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

Robin Künzler[1], Jörn Müller-Quade[2,⋆], and Dominik Raub[1,⋆⋆]

[1] ETH Zurich, Department of Computer Science, CH-8092 Zurich, Switzerland
robink@student.ethz.ch, raubd@inf.ethz.ch
[2] IKS/EISS, Fakultät für Informatik, Universität Karlsruhe (TH), Germany
muellerq@ira.uka.de

Abstract. While general secure function evaluation (SFE) with information-theoretical (IT) security is infeasible in presence of a corrupted majority in the standard model, 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, computational assumptions can usually be well justified at the time of protocol execution. The concern is rather a potential violation of the privacy of sensitive data by an attacker whose power increases over time. 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.

Towards a combinatorial characterization of this class of functions, we also 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.

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

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

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-642-00457-5_36

⋆ Thanks for financial support from the European Commission (SECOQC).
⋆⋆ Supported by the Swiss National Science Foundation (SNF), project no. 200020-113700/1.

O. Reingold (Ed.): TCC 2009, LNCS 5444, pp. 238–255, 2009.

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 arise that are no direct consequences of the results by Mayers [25] or Kitaev [1].

All results in this paper are constructive (whenever it is claimed that a class of functions is securely computable a protocol is given) and proven in a stand-alone simulatability based security model with a synchronous communication network (see e.g. [15]).

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* $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}}$ which can securely be computed by parties connected by authenticated channels in presence of a CO unlimited passive[2] 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*[3] 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 gives rise to novel impossibility results beyond those of Mayers [25] or Kitaev [1].

[1] However, quantum bit commitment is impossible [25] and hence no function implying bit commitment is computable.

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

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

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 a realistic communication infrastructure consisting 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 [21] (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 [9].

1.2 Related Work

Secure computability of functions was first discussed by [9]. 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 [23,24] and Beaver [2] presented the first results 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 [24]. Also, in the 2-party setting, [26] sketches a generalization of [24] 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 [9,2,24,26] 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. [17] 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 [17], 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 [19,20,3,21]. 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 [6].

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 [1].

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 [8,7] 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 [28]. In [13] 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 [5]. Additional assumptions, like a temporary bound on the quantum memory can again provide everlasting security for secure computations [11].

In this paper we investigate the power of temporary CO assumptions in the standard model. This is along the lines of [27]. However, in [27] 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 $\mathfrak{P} = \{P_1, \ldots, P_n\}$.[4] 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 sets $M = \{P_{m_1}, \ldots, P_{m_{|M|}}\} \subseteq \mathfrak{P}$ 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[5] (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.

We define *security* using a simulation based stand-alone[6] model (see e.g. [15]) 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*

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

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

[6] As opposed to a universally composable model.

Table 1. Basic Security Paradigms

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	$\epsilon(\kappa) < \mathsf{negl}$	$\pi \succcurlyeq^{\mathsf{LT}} I$

functionality I. A protocol π 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 which 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 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 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.[7] If now for any adversaries E from a class $\mathcal{E}$ controlling a set $\mathfrak{E}$ of parties there is a simulator S from a class $\mathcal{S}$ such that the advantage of any distinguisher D from a class $\mathcal{D}$ in distinguishing the real system $\mathsf{E} \circ \pi_{\mathfrak{H}} \circ \mathsf{R}$ and the ideal system $\mathsf{S}(\mathsf{E}) \circ I$ is bounded by an advantage function $\epsilon(\kappa)$, then we say that protocol π securely implements the ideal functionality I. 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. Denoting the class of efficient[8] algorithms by Poly, the class of arbitrary unbounded algorithms by Algo, and negligibility in the security parameter κ as $\epsilon(\kappa) < \mathsf{negl}$ we arrive at the security paradigms listed in Table 1.

