfrak{F}^{\mathsf{aut}}_{\mathsf{sh}}$. To obtain $f \in \mathfrak{F}_{2\mathsf{act}} \implies f \in \mathfrak{F}'_{2\mathsf{act}}$ we observe that for $f \notin \mathfrak{F}'_{2\mathsf{act}}$ the adversary can in any protocol induce an ouput distribution that is impossible to achieve in the ideal setting. The adversary does this by extracting information on the inputs of other participants from the protocol messages and adjusting his input according to that information.

The functions $f^{(7)}$ and $f^{(8)}$ in Fig. 1 are examples of actively computable functions. Especially compare $f^{(8)}$ with $f^{(2)} \in \mathfrak{F}_{2\mathsf{sh}}$ which is *not* actively computable. The lines in the tables for $f^{(7)}$ and $f^{(8)}$ represent messages which are to be sent in the protocol.

6 Quantum Protocols

In this section we will relate the class $\mathfrak{F}_{2\mathsf{sh}}$ of sh computable 2-party functions with the class of 2-party functions computable with quantum cryptography in presence of an active adversary. A similar result has been obtained by Louis Salvail, but is not published yet. Naturally, we have to adapt our model of security to the quantum case. All machines except for the distinguisher D will be quantum machines able to exchange quantum messages. Furthermore, all inputs and outputs must be classical and the distinguisher must try to distinguish the real and the ideal model based on this classical information.

Let $\mathfrak{F}_{2\mathsf{qu}}$ denote the set of functions $f \in \mathfrak{F}_2$ which can, with the help of a quantum channel, securely and efficiently be computed in presence of an unbounded active adversary. Then the following result holds.

Theorem 6.1. *The class $\mathfrak{F}_{2\mathsf{qu}}$ of quantum computable functions is strictly contained in the class of* sh *computable functions $\mathfrak{F}_{2\mathsf{sh}}$.*

A proof of this theorem is sketched in App. I. The strict inclusion $\mathfrak{F}_{2\mathsf{qu}} \subsetneq \mathfrak{F}_{2\mathsf{sh}}$ gives rise to new impossibility results. For instance, the function $f^{(6)} \notin \mathfrak{F}_{2\mathsf{sh}}$ in Fig. 1 cannot be computed by means of quantum cryptography. An interesting still open question is the power of temporary CO assumptions together with a quantum channel. It is known that this does not suffice to securely implement any function which could in turn be used to implement an IT secure bit commitment. However a secure implementation of the function $f^{(6)}$ in Fig. 1 is not precluded by this impossibility result.

[20] For a function class $\mathfrak{F}^{\mathsf{chan}}_{\mathsf{name}}$ we denote the 2-party subclass $\mathfrak{F}^{\mathsf{chan}}_{\mathsf{name}} \cap \mathfrak{F}_2$ by $\mathfrak{F}_{2\mathsf{name}}$. We drop the communication model specification chan as it is irrelevant for the 2-party setting.

7 Long-Term Security

Subsequently we characterize the n-party functions that can be computed LT securely (without fairness) in presence of active adversaries. LT security means we are willing to make CO assumptions, but only for the duration of the protocol interaction. Once the protocol has terminated we demand IT security. We look at different classes of LT securely computable functions, defined by different channel models. The most practical model, corresponding to the class $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\,pki}}$, is an internet-like setting, where insecure channels and a PKI are available to the parties. Furthermore we also discuss the classes $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}}$ where authenticated channels and $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$ where an authenticated BC channel are given. We find that all these classes $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\,pki}} = \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}} = \mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}}$ are equal to the class $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$ of sh computable functions.

Definition 7.1 ($\mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$, $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\,pki}}$, $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}}$: LT Computable Functions)**.** The classes of *LT computable* functions (i) $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$, (ii) $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\,pki}}$, (iii) $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}}$ consists of the functions $f \in \mathfrak{F}$ for which an efficient protocol $\pi \in \mathsf{Poly}$ exists that implements I_f^{ab} with LT security in presence of an active adversary from (i) an authenticated broadcast channel; (ii) a complete network of insecure channels and a PKI; (iii) a complete network of authenticated channels; respectively.

We now show that the classes defined in the previous section are all equivalent to $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$. First, we observe that once we allow CO assumptions during the protocol execution, we can force semi-honest behavior (i.e. that the adversary behaves according to the protocol) using an unconditionally hiding commitment scheme and zero-knowledge arguments of knowledge:

Theorem 7.2. *If one-way functions (OWF) exist, we have* $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} = \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$.

A proof of Thm. 7.2 can be found in App. J. Furthermore we claim:

Theorem 7.3. *We have* $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\,pki}} = \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}} = \mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}} = \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$.

We prove this by showing $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}} \subseteq \mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\,pki}}$, $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\,pki}} \subseteq \mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}}$, $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}} \subseteq \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$. First, $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}} \subseteq \mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\,pki}}$ holds as we can use the Dolev-Strong-Protocol [DS83] to obtain authenticated BC in the PKI setting. $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\,pki}} \subseteq \mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}}$ holds as using detectable precomputation [FHHW03] we can establish a PKI in the authenticated channels model.[21] $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}} \subseteq \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$ holds as given authenticated BC, authenticated channels can be implemented by simply broadcasting messages and instructing honest parties other than the intended recipient to ignore the message.

Thm. 7.3 is optimal in the sense that we cannot hope to implement all functions $f \in \mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\,pki}}$ with robustness or even fairness. Of course we have (by definition) robust LT (even IT) secure protocols for the functions $f \in \mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$. But e.g. the symmetric XOR function $f_{\mathrm{XOR}}(x_1, x_2) := (x_1 \,\mathrm{XOR}\, x_2, x_1 \,\mathrm{XOR}\, x_2)$ is by the combinatorial characterizations of the previous sections $f_{\mathrm{XOR}} \in \mathfrak{F}_{2\mathsf{sh}} \setminus \mathfrak{F}_{\mathsf{act}} \subset \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}} \setminus \mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$. Now a fair implementation of f_{XOR} would clearly imply a fair cointoss, which by [Cle86] cannot be implemented in the model under consideration. As such the security without fairness as guaranteed by Thm. 7.3 is indeed the best we can hope for.

7.1 Long Term Security with designated Aborter

As mentioned above we cannot generally guarantee robustness or even fairness for a LT secure protocol π_f computing $f \in \mathfrak{F}_{2\mathsf{lts}}$. However, under stronger CO assumptions, we can guarantee that only a specific designated party can abort the protocol after obtaining output and before the honest parties can generate output. This may be of practical relevance where a specific party is not trusted, but can be relied upon not to abort the protocol. For instance a party may have a vested interest in the successful termination of the protocol regardless of the outcome. One may think of an auctioneer that gets paid only if the auction terminates successfully. Or a party may act in an official capacity and cannot abort the protocol for legal reasons.

We will show that stronger guarantees of this type are obtainable if the underlying CO assumption allows for an oblivious transfer (OT) protocol which is LT secure against one of the participants. Enhanced trapdoor one-way permutations are an example of such an assumption [Gol04]. It is generally believed that OT is *not* implied by OWFs, meaning that LT security with designated aborter appears to require strictly stronger assumptions than plain LT security.

[21] Note that robustness is not required here: The establishment of the PKI may fail, but then the protocol simply aborts.

Lemma 7.4. *Any* sh *computable function* $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} = \mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\,pki}}$ *can be computed using a protocol* π *which is LTS-DA, i.e. implements* I_f^{des} *with CO security and simultaneously* I_f^{ab} *with LT security in the insecure channels model with PKI iff CO oblivious transfer LT-secure against one party (CO-OT+) exists.*

A proof of this lemma is sketched in Appendix K. Essentially we apply the protocol compiler of [GMW87, Gol04] to the distributed circuit of the sh secure protocol for f in such a fashion that gates owned by a specific party P_i are computed with CO primitives that IT protect P_i. Reconstruction is in the end done toward the designated party P_1, which then ensures that the remaining parties can reconstruct. As a result the protocol is CO correct, and IT no one learns more than in the sh secure protocol for f.

8 Classification of 2-party Functions

Combining the results of this work and of [KMQ08], we can derive a complete combinatorial classification of the 2-party functions $\mathfrak{F}_2$ by completeness and computability.

We first define an equivalence relation *renaming* on $\mathfrak{F}_2$ by $f^{(1)} \equiv f^{(2)}$ iff $f^{(2)}$ is obtained from $f^{(1)}$ by locally renaming input and output values. A formal definition can be found in [KMQ08] or App. G. It is easy to see that renamings are locally mutually reducible under all security paradigms considered in this work. In particular $f^{(1)} \equiv f^{(2)}$ implies $I_{f^{(1)}} \succcurlyeq_{\mathsf{act}}^{\mathsf{PFE}} I_{f^{(2)}} \succcurlyeq_{\mathsf{act}}^{\mathsf{PFE}} I_{f^{(1)}}$ and $I_{f^{(1)}} \succcurlyeq_{\mathsf{pas}}^{\mathsf{PFE}} I_{f^{(2)}} \succcurlyeq_{\mathsf{pas}}^{\mathsf{PFE}} I_{f^{(1)}}$.

Next we define an equivalence relation *matching* on the set of classes $\mathfrak{F}_2/\equiv$ (and thereby on $\mathfrak{F}_2$) by isolating inputs that lead to identical behavior and regarding functions as matching if, after eliminating such trivially redundant inputs, they are renamings:

Definition 8.1. Given a 2-party function $f \in \mathfrak{F}_2$ we say x_A *matches* x'_A for inputs $x_A, x'_A \in \mathcal{X}_A$, iff x_A dominates x'_A and x'_A dominates x_A. The matching relation is an equivalence relation on $\mathcal{X}_A$. By $\bar{\mathcal{X}}_A$ we designate a set of representatives. $\bar{\mathcal{X}}_B$ is defined analogously.

We then call $\bar{f} := f|_{\bar{\mathcal{X}}_A \times \bar{\mathcal{X}}_B}$ the *weakly redundancy-free version* of f and for $f^{(1)}, f^{(2)} \in \mathfrak{F}_2$ we write $f^{(1)} \cong f^{(2)}$ if $\bar{f}^{(1)} \equiv \bar{f}^{(2)}$ Furthermore for $x_A \in \mathcal{X}_A$ and $x_B \in \mathcal{X}_B$ let $\bar{x}_A \in \bar{\mathcal{X}}_A$ and $\bar{x}_B \in \bar{\mathcal{X}}_B$ be the (unique) elements that match x_A respectively x_B.

Like the redundancy-free version $\hat{f}$ of f, the weakly redundancy-free version $\bar{f}$ of f is well defined up to renaming. Before we can state the actual classification, we have to reiterate another result of [KMQ08]:

Theorem 8.2 (Complete Functions [KMQ08])**.** *The classes* $\mathfrak{C}_{\mathsf{2act}}$*,* $\mathfrak{C}_{\mathsf{2sh}}$ *and* $\mathfrak{C}_{\mathsf{2pas}}$ *of actively, semi-honestly, and passively complete 2-party functions are the classes of functions* $f \in \mathfrak{F}_2$ *to which all other 2-party functions can be securely reduced in presence of an active, semi-honest or passive adversary respectively.*

The classes $\mathfrak{C}_{\mathsf{2act}} = \mathfrak{C}_{\mathsf{2sh}}$ *consist of exactly the functions* $f \in \mathfrak{F}_2$ *where* $\hat{f} \in \mathfrak{C}_{\mathsf{2pas}}$*. The class* $\mathfrak{C}_{\mathsf{2pas}}$ *consists of exactly the functions* $f \in \mathfrak{F}_2$ *where* $\exists\, a_1, a_2 \in \mathcal{X}_A,\ b_1, b_2 \in \mathcal{X}_B$ *:*

$$f_A(a_1,b_1) = f_A(a_1,b_2) \ \wedge\ f_B(a_1,b_1) = f_B(a_2,b_1) \ \wedge\ \big(\, f_A(a_2,b_1) \neq f_A(a_2,b_2) \ \vee f_B(a_1,b_2) \neq f_B(a_2,b_2) \,\big).$$

We refer to this combinatorial structure as minimal OT.

Note that $f \in \mathfrak{C}_{\mathsf{2pas}}$ iff $f \in \mathfrak{C}_{\mathsf{2act}}$ or $\hat{f} \not\equiv \bar{f}$. This is clear from Kraschewki's result as stated above and from the observation that $\hat{f} \not\equiv \bar{f}$ implies a minimal OT. We then arrive at the following

Theorem 8.3 (Classification)**.** *The class of 2-party functions is a disjoint union of three sets* $\mathfrak{F}_2 = \mathfrak{C}_{\mathsf{2act}} \cup \mathfrak{F}_{\mathsf{2act}} \cup \mathfrak{F}_{\mathsf{2act}}^{\mathsf{nct}}$ *or* $\mathfrak{F}_2 = \mathfrak{C}_{\mathsf{2sh}} \cup \mathfrak{F}_{\mathsf{2sh}} \cup \mathfrak{F}_{\mathsf{2sh}}^{\mathsf{nct}}$ *or* $\mathfrak{F}_2 = \mathfrak{C}_{\mathsf{2pas}} \cup \mathfrak{F}_{\mathsf{2pas}} \cup \mathfrak{F}_{\mathsf{2pas}}^{\mathsf{nct}}$ *where* nct *stand for "neither complete nor computable". Now*

$$\emptyset \neq \mathfrak{F}_{\mathsf{2act}}, \mathfrak{F}_{\mathsf{2pas}} \subsetneq \mathfrak{F}_{\mathsf{2act}} \cup \mathfrak{F}_{\mathsf{2pas}} \subsetneq \mathfrak{F}_{\mathsf{2sh}}$$
$$\emptyset \neq \mathfrak{F}_{\mathsf{2pas}}^{\mathsf{nct}} \subsetneq \mathfrak{F}_{\mathsf{2sh}}^{\mathsf{nct}} \subsetneq \mathfrak{F}_{\mathsf{2act}}^{\mathsf{nct}}$$
$$\emptyset \neq \mathfrak{C}_{\mathsf{2act}} = \mathfrak{C}_{\mathsf{2sh}} \subsetneq \mathfrak{C}_{\mathsf{2pas}}$$

The above results are directly derived from the combinatorial descriptions of the function classes that can be found in the preceding sections and, as far as complete functions are concerned, in [KMQ08]. Additional details and examples can be found in Appendix L.

9 Conclusions

We defined the notion of long-term (LT) security, where we assume that the adversary is CO bounded *during* the execution of the protocol *only*. That is, we rely on CO assumptions, but only for the duration of the protocol execution; thereafter, a failure of the CO assumptions must not compromise security. We then gave a combinatorial description of the class $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\ pki}}$ of functions that can be computed LT securely in an internet-like setting, where a complete network of insecure channels and a PKI are available. Towards this goal, we characterized the classes $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$, $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$ and $\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$ of functions that can be computed with information theoretic (IT) security in the authenticated channels model (with broadcast) in presence of passive, semi-honest and active adversaries. Our results are constructive in that, for every function proven computable in a given setting, one can deduce a secure protocol.

More precisely, we showed that semi-honest computability and LT secure computability amount to the same, i.e. $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} = \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}} = \mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}} = \mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\ pki}}$, where the classes $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}}$ and $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$ are defined analogously to $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\ pki}}$, but rely on a network of authenticated channels or authenticated broadcast respectively as communication resources. We then characterized the class $\mathfrak{F}_{\mathsf{2act}}$ of actively computable 2-party functions in order to offset IT secure computability against LT secure computability. Indeed, we found $\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}} \subsetneq \mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\ pki}}$, meaning that in presence of corrupted majorities strictly more functions are computable with LT security than with IT security. We furthermore gave a necessary condition (that we conjecture also to be sufficient) for an n-party function to be in $\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$. As the functions in $\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$ are robustly (and therefore fairly) computable, these results can be interpreted along the lines Gordon et al. [GHKL08], who discuss the fair computability of binary 2-party functions in the CO setting. Our results apply to the IT scenario instead of the CO setting, there however, our results are much more general in that they pertain to arbitrary n-party functions. We showed that for the functions $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\ pki}}$ fairness is generally not achievable. However, for the functions $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\ pki}}$ we can guarantee LT security with designated aborter, where only a specific designated party can prematurely abort the protocol after having learned the output. Astonishingly, CO secure oblivious transfer (OT) is used in our construction, even though OT itself cannot be realized with full LT security.

We remark, that from a practical point of view, LT security is a useful notion if we deal with sensitive data that has to remain private beyond a limited time frame in a setting where a majority of the parties may be corrupted. In such a setting general IT secure SFE protocols like [BGW88] fail, as they do not tolerate corrupted majorities. CO protocols can tolerate corrupted majorities (if fairness is not required) but, as time passes, progress in hardware or algorithms may invalidate our CO assumptions and jeopardize the privacy of our computation. As the problem with CO assumptions is not so much that these could be unjustified right now, but rather their possible future invalidation, LT security is a viable alternative to IT security in this case. And indeed we could show that $\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}} \subsetneq \mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\ pki}}$, i.e. there are functions that cannot be computed with IT security in presence of dishonest majorities, but can be computed with LT security.

Furthermore, we found that quantum cryptography is not helpful in our context, i.e. the class $\mathfrak{F}_{\mathsf{2qu}}$ of 2-party functions which can be implemented with quantum cryptography is strictly contained in $\mathfrak{F}_{\mathsf{2sh}}$. This inclusion implies novel impossibility results beyond those of Mayers [May97] or Kitaev [ABDR04]. However, quantum cryptography can solve classically impossible problems in other models of security, like achieving a certain robustness to abort in a model with guaranteed message delivery or implementing deniable key exchange.

Finally, collecting results form the literature, especially [Kus92, KMQ08], and adding the results of this work, we obtain a complete taxonomy of 2-party functions by computability and completeness in the IT setting.

10 Acknowledgements

The authors wish to thank Daniel Kraschewski for helpful comments and discussions, and Ueli Maurer for encouragement and insightful comments on security models.

References

[ABDR04] Andris Ambainis, Harry Buhrman, Yevgeniy Dodis, and Hein Röhrig. Multiparty quantum coin flipping. In *IEEE Conference on Computational Complexity*, pages 250–259. IEEE Computer Society, 2004.

[BCMS99] Gilles Brassard, Claude Crépeau, Dominic Mayers, and Louis Salvail. Defeating classical bit commitments with a quantum computer. Los Alamos preprint archive quant-ph/9806031, May 1999.

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. 20th ACM Symposium on the Theory of Computing (STOC)*, pages 1–10, 1988.

[BMM99] Amos Beimel, Tal Malkin, and Silvio Micali. The all-or-nothing nature of two-party secure computation. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO'99*, volume 1666 of *LNCS*, pages 80–97. Springer, 1999.

[BT07] Anne Broadbent and Alain Tapp. Information-theoretic security without an honest majority. In *ASIACRYPT*, pages 410–426, 2007.

[CCM02] Christian Cachin, Claude Crépeau, and Julien Marcil. Oblivious transfer with a memory-bounded receiver. In *34th Annual ACM Symposium on Theory of Computing, Proceedings of STOC 2002*, pages 493–502. ACM Press, 2002.

[CK89] Chor and Kushilevitz. A zero-one law for boolean privacy. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 1989.

[Cle86] Richard Cleve. Limits on the security of coin flips when half the processors are faulty. In *STOC '86: Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 364–369, New York, NY, USA, 1986. ACM Press.

[CM97] Christian Cachin and Ueli Maurer. Unconditional security against memory-bounded adversaries. In Burton S. Kaliski Jr., editor, *Advances in Cryptology—CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 292–306. Springer-Verlag, 17–21 August 1997.

