Completeness Theorems with Constructive Proofs for Finite Deterministic 2-Party Functions

Daniel Kraschewski and Jörn Müller-Quade

Institute of Cryptography and Security, Faculty of Informatics,
Karlsruhe Institute of Technology, Germany
{kraschewski,mueller-quade}@kit.edu

Abstract. In this paper we present simple but comprehensive combinatorial criteria for completeness of finite deterministic 2-party functions with respect to information-theoretic security. We give a general protocol construction for efficient and statistically secure reduction of oblivious transfer to any finite deterministic 2-party function that fulfills our criteria. For the resulting protocols we prove universal composability. Our results are tight in the sense that our criteria still are necessary for any finite deterministic 2-party function to allow for implementation of oblivious transfer with statistical privacy and correctness.

We unify and generalize results of Joe Kilian (1991, 2000) in two ways. Firstly, we show that his completeness criteria also hold in the UC framework. Secondly, what is our main contribution, our criteria also cover a wide class of primitives that are not subject of previous criteria. We show that there are non-trivial examples of finite deterministic 2-party functions that are neither symmetric nor asymmetric and therefore have not been covered by existing completeness criteria so far.

As a corollary of our work, every finite deterministic 2-party function is either complete or can be considered equivalent to a non-complete symmetric 2-party function—this assertion holds true with respect to active adversaries as well as passive adversaries. Thereby known results on non-complete symmetric 2-party functions are strengthened.

Keywords: oblivious transfer, complete primitives, information-theoretic security, universal composability, secure function evaluation.

1 Introduction

Oblivious transfer in the sense of a trusted erasure channel (Rabin-OT) was introduced in [27] and later in [4] proven to be equivalent to $\binom{2}{1}$-OT, where a receiver Bob may learn only one of two bits sent by Alice. Oblivious transfer turned out to be complete in the sense that every secure multiparty computation can be implemented using OT [14,10,7,13]. Thereby, enduring interest in OT arised in cryptography and for numerous primitives it has been investigated, whether they allow for implementation of OT. In our work we exhaustively treat this question

Y. Ishai (Ed.): TCC 2011, LNCS 6597, pp. 364–381, 2011.

for the class of "finite deterministic 2-party functions", sometimes also referred to as "crypto gates". Such primitives are characterized by some finite alphabetes $\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}$ and some mappings $f_{\mathrm{A}} \in \Omega_{\mathrm{A}}^{\Upsilon_{\mathrm{A}} \times \Upsilon_{\mathrm{B}}}$, $f_{\mathrm{B}} \in \Omega_{\mathrm{B}}^{\Upsilon_{\mathrm{A}} \times \Upsilon_{\mathrm{B}}}$, such that on input $x \in \Upsilon_{\mathrm{A}}$ from Alice and $y \in \Upsilon_{\mathrm{B}}$ from Bob the primitive outputs $f_{\mathrm{A}}(x, y)$ to Alice and $f_{\mathrm{B}}(x, y)$ to Bob.

1.1 Related Work

In the literature one finds OT protocols for the bounded-classical-storage model [2] and the bounded-quantum-storage model [8] as well as noisy classical [6,30] and quantum channels [24,25], the latter taking commitments for granted. Further, there are reductions of $\binom{2}{1}$-OT to weaker OT versions that leak additional information [5,9,29] and to Rabin-OT [4]. OT-combiners implement OT from granted sets of OTs with faulty members [26,11]. For reversing the direction of $\binom{2}{1}$-OT a protocol is known with optimal number of OT calls [28]. Relative to complexity assumptions all-or-nothing laws have been shown [1,12,23], i.e. all non-trivial primitives are complete. Our work has several, nowadays folklore reduction techniques in common with all the aforementioned literature.

We unify and generalize the results of [15,16], where completeness criteria for symmetric (i.e. both parties receive the same output) and asymmetric (i.e. only one party learns the function output) 2-party functions were provided with respect to information-theoretic security. We import a main argument for the necessity of our criteria from [15]. Our sufficiency proof is independent from [15,16], since our results are more general and we use a very strict notion of security.

There are also results regarding whether various symmetric 2-party functions can be reduced to each other [22] and what can be implemented from scratch when there is only a passive adversary [21,20]. A corollary of our work extends all these results to non-symmetric primitives; some results of [20] already build on an early manuscript of our work [18].

1.2 Our Contribution

We expose a wide class of complete finite deterministic 2-party functions that are essentially neither symmetric nor asymmetric and hence are not subject of statistical completeness criteria in the literature so far. Further, by surprisingly simple combinatorial criteria to the respective function tables we give a precise characterization of *all* finite deterministic 2-party functions that allow for statistically secure implementation of OT. We provide an efficient and universally composable protocol scheme for OT from any finite deterministic 2-party function fulfilling our criteria. Our results are tight, as the necessity of our criteria still holds when only correctness and privacy of the implemented OT are required.

As a remarkable corollary of our work all non-complete finite deterministic 2-party functions turn out symmetric. This strengthens several known results for non-complete symmetric 2-party functions [21,22,20].

2 Presentation of Our Results

In this section we briefly present our results. Thereto, we first refer to the security notion that we use (Sec. 2.1), then introduce and motivate the notations needed for formulation of our results (Sec. 2.2) and, last but not least, state our completeness criteria in form of a Classification Theorem (Sec. 2.3).

2.1 Notion of Security

We prove our classification results in the UC framework [3] with static corruption and statistical security, i.e. the adversarial entities $\mathcal{A}, \mathcal{S}$ and the environment $\mathcal{Z}$ are computationally unbounded. Nonetheless, in our case the running time of a simulator $\mathcal{S}$ will always be polynomial in the running time of the according adversary $\mathcal{A}$. Since we implement $\binom{2}{1}$-OT from given 2-party functions, in the real model there always is a hybrid functionality that provides access to the latter (see Fig. 1). Since $\binom{2}{1}$-OT can be considered a special 2-party function that on input $(b_0, b_1) \in \{0,1\}^2$ from Alice and $c \in \{0,1\}$ from Bob outputs b_c to Bob and a special "nothing" symbol $\perp$ to Alice, we omit an explicit definition of the ideal functionality $\mathcal{F}_{\mathrm{OT}}$.