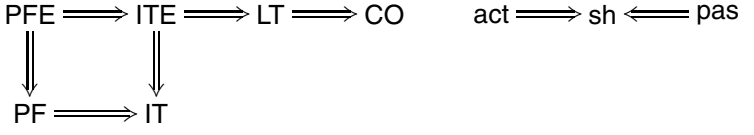
We refine these further by defining adversarial models, i.e. restrictions that we can impose on the adversaries and simulators for any of the above paradigms. We discuss active (act) adversaries, where adversaries and simulators in the classes $\mathcal{E}_{\mathsf{act}}$, $\mathcal{S}_{\mathsf{act}}$ are not restricted further; semi-honest (sh) adversaries[3], where adversaries in the class $\mathcal{E}_{\mathsf{sh}}$ are restricted to generate messages according to the prescribed protocol π with the inputs $x_{\mathfrak{E}}$ provided by the distinguisher D, and simulators in the class $\mathcal{S}_{\mathsf{sh}}$ are not restricted further; and passive (pas) adversaries[2], 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 $x_{\mathfrak{E}}$ provided by the distinguisher D to the ideal functionality I.

[7] 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.

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

We briefly motivate our definition of sh adversaries. When CO tools are applied to force active adversaries to behave passively, they can, in contrast to the pas setting, still substitute inputs. The sh setting is intended to model this scenario. 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. IT security against sh adversaries is described by $\mathcal{D}^{\mathsf{IT}}_{\mathsf{sh}} = \mathcal{D}^{\mathsf{IT}} \cap \mathcal{D}_{\mathsf{sh}}$, $\mathcal{S}^{\mathsf{IT}}_{\mathsf{sh}} = \mathcal{S}^{\mathsf{IT}} \cap \mathcal{S}_{\mathsf{sh}}$, $\mathcal{E}^{\mathsf{IT}}_{\mathsf{sh}} = \mathcal{E}^{\mathsf{IT}} \cap \mathcal{E}_{\mathsf{sh}}$, $\epsilon(\kappa) < \mathsf{negl}$ and denoted $\pi \succcurlyeq^{\mathsf{IT}}_{\mathsf{sh}} I$. By definition we have the following implications among security paradigms and adversarial models respectively:



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 function[9] and let $\mathfrak{C} \subset \mathfrak{P}$ be a set of corrupted players.

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 $y_{\mathfrak{C}}$ to the adversary (simulator). If $|\mathfrak{C}| > 0$, 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.[10]

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* (including robustness), 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.

[9] 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 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.

[10] 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.

Computing function f with a *designated aborter* (DA) is a slightly 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.

In the 2-party setting (but not for $n > 2$ parties) 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 can directly use it as implementation of I_f^{fair}. Thus robustness and fairness amount to the same:

Lemma 1. *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.*

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 to ignore the messages. By the same argument computability by public discussion and in the authenticated channels model *with BC* lead to identical results for active adversaries also.

Lemma 2. *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 passive, semi-honest, or 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 functions $f \in \mathfrak{F}$ that are computable IT securely in the authenticated channels model in presence of a passive adversary.

Definition 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.*

Note that by Lem. 2 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 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 (Characterization of $\mathfrak{F}_{\mathsf{loc}}$). *A function $f \in \mathfrak{F}$ is locally computable ($f \in \mathfrak{F}_{\mathsf{loc}}$) iff for every $i \in [n]$, $x_i \in \mathcal{X}_i$ 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 ($\mathfrak{F}'_{\mathsf{pas}}$: 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 $x_e \in \widetilde{\mathcal{X}}_e$ ($\mathfrak{E} := \{P_e\}$, $\mathfrak{H}' := \mathfrak{H} \setminus \{P_i\}$): $f_e(x_e, \widetilde{\mathcal{X}}_{\mathfrak{H}'}, \mathcal{X}'_i) \cap f_e(x_e, \widetilde{\mathcal{X}}_{\mathfrak{H}'}, \mathcal{X}''_i) = \emptyset$.*

The above definition only discusses adversary sets $\mathfrak{E}$ of cardinality $|\mathfrak{E}| = 1$. As we show next this is actually equivalent to quantifying over all sets $\mathfrak{E} \subseteq \mathfrak{P}$.