[DFSS05] Ivan Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. Cryptography in the bounded quantum-storage model. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 449–458. IEEE Computer Society, 2005.

[DM04] Stefan Dziembowski and Ueli M. Maurer. On generating the initial key in the bounded-storage model. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 126–137. Springer, 2004.

[DS83] Dolev and Strong. Authenticated algorithms for byzantine agreement. *SICOMP: SIAM Journal on Computing*, 12, 1983.

[FHHW03] Matthias Fitzi, Martin Hirt, Thomas Holenstein, and Jürg Wullschleger. Two-threshold broadcast and detectable multi-party computation. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, volume 265 of *Lecture Notes in Computer Science*, pages 51–67. Springer-Verlag, May 2003.

[GHKL08] S. Dov Gordon, Carmit Hazay, Jonathan Katz, and Yehuda Lindell. Complete fairness in secure two-party computation. In Richard E. Ladner and Cynthia Dwork, editors, *STOC*, pages 413–422. ACM, 2008.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game — a completeness theorem for protocols with honest majority. In *Proc. 19th ACM Symposium on the Theory of Computing (STOC)*, pages 218–229, 1987.

[Gol04] Oded Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, 2004.

[Kil91] Joe Kilian. A general completeness theorem for two-party games. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, STOC'91*, pages 553–560, New York, 1991. ACM Press.

[Kil00] Joe Kilian. More general completeness theorems for secure two-party computation. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, STOC'00*, pages 316–324, New York, 2000. ACM Press.

[KMQ08] Daniel Kraschewski and Jörn Müller-Quade. Completeness theorems with constructive proofs for symmetric, asymmetric and general 2-party-functions. Unpublished Manuscript, 2008. Online available at `http://iks.ira.uka.de/eiss/completeness`.

[Kus92] Eyal Kushilevitz. Privacy and communication complexity. *SIAM Journal on Discrete Mathematics*, 5(2):273–284, 1992.

[May97] D. Mayers. Unconditionally secure bit commitment is impossible. *Phys. Rev. Letters*, 78:3414–3417, 1997. A previous version was published at PhysComp96.

[MQ05] Jörn Müller-Quade. Temporary assumptions—quantum and classical. In *The 2005 IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security*, pages 31–33, 2005.

[MQU07] J. Müller-Quade and D. Unruh. Long-term security and universal composability. In *Theory of Cryptography, Proceedings of TCC 2004*, Lecture Notes in Computer Science. Springer-Verlag, 2007. to appear.

[Rab03] Michael O. Rabin. Hyper-encryption by virtual satellite. Science Center Research Lecture Series, December 2003. Online available at `http://athome.harvard.edu/dh/hvs.html`.

A Examples

$f^{(1)}$	0	1
0	0/0	0/0
1	0/0	1/0

$f^{(2)}$	0	1
0	0	1
1	0	2
2	3	2

$f^{(3)}$	0	1	2
0	0/0	1/1	1/0
1	0/0	2/2	2/0
2	3/3	2/2	2/0

$f^{(4)}$	0	1	2	3
0	1/1	1/1	2/2	2/0
1	4/4	5/5	2/2	2/0
2	4/4	3/3	3/3	3/0

$f^{(5)}$	0	1
0	0	0
1	0	1

$f^{(6)}$	0	1	2
0	1	1	2
1	4	5	2
2	4	3	3

$f^{(7)}$	4	2	0
3	4	3	3
1	4	2	1
0	4	2	0

$f^{(8)}$	0	1
0	0	1
1	0	2
2	3	2
3	3	1

$f^{(9)}$	z_1	z_2	z_3
x_1	$5/d$	$5/e$	$6/e$
x_2	$8/a$	$5/b$	$9/c$
x_3	$8/a$	$9/b$	$8/c$

Figure 1: Examples. Inputs for A are shown to the right, inputs for B on top. For asymmetric functions, outputs are denoted y_A/y_B; for symmetric functions only the common output of both parties is listed.

B Proof of Lemma 3.5

Before we give a proof of Lemma 3.5, we want to illustrate passive decomposability. Fig. 2 shows the function table of a restriction of f but only with the function outputs of $f_{\mathfrak{E}}$. The input domains are partitioned into three parts (namely the inputs of $\mathfrak{E}, \mathfrak{H} \setminus \{P_i\} = \mathfrak{H}'$ and P_i) and listed on the three axes. The definition says, that for any choice of $\mathfrak{E}$ and $x_{\mathfrak{E}}$, all the function outputs of $f_{\mathfrak{E}}$ in the densely dotted part have to be different from all the function values in the sparsely dotted part. As mentioned earlier, the intuition behind the above definition is that party P_i can now tell all the other parties if its input is in $\mathcal{X}_i'$ or in $\mathcal{X}_i''$. From this information, the other parties cannot learn anything more than they will know anyway, once they obtain their respective function outputs.

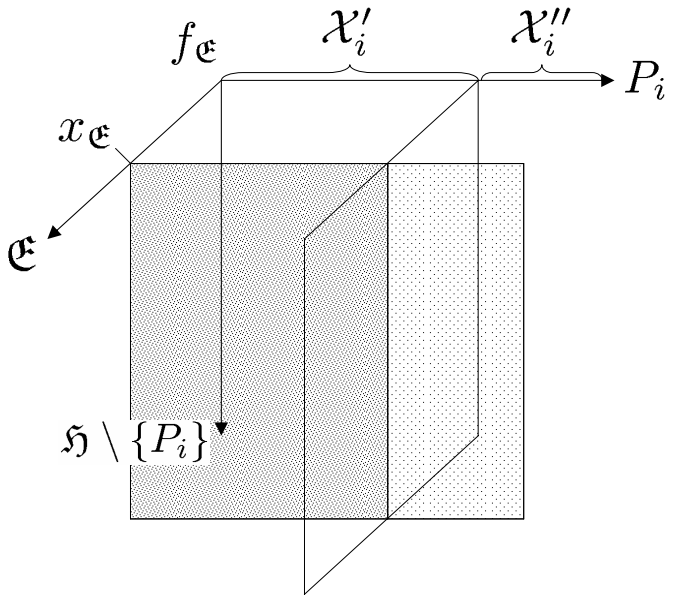

Figure 2: Illustration of Passive Decomposability

Proof. We prove the two implications separately. It is easy to see that Lemma 3.5 implies Definition 3.4. If (2.) holds for any set $\mathfrak{E}$ it also holds for sets $\mathfrak{E}$ where $|\mathfrak{E}| = 1$.

In order to prove the other direction, let f be passively decomposable according to Definition 3.4 and consider some restriction $\tilde{f}$ of f. If $\tilde{f}$ is locally computable, we are done because then (1.) of Lemma 3.5 holds. If (2.) of Definition 3.4 holds, we show by induction on the size of $\mathfrak{E}$ that indeed (2.) of Lemma 3.5 is true:
Base case: If we restrict to $|\mathfrak{E}| = 1$ in the lemma, the two characterizations are equivalent.
Induction hypothesis: Now assume that the lemma holds for sets $\mathfrak{E}$ where $|\mathfrak{E}| < m$ and $m < |\mathfrak{P}| - 1$.
Induction step: We show that the two characterizations are equivalent for any set $\mathfrak{E}_m$ where $|\mathfrak{E}_m| = m$. Let $\tilde{f} = f\,|_{\widetilde{\mathcal{X}}_1 \times \widetilde{\mathcal{X}}_2 \times \ldots \times \widetilde{\mathcal{X}}_n}$ be some restriction of f. Wlog (else rename the parties) let

$$
\begin{array}{lll}
\mathfrak{E}_m = \{P_1, P_2, \ldots, P_m\} & \mathfrak{E}_{m-1} = \{P_1, P_2, \ldots, P_{m-1}\} & \widetilde{\mathfrak{E}}_m = \{P_m\} \\
\mathfrak{H}'_m = \mathfrak{P} \setminus (\mathfrak{E}_m \cup \{P_i\}) & \mathfrak{H}'_{m-1} = \mathfrak{P} \setminus (\mathfrak{E}_{m-1} \cup \{P_i\}) & \widetilde{\mathfrak{H}}'_m = \mathfrak{P} \setminus (\widetilde{\mathfrak{E}}_m \cup \{P_i\}) \\
x_{\mathfrak{E}_m} \in \widetilde{\mathcal{X}}_{\mathfrak{E}_m} & x_{\mathfrak{E}_{m-1}} \in \widetilde{\mathcal{X}}_{\mathfrak{E}_{m-1}} & x_m \in \widetilde{\mathcal{X}}_{\widetilde{\mathfrak{E}}_m} = \mathcal{X}_m \\
x'_{\mathfrak{H}'_m} \in \widetilde{\mathcal{X}}_{\mathfrak{H}'_m} & x'_{\mathfrak{H}'_{m-1}} \in \widetilde{\mathcal{X}}_{\mathfrak{H}'_{m-1}} & x_{\widetilde{\mathfrak{H}}'_m} \in \widetilde{\mathcal{X}}_{\widetilde{\mathfrak{H}}'_m} \\
x''_{\mathfrak{H}'_m} \in \widetilde{\mathcal{X}}_{\mathfrak{H}'_m} & x''_{\mathfrak{H}'_{m-1}} \in \widetilde{\mathcal{X}}_{\mathfrak{H}'_{m-1}} & x''_{\widetilde{\mathfrak{H}}'_m} \in \widetilde{\mathcal{X}}_{\widetilde{\mathfrak{H}}'_m}
\end{array}
$$

Since $|\mathfrak{E}_{m-1}| = m-1 < m$ by induction hypothesis we know, that $\forall x_{\mathfrak{E}_{m-1}}, x'_{\mathfrak{H}'_{m-1}}, x''_{\mathfrak{H}'_{m-1}}$:

$$f_{\mathfrak{E}_{m-1}}(x_{\mathfrak{E}_{m-1}}, x'_{\mathfrak{H}'_{m-1}}, \mathcal{X}'_i) \cap f_{\mathfrak{E}_{m-1}}(x_{\mathfrak{E}_{m-1}}, x''_{\mathfrak{H}'_{m-1}}, \mathcal{X}''_i) = \emptyset$$

and also by induction hypothesis, since $|\widetilde{\mathfrak{E}}_m| = 1 < m$, we have $\forall x_m, x'_{\widetilde{\mathfrak{H}}'_m}, x''_{\widetilde{\mathfrak{H}}'_m}$:

$$f_m(x_m, x'_{\widetilde{\mathfrak{H}}'_m}, \mathcal{X}'_i) \cap f_m(x_m, x''_{\widetilde{\mathfrak{H}}'_m}, \mathcal{X}''_i) = \emptyset$$

The above two statements tell us that for the parties in $\mathfrak{E}_{m-1}$ and the party P_m the intersection is empty for any choice of input values of the parties in $\mathfrak{H}'_{m-1}$ and $\widetilde{\mathfrak{H}}'_m$ respectively. But now, since $\mathfrak{E}_m = \mathfrak{E}_{m-1} \cup \widetilde{\mathfrak{E}}_m$, $\mathfrak{H}'_m \subseteq \mathfrak{H}'_{m-1}$ and $\mathfrak{H}'_m \subseteq \widetilde{\mathfrak{H}}'_m$ this implies that $\forall x_{\mathfrak{E}_m}, x'_{\mathfrak{H}'_m}, x''_{\mathfrak{H}'_m}$:

$$f_{\mathfrak{E}_m}(x_{\mathfrak{E}_m}, x'_{\mathfrak{H}'_m}, \mathcal{X}'_i) \cap f_{\mathfrak{E}_m}(x_{\mathfrak{E}_m}, x''_{\mathfrak{H}'_m}, \mathcal{X}''_i) = \emptyset$$

which concludes the argument. □

C Proof of Theorem 3.6

By Lem. 2.3 we have $\mathfrak{F}^{\mathsf{aut}}_{\mathsf{pas}} = \mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}}$, so it suffices to show that $\mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}} = \mathfrak{F}'_{\mathsf{pas}}$ This is simplifies matters because in the public discussion setting we have a global transcript which we can refer to. We then first prove $\mathfrak{F}'_{\mathsf{pas}} \subseteq \mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}}$ by inductively describing a passively PFE secure protocol to compute any function in $\mathfrak{F}'_{\mathsf{pas}}$. Since the protocol is PFE secure, this implies that it is also IT secure. In the second part of the proof we show that $\mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}} \subseteq \mathfrak{F}'_{\mathsf{pas}}$, i.e. that any passively computable function has the structure described in Def. 3.4.

$f \in \mathfrak{F}'_{\mathsf{pas}} \implies f \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}}$**:** We prove the claim by induction over the size of the input space $|\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n|$ of the function f.
Base case: If the function f is locally computable ($f \in \mathfrak{F}_{\mathsf{loc}}$), we trivially have $f \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}}$ since all the parties can compute the function locally.
Induction hypothesis: We assume that for any $f' \in \mathfrak{F}'_{\mathsf{pas}}$ with input space smaller than $|\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n|$ we have $f' \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}}$.
Induction step: Since $f \in \mathfrak{F}'_{\mathsf{pas}}$ we have two cases: Either $f \in \mathfrak{F}_{\mathsf{loc}}$ (this is the base case and we are done) or f has a K-cut according to the definition of $\mathfrak{F}'_{\mathsf{pas}}$. This is the case we consider now.

Wlog we can assume (else interchange the parties) that the set $\mathcal{X}_1$ has K-cut $\mathcal{X}_1^{(1)} \dot\cup \mathcal{X}_1^{(2)} = \mathcal{X}_1$ (We choose the subset $\widetilde{\mathcal{X}}_1$ from the definition of $\mathfrak{F}'_{\mathsf{pas}}$ to be $\mathcal{X}_1$). Now we define

$$f^{(1)} := f\mid_{\mathcal{X}_1^{(1)} \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n}$$

$$f^{(2)} := f\mid_{\mathcal{X}_1^{(2)} \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n}$$

By the induction hypothesis, $f^{(1)}, f^{(2)} \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}}$ since $f^{(1)}, f^{(2)} \in \mathfrak{F}'_{\mathsf{pas}}$ and

$$|\mathcal{X}_1^{(1)} \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n| < |\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n|$$
$$|\mathcal{X}_1^{(2)} \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n| < |\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n|.$$

So there are protocols $\pi^{(1)}$ and $\pi^{(2)}$ that PFE securely implement $I_{f^{(1)}}$ and $I_{f^{(2)}}$ and there exist corresponding simulators $\mathsf{S}^{(1)}$ and $\mathsf{S}^{(2)}$ in $\mathcal{S}_{\mathsf{pas}}$. We construct the protocol π by defining a first round where P_1 sends a message $m_1 \in \{1,2\}$ to all other parties indicating whether P_1's input x_1 is $x_1 \in \mathcal{X}_1^{(1)}$ or $x_1 \in \mathcal{X}_1^{(2)}$. Subsequently all parties proceed to run $\pi^{(m_1)}$. To prove that the protocol π is PFE secure we provide an appropriate simulator $\mathsf{S} \in \mathcal{S}_{\mathsf{pas}}$ for the case where an arbitrary subset $\mathfrak{E}$ of the parties $\mathfrak{P}$ is corrupted by the adversary $\mathsf{E} \in \mathcal{E}_{\mathsf{pas}}$. It suffices to look at two cases. Either P_1 is in the set of corrupted parties ($P_1 \in \mathfrak{E}$), or it is not ($P_1 \notin \mathfrak{E}$).

If $P_1 \notin \mathfrak{E}$, we have $\mathfrak{E} = \{P_{e_1}, P_{e_2}, \ldots, P_{e_{|\mathfrak{E}|}}\}$ where $e_i \in \{2,3,\ldots,n\}$ and we assume $\mathfrak{E} \neq \emptyset$ (If $\mathfrak{E} = \emptyset$, no party is corrupted and security is trivially achieved). Now we construct the simulator $\mathsf{S}_{P_1 \notin \mathfrak{E}}$ as follows:

1. $\mathsf{S}_{P_1 \notin \mathfrak{E}}$ fixes the randomness of the adversary E.
2. It feeds the input $x_{\mathsf{E}} = (x_{\mathfrak{E}}, x'_{\mathsf{E}})$ of the distinguisher[22] D to E.
3. It extracts the input $x_{\mathfrak{E}} = (x_{e_1}, x_{e_2} \ldots, x_{e_{|\mathfrak{E}|}})$ from x_{E} and forwards it to the ideal system I_f.
4. After receiving the output $y_{\mathfrak{E}} = (y_{e_1}, y_{e_2} \ldots, y_{e_{|\mathfrak{E}|}})$ from I_f with $\mathfrak{H}' = \mathfrak{H} \setminus \{P_1\}$ and:

   $$y_{\mathfrak{E}} = f_{\mathfrak{E}}(x_{\mathfrak{E}}, x_{\mathfrak{H}'}, x_1),$$

   $\mathsf{S}_{P_1 \notin \mathfrak{E}}$ computes m_1 such that $y_{\mathfrak{E}} \in f_{\mathfrak{E}}(x_{\mathfrak{E}}, x_{\mathfrak{H}'}, \mathcal{X}_1^{(m_1)})$.
   Note that this is always possible since $f \in \mathfrak{F}'_{\mathsf{pas}}$ implies (using Lemma 3.5) in particular that for all $\mathfrak{E} \subseteq \mathfrak{P}$ and for any $x'_{\mathfrak{H}'}, x''_{\mathfrak{H}'}$ we have

   $$f_{\mathfrak{E}}(x_{\mathfrak{E}}, x'_{\mathfrak{H}'}, \mathcal{X}_1^{(1)}) \cap f_{\mathfrak{E}}(x_{\mathfrak{E}}, x''_{\mathfrak{H}'}, \mathcal{X}_1^{(2)}) = \emptyset.$$

5. Finally $\mathsf{S}_{P_1 \notin \mathfrak{E}}$ inputs m_1 to E and runs $\mathsf{S}^{(m_1)}(\mathsf{E})$ forwarding all inputs and outputs.

The correctness of the the simulation stems from the fact that the simulator $\mathsf{S}_{P_1 \notin \mathfrak{E}}$ can correctly discern the first message m_1 given the output $y_{\mathfrak{E}}$ of I_f. After simulating this message, the remainder of the simulation can be delegated to the subsimulator $\mathsf{S}^{(m_1)}$ that is guaranteed by our induction hypothesis.

It remains to look at the case where $P_1 \in \mathfrak{E}$. We then have $\mathfrak{E} = \{P_1, P_{e_1}, P_{e_2}, \ldots, P_{e_{|\mathfrak{E}\setminus\{1\}|}}\}$ where $e_i \in \{2, 3, \ldots, n\}$. The simulator $\mathsf{S}_{P_1 \in \mathfrak{E}}$ can be constructed as follows:

1. $\mathsf{S}_{P_1 \in \mathfrak{E}}$ fixes the randomness of the adversary E.
2. It feeds the input x_{E} of the distinguisher D to the adversary E and runs it until it outputs the message m_1.
3. $\mathsf{S}_{P_1 \in \mathfrak{E}}$ then runs the simulator $\mathsf{S}^{(m_1)}$ simply forwarding all inputs and outputs.

The simulation is correct since $\mathsf{S}^{(m_1)}$ (which is given by our induction hypothesis) now correctly simulates the execution with the honest parties that execute $\pi^{(m_1)}$.

This completes the first part of the proof.

$f \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}} \implies f \in \mathfrak{F}'_{\mathsf{pas}}$**:** We give a proof by contradiction. Assume a counterexample $f \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}}$ that is minimal in the size of the input space $|\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n|$, such that $f \notin \mathfrak{F}'_{\mathsf{pas}}$. Our goal is to show that such a counterexample does not exist.