Functionality: $\mathcal{F}_{\mathrm{SFE}}^{(F)}$

Let F be characterized by the output functions $f_{\mathrm{A}} : \Upsilon_{\mathrm{A}} \times \Upsilon_{\mathrm{B}} \to \Omega_{\mathrm{A}}$ and $f_{\mathrm{B}} : \Upsilon_{\mathrm{A}} \times \Upsilon_{\mathrm{B}} \to \Omega_{\mathrm{B}}$, where $\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}$ are Alice's input and output alphabet and $\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}}$ are Bob's input and output alphabet.

- Upon receiving input (x, i) from Alice, verify that $(x, i) \in \Upsilon_{\mathrm{A}} \times \mathbb{N}$ and that there is no recorded tuple $(\tilde{x}, i, \texttt{Alice})$; else ignore that input. Next, record $(x, i, \texttt{Alice})$ and send $(\texttt{processing}, \texttt{Alice}, i)$ to the adversary.
- Upon receiving input (y, i) from Bob, verify that $(y, i) \in \Upsilon_{\mathrm{B}} \times \mathbb{N}$ and that there is no recorded tuple $(\tilde{y}, i, \texttt{Bob})$; else ignore that input. Next, record $(y, i, \texttt{Bob})$ and send $(\texttt{processing}, \texttt{Bob}, i)$ to the adversary.
- Upon receiving a message $(\texttt{Delivery}, \texttt{Alice}, i)$ from the adversary, verify that there are recorded tuples $(x, i, \texttt{Alice})$ and $(y, i, \texttt{Bob})$ and the former is not marked; else ignore that input. Next, mark the recorded tuple $(x, i, \texttt{Alice})$, compute $a \leftarrow f_{\mathrm{A}}(x, y)$ and output (a, i) to Alice.
- Upon receiving a message $(\texttt{Delivery}, \texttt{Bob}, i)$ from the adversary, verify that there are recorded tuples $(x, i, \texttt{Alice})$ and $(y, i, \texttt{Bob})$ and the latter is not marked; else ignore that input. Next, mark the recorded tuple $(y, i, \texttt{Bob})$, compute $b \leftarrow f_{\mathrm{B}}(x, y)$ and output (b, i) to Bob.

When a party is corrupted, the adversary is granted unrestricted access to the channel between $\mathcal{F}_{\mathrm{SFE}}^{(F)}$ and the corrupted party, including the ability of deleting and/or forging arbitrary messages.

Fig. 1. The ideal functionality for secure evaluation of a 2-party function F. Adapted and simplified version of the Secure Function Evaluation functionality in [3]. Note that via the parameter i only the same multi-session ability is achieved as in [3] by multiple session IDs.

	0	1	2
0	0/0	0/0	0/0
1	0/0	1/1	0/1
2	0/0	0/1	1/1

	0	1	2
0	0/0	1/1	0/1
1	0/0	0/0	0/0
2	$\top$/0	$\top$/1	$\bot$/1

Fig. 2. Function tables of two 2-party functions that are consistent renamings of each other (Alice's inputs label the rows, Bob's inputs label the columns; outputs are denoted a/b, meaning that Alice learns a and Bob learns b). We just interchanged the first two rows and applied an injective function to Alice's outputs in the third row; i.e. $\sigma_{\mathrm{A}}(0) = 1$, $\sigma_{\mathrm{A}}(1) = 0$, $\rho_{\mathrm{A}}(2,0) = (2,\top)$, $\rho_{\mathrm{A}}(2,1) = (2,\bot)$, everything else just is mapped to itself.

2.2 Basic Concepts

A finite deterministic 2-party function can be characterized by its input and output alphabets and output functions (q.v. Fig. 1). By $\mathfrak{F}_{\mathsf{fin,det}}$ we denote the set of all tuples $(\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, f_{\mathrm{A}}, f_{\mathrm{B}})$, where $\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}$ are non-empty finite alphabets and $f_{\mathrm{A}}, f_{\mathrm{B}}$ are mappings from $\Upsilon_{\mathrm{A}} \times \Upsilon_{\mathrm{B}}$ to Ω_{A} and from $\Upsilon_{\mathrm{A}} \times \Upsilon_{\mathrm{B}}$ to Ω_{B} respectively. For convenience we will not always differentiate pedantically between the mathematical object $F \in \mathfrak{F}_{\mathsf{fin,det}}$ and the corresponding primitive $\mathcal{F}_{\mathrm{SFE}}^{(F)}$, but from the context should be always clear what is meant.

Our notion of $\mathfrak{F}_{\mathsf{fin,det}}$ turns out a bit too detailed, since Alice and Bob can always relabel their input-output tuples of a given 2-party function without any side effects. There is no need for distinguishing between some $F \in \mathfrak{F}_{\mathsf{fin,det}}$ and any relabelled version of F. By the following definition we can abstract from those irrelevant details (q.v. Fig. 2).

Definition 1 (Consistent renamings). *Let* $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, f_{\mathrm{A}}, f_{\mathrm{B}}) \in \mathfrak{F}_{\mathsf{fin,det}}$ *and* $F' := (\Upsilon'_{\mathrm{A}}, \Upsilon'_{\mathrm{B}}, \Omega'_{\mathrm{A}}, \Omega'_{\mathrm{B}}, f'_{\mathrm{A}}, f'_{\mathrm{B}}) \in \mathfrak{F}_{\mathsf{fin,det}}$*. Then* F *and* F' *are* consistent renamings *of each other, if there exist some injective mappings* $\rho_{\mathrm{A}} : \Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}} \to \Upsilon'_{\mathrm{A}} \times \Omega'_{\mathrm{A}}$ *and* $\rho_{\mathrm{B}} : \Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}} \to \Upsilon'_{\mathrm{B}} \times \Omega'_{\mathrm{B}}$ *and some bijective mappings* $\sigma_{\mathrm{A}} : \Upsilon_{\mathrm{A}} \to \Upsilon'_{\mathrm{A}}$ *and* $\sigma_{\mathrm{B}} : \Upsilon_{\mathrm{B}} \to \Upsilon'_{\mathrm{B}}$*, such that for all* $x \in \Upsilon_{\mathrm{A}}$*,* $y \in \Upsilon_{\mathrm{B}}$ *it holds:*

$$\rho_{\mathrm{A}}\big(x, f_{\mathrm{A}}(x,y)\big) = \big(\sigma_{\mathrm{A}}(x), f'_{\mathrm{A}}(\sigma_{\mathrm{A}}(x), \sigma_{\mathrm{B}}(y))\big)$$
$$\rho_{\mathrm{B}}\big(y, f_{\mathrm{B}}(x,y)\big) = \big(\sigma_{\mathrm{B}}(y), f'_{\mathrm{B}}(\sigma_{\mathrm{A}}(x), \sigma_{\mathrm{B}}(y))\big)$$

Moreover, there may exist input symbols that are kind of "redundant" in the sense that an actively corrupted party can always input some corresponding "dominating" input symbols and at the same time perfectly simulate honest behaviour. This concept plays an important role in our proofs and results. We formally grasp it by the following definition.