Lemma 4 (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 $x_{\mathfrak{E}} \in \widetilde{\mathcal{X}}_{\mathfrak{E}}$ ($\mathfrak{H}' := \mathfrak{H} \setminus \{P_i\}$): $f_{\mathfrak{E}}(x_{\mathfrak{E}}, \widetilde{\mathcal{X}}_{\mathfrak{H}'}, \mathcal{X}'_i) \cap f_{\mathfrak{E}}(x_{\mathfrak{E}}, \widetilde{\mathcal{X}}_{\mathfrak{H}'}, \mathcal{X}''_i) = \emptyset$.*

The proof of Lemma 4 is by induction over the size of the adversary set $\mathfrak{E}$ and can be found in the full version [22].

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

Theorem 1. *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 (in the security parameter κ) be computed with PFE security.*

The full proof of Thm. 1 can be found in [22]. $\mathfrak{F}_{\mathsf{pas}}^{\mathsf{aut}} \subseteq \mathfrak{F}'_{\mathsf{pas}}$ 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 [24] 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. 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 [24]) of the input domains in every round.

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 ($\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$: 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 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 [21]. 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 5 (Domination and Redundancy-Freeness). *Given an n-party function $f \in \mathfrak{F}_n$ we say $x_i \in \mathcal{X}_i$* dominates *$x'_i \in \mathcal{X}_i$ iff 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 $x_{\mathfrak{P}'}, x'_{\mathfrak{P}'} \in \mathcal{X}_{\mathfrak{P}'}$ (where $\mathfrak{P}' := \mathfrak{P} \setminus \{P_i\}$): $f_i(x'_i, x_{\mathfrak{P}'}) \neq f_i(x'_i, x'_{\mathfrak{P}'}) \implies f_i(x_i, x_{\mathfrak{P}'}) \neq f_i(x_i, x'_{\mathfrak{P}'})$.*

We define sets of dominating inputs $\widetilde{\mathcal{X}}_j := \{\mathcal{X} \subseteq \mathcal{X}_j \mid \forall x' \in \mathcal{X}_j \exists x \in \mathcal{X} : x \text{ dominates } x'\}$ ($j \in [n]$). Take 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 Sec. 8 or [22]). 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. 5 generalizes to the combined input of the corrupted parties $\mathfrak{E}$ as stated in Lem. 5 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 5. *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. 5 is by induction on $|\mathfrak{E}|$ and can be found in [22].

The following lemma states that the functions f and $\hat{f}$ are locally[11] 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 6. *The functions f and $\hat{f}$ are efficiently and PFE securely locally mutually reducible, even in presence of active adversaries.*

The proof of Lem. 6 is fairly straightforward, by showing how to implement $I_{\hat{f}}$ when I_f is given and vice versa. It can be found in [22]. One 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 1. *For any function $f \in \mathfrak{F}$ we have: $f \in \mathfrak{F}^{\text{aut}}_{\text{sh}} \iff \hat{f} \in \mathfrak{F}^{\text{aut}}_{\text{sh}}$.*

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

Theorem 2. *For a function $f \in \mathfrak{F}$ we have: $f \in \mathfrak{F}^{\text{aut}}_{\text{sh}} \iff \hat{f} \in \mathfrak{F}^{\text{aut}}_{\text{pas}}$.*

[11] without using any communication resources

$f^{(1)}$

	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	$8/a$	$9/b$	$8/c$

Fig. 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.

The full proof of Thm. 2 can be found in [22], we only give a sketch here. By Cor. 1 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. 1, 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 [24]. 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

We give a sufficient criterion for a function f to be in the class $\mathfrak{F}_{\mathsf{act}}^{\mathsf{bc}}$ of functions which can securely be computed by public discussion in presence of an unlimited active adversary. We conjecture that this criterion is also necessary and prove this fact for the 2-party case. As such we only obtain a full characterization of the class $\mathfrak{F}_{\mathsf{2act}}$ of actively computable 2-party functions, but this suffices 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 6 ($\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$: 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 $\pi \in$ 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 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 in the following assume an authenticated BC channel as the sole underlying resource.