Since $f \notin \mathfrak{F}'_{\mathsf{pas}}$, there is a restriction $\tilde{f} := f\,|_{\widetilde{\mathcal{X}}_1 \times \widetilde{\mathcal{X}}_2 \times \ldots \times \widetilde{\mathcal{X}}_n}$ of f such that

1. $\tilde{f}$ is **not** locally computable ($\tilde{f} \notin \mathfrak{F}_{\mathsf{loc}}$) **and**
2. $\forall i$: $\forall$ partitions of $\widetilde{\mathcal{X}}_i$ into non-empty sets $\mathcal{X}'_i \dot\cup \mathcal{X}''_i = \widetilde{\mathcal{X}}_i$
   $\exists P_e \in \mathfrak{P} \setminus \{P_i\}$ (where we define $\mathfrak{E} = \{P_e\}$ and thus $|\mathfrak{E}| = 1$)
   $\exists x_e \in \widetilde{\mathcal{X}}_e$ such that:

   $$\tilde{f}_e(x_e, \widetilde{\mathcal{X}}_{\mathfrak{H}'}, \mathcal{X}'_i) \cap \tilde{f}_e(x_e, \widetilde{\mathcal{X}}_{\mathfrak{H}'}, \mathcal{X}''_i) \neq \emptyset$$

   where $\mathfrak{H} = \mathfrak{P} \setminus \mathfrak{E}$ and $\mathfrak{H}' := \mathfrak{H} \setminus \{P_i\}$.

Claim C.1. *Due to the minimality of f, the above two facts (1.) and (2.) hold for the function $\tilde{f} = f$ itself, i.e. f is not locally computable and does not have a K-Cut.*

[22] x'_{E} stands for additional information that is passed along with the inputs by the distinguisher D and $x_{\mathfrak{E}}$ are the actual function inputs that the passive adversary E must use.

To see this, we observe that as $f \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}}$, so is any restriction $\tilde{f} := f\mid_{\widetilde{\mathcal{X}}_1 \times \widetilde{\mathcal{X}}_2 \times \ldots \times \widetilde{\mathcal{X}}_n}$ of f to subsets $\widetilde{\mathcal{X}}_1 \subseteq \mathcal{X}_1, \widetilde{\mathcal{X}}_2 \subseteq \mathcal{X}_2, \ldots, \widetilde{\mathcal{X}}_n \subseteq \mathcal{X}_n$:

$$f \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}} \implies \tilde{f} \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}} \tag{1}$$

Indeed this can be seen by using exactly the same protocol and simulator as for f, merely restricting the input domain. Now assume (1.) and (2.) both hold for a proper restriction $\tilde{f}$ (This means that $\tilde{f}$ is not locally computable and does not have a K-Cut). Therefore we know $\tilde{f} \notin \mathfrak{F}'_{\mathsf{pas}}$, which, using Equation (1) and the fact that $\tilde{f}$ is a proper restriction, contradicts the minimality of f. Therefore we know that (1.) and (2.) do not both hold in any proper restriction of f and thus the function $f = f\mid_{\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n}$ must itself satisfy both (1.) and (2.). This proves Claim C.1.

Now as $f \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{pas}}$ we have an efficient passively IT secure protocol π computing f. Then the number of rounds of protocol π is bounded by a polynomial $p_\pi(\kappa)$ in the security parameter κ. So we can wlog "pad" all protocol executions with dummy messages to have the same length $p_\pi(\kappa)$. We then define the set of transcripts

$$\begin{aligned}
\Pi := \pi(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n) &= \{t = m_1, m_2, \ldots, m_{p_\pi(\kappa)} \mid \\
&\qquad \exists x_1, c_1, x_2, c_2, \ldots, x_n, c_n : t = \pi(x_1, c_1, x_2, c_2, \ldots, x_n, c_n)\} \\
\Pi_r := \pi(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n)|_r &= \{t = m_1, m_2, \ldots, m_r \mid \\
&\qquad \exists m_{r+1}, \ldots, m_{p_\pi(\kappa)} : t, m_{r+1}, \ldots, m_{p_\pi(\kappa)} \in \Pi\} \\
&= \{t = m_1, m_2, \ldots, m_{p_r} \mid \\
&\qquad \exists x_1, c_1, x_2, c_2, \ldots, x_n, c_n : t = \pi(x_1, c_1, x_2, c_2, \ldots, x_n, c_n)|_r\}
\end{aligned}$$

where x_i denotes the input and c_i the random coin tosses of party P_i.

We define random variables $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$ for the input and output of the parties $P_1, \ldots, P_n$ respectively. For a set $M \subseteq \mathfrak{P}$ we define X_M and Y_M to be the random variable for the inputs and outputs of the parties in M. Then we let $T \in \Pi$ denote the random variable for the transcript, and $T_r \in \Pi_r$ the random variable on transcript prefixes of length r. We name the inputs $\mathcal{X}_l = \{x_l^{(1)}, x_l^{(2)}, \ldots, x_l^{(|\mathcal{X}_l|)}\}$ for party $P_l \in \mathfrak{P}$ and similarly $\mathcal{X}_M = \{x_M^{(1)}, x_M^{(2)}, \ldots, x_M^{(|\mathcal{X}_M|)}\}$ for a subset M of the parties $\mathfrak{P}$. We first show that for any $x_1^{(i_1)} \in \mathcal{X}_1, x_2^{(i_2)} \in \mathcal{X}_2, \ldots, x_n^{(i_n)} \in \mathcal{X}_n$ the statistical distance

$$\Delta(\mathrm{Pr}_{T|X_1,X_2,\ldots,X_n}(\cdot, x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}), \mathrm{Pr}_{T|X_1,X_2,\ldots,X_n}(\cdot, x_1^{(i_1)}, x_2^{(i_2)}, \ldots, x_n^{(i_n)})) < \mathsf{negl} \tag{2}$$

of these two distributions on the transcripts $t \in \Pi$ is negligible in the security parameter κ. We proceed by induction over the number of protocol rounds r and show that for any r

$$\delta_r := \max_{i_1, i_2, \ldots, i_n} \Delta(\mathrm{Pr}_{T_r|X_1,X_2,\ldots,X_n}(\cdot, x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}), \mathrm{Pr}_{T_r|X_1,X_2,\ldots,X_n}(\cdot, x_1^{(i_1)}, x_2^{(i_2)}, \ldots, x_n^{(i_n)})) < \mathsf{negl}. \tag{3}$$

Base case: For r=0 we have $\Pi_0 = \{\epsilon\}$ where ϵ is the empty transcript. Thus $\delta_0 = 0$.
Induction hypothesis: We assume $\delta_{r-1} < \mathsf{negl}$.
Induction step: Let $t_r = (m_1, m_2, \ldots, m_r) \in \Pi_r$ and define $t_{r-1} = (m_1, m_2, \ldots, m_{r-1}) \in \Pi_{r-1}$. Wlog (else rename the parties) the message in round r is sent from party P_1 to all other parties.

Recall the definition of $f \notin \mathfrak{F}'_{\mathsf{pas}}$ from the beginning of the proof. As we have seen in Claim C.1, f does not have a K-Cut. Using this, we can impose an ordering on the elements of $\mathcal{X}_1$ such that

$\forall x_1^{(j)}$ where $1 < j \leq |\mathcal{X}_1|$
$\exists x_1^{(k)}$ where $1 \leq k < j$
$\exists \mathfrak{E} = \{P_e\}$ where $P_e \in \{P_2, P_3, \ldots, P_n\}$
$\exists x_e^{(jk)} \in \mathcal{X}_{\mathfrak{E}} = \mathcal{X}_e$
$\exists \bar{x}_{\mathfrak{H}'}^{(jk)}, \bar{\bar{x}}_{\mathfrak{H}'}^{(jk)} \in \mathcal{X}_{\mathfrak{H}'}$ such that:

$$f_e(x_e^{(jk)}, \bar{x}_{\mathfrak{H}'}^{(jk)}, x_1^{(k)}) = f_e(x_e^{(jk)}, \bar{\bar{x}}_{\mathfrak{H}'}^{(jk)}, x_1^{(j)}) \tag{4}$$

Where we define $\mathfrak{H} = \mathfrak{P} \setminus \mathfrak{E}$ and $\mathfrak{H}' = \mathfrak{H} \setminus \{P_1\}$.

Now by the security of the passively IT secure protocol π we must have for any r that

$$\Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(jk)}, \bar{x}_{\mathfrak{H}'}^{(jk)}, x_1^{(k)}), \mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(jk)}, \bar{\bar{x}}_{\mathfrak{H}'}^{(jk)}, x_1^{(j)})) < \mathsf{negl} \tag{5}$$

This is because a simulator S_e for the corrupted party P_e sees exactly the same (see (4)) for the above inputs and is thus unable to emulate two *non-negligibly different* distributions of transcripts.

Now, since m_r travels from P_1 to all other parties, we find:

$$\begin{aligned}
&\Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(1)}, x_{\mathfrak{H}'}^{(1)}, x_1^{(k)}), \mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(jk)}, \bar{x}_{\mathfrak{H}'}^{(jk)}, x_1^{(k)})) && (6)\\
&= \sum_{t_r \in \Pi_r} |\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(t_r, x_e^{(1)}, x_{\mathfrak{H}'}^{(1)}, x_1^{(k)}) - \mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(t_r, x_e^{(jk)}, \bar{x}_{\mathfrak{H}'}^{(jk)}, x_1^{(k)})| && (7)\\
&= \sum_{t_r \in \Pi_r} |\mathrm{Pr}_{M_r|T_{r-1},X_1}(m_r, t_{r-1}, x_1^{(k)}) \cdot \mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1}, x_e^{(1)}, x_{\mathfrak{H}'}^{(1)}, x_1^{(k)}) && (8)\\
&\quad - \mathrm{Pr}_{M_r|T_{r-1},X_1}(m_r, t_{r-1}, x_1^{(k)}) \cdot \mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1}, x_e^{(jk)}, \bar{x}_{\mathfrak{H}'}^{(jk)}, x_1^{(k)})| && (9)\\
&= \sum_{t_{r-1} \in \Pi_{r-1}} \underbrace{\left(\sum_{m_r} \mathrm{Pr}_{M_r|T_{r-1},X_1}(m_r, t_{r-1}, x_1^{(k)}) \right)}_{=1} && (10)\\
&\quad |\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1}, x_e^{(1)}, x_{\mathfrak{H}'}^{(1)}, x_1^{(k)}) - \mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1}, x_e^{(jk)}, \bar{x}_{\mathfrak{H}'}^{(jk)}, x_1^{(k)})| && (11)\\
&= \Delta(\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1}, x_e^{(1)}, x_{\mathfrak{H}'}^{(1)}, x_1^{(k)}), \mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1}, x_e^{(jk)}, \bar{x}_{\mathfrak{H}'}^{(jk)}, x_1^{(k)})) && (12)\\
&\leq \Delta(\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1}, x_e^{(1)}, x_{\mathfrak{H}'}^{(1)}, x_1^{(1)}), \mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1}, x_e^{(1)}, x_{\mathfrak{H}'}^{(1)}, x_1^{(k)})) && (13)\\
&\quad + \Delta(\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1}, x_e^{(1)}, x_{\mathfrak{H}'}^{(1)}, x_1^{(1)}), \mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1}, x_e^{(jk)}, \bar{x}_{\mathfrak{H}'}^{(jk)}, x_1^{(k)})) && (14)\\
&= 2 \cdot \delta_{r-1} < \mathsf{negl}, && (15)
\end{aligned}$$

Now let $\tilde{x}_e \in \mathcal{X}_e$ and $\tilde{x}_{\mathfrak{H}'} \in \mathcal{X}_{\mathfrak{H}'}$ be arbitrary elements. In the same way as we derived Equation (15) we find:

$$\begin{aligned}
&\Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(jk)}, \bar{\bar{x}}_{\mathfrak{H}'}^{(jk)}, x_1^{(j)}), \mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, \tilde{x}_e, \tilde{x}_{\mathfrak{H}'}, x_1^{(j)})) && (16)\\
&\quad = 2 \cdot \delta_{r-1} < \mathsf{negl}, && (17)
\end{aligned}$$

so we find, using (5), (6) and (16), that

$$\begin{aligned}
&\Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(1)}, x_{\mathfrak{H}'}^{(1)}, x_1^{(k)}), \mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, \tilde{x}_e, \tilde{x}_{\mathfrak{H}'}, x_1^{(j)})) && (18)\\
&\quad \leq \Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(1)}, x_{\mathfrak{H}'}^{(1)}, x_1^{(k)}), \mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(jk)}, \bar{x}_{\mathfrak{H}'}^{(jk)}, x_1^{(k)})) && (19)\\
&\qquad + \Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(jk)}, \bar{x}_{\mathfrak{H}'}^{(jk)}, x_1^{(k)}), \mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(jk)}, \bar{\bar{x}}_{\mathfrak{H}'}^{(jk)}, x_1^{(j)})) && (20)\\
&\qquad + \Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(jk)}, \bar{\bar{x}}_{\mathfrak{H}'}^{(jk)}, x_1^{(j)}), \mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, \tilde{x}_e, \tilde{x}_{\mathfrak{H}'}, x_1^{(j)})) && (21)\\
&\quad \leq \mathsf{negl} + \mathsf{negl} + \mathsf{negl} && (22)
\end{aligned}$$

This is not yet what we claimed in (3) because we would like to obtain the above result (22) for $k = 1$. We can first observe that the choice of $\tilde{x}_e$ and $\tilde{x}_{\mathfrak{H}'}$ and j is arbitrary for the above argument, but $\mathfrak{E} = \{P_e\}$ is fixed for given j. To finally obtain what we claimed in (3) we use induction on j. The idea is to apply the result (22) step by step:

- To get from j to $k_1 < j$ there exists a set of corrupted parties $\mathfrak{E}_1 = P_{e_1}$ such that the statistical distance in (22) is negligible.
- To get from k_1 to $k_2 < k_1$ there exists a set of corrupted parties $\mathfrak{E}_2 = P_{e_2}$ such that the statistical distance in (22) is negligible.

- ...
- To get from k_i to $1 < k_i$ there exists a set of corrupted parties $\mathfrak{E}_i = P_{e_i}$ such that the statistical distance in (22) is negligible.

We claim that the sum of all these negligible distances is again negligible. This only holds if the length i of the induction chains is at most polynomial in the security parameter κ. But this is given, since we require the round number $p_\pi(\kappa)$ to be polynomially bounded and as for a given fixed function f the sizes of the input domains $|\mathcal{X}_1|, |\mathcal{X}_2|, \ldots, |\mathcal{X}_n|$ is of course constant.

This concludes our inductive argument and we proved the claim (3).

We know by (1.) from the beginning of the proof and the minimality of f that f is in particular not locally computable. Hence wlog there is a party P_i such that

$\exists x_i \in \widetilde{\mathcal{X}}_i$
$\exists x'_{\mathfrak{P}'}, x''_{\mathfrak{P}'} \in \widetilde{\mathcal{X}}_{\mathfrak{P}'}$ where $\mathfrak{P}' = \mathfrak{P} \setminus \{P_i\}$ such that

$$f_i(x_i, x'_{\mathfrak{P}'}) \neq f_i(x_i, x''_{\mathfrak{P}'}) \tag{23}$$

From the above result (2) and an application of the triangle inequality we obtain:

$$\Delta(\mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(\cdot, x_i, x'_{\mathfrak{P}'}), \mathrm{Pr}_{T|X_A,X_B}(\cdot, x_i, x''_{\mathfrak{P}'})) < \mathsf{negl}. \tag{24}$$

As the output of P_i is independent of the inputs of $\mathfrak{P}'$ given the transcript, we have:

$$\mathrm{Pr}_{Y_i|T,X_i,X_{\mathfrak{P}'}} = \mathrm{Pr}_{Y_i|T,X_i} \tag{25}$$

And so we find:

$$\begin{aligned}
&\Delta(\mathrm{Pr}_{Y_i|X_i,X_{\mathfrak{P}'}}(\cdot, x_i, x'_{\mathfrak{P}'}), \mathrm{Pr}_{Y_i|X_i,X_{\mathfrak{P}'}}(\cdot, x_i, x''_{\mathfrak{P}'})) && (26)\\
&= \sum_{y_i \in \mathcal{Y}_i} |\mathrm{Pr}_{Y_i|X_i,X_{\mathfrak{P}'}}(y_i, x_i, x'_{\mathfrak{P}'}) - \mathrm{Pr}_{Y_i|X_i,X_{\mathfrak{P}'}}(y_i, x_i, x''_{\mathfrak{P}'})| && (27)\\
&= \sum_{y_i \in \mathcal{Y}_i} |\sum_{t \in \Pi} \mathrm{Pr}_{Y_i,T|X_i,X_{\mathfrak{P}'}}(y_i, t, x_i, x'_{\mathfrak{P}'}) - \sum_{t \in \Pi} \mathrm{Pr}_{Y_i,T|X_i,X_{\mathfrak{P}'}}(y_i, t, x_i, x''_{\mathfrak{P}'})| && (28)\\
&\leq \sum_{y_i \in \mathcal{Y}_i, t \in \Pi} |\mathrm{Pr}_{Y_i|T,X_i,X_{\mathfrak{P}'}}(y_i, t, x_i, x'_{\mathfrak{P}'}) \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x'_{\mathfrak{P}'}) && (29)\\
&\qquad - \mathrm{Pr}_{Y_i|T,X_i,X_{\mathfrak{P}'}}(y_i, t, x_i, x''_{\mathfrak{P}'}) \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x''_{\mathfrak{P}'})|\\
&\overset{(25)}{=} \sum_{y_i \in \mathcal{Y}_i, t \in \Pi} |\mathrm{Pr}_{Y_i|T,X_i}(y_i, t, x_i) \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x'_{\mathfrak{P}'}) - \mathrm{Pr}_{Y_i|T,X_i}(y_i, t, x_i) \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x''_{\mathfrak{P}'})| && (30)\\
&= \sum_{y_i \in \mathcal{Y}_i, t \in \Pi} \mathrm{Pr}_{Y_i|T,X_i}(y_i, t, x_i) \, |\mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x'_{\mathfrak{P}'}) - \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x''_{\mathfrak{P}'})| && (31)\\
&= \sum_{t \in \Pi} \underbrace{\left(\sum_{y_i \in \mathcal{Y}_i} \mathrm{Pr}_{Y_i|T,X_i}(y_i, t, x_i) \right)}_{=1} |\mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x'_{\mathfrak{P}'}) - \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x''_{\mathfrak{P}'})| && (32)\\
&= \Delta(\mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(\cdot, x_i, x'_{\mathfrak{P}'}), \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(\cdot, x_i, x''_{\mathfrak{P}'})) \overset{(24)}{<} \mathsf{negl}. && (33)
\end{aligned}$$

This is in obvious contradiction to the correctness of the protocol π, so we must have $f \in \mathfrak{F}'_{\mathsf{pas}}$. Hence there is no counterexample and the claim is proven.