Definition 2 (Redundancy). *Given* $F = (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, f_{\mathrm{A}}, f_{\mathrm{B}}) \in \mathfrak{F}_{\mathsf{fin,det}}$*, an input symbol* $y' \in \Upsilon_{\mathrm{B}}$ *is* redundant*, if there exists some corresponding* dominating *input symbol* $y \in \Upsilon_{\mathrm{B}} \backslash \{y'\}$*, such that the following two conditions hold:*

1. *For all* $x \in \Upsilon_{\mathrm{A}}$ *we have that* $f_{\mathrm{A}}(x,y) = f_{\mathrm{A}}(x,y')$*, i.e. from her own output Alice does never learn whether Bob did input* y *or* y'*.*

	0	1
0	0/0	0/0
1	0/0	0/1

	0	1
0	0/0	0/0
1	0/0	1/0

	0	1
0	0/0	0/0
1	0/0	1/1

Fig. 3. Function tables of the three different types of OT-cores (up to consistent renaming)

2. *For all* $x, x' \in \Upsilon_A$ *with* $f_B(x, y') \neq f_B(x', y')$ *we have that* $f_B(x, y) \neq f_B(x', y)$*, i.e. by inputting* y *instead of* y' *Bob gets exactly the same or strictly more information.*

For input symbols $x \in \Upsilon_A$ *redundancy is defined analogously. If neither* Υ_A *nor* Υ_B *contains any redundant input symbols,* F *is called* redundancy-free*.*

W.l.o.g. actively corrupted parties always use dominating input symbols instead of the corresponding redundant ones. Also, there is no need to constrain what honest parties may learn. Therefore, in presence of an active adversary we can consider any 2-party functions equivalent when they only differ in some redundant input symbols.

Definition 3 (Equivalence). *Let* $F := (\Upsilon_A, \Upsilon_B, \Omega_A, \Omega_B, f_A, f_B) \in \mathfrak{F}_{\mathsf{fin,det}}$ *and* $F' := (\Upsilon'_A, \Upsilon'_B, \Omega'_A, \Omega'_B, f'_A, f'_B) \in \mathfrak{F}_{\mathsf{fin,det}}$*. Then* F *and* F' *are called* equivalent*, if they can be transformed into consistent renamings of each other by successive*[1] *removal of redundant input symbols from* Υ_A, Υ_B*,* Υ'_A, Υ'_B *and according adjustment of* f_A, f_B, f'_A, f'_B*. Let* $[F]$ *denote the resulting equivalence class.*

Given $F \in \mathfrak{F}_{\mathsf{fin,det}}$, one can show quite easily that all redundancy-free $\bar{F}, \bar{F}' \in [F]$ are consistent renamings of each other, i.e. the redundancy-free version of F is unique up to consistent renaming.

2.3 The Classification Theorem

With the concepts from Sec. 2.2 we can now formulate our completeness criteria.

Definition 4 (Symmetric 2-party functions). *Let* $F' \in \mathfrak{F}_{\mathsf{fin,det}}$*. If* F' *is a consistent renaming of some* $F = (\Upsilon_A, \Upsilon_B, \Omega_A, \Omega_B, f_A, f_B) \in \mathfrak{F}_{\mathsf{fin,det}}$ *with* $\Omega_A = \Omega_B$ *and* $f_A = f_B$*, then* F' *is called* symmetric*.*

Definition 5 (OT-cores). *Let* $F := (\Upsilon_A, \Upsilon_B, \Omega_A, \Omega_B, f_A, f_B) \in \mathfrak{F}_{\mathsf{fin,det}}$*. Then a quadruple* $(x, x', y, y') \in \Upsilon_A^2 \times \Upsilon_B^2$ *is an* OT-core *of* F*, if the following three conditions are met (q.v. Fig. 3):*

1. *We have that* $f_A(x, y) = f_A(x, y')$*.*

[1] Note that a step-by-step removal of one symbol at a time is crucial here. There may exist distinct input symbols that dominate each other but must not be removed both.

	1
0	0/0
1	0/1

	0	1
1	0/0	1/0

	0	1
0	0/0	0/0
1	0/0	1/1

Fig. 4. Redundancy-free versions of the three different types of OT-cores (cf. Fig. 3), when there are no other input symbols around

2. *We have that* $f_{\mathrm{B}}(x,y) = f_{\mathrm{B}}(x',y)$.
3. *We have that* $f_{\mathrm{A}}(x',y) \neq f_{\mathrm{A}}(x',y')$ *or* $f_{\mathrm{B}}(x,y') \neq f_{\mathrm{B}}(x',y')$ *(or both).*

Theorem 1 (Classification theorem). *For each* $F \in \mathfrak{F}_{\mathsf{fin,det}}$ *it holds:*

1. *For the* $\mathcal{F}_{\mathrm{SFE}}^{(F)}$*-hybrid model there exists an OT protocol that is statistically secure against passive adversaries, iff* F *has an OT-core.*
2. *If for the* $\mathcal{F}_{\mathrm{SFE}}^{(F)}$*-hybrid model there does not exist any OT protocol that is statistically secure against passive adversaries, then* F *is symmetric.*
3. *For the* $\mathcal{F}_{\mathrm{SFE}}^{(F)}$*-hybrid model there exists an OT protocol that is statistically secure against active adversaries, iff the redundancy-free version of* F *has an OT-core.*
4. *If for the* $\mathcal{F}_{\mathrm{SFE}}^{(F)}$*-hybrid model there does not exist any OT protocol that is statistically secure against active adversaries, then the redundancy-free version of* F *is symmetric.*

Note that, when there is an active adversary, only the third function in Fig. 3 is complete on its own. The redundancy free versions of the other two functions just collapse to simple binary channels (q.v. Fig. 4). This collapsing can be prevented by additional input symbols. In Fig. 5 one can see, how OT-cores can be complemented to redundancy-free 2-party functions of minimum size.

For *symmetric* and *asymmetric* 2-party functions our completeness criteria coincide with the criteria from [15,16]. More concretely, we can directly translate the completeness criteria of [15,16] to our notations as follows.

Completeness criteria of [15]: A *symmetric* 2-party function F is complete, iff it has an OT-core. This holds true, regardless whether the adversary is active or passive.