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 7 ($\mathfrak{F}'_{\mathsf{act}}$: 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 $\mathfrak{C} \subseteq \mathfrak{P} \setminus \{P_i\}$ and $\mathfrak{H}' := \mathfrak{H} \setminus \{P_i\}$ we have

$$\forall \bar{x}_{\mathfrak{C}} \in \mathcal{X}_{\mathfrak{C}} : f_{\mathfrak{C}}(\bar{x}_{\mathfrak{C}}, \mathcal{X}_{\mathfrak{H}'}, \mathcal{X}'_i) \cap f_{\mathfrak{C}}(\bar{x}_{\mathfrak{C}}, \mathcal{X}_{\mathfrak{H}'}, \mathcal{X}''_i) = \emptyset \qquad \textit{(K-cut)} \qquad \textit{and}$$

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

$$\forall x'_i \in \mathcal{X}'_i : f_{\mathfrak{H}}(x'_{\mathfrak{C}}, x_{\mathfrak{H}'}, x'_i) = f_{\mathfrak{H}}(x_{\mathfrak{C}}, x_{\mathfrak{H}'}, x'_i) \qquad \wedge$$

$$\forall x''_i \in \mathcal{X}''_i : f_{\mathfrak{H}}(x''_{\mathfrak{C}}, x_{\mathfrak{H}'}, x''_i) = f_{\mathfrak{H}}(x_{\mathfrak{C}}, x_{\mathfrak{H}'}, x''_i)$$

Active decomposability indeed characterizes the actively computable functions:

Theorem 3. *A function $f \in \mathfrak{F}$ is actively computable if it is actively decomposable. In short $\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}} \supseteq \mathfrak{F}'_{\mathsf{act}}$. In the 2-party case[12] we even have $\mathfrak{F}_{\mathsf{2act}} \subseteq \mathfrak{F}'_{\mathsf{2act}}$, i.e. $\mathfrak{F}_{\mathsf{2act}} = \mathfrak{F}'_{\mathsf{2act}}$. Furthermore, any function $f \in \mathfrak{F}'_{\mathsf{act}}$ can be computed efficiently with PFE security.*

Furthermore, we conjecture:

Conjecture 1. $\mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}} = \mathfrak{F}'_{\mathsf{act}}$.

The full proof of Thm. 3 can be found in [22]. The implication $f \in \mathfrak{F}'_{\mathsf{act}} \implies f \in \mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$ 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 $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}}$. To obtain $f \in \mathfrak{F}_{\mathsf{2act}} \implies f \in \mathfrak{F}'_{\mathsf{2act}}$ we observe that for $f \notin \mathfrak{F}'_{\mathsf{2act}}$ the adversary can in any protocol induce an output 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 $f^{(2)} \in \mathfrak{F}_{\mathsf{2sh}}$ 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.

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

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 4. *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 [22]. 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.

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 8 ($\mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$, $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{ins,\,pki}}$, $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}}$: 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 $\pi \in$ 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 [18] and zero-knowledge arguments of knowledge:

Theorem 5. *If one-way functions (OWF) exist, we have* $\mathfrak{F}_{\mathsf{sh}}^{\mathsf{aut}} = \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$.

A full proof of Thm. 5 can be found in [22]. We show that the semi-honest to active protocol compiler of [16] can be applied to a semi-honestly secure protocol in such a way that it becomes CO secure against active adversaries, while maintaining IT security against semi-honest adversaries. Furthermore we claim:

Theorem 6. *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 [12] 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 [14] we can establish a PKI in the authenticated channels model.[13] $\mathfrak{F}_{\mathsf{lts}}^{\mathsf{aut}} \subseteq \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}}$ follows from Lem. 2.