D Proof of Lemma 4.3

For any $x_{\mathfrak{E}} = (x_{e_1}, \ldots, x_{e_{|\mathfrak{E}|}})$, $\hat{x}_{\mathfrak{E}} = (\hat{x}_{e_1}, \ldots, \hat{x}_{e_{|\mathfrak{E}|}})$ where each $\hat{x}_{e_j}$ dominates x_{e_j} ($j \in [|\mathfrak{E}|]$), we have to show that:

1. for all $x_{\mathfrak{H}} \in \mathcal{X}_{\mathfrak{H}}$: $f_{\mathfrak{H}}(\hat{x}_{\mathfrak{E}}, x_{\mathfrak{H}}) = f_{\mathfrak{H}}(x_{\mathfrak{E}}, x_{\mathfrak{H}})$ and
2. for all $x_{\mathfrak{H}}, x'_{\mathfrak{H}} \in \mathcal{X}_{\mathfrak{H}}$: $f_{\mathfrak{E}}(x_{\mathfrak{E}}, x_{\mathfrak{H}}) \neq f_{\mathfrak{E}}(x_{\mathfrak{E}}, x'_{\mathfrak{H}}) \implies f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, x_{\mathfrak{H}}) \neq f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, x'_{\mathfrak{H}})$.

Each $\hat{x}_{e_j}$ dominates x_{e_j}, so we know the following:

(i) for all $x_{\mathfrak{P}'} \in \mathcal{X}_{\mathfrak{P}'}$: $f_{\mathfrak{P}'}(\hat{x}_{e_j}, x_{\mathfrak{P}'}) = f_{\mathfrak{P}'}(x_{e_j}, x_{\mathfrak{P}'})$ and

(ii) for all $x_{\mathfrak{P}'}, x'_{\mathfrak{P}'} \in \mathcal{X}_{\mathfrak{P}'}$: $f_{e_j}(x_{e_j}, x_{\mathfrak{P}'}) \neq f_{e_j}(x_{e_j}, x'_{\mathfrak{P}'}) \implies f_{e_j}(\hat{x}_{e_j}, x_{\mathfrak{P}'}) \neq f_{e_j}(\hat{x}_{e_j}, x'_{\mathfrak{P}'})$

where $\mathfrak{P}' = \mathfrak{P} \setminus \{P_{e_j}\}$.

To prove the first part (1.) of the lemma, we use induction on $|\mathfrak{E}|$.
Base case: We have $|\mathfrak{E}| = 1$ and the claim is exactly what is given by (i) in the above statement.
Induction hypothesis: Assume the claim (1.) holds for $|\mathfrak{E}| = m - 1$.
Induction step: We show that the claim holds for $|\mathfrak{E}| = m$. By our induction hypothesis we have

$$\forall x_{\mathfrak{H}} \in \mathcal{X}_{\mathfrak{H}} : f_{\mathfrak{H}}(\hat{x}_{\mathfrak{E}}, x_{\mathfrak{H}}) = f_{\mathfrak{H}}(x_{\mathfrak{E}}, x_{\mathfrak{H}}) \tag{34}$$

for $x_{\mathfrak{E}} = (x_{e_1}, x_{e_2}, \ldots, x_{e_{m-1}})$. Now we add the corrupted party P_{e_m} and let $\mathfrak{H}' = \mathfrak{H} \setminus \{P_{e_m}\}$. From the above we know in particular:

$$\forall x_{\mathfrak{H}'} \in \mathcal{X}_{\mathfrak{H}'} : f_{\mathfrak{H}}(\hat{x}_{\mathfrak{E}}, x_{e_m}, x_{\mathfrak{H}'}) = f_{\mathfrak{H}}(x_{\mathfrak{E}}, x_{e_m}, x_{\mathfrak{H}'}) \tag{35}$$

But now by (i) for x_{e_m} we can conclude that

$$\forall x_{\mathfrak{H}'} \in \mathcal{X}_{\mathfrak{H}'} : f_{\mathfrak{H}'}(\hat{x}_{\mathfrak{E}}, \hat{x}_{e_m}, x_{\mathfrak{H}'}) \overset{\text{(i)}}{=} f_{\mathfrak{H}'}(\hat{x}_{\mathfrak{E}}, x_{e_m}, x_{\mathfrak{H}'}) \overset{(35)}{=} f_{\mathfrak{H}'}(x_{\mathfrak{E}}, x_{e_m}, x_{\mathfrak{H}'}) \tag{36}$$

which is exactly what we wanted to show.

To prove the second part (2.), we again use induction on $|\mathfrak{E}|$.
Base case: We have $|\mathfrak{E}| = 1$ and the claim is exactly what is given by (ii) in the above statement.
Induction hypothesis: Assume the claim (2.) holds for $|\mathfrak{E}| = m - 1$.
Induction step: We show that the claim holds for $|\mathfrak{E}| = m$. By our induction hypothesis we have

$$\forall x_{\mathfrak{H}}, x'_{\mathfrak{H}} \in \mathcal{X}_{\mathfrak{H}} : f_{\mathfrak{E}}(x_{\mathfrak{E}}, x_{\mathfrak{H}}) \neq f_{\mathfrak{E}}(x_{\mathfrak{E}}, x'_{\mathfrak{H}}) \implies f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, x_{\mathfrak{H}}) \neq f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, x'_{\mathfrak{H}}) \tag{37}$$

for $x_{\mathfrak{E}} = (x_{e_1}, x_{e_2}, \ldots, x_{e_{m-1}})$. Now we add the corrupted party P_{e_m} and let $\mathfrak{H}' = \mathfrak{H} \setminus \{P_{e_m}\}$. Then we can see the following:

$$\begin{aligned} \forall x_{\mathfrak{H}'}, x'_{\mathfrak{H}'} \in \mathcal{X}_{\mathfrak{H}'} : & \\ & f_{\mathfrak{E}}(x_{\mathfrak{E}}, x_{e_m}, x_{\mathfrak{H}'}) \neq f_{\mathfrak{E}}(x_{\mathfrak{E}}, x_{e_m}, x'_{\mathfrak{H}'}) \\ \implies & f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, x_{e_m}, x_{\mathfrak{H}'}) \neq f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, x_{e_m}, x'_{\mathfrak{H}'}) \\ \implies & f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, \hat{x}_{e_m}, x_{\mathfrak{H}'}) \neq f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, \hat{x}_{e_m}, x'_{\mathfrak{H}'}) \end{aligned} \tag{38}$$

We get the first implication from Equation (37) and the second implication becomes clear from the following observation: By (i) for x_{e_m} we have:

$$\begin{aligned} f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, x_{e_m}, x_{\mathfrak{H}'}) &= f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, \hat{x}_{e_m}, x_{\mathfrak{H}'}) \\ f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, x_{e_m}, x'_{\mathfrak{H}'}) &= f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, \hat{x}_{e_m}, x'_{\mathfrak{H}'}) \end{aligned}$$

Now on the other hand we find:

$$\begin{aligned}\forall x_{\mathfrak{H}'}, x'_{\mathfrak{H}'} \in \mathcal{X}_{\mathfrak{H}'} :& \\ & f_{e_m}(x_{\mathfrak{E}}, x_{e_m}, x_{\mathfrak{H}'}) \neq f_{e_m}(x_{\mathfrak{E}}, x_{e_m}, x'_{\mathfrak{H}'}) \\ \implies & f_{e_m}(x_{\mathfrak{E}}, \hat{x}_{e_m}, x_{\mathfrak{H}'}) \neq f_{e_m}(x_{\mathfrak{E}}, \hat{x}_{e_m}, x'_{\mathfrak{H}'}) \\ \implies & f_{e_m}(\hat{x}_{\mathfrak{E}}, \hat{x}_{e_m}, x_{\mathfrak{H}'}) \neq f_{e_m}(\hat{x}_{\mathfrak{E}}, \hat{x}_{e_m}, x'_{\mathfrak{H}'}) \end{aligned} \tag{39}$$

Where the first implication is a special case of property (1.) of the definition of $\hat{x}_{e_m}$ and the second implication is given by the definition of $\hat{x}_{e_1}, \hat{x}_{e_2}, \ldots, \hat{x}_{e_{m-1}}$.
Now we can conclude from Equations (38) and (39) that for $\mathfrak{E}' = \mathfrak{E} \cup \{P_{e_m}\}$

$$\forall x_{\mathfrak{H}'}, x'_{\mathfrak{H}'} \in \mathcal{X}_{\mathfrak{H}'} : f_{\mathfrak{E}'}(x_{\mathfrak{E}'}, x_{\mathfrak{H}'}) \neq f_{\mathfrak{E}'}(x_{\mathfrak{E}'}, x'_{\mathfrak{H}'}) \implies f_{\mathfrak{E}'}(\hat{x}_{\mathfrak{E}'}, x_{\mathfrak{H}'}) \neq f_{\mathfrak{E}'}(\hat{x}_{\mathfrak{E}'}, x'_{\mathfrak{H}'}) \tag{40}$$

holds and this is what we wanted to prove.

E Proof of Lemma 4.4

We have to show that I_f and $I_{\hat{f}}$ are locally mutually reducible. We prove each direction separately.

We proceed as follows:

1. We show how to implement $I_{\hat{f}}$ when I_f is given.
2. We show how to implement I_f when $I_{\hat{f}}$ is given.

For both cases we show that the implementation is correct and secure.

Let I_f be given. We implement $I_{\hat{f}}$ using a protocol $\pi_{\hat{f}}$ that simply restricts the input space for I_f. So in this case the real system is $\pi_{\hat{f}} \circ I_f$ and the ideal system is $I_{\hat{f}}$. Correctness (for the all honest case) is obvious. To prove security we have to provide a simulator $\mathsf{S} \in \mathsf{Poly}$ that interacts with the ideal system $I_{\hat{f}}$. We assume some nonempty set of parties $\mathfrak{E} = \{P_{e_1}, P_{e_2}, \ldots, P_{e_{|\mathfrak{E}|}}\} \subseteq \mathfrak{P}$ is corrupted. The simulator $\mathsf{S}_{\mathfrak{E}}$ simply runs the adversary E until E produces an input $x_{\mathfrak{E}} = (x_{e_1}, x_{e_2}, \ldots, x_{e_{|\mathfrak{E}|}})$ intended for I_f. $\mathsf{S}_{\mathfrak{E}}$ then inputs $\hat{x}_{\mathfrak{E}} = (\hat{x}_{e_1}, \hat{x}_{e_2}, \ldots, \hat{x}_{e_{|\mathfrak{E}|}})$ to $I_{\hat{f}}$ and receives an output $\hat{y}_{\mathfrak{E}} = \hat{f}_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, x_{\mathfrak{H}})$ where $x_{\mathfrak{H}}$ is the input of the honest parties $\mathfrak{H} = \mathfrak{P} \setminus \mathfrak{E}$. Now by Lemma 4.3 we have:

1. for all $x_{\mathfrak{H}} \in \mathcal{X}_{\mathfrak{H}}$: $f_{\mathfrak{H}}(\hat{x}_{\mathfrak{E}}, x_{\mathfrak{H}}) = f_{\mathfrak{H}}(x_{\mathfrak{E}}, x_{\mathfrak{H}})$ and
2. for all $x_{\mathfrak{H}}, x'_{\mathfrak{H}} \in \mathcal{X}_{\mathfrak{H}}$: $f_{\mathfrak{E}}(x_{\mathfrak{E}}, x_{\mathfrak{H}}) \neq f_{\mathfrak{E}}(x_{\mathfrak{E}}, x'_{\mathfrak{H}}) \implies f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, x_{\mathfrak{H}}) \neq f_{\mathfrak{E}}(\hat{x}_{\mathfrak{E}}, x'_{\mathfrak{H}})$.

By the second point $\mathsf{S}_{\mathfrak{E}}$ can deduce $y_{\mathfrak{E}} = f(x_{\mathfrak{E}}, x_{\mathfrak{H}})$ from $\hat{y}_{\mathfrak{E}}$. The simulator $\mathsf{S}_{\mathfrak{E}}$ then feeds $y_{\mathfrak{E}}$ to the adversary E and forwards its output y_{E} to the distinguisher D. This clearly results in a perfectly indistinguishable interaction for E and hence in indistinguishable output y_{E} for D. Due to the first point the simulation is also indistinguishable by the outputs of the honest parties $y_{\mathfrak{H}}$, thus $\pi_{\hat{f}} \circ I_f$ indeed perfectly securely implements $I_{\hat{f}}$ and the simulator $\mathsf{S}_{\mathfrak{E}}$ is efficient.

Now let $I_{\hat{f}}$ be given. We describe the protocol π for implementing I_f from $I_{\hat{f}}$ by describing P_i's protocol π_i for any i. The protocol π_i takes input x_i and inputs $\hat{x}_i$ to $I_{\hat{f}}$, receiving $\hat{y}_i = \hat{f}_i(\hat{x}_i, \hat{x}_{\mathfrak{P}'})$ in turn (recall that $\mathfrak{P}' = \mathfrak{P} \setminus \{P_i\}$). Now by definition of $\hat{x}_i$ (1.) and $\hat{f}$ we have

$$\hat{y}_i = \hat{f}_i(\hat{x}_i, \hat{x}_{\mathfrak{P}'}) = f_i(\hat{x}_i, x_{\mathfrak{P}'}).$$

By definition of $\hat{x}_i$ (2.) then for all $x_{\mathfrak{P}'}, x'_{\mathfrak{P}'} \in \mathcal{X}_{\mathfrak{P}'}$:

$$f_i(x_i, x_{\mathfrak{P}'}) \neq f_i(x_i, x'_{\mathfrak{P}'}) \implies f_i(\hat{x}_i, x_{\mathfrak{P}'}) \neq f_i(\hat{x}_i, x'_{\mathfrak{P}'}).$$

Once again this implies that π_i can recover $y_i = f_i(x_i, x_{\mathfrak{P}'})$ from $\hat{y}_i$. Finally the protocol π_i outputs y_i to the distinguisher. The correctness of the protocol for the all honest case is immediate. For the security proof, we note that in this case the ideal system is I_f and the real system is $\pi_i \circ I_{\hat{f}}$. Then the security is follows trivially: The simulator S_i simply restricts the input space for I_f to $\hat{\mathcal{X}}_i$.

F Proof of Lemma 4.6

From Corollary 4.5 we know that $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} \iff \hat{f} \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$. Therefore it suffices to show that for redundancy-free functions $f = \hat{f}$

$$f = \hat{f} \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} \iff f = \hat{f} \in \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}} \tag{41}$$

because then we have:

$$f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} \overset{\text{Cor. 4.5}}{\iff} \hat{f} \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} \iff \hat{f} \in \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}.$$

So wlog we assume for the remainder of this proof that the function f is redundancy-free. Now by Lem. 2.3 we have $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} = \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}}$, so it suffices to show

$$f = \hat{f} \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}} \iff f = \hat{f} \in \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}. \tag{42}$$

This is simplifies matters because in the public discussion setting we have a global transcript which we can refer to. We proceed by showing each implication separately.

$f \in \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}} \implies f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}}$**:** Since passive security implies semi-honest security this is given.

$f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}} \implies f \in \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$**:** The argument we give below is very similar to the second part the proof of Theorem 3.6 where we showed that $f \in \mathfrak{F}_{\mathsf{pas}}^{\mathsf{bc}} \implies f \in \mathfrak{F}'_{\mathsf{pas}}$. But there are two main differences:

(i) Here we do not assume a counterexample that is minimal in the size of the input space and we use our argument on the proper restriction $\tilde{f}$.
The reason for this change is that $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}}$ does not directly imply that any restriction of f is also in $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}}$ and therefore we cannot immediately show an analogue of Claim C.1.

(ii) Since now we are in the semi-honest setting, the simulator is allowed to substitute the inputs provided by the distinguisher for different ones. Therefore, compared to the proof in the passive setting, we have to give an additional argument to prove Equation (46), which was Equation (5) in the proof for the passive case. At that point we use the fact that $f = \hat{f}$ is redundancy-free to show that the simulator indeed has to forward the inputs provided by the distinguisher.

We give a proof by contradiction. Assume a counterexample $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}}$ such that $f \notin \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$. Our goal is to show that such a counterexample does not exist.

Recall that wlog $f = \hat{f}$ is redundancy-free. Since $f \notin \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$, there is a restriction $\tilde{f} := f\,|_{\widetilde{\mathcal{X}}_1 \times \widetilde{\mathcal{X}}_2 \times \ldots \times \widetilde{\mathcal{X}}_n}$ of f such that

1. $\tilde{f}$ is **not** locally computable ($\tilde{f} \notin \mathfrak{F}_{\mathsf{loc}}$) **and**
2. $\forall i : \forall$ partitions of $\widetilde{\mathcal{X}}_i$ into non-empty sets $\mathcal{X}'_i \dot{\cup} \mathcal{X}''_i = \widetilde{\mathcal{X}}_i$
   $\exists P_e \in \mathfrak{P} \setminus \{P_i\}$ (where we define $\mathfrak{E} = \{P_e\}$ and thus $|\mathfrak{E}| = 1$)
   $\exists x_e \in \widetilde{\mathcal{X}}_e$ such that:
   $$\tilde{f}_e(x_e, \widetilde{\mathcal{X}}_{\mathfrak{H}'}, \mathcal{X}'_i) \cap \tilde{f}_e(x_e, \widetilde{\mathcal{X}}_{\mathfrak{H}'}, \mathcal{X}''_i) \neq \emptyset$$
   where $\mathfrak{H} = \mathfrak{P} \setminus \mathfrak{E}$ and $\mathfrak{H}' := \mathfrak{H} \setminus \{P_i\}$.

Now as $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}}$ we have an efficient semi-honestly IT secure protocol π computing f. Then the number of rounds of the protocol π is bounded by a polynomial $p_\pi(\kappa)$ in the security parameter κ. So we can wlog "pad" all

protocol executions with dummy messages to have the same length $p_\pi(\kappa)$. We then define the set of transcripts

$$\begin{aligned}\Pi := \pi(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n) = \{t = m_1, m_2, \ldots, m_{p_\pi(\kappa)} | \\ \exists x_1, c_1, x_2, c_2, \ldots, x_n, c_n : t = \pi(x_1, c_1, x_2, c_2, \ldots, x_n, c_n)\} \\ \Pi_r := \pi(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n)|_r = \{t = m_1, m_2, \ldots, m_r | \\ \exists m_{r+1}, \ldots, m_{p_\pi(\kappa)} : t, m_{r+1}, \ldots, m_{p_\pi(\kappa)} \in \Pi\} \\ = \{t = m_1, m_2, \ldots, m_{p_r} | \\ \exists x_1, c_1, x_2, c_2, \ldots, x_n, c_n : t = \pi(x_1, c_1, x_2, c_2, \ldots, x_n, c_n)|_r\}\end{aligned}$$

where x_i denotes the input and c_i the random coin tosses of party P_i.