Completeness criteria of [16]: Given an active adversary, an *asymmetric* 2-party function F' (with Bob being the receiver) is complete, iff for every input symbol $y \in \Upsilon_{\mathrm{B}}$ there exists some other input symbol $y' \in \Upsilon_{\mathrm{B}}$ that is not dominated by y; in other words, F' is complete, iff its redundancy-free version is non-trivial in the sense that both input alphabets have cardinality 2 or more. Given only a passive adversary, an *asymmetric* 2-party function F' is complete, iff it has an OT-core.

However, our criteria are much more comprehensive than that of [15,16], since ours also cover 2-party functions that are neither *symmetric* nor *asymmetric*. An illustrating example is the third function in Fig. 5, which is complete but not subject of the criteria in [15,16].

	0	1
0	0/0	0/0
1	0/0	0/1
2	0/1	0/0

	0	1
0	0/0	0/0
1	0/0	0/1
2	0/1	0/2
3	0/2	0/2

	0	1
0	0/0	0/0
1	0/0	0/1
2	0/1	1/2

	0	1
0	0/0	0/0
1	0/0	1/1

Fig. 5. Function tables of the four minimal complete 2-party functions. Up to consistent renaming and interchanging the roles of Alice and Bob every function table of a complete 2-party function $F \in \mathfrak{F}_{\mathsf{fin},\mathsf{det}}$ contains at least one of these examples as a submatrix.

3 How to Prove the Classification Theorem

In this section we give an intuitive exposition of how we prove our Classification Theorem. Due to space limitations we can only sketch the main ideas; for formal proofs we refer to the full version [19].

A fundamental tool in our proof strategy is the connection between presence of OT-cores and the question whether a 2-party function is symmetric.

Lemma 1 (Symmetrization lemma). *Each $F \in \mathfrak{F}_{\mathsf{fin},\mathsf{det}}$ that does not have any OT-core is symmetric (in the sense of Definition 4).*

One way to prove this lemma can be sketched as follows. Given a 2-party function $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, f_{\mathrm{A}}, f_{\mathrm{B}}) \in \mathfrak{F}_{\mathsf{fin},\mathsf{det}}$, we can define an equivalence relation on $(\Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}}) \cup (\Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}})$ induced as follows:

$$(x,a) \sim (y,b) \quad :\Leftarrow \quad f_{\mathrm{A}}(x,y) = a \ \wedge \ f_{\mathrm{B}}(x,y) = b$$

Let the according equivalence classes be denoted by $[x,a]$ or $[y,b]$. For all $x, x' \in \Upsilon_{\mathrm{A}}$, $a, a' \in \Omega_{\mathrm{A}}$ some simple induction yields the following implication (else F would have an OT-core):

$$(x,a) \sim (x',a') \quad \Rightarrow \quad \big\{y \in \Upsilon_{\mathrm{B}} \,\big|\, f_{\mathrm{A}}(x,y) = a\big\} = \big\{y \in \Upsilon_{\mathrm{B}} \,\big|\, f_{\mathrm{A}}(x',y) = a'\big\}$$

Thereby, we cannot find any $x \in \Upsilon_{\mathrm{A}}$, $a, a' \in \Omega_{\mathrm{A}}$ with $a \neq a'$ and $(x,a) \sim (x,a')$; the analog holds for $y \in \Upsilon_{\mathrm{A}}$, $b, b' \in \Omega_{\mathrm{A}}$. Hence, via the mappings $\rho_{\mathrm{A}} : (x,a) \mapsto \big(x, [x,a]\big)$ and $\rho_{\mathrm{B}} : (y,b) \mapsto \big(y, [y,b]\big)$ we get a consistent renaming of F and this consistent renaming is obviously symmetric.

By the Symmetrization Lemma and some results in the literature we can already argue for the assertions 1 and 2 of our Classification Theorem. On the one hand, when F has no OT-core, F can be considered symmetric by our Symmetrization Lemma. However, in [15] it has been shown that no reduction of OT to a symmetric 2-party function without OT-core can yield correctness and privacy at the same time, even if there is only a passive adversary—Alice can always exactly determine Bob's information about her inputs to the underlying 2-party function and vice versa.

	0	1	2	3
0	0/0	0/0	0/0	0/0
1	0/0	1/0	0/1	1/1
2	0/1	1/1	1/2	0/2

Fig. 6. A complete 2-party function that needs some carefully chosen, non-symmetric input distribution

On the other hand, when F has an OT-core and there is only a passive adversary, we can trivially implement one of the 2-party functions in Fig. 3. However, each of them can be transformed into a non-trivial noisy channel (shown to be complete in [6]) by the following protocol with expected 4 function calls. Alice first inputs a random bit b and then the inverse $\neg b$; Bob inputs independent random bits in both steps. The protocol is restarted until nowhere output "1" occurs. Afterwards Alice uses the last value of b as a one-time pad, which Bob knows with probability $\frac{2}{3}$.

Once assertion 1 of the Classification Theorem is shown, assertion 2 follows by the Symmetrization Lemma. Analogously assertion 4 follows from assertion 3, so all we have to do is proving assertion 3. One direction, the necessity of OT-cores, already follows from the passive case. Proving sufficiency for the active case is much more challenging and can be seen as our main contribution.

Our overall strategy for reducing OT in presence of an active adversary to a finite deterministic 2-party function having an OT-core proceeds in two steps. First, Alice and Bob generate some amount of correlated data by repeated invocation of the 2-party function with randomized input. Within a subsequent test step each party has to partially unveil its data, so that significant cheating can be detected. Then, on top of the remaining data an invocation of OT is built. In Sec. 3.1 we examine what input distributions are adequate and how the test step has to be performed. In Sec. 3.2 we construct a protocol for OT from such correlated data and we examine its security.

3.1 Secure Generation of Correlated Data

We start our examination with some negative example (see Fig. 6), which shows that choosing an adequate input distribution is not trivial. In the first place, the example in Fig. 6 shows that letting Alice and Bob use uniformly random input is not necessarily secure. In our example there would be an undetectable cheating strategy[2] for a corrupted Bob: He picks a uniformly random input symbol from $\{2,3\}$ instead of $\{0,1,2,3\}$ and after each invocation of the 2-party function with probability $\frac{1}{2}$ locally relabels his input-output tuple by $(2,0) \mapsto (0,0)$, $(2,1) \mapsto (0,0)$, $(2,2) \mapsto (1,1)$, $(3,0) \mapsto (1,0)$, $(3,1) \mapsto (1,0)$, $(3,2) \mapsto (0,1)$. Thereby he can perfectly simulate honest behaviour, but at the same time does learn all of Alice's inputs to the 2-party function.