Thm. 6 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 $f_{\mathrm{XOR}} \in \mathfrak{F}_{\mathsf{2sh}} \setminus \mathfrak{F}_{\mathsf{act}} \subset \mathfrak{F}_{\mathsf{lts}}^{\mathsf{bc}} \setminus \mathfrak{F}_{\mathsf{act}}^{\mathsf{aut}}$. Now a fair implementation of f_{XOR} would clearly imply a fair cointoss, which by [10] cannot be implemented in the model under consideration. As such the security without fairness as guaranteed by Thm. 6 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 $f \in \mathfrak{F}_{\mathsf{2lts}}$. 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 [15]. 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.

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

Lemma 7. *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 [22]. Essentially we apply the protocol compiler of [16] 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 [21], 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 [21] or [22]. 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 9. *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 [21]:

Theorem 7 (Complete Functions [21]). *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* $\exists\, a_1, a_2 \in \mathcal{X}_A,\ b_1, b_2 \in \mathcal{X}_B$ *:*

$$\exists\, a_1, a_2 \in \mathcal{X}_A, b_1, b_2 \in \mathcal{X}_B :\ f_A(a_1, b_1) = f_A(a_1, b_2) \ \wedge\ f_B(a_1, b_1) = f_B(a_2, b_1)$$
$$\wedge\ (\ f_A(a_2, b_1) \neq f_A(a_2, b_2) \ \vee f_B(a_1, b_2) \neq f_B(a_2, b_2)\ ).$$

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

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

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

$$\emptyset \neq \mathfrak{C}_{\mathsf{2act}} = \mathfrak{C}_{\mathsf{2sh}} \subsetneq \mathfrak{C}_{\mathsf{2pas}}$$

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 [21]. Additional details and examples can be found in [22].

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 $\mathfrak{F}^{\mathsf{ins,\ pki}}_{\mathsf{lts}}$ 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 $\mathfrak{F}^{\mathsf{aut}}_{\mathsf{pas}}$, $\mathfrak{F}^{\mathsf{aut}}_{\mathsf{sh}}$ and $\mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}}$ 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. $\mathfrak{F}^{\mathsf{aut}}_{\mathsf{sh}} = \mathfrak{F}^{\mathsf{bc}}_{\mathsf{lts}} = \mathfrak{F}^{\mathsf{aut}}_{\mathsf{lts}} = \mathfrak{F}^{\mathsf{ins,\ pki}}_{\mathsf{lts}}$, where the classes $\mathfrak{F}^{\mathsf{aut}}_{\mathsf{lts}}$ and $\mathfrak{F}^{\mathsf{bc}}_{\mathsf{lts}}$ are defined analogously to $\mathfrak{F}^{\mathsf{ins,\ pki}}_{\mathsf{lts}}$, 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 $\mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}} \subsetneq \mathfrak{F}^{\mathsf{ins,\ pki}}_{\mathsf{lts}}$, 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 $\mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}}$. As the functions in $\mathfrak{F}^{\mathsf{aut}}_{\mathsf{act}}$ are robustly (and therefore fairly) computable, these results can be interpreted along the lines Gordon et al. [17], 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 $\mathfrak{F}^{\mathsf{ins,\ pki}}_{\mathsf{lts}}$ fairness is generally not achievable. However, for the functions $\mathfrak{F}^{\mathsf{ins,\ pki}}_{\mathsf{lts}}$ 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 [4] 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 [25] or Kitaev [1]. 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 from the literature, especially [24,21], and adding the results of this work, we obtain a complete taxonomy of 2-party functions by computability and completeness in the IT setting.