We define random variables $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$ for the input and output of the parties $P_1, \ldots, P_n$ respectively. For a set $M \subseteq \mathfrak{P}$ we define X_M and Y_M to be the random variables for the inputs and outputs of the parties in M. Then we let $T \in \Pi$ denote the random variable for the transcript, and $T_r \in \Pi_r$ the random variable on transcript prefixes of length r. Recall that $\tilde{f} = f\,|_{\widetilde{\mathcal{X}}_1 \times \ldots \times \widetilde{\mathcal{X}}_n}$ from the beginning of the proof. We name the inputs $\widetilde{\mathcal{X}}_l = \{x_l^{(1)}, x_l^{(2)}, \ldots, x_l^{(|\widetilde{\mathcal{X}}_l|)}\}$ for party $P_l \in \mathfrak{P}$ and similarly $\widetilde{\mathcal{X}}_M = \{x_M^{(1)}, x_M^{(2)}, \ldots, x_M^{(|\widetilde{\mathcal{X}}_M|)}\}$ for a subset M of the set of parties $\mathfrak{P}$. We first show that for any $x_1^{(i_1)} \in \widetilde{\mathcal{X}}_1, x_2^{(i_2)} \in \widetilde{\mathcal{X}}_2, \ldots, x_n^{(i_n)} \in \widetilde{\mathcal{X}}_n$ (from the restricted input space) the statistical distance

$$\Delta(\mathrm{Pr}_{T|X_1,X_2,\ldots,X_n}(\cdot, x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}), \mathrm{Pr}_{T|X_1,X_2,\ldots,X_n}(\cdot, x_1^{(i_1)}, x_2^{(i_2)}, \ldots, x_n^{(i_n)})) < \mathsf{negl} \tag{43}$$

of these two distributions on the transcripts $t \in \Pi$ is negligible in the security parameter κ. We proceed by induction over the number of protocol rounds r and show that for any r

$$\delta_r := \max_{i_1,i_2,\ldots,i_n} \Delta(\mathrm{Pr}_{T_r|X_1,X_2,\ldots,X_n}(\cdot, x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}), \mathrm{Pr}_{T_r|X_1,X_2,\ldots,X_n}(\cdot, x_1^{(i_1)}, x_2^{(i_2)}, \ldots, x_n^{(i_n)})) < \mathsf{negl}. \tag{44}$$

Base case: For r=0 we have $\Pi_0 = \{\epsilon\}$ where ϵ is the empty transcript. Thus $\delta_0 = 0$.
Induction hypothesis: We assume $\delta_{r-1} < \mathsf{negl}$.
Induction step: Let $t_r = (m_1, m_2, \ldots, m_r) \in \Pi_r$ and consider $t_{r-1} = (m_1, m_2, \ldots, m_{r-1}) \in \Pi_{r-1}$. Wlog (else rename the parties) the message in round r is sent from party P_1 to all other parties.

Recall the definition of $f \notin \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$ from the beginning of the proof. We know by property (2.) of $\tilde{f}$ that the restricted function $\tilde{f}$ does not have a K-Cut. Using this, we can impose an ordering on the elements of $\widetilde{\mathcal{X}}_1$ such that

$\forall x_1^{(j)} \in \widetilde{\mathcal{X}}_1$ where $1 < j \leq |\widetilde{\mathcal{X}}_1|$
$\exists x_1^{(k)} \in \widetilde{\mathcal{X}}_1$ where $1 \leq k < j$
$\exists \mathfrak{E} = \{P_e\}$ where $P_e \in \{P_2, P_3, \ldots, P_n\}$
$\exists x_e^{(jk)} \in \widetilde{\mathcal{X}}_{\mathfrak{E}} = \widetilde{\mathcal{X}}_e$
$\exists \bar{x}_{\mathfrak{H}'}^{(jk)}, \bar{\bar{x}}_{\mathfrak{H}'}^{(jk)} \in \widetilde{\mathcal{X}}_{\mathfrak{H}'}$ such that:

$$f_e(x_e^{(jk)}, \bar{x}_{\mathfrak{H}'}^{(jk)}, x_1^{(k)}) = f_e(x_e^{(jk)}, \bar{\bar{x}}_{\mathfrak{H}'}^{(jk)}, x_1^{(j)}) \tag{45}$$

Where we define $\mathfrak{H} = \mathfrak{P} \setminus \mathfrak{E}$ and $\mathfrak{H}' = \mathfrak{H} \setminus \{P_1\}$.

We proceed by showing that for any r we have

$$\Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(jk)}, \bar{x}_{\mathfrak{H}'}^{(jk)}, x_1^{(k)}), \mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot, x_e^{(jk)}, \bar{\bar{x}}_{\mathfrak{H}'}^{(jk)}, x_1^{(j)})) < \mathsf{negl}. \tag{46}$$

In order to prove this claim, we make use of the fact that the protocol π is semi-honestly secure. By the security of π we know that for any set of corrupted parties $\mathfrak{E} = \{P_e\}$ there exists a simulator $\mathsf{S}_e \in \mathcal{S}_{\mathsf{sh}}$ such that no distinguisher is able to distinguish the ideal from the real setting.

We observe that if the simulator S_e forwards the inputs that are provided by D, then the claim holds because S_e for the corrupted party P_e sees exactly the same (see Equation (45)). But since π is semi-honestly secure, the simulator S_e could input a value $\tilde{x}_e$ different from $x_e^{(jk)}$ to I_f and therefore it could observe a different behavior if the function values in Equation (45) for the new input $\tilde{x}_e$ are not equal. Now we show that due to the fact that f is redundancy-free, S_e must forward the input $x_e^{(jk)}$ provided by the distinguisher D to the ideal functionality I_f. The following argument shows that if S_e forwards a different input, then there exists a distinguisher telling the real and ideal settings apart with non-negligible probability, contradicting our assumption that using S_e, the settings are indistinguishable.

Towards a contradiction assume that S_e inputs $\tilde{x}_e \neq x_e^{(jk)}$ to I_f with non-negligible probability. As wlog f is redundancy-free, we know that $x_e^{(jk)}$ does not dominate $\tilde{x}_e$ so there are two possibilities:

$$\exists x_{\mathfrak{H}} \in \mathcal{X}_{\mathfrak{H}} \,:\, f_{\mathfrak{H}}(x_e^{(jk)}, x_{\mathfrak{H}}) \neq f_{\mathfrak{H}}(\tilde{x}_e, x_{\mathfrak{H}}) \qquad \text{or} \tag{47}$$

$$\exists x'_{\mathfrak{H}}, x''_{\mathfrak{H}} \in \mathcal{X}_{\mathfrak{H}} \,:\, f_e(x_e^{(jk)}, x'_{\mathfrak{H}}) \neq f_e(x_e^{(jk)}, x''_{\mathfrak{H}}) \wedge f_e(\tilde{x}_e, x'_{\mathfrak{H}}) = f_e(\tilde{x}_e, x''_{\mathfrak{H}}). \tag{48}$$

If Equation (47) holds with non-negligible probability, a distinguisher D can easily distinguish the two settings by inputting $x_{\mathfrak{H}}$ and comparing the outputs of the honest parties, contradicting our assumption that using S_e, the real and ideal settings are indistinguishable.

So assume Equation (48) holds with non-negligible probability. We know that the adversary E runs the protocol π_e because we are in the semi-honest setting. Furthermore we can assume that E includes the protocol result y_e in its output y_{E}, as a simulator that works for such an adversary E will also work if the protocol output y_e is not included in y_{E}.

Now we construct a distinguisher D that selects the input for the honest parties $\tilde{x}_{\mathfrak{H}}$ uniformly at random from $\{x'_{\mathfrak{H}}, x''_{\mathfrak{H}}\}$. In the real setting we find $y_e = f_e(x_e^{(jk)}, \tilde{x}_{\mathfrak{H}})$ with overwhelming probability. But in the ideal setting we find $y_e \neq f_e(x_e^{(jk)}, \tilde{x}_{\mathfrak{H}})$ with non-negligible probability because the simulator receives

$$y'_e = f_e(\tilde{x}_e, x'_{\mathfrak{H}}) = f_e(\tilde{x}_e, x''_{\mathfrak{H}})$$

with non-negligible probability. Then, in the ideal setting y_e is independent of $\tilde{x}_{\mathfrak{H}}$ whereas in the real setting it is directly depending on $\tilde{x}_{\mathfrak{H}}$. So the distinguisher D can distinguish the two settings by checking whether $y_e = f_e(x_e^{(jk)}, \tilde{x}_{\mathfrak{H}})$. This is a contradiction to the indistinguishability of $\mathsf{S}_e(\mathsf{E}) \circ I_f$ and $\mathsf{E} \circ \pi$.

This concludes the argument and we can be sure that S_e forwards the input it receives with non-negligible probability. So the claim (46) is proved.

Now, since m_r travels from P_1 to all other parties, we find:

$$\Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot,x_e^{(1)},x_{\mathfrak{H}'}^{(1)},x_1^{(k)}),\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot,x_e^{(jk)},\bar{x}_{\mathfrak{H}'}^{(jk)},x_1^{(k)})) \tag{49}$$

$$= \sum_{t_r\in\Pi_r} |\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(t_r,x_e^{(1)},x_{\mathfrak{H}'}^{(1)},x_1^{(k)}) - \mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(t_r,x_e^{(jk)},\bar{x}_{\mathfrak{H}'}^{(jk)},x_1^{(k)})| \tag{50}$$

$$= \sum_{t_r\in\Pi_r} |\mathrm{Pr}_{M_r|T_{r-1},X_1}(m_r,t_{r-1},x_1^{(k)})\cdot\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1},x_e^{(1)},x_{\mathfrak{H}'}^{(1)},x_1^{(k)}) \tag{51}$$

$$- \mathrm{Pr}_{M_r|T_{r-1},X_1}(m_r,t_{r-1},x_1^{(k)})\cdot\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1},x_e^{(jk)},\bar{x}_{\mathfrak{H}'}^{(jk)},x_1^{(k)})| \tag{52}$$

$$= \sum_{t_{r-1}\in\Pi_{r-1}} \underbrace{\left(\sum_{m_r}\mathrm{Pr}_{M_r|T_{r-1},X_1}(m_r,t_{r-1},x_1^{(k)})\right)}_{=1} \tag{53}$$

$$|\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1},x_e^{(1)},x_{\mathfrak{H}'}^{(1)},x_1^{(k)}) - \mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1},x_e^{(jk)},\bar{x}_{\mathfrak{H}'}^{(jk)},x_1^{(k)})| \tag{54}$$

$$= \Delta(\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1},x_e^{(1)},x_{\mathfrak{H}'}^{(1)},x_1^{(k)}),\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1},x_e^{(jk)},\bar{x}_{\mathfrak{H}'}^{(jk)},x_1^{(k)})) \tag{55}$$

$$\leq \Delta(\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1},x_e^{(1)},x_{\mathfrak{H}'}^{(1)},x_1^{(1)}),\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1},x_e^{(1)},x_{\mathfrak{H}'}^{(1)},x_1^{(k)})) \tag{56}$$

$$+ \Delta(\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1},x_e^{(1)},x_{\mathfrak{H}'}^{(1)},x_1^{(1)}),\mathrm{Pr}_{T_{r-1}|X_e,X_{\mathfrak{H}'},X_1}(t_{r-1},x_e^{(jk)},\bar{x}_{\mathfrak{H}'}^{(jk)},x_1^{(k)})) \tag{57}$$

$$= 2\cdot\delta_{r-1} < \mathsf{negl}, \tag{58}$$

Now let $\tilde{x}_e \in \widetilde{\mathcal{X}}_e$ and $\tilde{x}_{\mathfrak{H}'} \in \widetilde{\mathcal{X}}_{\mathfrak{H}'}$ be arbitrary elements. In the same way as we derived equation (58) we find:

$$\Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot,x_e^{(jk)},\bar{\bar{x}}_{\mathfrak{H}'}^{(jk)},x_1^{(j)}),\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot,\tilde{x}_e,\tilde{x}_{\mathfrak{H}'},x_1^{(j)})) \tag{59}$$

$$= 2\cdot\delta_{r-1} < \mathsf{negl}, \tag{60}$$

so we find, using (46), (49) and (59), that

$$\Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot,x_e^{(1)},x_{\mathfrak{H}'}^{(1)},x_1^{(k)}),\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot,\tilde{x}_e,\tilde{x}_{\mathfrak{H}'},x_1^{(j)})) \tag{61}$$

$$\leq \Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot,x_e^{(1)},x_{\mathfrak{H}'}^{(1)},x_1^{(k)}),\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot,x_e^{(jk)},\bar{x}_{\mathfrak{H}'}^{(jk)},x_1^{(k)})) \tag{62}$$

$$+ \Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot,x_e^{(jk)},\bar{x}_{\mathfrak{H}'}^{(jk)},x_1^{(k)}),\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot,x_e^{(jk)},\bar{\bar{x}}_{\mathfrak{H}'}^{(jk)},x_1^{(j)})) \tag{63}$$

$$+ \Delta(\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot,x_e^{(jk)},\bar{\bar{x}}_{\mathfrak{H}'}^{(jk)},x_1^{(j)}),\mathrm{Pr}_{T_r|X_e,X_{\mathfrak{H}'},X_1}(\cdot,\tilde{x}_e,\tilde{x}_{\mathfrak{H}'},x_1^{(j)})) \tag{64}$$

$$\leq \mathsf{negl} + \mathsf{negl} + \mathsf{negl} \tag{65}$$

This is not yet what we claimed in (44) because we would like to obtain the above result (65) for $k = 1$. We can first observe that the choice of $\tilde{x}_e$, $\tilde{x}_{\mathfrak{H}'}$ and j is arbitrary for the above argument, but $\mathfrak{E} = \{P_e\}$ is fixed for given j. To finally obtain what we claimed in (44) we use induction on j. The idea is to apply the result (65) step by step:

- To get from j to $k_1 < j$ there exists a set of corrupted parties $\mathfrak{E}_1 = P_{e_1}$ such that the statistical distance in (65) is negligible.
- To get from k_1 to $k_2 < k_1$ there exists a set of corrupted parties $\mathfrak{E}_2 = P_{e_2}$ such that the statistical distance in (65) is negligible.
- ...
- To get from k_i to $1 < k_i$ there exists a set of of corrupted parties $\mathfrak{E}_i = P_{e_i}$ such that the statistical distance in (65) is negligible.

We claim that the sum of all these negligible distances is again negligible. This only holds if the length i of the induction chains is at most polynomial in the security parameter κ. But this is given, since we require the round number $p_\pi(\kappa)$ to be polynomially bounded and as for a given fixed function f the sizes of the input domains $|\mathcal{X}_1|, |\mathcal{X}_2|, \ldots, |\mathcal{X}_n|$ is of course constant.

This concludes our inductive argument and we proved the claim (44).

We know by definition that $\tilde{f}$ is in particular not locally computable. Hence wlog there is a party P_i such that

$\exists x_i \in \widetilde{\mathcal{X}}_i$
$\exists x'_{\mathfrak{P}'}, x''_{\mathfrak{P}'} \in \widetilde{\mathcal{X}}_{\mathfrak{P}'}$ where $\mathfrak{P}' = \mathfrak{P} \setminus \{P_i\}$ such that

$$f_i(x_i, x'_{\mathfrak{P}'}) \neq f_i(x_i, x''_{\mathfrak{P}'}) \tag{66}$$

From the above result (43) and an application of the triangle inequality we obtain:

$$\Delta(\mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(\cdot, x_i, x'_{\mathfrak{P}'}), \mathrm{Pr}_{T|X_A,X_B}(\cdot, x_i, x''_{\mathfrak{P}'})) < \mathsf{negl}. \tag{67}$$

As the output of P_i is independent of the inputs of $\mathfrak{P}'$ given the transcript, we have:

$$\mathrm{Pr}_{Y_i|T,X_i,X_{\mathfrak{P}'}} = \mathrm{Pr}_{Y_i|T,X_i} \tag{68}$$

And so we find:

$$\begin{aligned}
&\Delta(\mathrm{Pr}_{Y_i|X_i,X_{\mathfrak{P}'}}(\cdot, x_i, x'_{\mathfrak{P}'}), \mathrm{Pr}_{Y_i|X_i,X_{\mathfrak{P}'}}(\cdot, x_i, x''_{\mathfrak{P}'})) && (69)\\
&= \sum_{y_i \in \mathcal{Y}_i} |\mathrm{Pr}_{Y_i|X_i,X_{\mathfrak{P}'}}(y_i, x_i, x'_{\mathfrak{P}'}) - \mathrm{Pr}_{Y_i|X_i,X_{\mathfrak{P}'}}(y_i, x_i, x''_{\mathfrak{P}'})| && (70)\\
&= \sum_{y_i \in \mathcal{Y}_i} |\sum_{t \in \Pi} \mathrm{Pr}_{Y_i,T|X_i,X_{\mathfrak{P}'}}(y_i, t, x_i, x'_{\mathfrak{P}'}) - \sum_{t \in \Pi} \mathrm{Pr}_{Y_i,T|X_i,X_{\mathfrak{P}'}}(y_i, t, x_i, x''_{\mathfrak{P}'})| && (71)\\
&\leq \sum_{y_i \in \mathcal{Y}_i, t \in \Pi} |\mathrm{Pr}_{Y_i|T,X_i,X_{\mathfrak{P}'}}(y_i, t, x_i, x'_{\mathfrak{P}'}) \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x'_{\mathfrak{P}'}) && (72)\\
&\qquad\qquad - \mathrm{Pr}_{Y_i|T,X_i,X_{\mathfrak{P}'}}(y_i, t, x_i, x''_{\mathfrak{P}'}) \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x''_{\mathfrak{P}'})| \\
&\overset{(68)}{=} \sum_{y_i \in \mathcal{Y}_i, t \in \Pi} |\mathrm{Pr}_{Y_i|T,X_i}(y_i, t, x_i) \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x'_{\mathfrak{P}'}) - \mathrm{Pr}_{Y_i|T,X_i}(y_i, t, x_i) \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x''_{\mathfrak{P}'})| && (73)\\
&= \sum_{y_i \in \mathcal{Y}_i, t \in \Pi} \mathrm{Pr}_{Y_i|T,X_i}(y_i, t, x_i) \, |\mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x'_{\mathfrak{P}'}) - \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x''_{\mathfrak{P}'})| && (74)\\
&= \sum_{t \in \Pi} \underbrace{\left(\sum_{y_i \in \mathcal{Y}_i} \mathrm{Pr}_{Y_i|T,X_i}(y_i, t, x_i) \right)}_{=1} |\mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x'_{\mathfrak{P}'}) - \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(t, x_i, x''_{\mathfrak{P}'})| && (75)\\
&= \Delta(\mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(\cdot, x_i, x'_{\mathfrak{P}'}), \mathrm{Pr}_{T|X_i,X_{\mathfrak{P}'}}(\cdot, x_i, x''_{\mathfrak{P}'})) \overset{(67)}{<} \mathsf{negl}. && (76)
\end{aligned}$$

This is in obvious contradiction to the correctness of the protocol π, so we must have $f \in \mathfrak{F}^{\mathsf{aut}}_{\mathsf{pas}}$. Hence there is no counterexample and the claim is proven.

G Symmetrization

We present a generalization of the Symmetrization-Lemma of [KMQ08] to the n-party setting.

Lemma G.1. *(Symmetrization) For any function $f \in \mathfrak{F}^{\mathsf{aut}}_{\mathsf{sh}}$ there is a redundancy free, symmetric function $\mathrm{sym}(f) \in \mathfrak{F}$ such that the functions f and $\mathrm{sym}(f)$ are efficiently and PFE securely locally mutually reducible, even in presence of active adversaries.*

Proof. By Lem. 4.4 it is sufficient to show Lem. G.1 for redundancy-free functions f where $f = \hat{f}$. So in the following let $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$ and let $f = \hat{f}$ be redundancy-free. Then by Thm. 4.6 we already have $f \in \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$.

We now follow the proof of [KMQ08]. Let

$$\begin{aligned}
&\mathfrak{P}_{-i} := \mathfrak{P} \setminus \{P_i\},\\
&\vec{x} := x_1, \dots, x_n \in \vec{\mathcal{X}} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_n,\\
&\vec{x}_{-i} := x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n \in \vec{\mathcal{X}}_{-i} := \mathcal{X}_1 \times \ldots \times \mathcal{X}_{i-1} \times \mathcal{X}_{i+1} \times \cdots \times \mathcal{X}_n\\
&\vec{x}_{-i,j} := x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_{j-1}, x_{j+1}, \dots, x_n\\
&\qquad \in \vec{\mathcal{X}}_{-i,j} := \mathcal{X}_1 \times \ldots \times \mathcal{X}_{i-1} \times \mathcal{X}_{i+1} \times \ldots \times \mathcal{X}_{j-1} \times \mathcal{X}_{j+1} \times \cdots \times \mathcal{X}_n.
\end{aligned}$$

We begin by defining a consistent renaming, generalizing the definition of [KMQ08] to the multi-party case.