[2] Note that such an undetectable cheating strategy cannot exist for symmetric 2-party functions, as there Alice will notice any change in Bob's output distribution.

We circumvent this problem by more asymmetric input distributions: We pick an OT-core and let the corresponding input symbols be input with relatively high probability, while all other input symbols have a relatively low probability and are only needed for the test step. However, the example in Fig. 6 also shows that we must choose the OT-core carefully. E.g. the OT-core in the upper left corner would be a bad choice, since the abovementioned cheating strategy can be adjusted to every protocol that assigns equal probability to Bob's input symbols "0" and "1". Still, significant cheating is possible for any input distribution with high probability for "0" and "1", as inputting "0" and "1" each once can be perfectly simulated by inputting "2" and "3" each once.

Actually, a main part of our work consists in proving that there always exists a "good" OT-core, if only the redundancy-free version of the considered 2-party function has any OT-core at all. In Fig. 6 one "good" OT-core corresponds to inputs $\{0, 1\}$ from Alice and $\{1, 2\}$ from Bob: By occasionally inputting "2" Alice can check that Bob does not too often use other input symbols than $\{1, 2\}$ (on input "2" she must not get output "0" too often) and that he does input "1" and "2" each with the right frequency (on input "1" she must get output "1" and "0" with according frequency), while Bob also sees Alice's actual input distribution (it is close to Bob's output distribution on input "2"). However, as the first function in Fig. 5 shows, in general it will not suffice that the participants only pay attention to their own input-output distributions. Since in this example Alice's output always is "0", only by unveiling some random subset of his input-output tuples Bob can prove that he did use a prescribed input distribution; e.g. he will be caught cheating with high probability when he claims to have input "0" sufficiently often but can never distinguish whether Alice did input "0" or "2". Again, for a meaningful test it is necessary that Alice uses her complete input alphabet.

These examples motivate that always all input symbols should be used with some non-zero probability. In the following we first sketch our protocol for generation of correlated data, then we introduce some algebraic structure that abstractly represents how a corrupted party can deviate from the protocol; finally we argue that there always is an OT-core that is "robust" against all such cheating strategies.

Our protocol for generating correlated data basically proceeds as follows:

1. **Invocation of F:** Alice and Bob call the underlying 2-party function F with randomized input for k times (k being the security parameter) and record their respective input-output tuples. A protocol parameter assigns what probability mass functions are to be used.
2. **Control A:** Alice challenges Bob on some polynomial subset of the recorded data, where he has to reveal his input-output tuples. Alice aborts the protocol if Bob obviously lies (i.e. his announcement is inconsistent with Alice's recorded input-output tuples) or his input distribution appears faulty. The test set is then removed from the recorded data.
3. **Control B:** This step equals the previous one with interchanged roles of Alice and Bob.

4. **Output:** Both parties announce where they have used input symbols that were only for test purposes. All corresponding elements are removed from the recorded input-output tuples by both parties. When too much of the recorded data has been deleted, the protocol is aborted; else each party outputs its remaining string of recorded input-output tuples.

We call this scheme *offline protocol*, since after the protocol step **Invocation of** F never again access to F is needed.

At this point we want to emphasize that although offline protocols are not completely symmetric in Alice and Bob, all of our arguments are. This convenient circumstance is predicated on the fact that a corrupted party only can get some polynomially small advantage by adversarial choice of the test set in protocol step **Control A** or **Control B** respectively. Our protocol in Sec. 3.2 for reduction of OT to correlated data is robust against such polynomially small advantages.

Now we define and investigate a class of functions $\eta : \Upsilon_{\mathrm{A}} \times \Upsilon_{\mathrm{B}}^2 \to \mathbb{R}_{\geq 0}$ that characterize how a corrupted Bob may cheat in an offline protocol. For symmetry reasons our results will directly carry over to the case that Alice is corrupted. Our intuition is that $\eta(x, y, y')$ quantifies the relative frequency of events in protocol step **Control A**, where F was invoked with input (x, y), but Bob successfully claims that he did input y'. We call such functions *cheating situations*. For convenience we use the notation $\eta(X, Y, Y') := \sum_{x \in X, y \in Y, y' \in Y'} \eta(x, y, y')$ for any $X \subseteq \Upsilon_{\mathrm{A}}$, $Y, Y' \subseteq \Upsilon_{\mathrm{B}}$. Also for convenience, we speak of a situation $(x, y)_F$ when we mean that F was called with input x from Alice and input y from Bob. We have the following six conditions to cheating situations:

1. It holds that $\eta(\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Upsilon_{\mathrm{B}}) = 1$.
2. For all $x \in \Upsilon_{\mathrm{A}}$ it holds that $\eta(x, \Upsilon_{\mathrm{B}}, \Upsilon_{\mathrm{B}}) > 0$, i.e. Alice did use her complete input alphabet.
3. For all $x \in \Upsilon_{\mathrm{A}}$, $y \in \Upsilon_{\mathrm{B}}$ it holds that $\eta(x, y, \Upsilon_{\mathrm{B}}) = \eta(x, \Upsilon_{\mathrm{B}}, \Upsilon_{\mathrm{B}}) \cdot \eta(\Upsilon_{\mathrm{A}}, y, \Upsilon_{\mathrm{B}})$, i.e. Bob's actual input distribution is independent of Alice's input distribution.
4. For all $x \in \Upsilon_{\mathrm{A}}$, $y' \in \Upsilon_{\mathrm{B}}$ it holds that $\eta(x, \Upsilon_{\mathrm{B}}, y') = \eta(x, \Upsilon_{\mathrm{B}}, \Upsilon_{\mathrm{B}}) \cdot \eta(\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, y')$, i.e. Bob's claimed input distribution appears independent of Alice's input distribution.
5. (a) For all $x \in \Upsilon_{\mathrm{A}}$, $y, y' \in \Upsilon_{\mathrm{B}}$ with $f_{\mathrm{A}}(x, y) \neq f_{\mathrm{A}}(x, y')$ it holds that $\eta(x, y, y') = 0$; else in the test step **Control A** Bob would be caught cheating immediately.
   (b) For all $x, x' \in \Upsilon_{\mathrm{A}}$, $y, y' \in \Upsilon_{\mathrm{B}}$ that fulfill $f_{\mathrm{B}}(x, y) = f_{\mathrm{B}}(x', y)$ and $f_{\mathrm{B}}(x, y') \neq f_{\mathrm{B}}(x', y')$, it holds that $\eta(x, y, y') = \eta(x', y, y') = 0$; else Bob would run an overwhelming risk of being caught cheating, since he cannot distinguish between situations $(x, y)_F$ and $(x', y)_F$ but must perfectly distinguish between these situations in the test step **Control A**.