Acknowledgments

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

References

1. Ambainis, A., Buhrman, H., Dodis, Y., Röhrig, H.: Multiparty quantum coin flipping. In: IEEE Conference on Computational Complexity, pp. 250–259. IEEE, Los Alamitos (2004)
2. Beaver, D.: Perfect privacy for two-party protocols. In: Proceedings of the DIMACS Workshop on Distributed Computing and Cryptography (1989)
3. Beimel, A., Malkin, T., Micali, S.: The all-or-nothing nature of two-party secure computation. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 80–97. Springer, Heidelberg (1999)
4. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: STOC 1988, pp. 1–10 (1988)
5. Brassard, G., Crépeau, C., Mayers, D., Salvail, L.: Defeating classical bit commitments with a quantum computer. Los Alamos preprint archive quant-ph/9806031 (May 1999)
6. Broadbent, A., Tapp, A.: Information-theoretic security without an honest majority. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 410–426. Springer, Heidelberg (2007)

7. Cachin, C., Crépeau, C., Marcil, J.: Oblivious transfer with a memory-bounded receiver. In: STOC 2002, pp. 493–502. ACM Press, New York (2002)
8. Cachin, C., Maurer, U.: Unconditional security against memory-bounded adversaries. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 292–306. Springer, Heidelberg (1997)
9. Chor, B., Kushilevitz, E.: A zero-one law for boolean privacy. In: STOC 1989 (1989)
10. Cleve, R.: Limits on the security of coin flips when half the processors are faulty. In: STOC 1986, pp. 364–369. ACM Press, New York (1986)
11. Damgård, I., Fehr, S., Salvail, L., Schaffner, C.: Cryptography in the bounded quantum-storage model. In: FOCS 2005, pp. 449–458. IEEE, Los Alamitos (2005)
12. Dolev, D., Strong, R.: Authenticated algorithms for byzantine agreement. SICOMP: SIAM Journal on Computing, 12 (1983)
13. Dziembowski, S., Maurer, U.: On generating the initial key in the bounded-storage model. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 126–137. Springer, Heidelberg (2004)
14. Fitzi, M., Hirt, M., Holenstein, T., Wullschleger, J.: Two-threshold broadcast and detectable multi-party computation. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 51–67. Springer, Heidelberg (2003)
15. Goldreich, O.: Foundations of Cryptography: Basic Applications, vol. 2. Cambridge University Press, Cambridge (2004)
16. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game — a completeness theorem for protocols with honest majority. In: STOC 1987, pp. 218–229 (1987)
17. Gordon, S.D., Hazay, C., Katz, J., Lindell, Y.: Complete fairness in secure two-party computation. In: STOC 2008, pp. 413–422. ACM, New York (2008)
18. Haitner, I., Reingold, O.: Statistically-hiding commitment from any one-way function. In: STOC 2007, pp. 1–10. ACM, New York (2007)
19. Kilian, J.: A general completeness theorem for two-party games. In: STOC 1991, pp. 553–560. ACM Press, New York (1991)
20. Kilian, J.: More general completeness theorems for secure two-party computation. In: STOC 2000, pp. 316–324. ACM Press, New York (2000)
21. Kraschewski, D., Müller-Quade, J.: Completeness theorems with constructive proofs for symmetric, asymmetric and general 2-party-functions (unpublished manuscript, 2008)
22. Künzler, R., Müller-Quade, J., Raub, D.: Secure computability of functions in the IT setting with dishonest majority and applications to long-term security. Cryptology ePrint Archive, Report 2008/264 (2008), http://eprint.iacr.org/2008/264
23. Kushilevitz, E.: Privacy and communication complexity. In: FOCS 1989, pp. 416–421. IEEE, Los Alamitos (1989)
24. Kushilevitz, E.: Privacy and communication complexity. SIAM Journal on Discrete Mathematics 5(2), 273–284 (1992)
25. Mayers, D.: Unconditionally secure bit commitment is impossible. Phys. Rev. Letters 78, 3414–3417 (1997)
26. Müller-Quade, J.: Temporary assumptions—quantum and classical. In: The 2005 IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security, pp. 31–33 (2005)
27. Müller-Quade, J., Unruh, D.: Long-term security and universal composability. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 41–60. Springer, Heidelberg (2007)
28. Rabin, M.: Hyper-encryption by virtual satellite. Science Center Research Lecture Series (2003)