Definition G.2 (Consistent Renaming)**.** A function $f^{(1)} \in \mathfrak{F}$ is a consistent renaming of a function $f^{(2)} \in \mathfrak{F}$ iff for all $i \in [n]$ there are

1. bijective maps $\sigma_i : \mathcal{X}_i^{(1)} \to \mathcal{X}_i^{(2)}$,
2. $\forall x_i \in \mathcal{X}_i^{(1)}$ a bijective map $\sigma_{x_i} : f_i^{(1)}(x_i, \mathcal{X}_{-i}^{(1)}) \to f_i^{(2)}(\sigma_i(x_i), \mathcal{X}_{-i}^{(2)})$,

We then say $f^{(1)}$ and $f^{(2)}$ are renamings: $f^{(1)} \equiv f^{(2)}$.

If $f \equiv g$, then clearly functions f, g are efficiently and PFE securely locally mutually reducible, even in presence of active adversaries.

We define new functions $g, h \in \mathfrak{F}$ by

$$\begin{aligned}
g_i(\vec{x}) &:= \{\vec{x}'_{-i} \in \vec{\mathcal{X}}_{-i} \mid f_i(\vec{x}) = f_i(\vec{x}'_{-i}, x_i)\},\\
h_i(\vec{x}) &:= (g_j(\vec{x}))_{j \in [n]},
\end{aligned}$$

where h is by construction symmetric. Clearly $f \equiv g$. We now show that $g \equiv h$. It suffices to prove

$$\forall i \in [n]\ \forall x_i \in \mathcal{X}_i,\ \vec{x}_{-i}, \vec{x}'_{-i} \in \mathcal{X}_{-i} :\ g_i(x_i, \vec{x}_{-i}) = g_i(x_i, \vec{x}'_{-i}) \iff h_i(x_i, \vec{x}_{-i}) = h_i(x_i, \vec{x}'_{-i})$$

By definition of h, this is equivalent to

$$\forall i \in [n]\ \forall x_i \in \mathcal{X}_i,\ \vec{x}_{-i}, \vec{x}'_{-i} \in \mathcal{X}_{-i} :\ g_i(x_i, \vec{x}_{-i}) = g_i(x_i, \vec{x}'_{-i}) \implies g_{-i}(x_i, \vec{x}_{-i}) = g_{-i}(x_i, \vec{x}'_{-i}) \tag{77}$$

Towards a contradiction assume we have an $i \in [n]$ and $x_i \in \mathcal{X}_i$, $\vec{x}_{-i}, \vec{x}'_{-i} \in \mathcal{X}_{-i}$ such that for $\vec{x} := (x_i, \vec{x}_{-i})$ and $\vec{x}' := (x_i, \vec{x}'_{-i})$ we have $g_i(\vec{x}) = g_i(\vec{x}')$ but $g_{-i}(\vec{x}) \neq g_{-i}(\vec{x}')$. So there is a $j \in [n] \setminus \{i\}$ such that $g_j(\vec{x}) \neq g_j(\vec{x}')$.

Because of $g_j(\vec{x}) \neq g_j(\vec{x}')$ we find $\vec{x}''_{-j} \in \mathcal{X}_{-j}$ such that wlog (else interchange $\vec{x}$ and $\vec{x}'$) $\vec{x}''_{-j} \in g_j(\vec{x}')$, $\vec{x}''_{-j} \notin g_j(\vec{x})$. We now define $\vec{x}'' := (x_j, \vec{x}''_{-j})$, $\vec{x}''' := (x'_j, \vec{x}''_{-j})$,and

$$\vec{\mathcal{X}}' := \{x_i, x''_i\} \times \{x_j, x'_j\} \times \{\vec{x}_{-i,j}, \vec{x}'_{-i,j}, \vec{x}''_{-i,j}\}$$

and claim that $f\,|_{\vec{\mathcal{X}}'} \notin \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$. By definition of g and choice of $\vec{x}$, $\vec{x}'$, $\vec{x}''$, $\vec{x}'''$ we have

$$f_i(\vec{x}) = f_i(\vec{x}') \qquad \text{by choice of } \vec{x} = (x_i, \vec{x}_{-i}),\ \vec{x}' = (x_i, \vec{x}'_{-i}) \tag{78}$$

$$f_j(\vec{x}) \neq f_j(\vec{x}'') \qquad \text{because } x''_{-j} \notin g_j(\vec{x}) \tag{79}$$

$$f_j(\vec{x}') = f_j(\vec{x}''') \qquad \text{because } x''_{-j} \in g_j(\vec{x}'). \tag{80}$$

So $f\,|_{\vec{\mathcal{X}}'}$ is not locally computable due to $f_j(\vec{x}) \neq f_j(\vec{x}'')$, and there is no K-cut for P_i due to $f_j(\vec{x}') = f_j(\vec{x}''')$, no K-cut for P_j due to $f_i(\vec{x}) = f_i(\vec{x}')$, and no K-cut for $P_{-i,j}$ due to $f_i(\vec{x}) = f_i(\vec{x}')$ and $f_j(\vec{x}') = f_j(\vec{x}''')$.

So by Thm.3.6 we have $f\,|_{\vec{\mathcal{X}}'} \notin \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$ and moreover $f\,|_{\vec{\mathcal{X}}'}$ being a restriction of f we find $f \notin \mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$ in contradiction to the choice of f. We conclude that Eq. (77) holds and thus $f \equiv h =: \mathrm{sym}(f)$. The statement of Lem. G.1 immediately follows. □

H Proof of Theorem 5.3

By Lem. 4.4 it is sufficient to show for redundancy-free functions f where $f = \hat{f}$ that $f \in \mathfrak{F}'_{\mathsf{act}} \implies f \in \mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}}$ and $f \in \mathfrak{F}_{2\mathsf{act}} \implies f \in \mathfrak{F}'_{2\mathsf{act}}$. For all other (not redundancy-free) functions we then find

$$f \in \mathfrak{F}'_{\mathsf{act}} \overset{\text{Def. 5.2}}{\iff} \hat{f} \in \mathfrak{F}'_{\mathsf{act}} \implies \hat{f} \in \mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}} \overset{\text{Lem. 4.4}}{\iff} f \in \mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}} \tag{81}$$

$$f \in \mathfrak{F}_{2\mathsf{act}} \overset{\text{Lem. 4.4}}{\iff} \hat{f} \in \mathfrak{F}_{2\mathsf{act}} \implies \hat{f} \in \mathfrak{F}'_{2\mathsf{act}} \overset{\text{Def. 5.2}}{\iff} f \in \mathfrak{F}'_{2\mathsf{act}}. \tag{82}$$

So in the following let $f = \hat{f}$ be redundancy-free.

$f \in \mathfrak{F}'_{\mathsf{act}} \implies f \in \mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}}$: This part of the proof is considerably simplified by use of the symmetrization-lemma Lem. G.1. Said lemma states that any function $f \in \mathfrak{F}^{\mathsf{aut}}_{\mathsf{sh}}$ is locally mutually reducible to a symmetric, redundancy-free function $\mathrm{sym}(f)$, i.e. for all $i, j \in [n]$ $\mathrm{sym}(f)_i = \mathrm{sym}(f)_j$. Now it is easily seen that $\mathfrak{F}'_{\mathsf{act}}, \mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}} \subseteq \mathfrak{F}^{\mathsf{aut}}_{\mathsf{sh}}$ and it suffices to show that $\mathrm{sym}(f) \in \mathfrak{F}'_{\mathsf{act}} \implies \mathrm{sym}(f) \in \mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}}$ because using Lem. G.1 we have

$$f \in \mathfrak{F}'_{\mathsf{act}} \overset{(*)}{\iff} \mathrm{sym}(f) \in \mathfrak{F}'_{\mathsf{act}} \implies \mathrm{sym}(f) \in \mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}} \overset{\text{Lem. G.1}}{\iff} f \in \mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}}.$$

Implication (*) holds as we have seen in the proof of Lem. G.1, that $\mathrm{sym}(f)$ is a consistent renaming of $\hat{f}$ so

$$\mathrm{sym}(f) \in \mathfrak{F}'_{\mathsf{act}} \iff \hat{f} \in \mathfrak{F}'_{\mathsf{act}} \overset{\text{Def. 5.2}}{\iff} f \in \mathfrak{F}'_{\mathsf{act}}.$$

So wlog let $f \in \mathfrak{F}'_{\mathsf{act}}$ be symmetric and redundancy-free.

We prove $f \in \mathfrak{F}'_{\mathsf{act}} \implies f \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{act}} \overset{\text{Lem. 2.3}}{=} \mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}}$ by induction over the size of the input space $|\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n|$. Now $f \in \mathfrak{F}'_{\mathsf{act}}$ implies that f is either $f \in \mathfrak{F}_{\mathsf{loc}}$ locally computable, in which case clearly $f \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{act}}$ actively computable and we are done, or f has a T-cut.

In the second case, by definition of the class $\mathfrak{F}'_{\mathsf{act}}$ for some $P_i \in \mathfrak{P}$ the set $\mathcal{X}_i$ has a partition $\mathcal{X}_i^{(1)} \dot{\cup} \mathcal{X}_i^{(2)} = \mathcal{X}_i$ such that

(i) $f^{(1)} := f|_{\mathcal{X}_1 \times \ldots \times \mathcal{X}_i^{(1)} \times \ldots \times \mathcal{X}_n}$, $f^{(2)} := f|_{\mathcal{X}_1 \times \ldots \times \mathcal{X}_i^{(2)} \times \ldots \times \mathcal{X}_n} \in \mathfrak{F}'_{\mathsf{act}}$ and

(ii) $\forall \mathfrak{E} \subseteq \mathfrak{P} \setminus \{P_i\}$ and $\mathfrak{H}' = \mathfrak{H} \setminus \{P_i\}$ we have

$$\forall \bar{x}_{\mathfrak{E}} \in \mathcal{X}_{\mathfrak{E}} : f_{\mathfrak{E}}(\bar{x}_{\mathfrak{E}}, \mathcal{X}_{\mathfrak{H}'}, \mathcal{X}_i^{(1)}) \cap f_{\mathfrak{E}}(\bar{x}_{\mathfrak{E}}, \mathcal{X}_{\mathfrak{H}'}, \mathcal{X}_i^{(2)}) = \emptyset$$

and $\forall x'_{\mathfrak{E}}, x''_{\mathfrak{E}} \in \mathcal{X}_{\mathfrak{E}}$ $\exists x_{\mathfrak{E}} \in \mathcal{X}_{\mathfrak{E}}$ $\forall x_{\mathfrak{H}'} \in \mathcal{X}_{\mathfrak{H}'}$

$$\begin{aligned} &\forall x_i^{(1)} \in \mathcal{X}_i^{(1)} : && f_{\mathfrak{H}}(x'_{\mathfrak{E}}, x_{\mathfrak{H}'}, x_i^{(1)}) = f_{\mathfrak{H}}(x_{\mathfrak{E}}, x_{\mathfrak{H}'}, x_i^{(1)}) \quad \wedge \\ &\forall x_i^{(2)} \in \mathcal{X}_i^{(2)} : && f_{\mathfrak{H}}(x''_{\mathfrak{E}}, x_{\mathfrak{H}'}, x_i^{(2)}) = f_{\mathfrak{H}}(x_{\mathfrak{E}}, x_{\mathfrak{H}'}, x_i^{(2)}) \end{aligned}$$

By induction hypothesis we then have $f^{(1)}, f^{(2)} \in \mathfrak{F}^{\mathsf{bc}}_{\mathsf{act}}$ as

$$|\mathcal{X}_1 \times \ldots \times \mathcal{X}_i^{(1)} \times \ldots \times \mathcal{X}_n| < |\mathcal{X}_1 \times \ldots \times \mathcal{X}_i \times \ldots \times \mathcal{X}_n|$$

$$|\mathcal{X}_1 \times \ldots \times \mathcal{X}_i^{(2)} \times \ldots \times \mathcal{X}_n| < |\mathcal{X}_1 \times \ldots \times \mathcal{X}_i \times \ldots \times \mathcal{X}_n|$$

and there are protocols $\pi^{(1)}, \pi^{(2)}$ actively PFE securely implementing $I_{f^{(1)}}, I_{f^{(2)}}$ under simulators $\mathsf{S}^{(1)}, \mathsf{S}^{(2)}$.

We now construct a protocol π by defining a first protocol round where P_i broadcasts a message $m_1 \in \{1, 2\}$ indicating whether P_i's input x_i is $x_i \in \mathcal{X}_i^{(1)}$ or $x_i \in \mathcal{X}_i^{(2)}$. If P_i sends no message, the other parties just assume $m_1 = 1$. Subsequently the parties proceed to run $\pi^{(m_1)}$.

It remains to prove that this protocol π is PFE secure by providing an appropriate simulator $\mathsf{S} \in \mathcal{S}_{\mathsf{act}}$ for the case where an arbitrary subset $\mathfrak{E}$ of the parties $\mathfrak{P}$ is corrupted by the adversary $\mathsf{E} \in \mathcal{E}_{\mathsf{act}}$. It suffices to consider two cases. Either P_i is in the set of corrupted parties ($P_i \in \mathfrak{E}$), or it is not ($P_i \notin \mathfrak{E}$).

Let us first consider the case where $P_i \notin \mathfrak{E}$. We construct the simulator $\mathsf{S}_{P_i \notin \mathfrak{E}}$ as follows:

1. $\mathsf{S}_{P_i \notin \mathfrak{C}}$ fixes the randomness of the adversary E
2. It feeds the input x_{E} of the distinguisher D to E
3. It generates two copies $\mathsf{E}^{(1)}$ and $\mathsf{E}^{(2)}$ of E and feeds $m_1 = 1$ to $\mathsf{E}^{(1)}$ and $m_1 = 2$ to $\mathsf{E}^{(2)}$
4. Now $\mathsf{S}_{P_i \notin \mathfrak{C}}$ runs $\mathsf{S}^{(1)}(\mathsf{E}^{(1)})$ and $\mathsf{S}^{(2)}(\mathsf{E}^{(2)})$ until these simulators produce inputs $x_{\mathfrak{C}}^{(1)}$ and $x_{\mathfrak{C}}^{(2)}$ for $I_{f^{(1)}}$ and $I_{f^{(2)}}$ respectively
5. Since $f \in \mathfrak{F}'_{\mathsf{act}}$ we know:

$$\forall \bar{x}_{\mathfrak{C}} \in \mathcal{X}_{\mathfrak{C}} : f_{\mathfrak{C}}(\bar{x}_{\mathfrak{C}}, \mathcal{X}_{\mathfrak{H}'}, \mathcal{X}_i^{(1)}) \cap f_{\mathfrak{C}}(\bar{x}_{\mathfrak{C}}, \mathcal{X}_{\mathfrak{H}'}, \mathcal{X}_i^{(2)}) = \emptyset \tag{83}$$

   and $\exists x_{\mathfrak{C}} \in \mathcal{X}_{\mathfrak{C}}\, \forall x_{\mathfrak{H}'} \in \mathcal{X}_{\mathfrak{H}'}$

$$\begin{aligned} &\forall x_i^{(1)} \in \mathcal{X}_i^{(1)} : && f_{\mathfrak{H}}(x_{\mathfrak{C}}^{(1)}, x_{\mathfrak{H}'}, x_i^{(1)}) = f_{\mathfrak{H}}(x_{\mathfrak{C}}, x_{\mathfrak{H}'}, x_i^{(1)}) \quad \wedge \\ &\forall x_i^{(2)} \in \mathcal{X}_i^{(2)} : && f_{\mathfrak{H}}(x_{\mathfrak{C}}^{(2)}, x_{\mathfrak{H}'}, x_i^{(2)}) = f_{\mathfrak{H}}(x_{\mathfrak{C}}, x_{\mathfrak{H}'}, x_i^{(2)}) \end{aligned} \tag{84}$$

   The simulator $\mathsf{S}_{P_i \notin \mathfrak{C}}$ computes this $x_{\mathfrak{C}}$, inputs it to I_f and receives one of the following two outputs:

$$\begin{aligned} y_{\mathfrak{C}} = f_{\mathfrak{C}}(x_{\mathfrak{C}}, x_{\mathfrak{H}'}, x_i) \overset{(*)}{=} f_{\mathfrak{H}}(x_{\mathfrak{C}}, x_{\mathfrak{H}'}, x_i) \overset{(84)}{=} f_{\mathfrak{H}}(x_{\mathfrak{C}}^{(1)}, x_{\mathfrak{H}'}, x_i) \overset{(*)}{=} f_{\mathfrak{C}}(x_{\mathfrak{C}}^{(1)}, x_{\mathfrak{H}'}, x_i) \text{ if } x_i \in \mathcal{X}_i^{(1)} \\ y_{\mathfrak{C}} = f_{\mathfrak{C}}(x_{\mathfrak{C}}, x_{\mathfrak{H}'}, x_i) \overset{(*)}{=} f_{\mathfrak{H}}(x_{\mathfrak{C}}, x_{\mathfrak{H}'}, x_i) \overset{(84)}{=} f_{\mathfrak{H}}(x_{\mathfrak{C}}^{(2)}, x_{\mathfrak{H}'}, x_i) \overset{(*)}{=} f_{\mathfrak{C}}(x_{\mathfrak{C}}^{(2)}, x_{\mathfrak{H}'}, x_i) \text{ if } x_i \in \mathcal{X}_i^{(2)} \end{aligned} \tag{85}$$

   where the equalities marked by $(*)$ follow from the symmetry of the function f, which we assumed in the beginning of the proof.
6. Now if $y_{\mathfrak{C}} \in f_{\mathfrak{C}}(x_{\mathfrak{C}}, \mathcal{X}_{\mathfrak{H}'}, \mathcal{X}_i^{(1)})$ (by Eqn. (83) this check is always unambiguous) then $\mathsf{S}_{P_i \notin \mathfrak{C}}$ passes $y_{\mathfrak{C}}$ to $\mathsf{S}^{(1)}(\mathsf{E}^{(1)})$ and returns the output of that simulator. Note that the above Eqn (85) shows that the input $y_{\mathfrak{C}}$ is correct for $\mathsf{S}^{(1)}(\mathsf{E}^{(1)})$.
   If $y_{\mathfrak{C}} \in f_{\mathfrak{C}}(x_{\mathfrak{C}}, \mathcal{X}_{\mathfrak{H}'}, \mathcal{X}_i^{(2)})$ then $\mathsf{S}_{P_i \notin \mathfrak{C}}$ proceeds in the same way with $\mathsf{S}^{(2)}(\mathsf{E}^{(2)})$.

The correctness of the above simulation is obvious.

We now turn to the second case, where P_i is corrupted, i.e. we have $P_i \in \mathfrak{C}$. The simulator $\mathsf{S}_{P_i \in \mathfrak{C}}$ can be constructed as follows:

1. $\mathsf{S}_{P_i \in \mathfrak{C}}$ fixes the randomness of the adversary E
2. It passes the input x_{E} provided by the distinguisher D to E
3. It runs the adversary E until it outputs a message m_1
4. If the adversary makes no output $m_1 \in \{1, 2\}$, $\mathsf{S}_{P_i \in \mathfrak{C}}$ uses $m_1 = 1$
5. Now the simulator $\mathsf{S}_{P_i \in \mathfrak{C}}$ runs $\mathsf{S}^{(m_1)}(\mathsf{E})$ and simply forwards all inputs and outputs

This simulates the real execution, as depending on the message m_1 the protocol $\pi^{(m_1)}$ is executed by the honest parties. The interaction with this protocol is then faithfully simulated by $\mathsf{S}_{P_i \in \mathfrak{C}}$.