Given some 2-party function $F \in \mathfrak{F}_{\mathsf{fin},\mathsf{det}}$, the set $\mathfrak{N}_F$ of all according cheating sitations has a very handy algebraic structure. On the one hand, cheating situations can be considered independent of (honest) Alice's input distribution, since they can canonically be rescaled to every input distribution that has non-zero

probability for all $x \in \Upsilon_A$. On the other hand, when we fix Alice's input distribution, i.e. for all $x \in \Upsilon_A$ the $\eta(x, \Upsilon_B, \Upsilon_B)$ are fixed, then our six conditions can be subsumed by a linear equation system, i.e. the set of all remaining cheating situations is a convex and bounded polytope in the linear space $\mathbb{R}^{\Upsilon_A \times \Upsilon_B^2}$.

Also the abovementioned conditions 5a and 5b play a fundamental role in our proofs. Therefore we sum them up by an extra notation. Given $F = (\Upsilon_A, \Upsilon_B, \Omega_A, \Omega_B, f_A, f_B) \in \mathfrak{F}_{\mathsf{fin,det}}$ and $x \in \Upsilon_A$, $y, y' \in \Upsilon_B$, let $(x, y) \overset{F}{\rightsquigarrow} (x, y')$ denote that the following two conditions are fulfilled:

- It holds that $f_A(x, y) = f_A(x, y')$.
- For all $\tilde{x} \in \Upsilon_A$ with $f_B(x, y) = f_B(\tilde{x}, y)$ it holds that $f_B(x, y') = f_B(\tilde{x}, y')$.

The intuition behind that notation is that Bob can claim a situation $(x, y)_F$ to be a situation $(x, y')_F$, iff $(x, y) \overset{F}{\rightsquigarrow} (x, y')$. At least he cannot do so too often, if $(x, y) \overset{F}{\not\rightsquigarrow} (x, y')$. For all cheating situations η and all $x \in \Upsilon_A$, $y, y' \in \Upsilon_B$ with $(x, y) \overset{F}{\not\rightsquigarrow} (x, y')$ it holds that $\eta(x, y, y') = 0$.

Note that the "$\overset{F}{\rightsquigarrow}$"-relation links cheating situations to redundancy matters, since an input symbol $y' \in \Upsilon_B$ is redundant, iff there exists some $y \in \Upsilon_B \setminus \{y'\}$ with $(x, y) \overset{F}{\rightsquigarrow} (x, y')$ for all $x \in \Upsilon_A$. In other words, the "$\overset{F}{\rightsquigarrow}$"-relation describes some kind of "local redundancy".

Given that Alice is uncorrupted, for every non-aborted run of an offline protocol there exists with overwhelming probability some cheating situation η, such that up to some polynomially small error the mappings $(x, y) \mapsto \eta(x, \Upsilon_B, y)$ and $(x, y) \mapsto \eta(x, y, \Upsilon_B)$ describe the prescribed and the actual joint input distribution to the underlying 2-party function respectively. Thus we have to look for some kind of "robust" OT-cores $(\tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}')$, so that there does not exist any essentially non-trivial cheating situation η with $\eta(\Upsilon_A, \Upsilon_B, \{\tilde{y}, \tilde{y}'\}) = 1$. More concretely, we will show that whenever a redundancy-free 2-party function $F \in \mathfrak{F}_{\mathsf{fin,det}}$ has any OT-core $(\tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}')$, then F also has an OT-core $(\tilde{x}, \tilde{x}', \bar{y}, \bar{y}')$, such that for every cheating situation η with $\eta(\Upsilon_A, \Upsilon_B, \{\bar{y}, \bar{y}'\}) = 1$ it holds that $\eta(\Upsilon_A, \Upsilon_B, y) = \eta(\Upsilon_A, y, \Upsilon_B)$ for all $y \in \Upsilon_B$, i.e. Bob practically cannot lie about his actual input distribution when he is demanded to use no other input symbols than $\bar{y}, \bar{y}'$.

Note that Alice's input symbols $\tilde{x}, \tilde{x}'$ have remained the same; hence in a second step we can analogously find an OT-core $(\bar{x}, \bar{x}', \bar{y}, \bar{y}')$ that is also "robust" against all relevant cheating attempts of Alice and stays "robust" against a possibly malicious Bob.

Given an OT-core $(\tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}')$ of a redundancy-free 2-party function $F \in \mathfrak{F}_{\mathsf{fin,det}}$, we can find an OT-core with the desired "robustness" by just picking some $\bar{y}, \bar{y}' \in \Upsilon_B$, such that $(\tilde{x}, \tilde{x}', \bar{y}, \bar{y}')$ is an OT-core and the following set has minimum size (q.v. Fig. 7):

$$\Phi(\bar{y}, \bar{y}') \;:=\; \left\{ y \in \Upsilon_B \;\middle|\; \forall\, x \in \Upsilon_A : (x, y) \overset{F}{\rightsquigarrow} (x, \bar{y}) \vee (x, y) \overset{F}{\rightsquigarrow} (x, \bar{y}') \right\}$$

Intuitively spoken, within an offline protocol that assigns high input probability only to $\bar{y}, \bar{y}'$ Bob cannot use any input symbol $y \in \Upsilon_B \setminus \Phi(\bar{y}, \bar{y}')$ too often; at

	0	1	2	3	4	5
0	0/0	0/0	0/0	0/0	0/0	1/∗
1	0/0	1/0	1/0	0/0	0/1	∗/∗
2	0/1	0/1	0/1	0/1	0/2	∗/∗
3	0/1	0/1	0/1	0/2	0/2	∗/∗
4	0/2	0/1	0/1	0/2	0/2	∗/∗
5	0/3	0/2	0/2	0/3	0/3	∗/∗
6	0/3	0/2	0/3	0/3	0/3	∗/∗
7	0/3	0/3	0/3	0/3	0/3	∗/∗

Fig. 7. Example for illustration of the construction of Φ and Y, Y'. From the first two rows one can infer that $(0,1,0,1)$ is an OT-core and $\Phi(0,1) \subseteq \{0,1,2,3,4\}$, regardless of the wildcards "∗". The other six rows just make the function redundancy-free, but still allow that $\Phi(0,1) \supseteq \{0,1,2,3,4\}$. Thereby, for the OT-core in the upper left corner we have that $\Phi(0,1) = \{0,1,2,3,4\}$ and $Y = \{0,3\}$ and $Y' = \{1,2,4\}$. However, we would not pick this OT-core but $(0,1,0,4)$ or $(0,1,3,4)$ instead, since $\Phi(0,4) = \Phi(3,4) = \{0,3,4\} \subsetneq \Phi(0,1)$, as Alice can distinguish between $\{0,3,4\}$ and $\{1,2\}$ by her output in the second row. Note that analogously $\Phi(1,2) = \{1,2\}$, but $(0,1,1,2)$ is not an OT-core.

least for some specific $x \in \Upsilon_{\mathrm{A}}$ he practically cannot claim a situation $(x,y)_F$ to be $(x,\bar{y})_F$ or $(x,\bar{y}')_F$ without being caught cheating. In general it will not necessarily hold that $\Phi(\bar{y},\bar{y}') = \{\bar{y},\bar{y}'\}$, nonetheless we can show now that the chosen OT-core $(\tilde{x},\tilde{x}',\bar{y},\bar{y}')$ is "robust" in the abovementioned sense. So, let some arbitrary cheating situation η with $\eta(\Upsilon_{\mathrm{A}},\Upsilon_{\mathrm{B}},\{\bar{y},\bar{y}'\}) = 1$ be given. By the following eight steps we show that $\eta(\Upsilon_{\mathrm{A}},\Upsilon_{\mathrm{B}},y) = \eta(\Upsilon_{\mathrm{A}},y,\Upsilon_{\mathrm{B}})$ for all $y \in \Upsilon_{\mathrm{B}}$.

1. Since the "$\overset{F}{\rightsquigarrow}$"-relation is transitive, we observe that $\Phi(y,y') \subseteq \Phi(\bar{y},\bar{y}')$ for all $y, y' \in \Phi(\bar{y},\bar{y}')$.
2. We want to exploit the minimality of $\Phi(\bar{y},\bar{y}')$, but it yields that $\left|\Phi(\bar{y},\bar{y}')\right| \leq \left|\Phi(y,y')\right|$ only in case that $(\tilde{x},\tilde{x}',y,y')$ is an OT-core. However, note that $f_{\mathrm{A}}(\tilde{x},\bar{y}) = f_{\mathrm{A}}(\tilde{x},\bar{y}')$, since $(\tilde{x},\tilde{x}',\bar{y},\bar{y}')$ is an OT-core. Furthermore, for all $y \in \Phi(\bar{y},\bar{y}')$ by definition of Φ we especially have that $(\tilde{x},y) \overset{F}{\rightsquigarrow} (\tilde{x},\bar{y})$ or $(\tilde{x},y) \overset{F}{\rightsquigarrow} (\tilde{x},\bar{y}')$, what in turn implies that $f_{\mathrm{A}}(\tilde{x},y) = f_{\mathrm{A}}(\tilde{x},\bar{y})$ or $f_{\mathrm{A}}(\tilde{x},y) = f_{\mathrm{A}}(\tilde{x},\bar{y}')$. Putting things together, we can conclude that $f_{\mathrm{A}}(\tilde{x},y) = f_{\mathrm{A}}(\tilde{x},y')$ for all $y, y' \in \Phi(\bar{y},\bar{y}')$. Therefore, by the following construction we can split $\Phi(\bar{y},\bar{y}')$ into disjoint subsets Y, Y', such that $(\tilde{x},\tilde{x}',y,y')$ actually is an OT-core for all $y \in Y$, $y' \in Y'$. We define (q.v. Fig. 7):

$$Y := \left\{y \in \Phi(\bar{y},\bar{y}') \;\middle|\; f_{\mathrm{A}}(\tilde{x}',\bar{y}) = f_{\mathrm{A}}(\tilde{x}',y) \;\wedge\; f_{\mathrm{B}}(\tilde{x},y) = f_{\mathrm{B}}(\tilde{x}',y)\right\}$$
$$Y' := \left\{y' \in \Phi(\bar{y},\bar{y}') \;\middle|\; f_{\mathrm{A}}(\tilde{x}',\bar{y}) \neq f_{\mathrm{A}}(\tilde{x}',y') \;\vee\; f_{\mathrm{B}}(\tilde{x},y') \neq f_{\mathrm{B}}(\tilde{x}',y')\right\}$$

   Now, by the minimality of $\Phi(\bar{y},\bar{y}')$ and our observation in step 1 it follows that $\Phi(\bar{y},\bar{y}') = \Phi(y,y')$ for all $y \in Y$, $y' \in Y'$.

3. Now, for each $(x, \hat{y}) \in \Upsilon_A \times \Phi(\bar{y}, \bar{y}')$ at least one of the following assertions must hold true:

$$\forall y \in Y : (x, \hat{y}) \overset{F}{\rightsquigarrow} (x, y) \qquad \forall y' \in Y' : (x, \hat{y}) \overset{F}{\rightsquigarrow} (x, y')$$

   Otherwise we had some $x \in \Upsilon_A$, $\hat{y} \in \Phi(\bar{y}, \bar{y}')$, $y \in Y$, $y' \in Y'$, such that $(x, \hat{y}) \overset{F}{\not\rightsquigarrow} (x, y)$ and $(x, \hat{y}) \overset{F}{\not\rightsquigarrow} (x, y')$ and thereby $\hat{y} \notin \Phi(y, y')$, what would be a contradiction to $\hat{y} \in \Phi(\bar{y}, \bar{y}') = \Phi(y, y')$ (cf. the final sentence of step 2).
4. For every $\hat{y} \in \Phi(\bar{y}, \bar{y}') \setminus \{\bar{y}\}$ we find some $x \in \Upsilon_A$, such that $(x, \hat{y}) \overset{F}{\not\rightsquigarrow} (x, \bar{y})$ and $\forall y' \in Y' : (x, y') \overset{F}{\not\rightsquigarrow} (x, \bar{y})$.