Note that we actually prove active PFE security above. The computation of locally computable functions is perfectly secure (induction base) and in the above argument if the subsimulators are perfectly secure and efficient, so are the resulting simulators (induction step).

$f \in \mathfrak{F}_{\mathsf{2act}} \implies f \in \mathfrak{F}'_{\mathsf{2act}}$**:** Recall that wlog $f = \hat{f}$ is redundancy-free. If $f \in \mathfrak{F}_{\mathsf{2loc}}$ is locally computable then $f \in \mathfrak{F}'_{\mathsf{2act}}$ and we are done. So consider an $f \notin \mathfrak{F}_{\mathsf{2loc}}$. We show that f has a T-cut and then apply an inductive argument. As f is actively computable we have $f \in \mathfrak{F}_{\mathsf{2sh}}$ and $f \notin \mathfrak{F}_{\mathsf{2loc}}$, so f has a K-cut. Wlog we have $\mathcal{X}_B = \mathcal{X}'_B \cup \mathcal{X}''_B$ and

$$\forall x_A \in \mathcal{X}_A \ : \ f_A(x_A, \mathcal{X}'_B) \cap f_A(x_A, \mathcal{X}''_B) = \emptyset \tag{86}$$

Now for any $x_A \in \mathcal{X}_A$ and regardless of the a priori distribution $\Pr_{X_A,X_B}$ on the inputs, as the function outputs on $\mathcal{X}'_B$ and $\mathcal{X}''_B$ differ, so must the protocol outputs y_A for A with overwhelming probability. As a result the statistical distance of the transcripts under input taken from $x_A, \mathcal{X}'_B$ or $x_A, \mathcal{X}''_B$ must be overwhelming as well:

$$\begin{aligned}
1 - \mathsf{negl} &< \Delta(\Pr_{Y_A|X_A,X_B}(\cdot, x_A, \mathcal{X}'_B), \Pr_{Y_A|X_A,X_B}(\cdot, x_A, \mathcal{X}''_B)) && (87)\\
&= \sum_{y_A \in \mathcal{Y}_A} |\Pr_{Y_A|X_A,X_B}(y_A, x_A, \mathcal{X}'_B) - \Pr_{Y_A|X_A,X_B}(y_A, x_A, \mathcal{X}''_B)| && (88)\\
&= \sum_{y_A \in \mathcal{Y}_A} |\sum_{t \in \Pi} \Pr_{Y_A,T|X_A,X_B}(y_A, t, x_A, \mathcal{X}'_B) - \sum_{t \in \Pi} \Pr_{Y_A,T|X_A,X_B}(y_A, t, x_A, \mathcal{X}''_B)| && (89)\\
&\leq \sum_{y_A \in \mathcal{Y}_A, t \in \Pi} |\Pr_{Y_A|T,X_A,X_B}(y_A, t, x_A, \mathcal{X}'_B)\Pr_{T|X_A,X_B}(t, x_A, \mathcal{X}'_B) && (90)\\
&\qquad - \Pr_{Y_A|T,X_A,X_B}(y_A, t, x_A, \mathcal{X}''_B)\Pr_{T|X_A,X_B}(t, x_A, \mathcal{X}''_B)|\\
&= \sum_{y_A \in \mathcal{Y}_A, t \in \Pi} |\Pr_{Y_A|T,X_A}(y_A, t, x_A)\Pr_{T|X_A,X_B}(t, x_A, \mathcal{X}'_B) && (91)\\
&\qquad - \Pr_{Y_A|T,X_A}(y_A, t, x_A)\Pr_{T|X_A,X_B}(t, x_A, \mathcal{X}''_B)|\\
&= \sum_{y_A \in \mathcal{Y}_A, t \in \Pi} \Pr_{Y_A|T,X_A}(y_A, t, x_A) \, |\Pr_{T|X_A,X_B}(t, x_A, \mathcal{X}'_B) - \Pr_{T|X_A,X_B}(t, x_A, \mathcal{X}''_B)| && (92)\\
&= \sum_{t \in \Pi} \underbrace{\sum_{y_A \in \mathcal{Y}_A} \Pr_{Y_A|T,X_A}(y_A, t, x_A)}_{=1} \, |\Pr_{T|X_A,X_B}(t, x_A, \mathcal{X}'_B) - \Pr_{T|X_A,X_B}(t, x_A, \mathcal{X}''_B)| && (93)\\
&= \Delta(\Pr_{T|X_A,X_B}(\cdot, x_A, \mathcal{X}'_B), \Pr_{T|X_A,X_B}(\cdot, x_A, \mathcal{X}''_B)). && (94)
\end{aligned}$$

Note, that being independent from the priori distribution $\Pr_{X_A,X_B}$, this result applies to all subsets of $\mathcal{X}'_B$, $\mathcal{X}''_B$ as well. In particular for any $x'_B \in \mathcal{X}'_B$, $x''_B \in \mathcal{X}''_B$, $x_A \in \mathcal{X}_A$

$$\begin{aligned}
1 - \mathsf{negl} &< \Delta(\Pr_{Y_A|X_A,X_B}(\cdot, x_A, x'_B), \Pr_{Y_A|X_A,X_B}(\cdot, x_A, x''_B)) && (95)\\
&\leq \Delta(\Pr_{T|X_A,X_B}(\cdot, x_A, x'_B), \Pr_{T|X_A,X_B}(\cdot, x_A, x''_B)). && (96)
\end{aligned}$$

Now for any $\mathcal{X}'_B \cup \mathcal{X}''_B = \mathcal{X}_B$, $a \in \mathcal{X}_A$

$$\Delta(\Pr_{T_0|X_A,X_B}(\cdot, a, \mathcal{X}'_B), \Pr_{T_0|X_A,X_B}(\cdot, a, \mathcal{X}''_B)) = 0. \tag{97}$$

Hence considering all possible choices of values $x_A \in \mathcal{X}_A$ and cuts $\mathcal{X}'_B \cup \mathcal{X}''_B = \mathcal{X}_B$, as well as symmetrically $x_B \in \mathcal{X}_B$ and cuts $\mathcal{X}'_A \cup \mathcal{X}''_A = \mathcal{X}_A$, there must be a minimal round number r for which there is a distribution $\Pr_{X_A,X_B}$ on the inputs such that we have

$$\begin{aligned}
\alpha :=& \Delta(\Pr_{T_r|X_A,X_B}(\cdot, x_A, \mathcal{X}'_B), \Pr_{T_r|X_A,X_B}(\cdot, x_A, \mathcal{X}''_B)) > \mathsf{ntcbl} \qquad \text{or} && (98)\\
&\Delta(\Pr_{T_r|X_A,X_B}(\cdot, \mathcal{X}'_A, x_B), \Pr_{T_r|X_A,X_B}(\cdot, \mathcal{X}''_A, x_B)) > \mathsf{ntcbl} && (99)
\end{aligned}$$

Wlog we assume that the minimum r is achieved for some particular $x_A \in \mathcal{X}_A$ and a cut $\mathcal{X}'_B \cup \mathcal{X}''_B = \mathcal{X}_B$. Then, by minimality of the round number r, for all $r' < r$

$$\begin{aligned}
\forall x_B \in \mathcal{X}_B, \ x_A \in \mathcal{X}_A, \mathcal{X}'_B \cup \mathcal{X}''_B = \mathcal{X}_B \ \mathcal{X}'_A \cup \mathcal{X}''_A = \mathcal{X}_A \ :& \\
\Delta(\Pr_{T_{r'}|X_A,X_B}(\cdot, x_A, \mathcal{X}'_B), \Pr_{T_{r'}|X_A,X_B}(\cdot, x_A, \mathcal{X}''_B)) < \mathsf{negl} \qquad \text{and} & \qquad (100)\\
\Delta(\Pr_{T_r|X_A,X_B}(\cdot, \mathcal{X}'_A, x_B), \Pr_{T_r|X_A,X_B}(\cdot, \mathcal{X}''_A, x_B)) < \mathsf{negl}. &
\end{aligned}$$

So the message m_r travels from B to A, as A cannot construct m_r for lack of information about x_B. So Eq. (100) and Eq. (98) hold for the given cut $\mathcal{X}'_B \cup \mathcal{X}''_B = \mathcal{X}_B$ and an *arbitrary* $x_A \in \mathcal{X}_A$, as m_r is constructed by B without information about x_A.

Furthermore note that we can find a cut $\mathcal{X}'_B \cup \mathcal{X}''_B = \mathcal{X}_B$ such that the above holds for every distribution $\Pr_{X_A,X_B}$ on the inputs. This can be seen using a hybrid argument over the inputs.

We need to show now

1. $\mathcal{X}'_B \cup \mathcal{X}''_B = \mathcal{X}_B$ constitutes a K-cut
2. $\mathcal{X}'_B \cup \mathcal{X}''_B = \mathcal{X}_B$ constitutes a T-cut
3. the argument perpetuates inductively

The cut $\mathcal{X}'_B \cup \mathcal{X}''_B = \mathcal{X}_B$ constitutes a K-cut. Towards a contradiction assume we have $b' \in \mathcal{X}'_B$, $b'' \in \mathcal{X}''_B$, $a \in \mathcal{X}_A$ such that $f_A(a, b') = f_A(a, b'')$. Then for any simulated transcript T_{S}

$$\Pr_{T_{\mathsf{S}}|X_A,X_B}(\cdot, a, b') = \Pr_{T_{\mathsf{S}}|X_A,X_B}(\cdot, a, b'') \tag{101}$$

Now by security there exists a simulator S such that

$$\Delta(\Pr_{T_{\mathsf{S}}|X_A,X_B}(\cdot, a, b'), \Pr_{T|X_A,X_B}(\cdot, a, b')) < \mathsf{negl}, \tag{102}$$
$$\Delta(\Pr_{T_{\mathsf{S}}|X_A,X_B}(\cdot, a, b''), \Pr_{T|X_A,X_B}(\cdot, a, b'')) < \mathsf{negl}. \tag{103}$$

Then by transitivity

$$\Delta(\Pr_{T|X_A,X_B}(\cdot, a, b'), \Pr_{T|X_A,X_B}(\cdot, a, b'')) < \mathsf{negl} \tag{104}$$

which implies

$$\Delta(\Pr_{T_r|X_A,X_B}(\cdot, a, b'), \Pr_{T_r|X_A,X_B}(\cdot, a, b'')) < \mathsf{negl} \tag{105}$$

in contradiction to Eq. (98).

The cut $\mathcal{X}'_B \cup \mathcal{X}''_B = \mathcal{X}_B$ is a T-cut. Consider an $a'' \in \mathcal{X}_A$ and the cut $\mathcal{X}'_B \cup \mathcal{X}''_B = \mathcal{X}_B$ from Eq. (98). From Eq. (100) we have for any $a' \in \mathcal{X}_A$

$$\Delta(\Pr_{T_r|X_A,X_B}(\cdot, a', \mathcal{X}'_B), \Pr_{T_r|X_A,X_B}(\cdot, a'', \mathcal{X}'_B)) < \mathsf{negl}. \tag{106}$$

Thus at round r the distribution of transcripts for different inputs of A differs still only negligibly. So with overwhelming probability a corrupted A can choose (uniformly among the matching ones) a new random string c'_A such that under the random string c'_A the input a' is consistent with the transcript T_r observed so far. The execution then proceeds according to π with the new values a' as input and c'_A as random string. We now fix an $a' \in \mathcal{X}_A$ and name the procedure just described π' and the induced transcript random variable T'.

We find that the distributions on the transcript T' (and thus also the output distributions) resulting from this procedure differ only negligibly from the transcript distributions obtained by initiating a normal protocol run with

input a':

$$\Delta(\Pr_{T|X_A,X_B}(\cdot,a',\mathcal{X}'_B),\Pr_{T'|X_A,X_B}(\cdot,a'',\mathcal{X}'_B)) \tag{107}$$

$$= \sum_{t\in\Pi} |\Pr_{T|X_A,X_B}(t,a',\mathcal{X}'_B) - \Pr_{T'|X_A,X_B}(t,a'',\mathcal{X}'_B)| \tag{108}$$

$$= \sum_{t\in\Pi} |\Pr_{T|T_r,X_A,X_B}(t,t_r,a',\mathcal{X}'_B)\Pr_{T_r|X_A,X_B}(t,a',\mathcal{X}'_B) \tag{109}$$

$$- \Pr_{T'|T'_r,X_A,X_B}(t,t_r,a'',\mathcal{X}'_B)\Pr_{T'_r|X_A,X_B}(t,a'',\mathcal{X}'_B)|$$

$$= \sum_{t\in\Pi} |\Pr_{T|T_r,X_A,X_B}(t,t_r,a',\mathcal{X}'_B)\Pr_{T_r|X_A,X_B}(t,a',\mathcal{X}'_B) \tag{110}$$

$$- \Pr_{T|T_r,X_A,X_B}(t,t_r,a',\mathcal{X}'_B)\Pr_{T_r|X_A,X_B}(t,a'',\mathcal{X}'_B)|$$

$$= \sum_{t\in\Pi} \Pr_{T|T_r,X_A,X_B}(t,t_r,a',\mathcal{X}'_B)|\Pr_{T_r|X_A,X_B}(t,a',\mathcal{X}'_B) - \Pr_{T_r|X_A,X_B}(t,a'',\mathcal{X}'_B)| \tag{111}$$

$$= \Delta(\Pr_{T_r|X_A,X_B}(\cdot,a',\mathcal{X}'_B),\Pr_{T_r|X_A,X_B}(\cdot,a'',\mathcal{X}'_B)) < \mathsf{negl}. \tag{112}$$

We can proceed analogously for $\mathcal{X}''_B$ and thus have

$$\Delta(\Pr_{T|X_A,X_B}(\cdot,a',\mathcal{X}'_B),\Pr_{T'|X_A,X_B}(\cdot,a'',\mathcal{X}'_B)) < \mathsf{negl}, \tag{113}$$

$$\Delta(\Pr_{T|X_A,X_B}(\cdot,a',\mathcal{X}''_B),\Pr_{T'|X_A,X_B}(\cdot,a'',\mathcal{X}''_B)) < \mathsf{negl}. \tag{114}$$

A corrupted A may then mount the following attack: A executes π with input a up to round r, obtaining a transcript t_r. Now if $\Pr_{T_r|X_A,X_B}(t_r,a'',\mathcal{X}'_B) > \Pr_{T_r|X_A,X_B}(t_r,a'',\mathcal{X}''_B)$ (assuming uniform distribution on each set $\mathcal{X}'_B$ or $\mathcal{X}''_B$ of the partition) then A runs π' and π else. The resulting output distributions are indistinguishable (negligible statistical distance) from $\pi(a',X_B)$ and $\pi(a'',X_B)$ respectively as seen above. We designate this protocol execution by $\breve{\pi}$ and the induced random variable for the transcripts $\breve{T}$. Now take a distinguisher D that with probability $\frac{1}{2}$ each chooses X_B from $\mathcal{X}'_B$ or $\mathcal{X}''_B$ respectively with uniform distribution on each set $\mathcal{X}'_B$ or $\mathcal{X}''_B$ of the partition and provides A with no information.

Consider the conditional distribution of B's output under the given adverserial A and distinguisher D where we

first take $b' \in \mathcal{X}'_B$ and then $b'' \in \mathcal{X}''_B$:

$$\Pr_{Y_B|X_B}(y,b') = \sum_{t\in\Pi} \Pr_{Y_B|\breve{T},X_B}(y,t,b')\Pr_{\breve{T}|X_B}(t,b') \tag{115}$$

$$= \sum_{t\in\Pi} \Pr_{Y_B|T,X_B}(y,t,b')\Pr_{\breve{T}|X_B}(t,b') \tag{116}$$

$$\overset{(100)}{\approx} \sum_{t\in\Pi} \Pr_{Y_B|T,X_B}(y,t,b') \left(\frac{1+\alpha}{2}\Pr_{T|X_A,X_B}(t,a',b') + \frac{1-\alpha}{2}\Pr_{T|X_A,X_B}(t,a'',b')\right) \tag{117}$$

$$\approx \frac{1+\alpha}{2}\Pr_{Y_B|X_A,X_B}(y,a',b') + \frac{1-\alpha}{2}\Pr_{Y_B|X_A,X_B}(y,a'',b') \tag{118}$$

$$\approx \frac{1+\alpha}{2}1_{y=f_B(a',b')} + \frac{1-\alpha}{2}1_{y=f_B(a'',b')}; \tag{119}$$

$$\Pr_{Y_B|X_B}(y,b'') = \sum_{t\in\Pi} \Pr_{Y_B|\breve{T},X_B}(y,t,b'')\Pr_{\breve{T}|X_B}(t,b'') \tag{120}$$

$$= \sum_{t\in\Pi} \Pr_{Y_B|T,X_B}(y,t,b'')\Pr_{\breve{T}|X_B}(t,b'') \tag{121}$$

$$\overset{(100)}{\approx} \sum_{t\in\Pi} \Pr_{Y_B|T,X_B}(y,t,b'') \left(\frac{1-\alpha}{2}\Pr_{T|X_A,X_B}(t,a',b'') + \frac{1+\alpha}{2}\Pr_{T|X_A,X_B}(t,a'',b'')\right) \tag{122}$$

$$\approx \frac{1-\alpha}{2}\Pr_{Y_B|X_A,X_B}(y,a',b'') + \frac{1+\alpha}{2}\Pr_{Y_B|X_A,X_B}(y,a'',b'') \tag{123}$$

$$\approx \frac{1-\alpha}{2}1_{y=f_B(a',b'')} + \frac{1+\alpha}{2}1_{y=f_B(a'',b'')}. \tag{124}$$

In the ideal setting and for the distinguisher D as described above, the input a of A as provided to the ideal functionality by the simulator is independent of the input b of B. Thus we find for the conditional distribution of the output Y_B^I of B in the ideal case, both for $b \in \mathcal{X}'_B$ and $b \in \mathcal{X}''_B$

$$\Pr_{Y_B^I|X_B}(y,b) = \sum_{a\in\mathcal{X}_A \,:\, y=f_B(a,b)} \Pr_{X_A}(a) \tag{125}$$

where the simulator can only adjust $\Pr_{X_A}(a)$.

By security of the protocol (simulatability) we must have a *single* distribution $\Pr_{X_A}$ such that

$$\mathsf{negl} > \Delta(\Pr_{Y_B|X_B}(y,b'), \Pr_{Y_B^I|X_B}(y,b')) \tag{126}$$

$$\mathsf{negl} > \Delta(\Pr_{Y_B|X_B}(y,b''), \Pr_{Y_B^I|X_B}(y,b'')) \tag{127}$$

We can conclude that there is an $a \in \mathcal{X}_A$ such that $\Pr_{X_A}(a) > \mathsf{ntcbl}$ and $f_B(a,b') = f_B(a',b')$, $f_B(a,b'') = f_B(a'',b'')$ for all $b' \in \mathcal{X}'_B$ and $b'' \in \mathcal{X}''_B$:

$$\mathsf{negl} > \Delta(\Pr_{Y_B|X_B}(y,b'), \Pr_{Y_B^I|X_B}(y,b')) \tag{128}$$

$$= \sum_{y\in\mathcal{Y}_B} |\Pr_{Y_B|X_B}(y,b') - \Pr_{Y_B^I|X_B}(y,b')| \tag{129}$$

$$\approx \sum_{y\in\mathcal{Y}_B} |\frac{1+\alpha}{2}1_{y=f(a',b')} + \frac{1-\alpha}{2}1_{y=f_B(a'',b')} - \sum_{a\in\mathcal{X}_A \,:\, y=f_B(a,b')} \Pr_{X_A}(a)| \tag{130}$$

$$= \sum_{y\in\mathcal{Y}_B} |\frac{1+\alpha}{2}1_{y=f(a',b')} + \frac{1-\alpha}{2}1_{y=f_B(a'',b')} - \sum_{a\in\mathcal{X}_A} \Pr_{X_A}(a)1_{y=f_B(a,b')}|. \tag{131}$$

If $f_B|_{\mathcal{X}_A\times\mathcal{X}'_B}$ or $f_B|_{\mathcal{X}_A\times\mathcal{X}''_B}$ are constant, then f already has a T-cut. So wlog we consider $a', a'' \in \mathcal{X}_A$ such that there is a $b' \in \mathcal{X}'_B$ and a $b'' \in \mathcal{X}''_B$ where $f_B(a',b') \neq f_B(a'',b')$ and $f_B(a',b'') \neq f_B(a'',b'')$. But then the obvious

solution $\Pr_{X_A}(a') \approx \frac{1+\alpha}{2}$, $\Pr_{X_A}(a'') \approx \frac{1-\alpha}{2}$, $\Pr_{X_A}(a) \approx 0$ for $a' \neq a \neq a''$ leads to a contradiction with

$$\mathsf{negl} > \Delta(\Pr_{Y_B|X_B}(y,b''), \Pr_{Y_B^I|X_B}(y,b'')) \tag{132}$$

$$\approx \sum_{y\in\mathcal{Y}_B} |\frac{1-\alpha}{2} 1_{y=f(a',b'')} + \frac{1+\alpha}{2} 1_{y=f_B(a'',b'')} - \sum_{a\in\mathcal{X}_A} \Pr_{X_A}(a) 1_{y=f_B(a,b'')}|. \tag{133}$$

So there must be an $a \in \mathcal{X}_A$ where $\Pr_{X_A}(a) > \mathsf{ntcbl}$. But then also $f(a,b') = f(a',b')$, $f(a,b'') = f(a'',b'')$. In fact, if there is only one such a, we find

$$\Pr_{X_A}(a) \approx \alpha, \tag{134}$$

$$\Pr_{X_A}(a') \approx \frac{1-\alpha}{2}, \tag{135}$$

$$\Pr_{X_A}(a'') \approx \frac{1-\alpha}{2}. \tag{136}$$

Extending the Argument Finally we need to show, that the argument presented above applies inductively until we are left with locally computable restrictions of the original function.

To this end we consider $f' = f|_{\mathcal{X}_A\times\mathcal{X}'_B}$ and $f'' = f|_{\mathcal{X}_A\times\mathcal{X}''_B}$. Unfortunately we cannot readily conclude that $f \in \mathfrak{F}_{2\mathsf{act}}$ implies $f', f'' \in \mathfrak{F}_{2\mathsf{act}}$. Take for instance f' and some corrupted B. Then S simulates correctly for I_f by security of π for f. On $I_{f'}$ however the simulation may fail, as S may rely on making an input $b \in \mathcal{X}''_B$ to I_f, which will be rejected by $I_{f'}$.

However, for our proof above we employ only a specific class $\mathcal{E}$ of adversaries, namely semi-honest adversaries and adversaries that generally adhere to the protocol π, but possibly attempt to "switch" their input. We show that security against this class $\mathcal{E}$ of adversaries already implies $f \in \mathfrak{F}'_{2\mathsf{act}}$, so in particular $f \in \mathfrak{F}_{2\mathsf{act}} \implies f \in \mathfrak{F}'_{2\mathsf{act}}$. So it is sufficient to argue that both f' and f'' are indeed in the class $\mathfrak{F}''_{2\mathsf{act}}$ of functions secure against this specific subclass $\mathcal{E}$ of adversaries. Then we can proceed inductively or simply argue by contraposition: Take an $f \in \mathfrak{F}''_{2\mathsf{act}}$, $f \notin \mathfrak{F}'_{2\mathsf{act}}$ where $|\mathcal{X}_A \times \mathcal{X}_B|$ minimal. We have shown that such an f must have a T-cut decomposing f into f' and f''. Now $f', f'' \in \mathfrak{F}''_{2\mathsf{act}}$ (this we show below) and hence by minimality of f we have $f', f'' \in \mathfrak{F}'_{2\mathsf{act}}$. But then $f' \in \mathfrak{F}'_{2\mathsf{act}}$ by definition of $\mathfrak{F}'_{2\mathsf{act}}$ in contradiction to the choice of f. We hence find

$$f \in \mathfrak{F}_{2\mathsf{act}} \implies f \in \mathfrak{F}''_{2\mathsf{act}} \implies f \in \mathfrak{F}'_{2\mathsf{act}}. \tag{137}$$

So we need to show that f' and f'' are indeed in the class $\mathfrak{F}''_{2\mathsf{act}}$ of functions secure against the subclass of adversaries $\mathcal{E}$ used in the proof above. Now clearly for corrupted A we have f' and f'' as secure as f, since the same simulators can be applied. For a corrupted B consider adversaries and distinguishers as in the argument above where simply the set $\mathcal{X}_A$ is replaced by $\mathcal{X}'_B$ and the set $\mathcal{X}_B$ by $\mathcal{X}_A$. Then the outcome Y of the real protocol execution will be $Y \in f(\mathcal{X}_A, \mathcal{X}'_B)$ with overwhelming probability. Now $\mathcal{X}'_B \cup \mathcal{X}''_B = \mathcal{X}_B$ is a K-cut, meaning

$$\forall x_A \in \mathcal{X}_A \ : \ f(x_A, \mathcal{X}'_B) \cap f(x_A, \mathcal{X}''_B) = \emptyset. \tag{138}$$

But then, as $\mathsf{S}(B)$ is a valid simulator for I_f under π, $\mathsf{S}(B)$ must with overwhelming probability give an input $b \in \mathcal{X}'_B$ to I_f. Hence $\mathsf{S}(B)$ is actually a valid simulator for $I_{f'}$ with respect to π. In other words $f' \in \mathfrak{F}''_{2\mathsf{act}}$.

I Proofsketch of Theorem 6.1

We first show $\mathfrak{F}_{2\mathsf{qu}} \neq \mathfrak{F}_{2\mathsf{sh}}$. Due to the impossibility of implementing unbiased coin flipping (Kitaev, quoted in [ABDR04]) it is impossible to have XOR$\in \mathfrak{F}_{2\mathsf{qu}}$ (which is sufficient to implement coin flipping), however XOR is in $\mathfrak{F}_{2\mathsf{sh}}$.

Next we sketch how to prove $\mathfrak{F}_{2\mathsf{qu}} \subseteq \mathfrak{F}_{2\mathsf{sh}}$. Towards a contradiction assume there is a function $f \in \mathfrak{F}_{2\mathsf{qu}} \setminus \mathfrak{F}_{2\mathsf{sh}}$. Then there is an effcient secure quantum protocol π for f, but no classical protocol secure against a semi-honest adversary. We can assume f to be minimal in the size of the input space $|\mathcal{X}_A \times \mathcal{X}_B|$. Then f must already be redundancy-free and hence, according to Lemma 4.6 we have $f \notin \mathfrak{F}_{2\mathsf{pas}}$. This implies that f does not have a K-cut.

The secure quantum protocol π runs in a polynomial number of rounds, and we can assume the number of rounds to be padded to $p(k)$ rounds independent of the inputs.

The proof will be by induction over the number of rounds, analogously to the proof of Lemma 3.6. However we will have to consider measurements on the quantum data.

For round number r of a protocol π and a quantum non-demolition measurement[23] M which measures just one bit we define an adversary $\mathsf{E}_{r,M}$. This adversary honestly follows the protocol π until round r. In round r (before sending or after receiving a message) the adversary $\mathsf{E}_{r,M}$ performs the measurement M. Then the adversary continues to honestly follow the protocol, however, his quantum data might be disturbed by the measurement M and we have to see that this will remain undetected by the uncorrupted party with noticeable probability:

A quantum state before a quantum non-demolition measurement yielding one bit of output and the quantum state after the measurement have a noticeable overlap and the probability for any measurement to give identical results for the two states is noticeable. This noticeable probability upper bounds the probability that the uncorrupted party will detect any difference between the behaviour of $\mathsf{E}_{r,M}$ and honest behaviour.

Let the inputs x_A, x_B for the parties be chosen uniformly at random. According to the above discussion of the probability with which the measurement performed by the adversary remains undetected we have that for every input x_A, x_B the protocol π will not abort (i.e. terminate successfully) with noticeable probability. We will next prove by induction that for these non-aborting protocol runs it holds that the bit measured by the adversary is statistically independent of the input of the uncorrupted party. This will then lead to the desired contradiction.

In round $r = 0$ before any message was sent any bit resulting from a measurement performed by the adversary is statistically independent of the input of the uncorrupted party.

Now we take as induction hypothesis that no statistical difference could be observed by any adversary up to round r. In round $r + 1$ a message is sent. If this message is sent by the adversary then still no statistical difference can be detected, hence we only need to consider the case where the message in round $r+1$ is sent by the uncorrupted party to the adversary (for notational convenience we assume in the following that B is the corrupted party). We assume that there is an adversary $\mathsf{E}_{r+1,M}$ who can after receiving the message observe a statistical difference with respect to different inputs of the uncorrupted party. There is one party which, if corrupted, would be able to observe the statistical difference first with a probability $\geq 1/2$ and we can thus conclude that with a probability $\geq 1/2$ the result of any measurement which could be performed on the quantum data of the uncorrupted party is statistically independent of the input of the adversary (otherwise we just change the roles of the parties). Hence the (noticeable) probability of observing this statistical difference in round $r + 1$ is almost independent of the input of the adversary $\mathsf{E}_{r+1,M}$. Due to this statistical difference the set of inputs $\mathcal{X}_A$ of the uncorrupted party can be partitioned into two disjoint subsets $\mathcal{X}'_A, \mathcal{X}''_A$ ($\mathcal{X}_A = \mathcal{X}'_A \cup \mathcal{X}''_A$) such that the conditional probability of any input in $\mathcal{X}'_A$ is noticeably higher than the conditional probability of any input in $\mathcal{X}''_A$ given the measured bit. Since the function f has no K-cut there exist inputs $x'_A \in \mathcal{X}'_A$ and $x''_A \in \mathcal{X}''_A$ for the uncorrupted party and an input x_B for the adversary such that $f_B(x'_A, x_B) = f_B(x''_A, x_B)$. So with noticeable probability the protocol will not abort and the adversary will have a noticeable advantage in distinguishing x'_A and x''_A while using input x_B. This is impossible in an ideal model secure function evaluation of f. We can conclude that for every adversary $\mathsf{E}_{r,M}$ and every round r the bit measured by the adversary is statistically independent of the input of the uncorrupted party.

However, there is an adversary who is looking at his output in the last round (and otherwise runs the protocol). This adversary will be able to see a statistical difference with respect to the input of the uncorrupted party or the function is locally computable. This contradicts the statistical independence of every measured bit and concludes the proof.

J Proofsketch of Theorem 7.2

By Lem. 2.3 we have $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} = \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}}$, so it suffices to show $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}} \iff f \in \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$. We proceed to show each implication separately.

[23] I.e. a textbook measurement which disturbs the quantum state just as much as required by the laws of quantum mechanics.

$f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}} \implies f \in \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$**:** A long-term secure protocol is already by definition secure against unbounded semi-honest adversaries.

$f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}} \implies f \in \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$**:** Let $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}}$. We show that $f \in \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$ by constructing a protocol for computing f and prove its security.

The protocol works as follows: First, we obtain an unconditionally hiding commitment scheme from the given OWF. Then the parties commit to their inputs and give a zero-knowledge argument of knowledge of their inputs. Now the parties execute the semi-honestly secure protocol π_f for f, that exists as $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{bc}}$, with the following modifications: After each computation step they commit to the result and give a perfect ZK argument that it was computed correctly. Instead of sending messages they now simply open commitments. If at any point a zero-knowledge argument fails, the protocol aborts.

To show that this protocol is LT secure (security with abort), we construct an efficient simulator $\mathsf{S} \in \mathsf{Poly}$ such that for any efficient adversary $\mathsf{E} \in \mathsf{Poly}$ every distinguisher $\mathsf{D} \in \mathsf{Algo}$ has only negligible distinguishing advantage.

The simulator S feeds the input x_{E} received from D to E. Now E has to commit to his inputs $x_{\mathfrak{E}}$ and give a zero-knowledge argument of knowledge of the inputs $x_{\mathfrak{E}}$. By the definition of a proof of knowledge, there exists a knowledge-extractor W interacting with E that can extract the knowledge $x_{\mathfrak{E}}$ of E. Since S can execute multiple instances of E and branch its execution with E's randomness fixed, the simulator can use W to learn $x_{\mathfrak{E}}$. Now S forwards $x_{\mathfrak{E}}$ to the ideal functionality $I_f^{\mathsf{ab}}(\mathfrak{E})$ and in return gets the outputs $y_{\mathfrak{E}}$. Now the simulator determines some $\tilde{x}_{\mathfrak{H}}$ such that

$$f_{\mathfrak{E}}(\tilde{x}_{\mathfrak{H}}, x_{\mathfrak{E}}) = y_{\mathfrak{E}}$$

using the function table of f. S now proceeds to run the protocol $\pi_{\mathfrak{H}}(\tilde{x}_{\mathfrak{H}})$ of the honest parties with the adversary E. S forwards the output y_{E} of E to D. If E deviates from the protocol π (i.e. a zero-knowledge argument fails), S sends the abort signal to $I_f^{\mathsf{ab}}(\mathfrak{E})$.

Now we have to check if this simulator S meets our requirements:
It is clear that S is efficient, since E, W and π are efficient and finding $\tilde{x}_{\mathfrak{H}}$ in the constant size function table can be done efficiently.
It remains to see that the real and the ideal settings are indistinguishable for any distinguisher D:
Let $x_{\mathfrak{H}}$ be the input of D for the honest parties. By the way $\tilde{x}_{\mathfrak{H}}$ is chosen by S the protocol execution between S and E must lead to a protocol execution that only negligibly differs from the real protocol execution and therefore the output of the adversary differs only negligibly from the output in the real setting. This directly follows from the security of the underlying semi-honestly secure protocol.
Since E must be efficient, it cannot forge the zero-knowledge arguments with non-negligible probability. Thus, if the adversary E deviates from the protocol π, this results in a failed zero-knowledge argument with overwhelming probability and then the protocol aborts. So our use of zero-knowledge arguments actually forces semi-honest behavior (with abort).
Since we use a perfectly hiding bit commitment scheme and the arguments are perfect zero-knowledge, no additional information is flowing compared to the original protocol.
Finally, we see that the security cannot be violated anymore after protocol termination, even by unbounded adversaries.

K Proofsketch of Lemma 7.4

Let $f \in \mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$ be a sh computable two party function. From Theorem 4.6 we obtain a sh secure protocol $\widetilde{\pi}$. We "pad" the protocol with dummy-messages, such that the number of rounds does not depend on the inputs anymore. We view the resulting protocol as a distributed circuit, where the gates are owned by the parties P_i, depending on which party executes a specific computation. Communication is then nothing else than a wire between gates owned by different parties.

From this distributed circuit we can obtain a CO secure protocol for the function f using the Goldreich compiler [Gol04], that (CO) securely computes any circuit using a CO-OT+ functionality.

As in the case of plain LT security each party first commits to their input using unconditionally hiding bit-commitments and proofs knowledge of the input using a perfectly zero-knowledge argument of knowledge. However, in the following, since we are interested in an LT secure protocol π for f with designated aborter P_1, we need to take special care when applying the Goldreich compiler. Concretely, we only use unconditionally hiding bit-commitments and perfect zero-knowledge arguments and we take care to apply the CO-OT+ in such a fashion that it is LT secure for P_i when computing a gate owned by P_i. When the protocol is complete, the result is first opened towards P_1, who then ensures that the result is opened to the remaining parties.

As shown in [Gol04] the protocol π computes f CO securely and the designated abort property is provided through the opening procedure, since as Goldreich shows no party learns anything until the opening phase. It remains to argue LT security.

In π the structure of the underlying passively IT secure protocol $\widetilde{\pi}$ is preserved. Due to the way we employ CO-OT+, information-theoretically, each party only receives information about inputs and outputs of gates it computes itself in $\widetilde{\pi}$. Therefore no party receives more information than in $\widetilde{\pi}$, and after termination of the protocol security, even against unbounded distinguishers, is guaranteed.

Simulators as demanded by the definition of security are easily constructed from the above. The proofs of knowledge in the beginning of the protocol are used to extract an input. A simulator passes this input to the ideal functionality and obtain output from the ideal functionality. The simulator then determines an input of the honest party that is consistent with the input and output extracted from the adversary respectively received from ideal functionality. Using this input the simulator can simulate a regular protocol execution toward the adversary and use the output of the adversary as its own. Indistiguishability of such a simulation follows once again directly from the above arguments.

L Classification of 2-party Functions

Note that in Theorem 8.3 we actually have

$$\begin{aligned}
\mathfrak{F}_{\mathsf{2act}} \setminus \mathfrak{F}_{\mathsf{2pas}} &= \{f \in \mathfrak{F}_{\mathsf{2act}} \mid \bar{f} \not\equiv \hat{f}\} \\
\mathfrak{F}_{\mathsf{2sh}} \setminus \mathfrak{F}_{\mathsf{2pas}} &= \{f \in \mathfrak{F}_{\mathsf{2sh}} \mid \bar{f} \not\equiv \hat{f}\} \\
\mathfrak{F}^{\mathsf{nct}}_{\mathsf{2pas}} &= \{f \in \mathfrak{F}^{\mathsf{nct}}_{\mathsf{2sh}} \mid \bar{f} \equiv \hat{f}\} \\
\mathfrak{F}^{\mathsf{nct}}_{\mathsf{2act}} &= \mathfrak{F}^{\mathsf{nct}}_{\mathsf{2sh}} \cup (\mathfrak{F}_{\mathsf{2sh}} \setminus \mathfrak{F}_{\mathsf{2act}}) \\
\mathfrak{C}_{\mathsf{2pas}} &= \mathfrak{C}_{\mathsf{2act}} \cup \{f \in \mathfrak{F}_2 \mid \hat{f} \not\equiv \bar{f}\}.
\end{aligned}$$

Using the examples in Fig. 1 we can illustrate that the inclusions of Theorem 8.3 are indeed proper:

$$f^{(1)} \in \mathfrak{F}_{\mathsf{2act}} \setminus \mathfrak{F}_{\mathsf{2pas}} \subsetneq \mathfrak{C}_{\mathsf{2pas}} \setminus \mathfrak{C}_{\mathsf{2act}} \tag{139}$$

$$f^{(2)} \in \mathfrak{F}_{\mathsf{2pas}} \setminus \mathfrak{F}_{\mathsf{2act}} \subsetneq \mathfrak{F}^{\mathsf{nct}}_{\mathsf{2act}} \setminus \mathfrak{F}^{\mathsf{nct}}_{\mathsf{2sh}} \tag{140}$$

$$f^{(3)} \in \mathfrak{F}_{\mathsf{2sh}} \setminus \mathfrak{F}_{\mathsf{2act}} \cup \mathfrak{F}_{\mathsf{2pas}} \tag{141}$$

$$f^{(4)} \in \mathfrak{C}_{\mathsf{2pas}} \cap \mathfrak{F}^{\mathsf{nct}}_{\mathsf{2sh}} \subsetneq \mathfrak{C}_{\mathsf{2pas}} \setminus \mathfrak{C}_{\mathsf{2act}}. \tag{142}$$