   This follows from our observation in step 3, F being redundancy-free and the transitivity of the "$\overset{F}{\rightsquigarrow}$"-relation. Since F is redundancy-free, we find some $x \in \Upsilon_A$, such that $(x, \hat{y}) \overset{F}{\not\rightsquigarrow} (x, \bar{y})$. This not only is one part of the claim above, but it also yields by step 3 that $(x, \hat{y}) \overset{F}{\rightsquigarrow} (x, y')$ for all $y' \in Y'$, since $\bar{y} \in Y$ by construction of Y. Now, if we could find any $y' \in Y'$ with $(x, y') \overset{F}{\rightsquigarrow} (x, \bar{y})$, in contradiction to our choice of x this would imply that $(x, \hat{y}) \overset{F}{\rightsquigarrow} (x, \bar{y})$, due to the transitivity of the "$\overset{F}{\rightsquigarrow}$"-relation.
5. For all $\hat{y} \in \Phi(\bar{y}, \bar{y}') \setminus \{\bar{y}\}$ we have that $\eta(\Upsilon_A, Y \setminus \{\hat{y}\}, \Upsilon_B) \geq \eta(\Upsilon_A, \Upsilon_B, \bar{y})$, i.e. Bob's claimed input frequency of $\bar{y}$ cannot be greater than his actual overall input frequency of symbols in $Y \setminus \{\hat{y}\}$.

   Otherwise we could find some $\hat{y} \in \Phi(\bar{y}, \bar{y}') \setminus \{\bar{y}\}$, such that $\eta(x, \Upsilon_B, \bar{y}) > \eta(x, Y \setminus \{\hat{y}\}, \Upsilon_B)$ for all $x \in \Upsilon_A$ (cf. the conditions 3 and 4 to cheating situations). However, by step 4 we can choose x such that Bob cannot claim any situation $(x, y')_F$ with $y' \in Y' \cup \{\hat{y}\}$ to be a situation $(x, \bar{y})_F$; the same holds for $y' \in \Upsilon_B \setminus \Phi(\bar{y}, \bar{y}')$ by definition of Φ. He may do so only for situations $(x, y')_F$ with $y' \in Y \setminus \{\hat{y}\}$, but these are too few, as $\eta(x, \Upsilon_B, \bar{y}) > \eta(x, Y \setminus \{\hat{y}\}, \Upsilon_B)$.
6. We observe that $\eta(\Upsilon_A, \Upsilon_B \setminus \Phi(\bar{y}, \bar{y}'), \Upsilon_B) = 0$, since $\eta(\Upsilon_A, \Upsilon_B, \{\bar{y}, \bar{y}'\}) = 1$ by assumption, i.e. $\eta(\Upsilon_A, \Upsilon_B, \Upsilon_B \setminus \{\bar{y}, \bar{y}'\}) = 0$, and $\eta(\Upsilon_A, \Upsilon_B \setminus \Phi(\bar{y}, \bar{y}'), \{\bar{y}, \bar{y}'\}) = 0$ by construction of Φ.
7. For every $\hat{y}' \in \Phi(\bar{y}, \bar{y}') \setminus \{\bar{y}'\}$ we have that $\eta(\Upsilon_A, Y \cup \{\hat{y}'\}, \Upsilon_B) \leq \eta(\Upsilon_A, \Upsilon_B, \bar{y})$, i.e. Bob's claimed input frequency of $\bar{y}$ cannot be less than his actual overall input frequency of symbols in $Y \cup \{\hat{y}'\}$.

   Since the assertion of step 3 is symmetric in Y and Y', analogously to step 4 for every $\hat{y}' \in \Phi(\bar{y}, \bar{y}') \setminus \{\bar{y}'\}$ we find some $x \in \Upsilon_A$, such that $\forall y \in Y \cup \{\hat{y}'\} : (x, y) \overset{F}{\not\rightsquigarrow} (x, \bar{y}')$. We can use that to prove the analog of step 5: For every $\hat{y}' \in \Phi(\bar{y}, \bar{y}') \setminus \{\bar{y}'\}$ we have that $\eta(\Upsilon_A, Y' \setminus \{\hat{y}'\}, \Upsilon_B) \geq \eta(\Upsilon_A, \Upsilon_B, \bar{y}')$. Moreover, we have that $\eta(\Upsilon_A, \Phi(\bar{y}, \bar{y}'), \Upsilon_B) = 1$ by step 6 and that $\eta(\Upsilon_A, \Upsilon_B, \{\bar{y}, \bar{y}'\}) = 1$ by assumption. Conclusively, for all $\hat{y}' \in \Phi(\bar{y}, \bar{y}') \setminus \{\bar{y}'\}$ we get that $\eta(\Upsilon_A, \Upsilon_B, \bar{y}) = 1 - \eta(\Upsilon_A, \Upsilon_B, \bar{y}') \geq 1 - \eta(\Upsilon_A, Y' \setminus \{\hat{y}'\}, \Upsilon_B) = \eta(\Upsilon_A, Y \cup \{\hat{y}'\}, \Upsilon_B)$.
8. By combination of step 5 and step 7, for all $\hat{y}, \hat{y}' \in \Phi(\bar{y}, \bar{y}')$ with $\hat{y}' \neq \bar{y}'$ and $\hat{y} \neq \bar{y}$ we can now conclude that $\eta(\Upsilon_A, Y \cup \{\hat{y}'\}, \Upsilon_B) \leq \eta(\Upsilon_A, Y \setminus \{\hat{y}\}, \Upsilon_B)$. This can be exploited as follows. On the one hand, we can choose $\hat{y} = \bar{y}'$, i.e. $Y \setminus \{\hat{y}\} = Y$, whereby for all $\hat{y}' \in Y' \setminus \{\bar{y}'\}$ it follows that $\eta\big(\Upsilon_A, \hat{y}', \Upsilon_B\big) \leq 0$, i.e. $\eta\big(\Upsilon_A, Y' \setminus \{\bar{y}'\}, \Upsilon_B\big) = 0$. On the other hand, we can choose $\hat{y}' = \bar{y}$, i.e.

$Y \cup \{\hat{y}'\} = Y$, whereby for all $\hat{y} \in Y \backslash \{\bar{y}\}$ it follows that $\eta\big(\Upsilon_{\mathrm{A}}, \hat{y}, \Upsilon_{\mathrm{B}}\big) \leq 0$, i.e. $\eta\big(\Upsilon_{\mathrm{A}}, Y \backslash \{\bar{y}\}, \Upsilon_{\mathrm{B}}\big) = 0$. Conclusively, using that $\eta(\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}} \backslash \Phi(\bar{y}, \bar{y}'), \Upsilon_{\mathrm{B}}) = 0$ by step 6, we get that $\eta\big(\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}} \backslash \{\bar{y}, \bar{y}'\}, \Upsilon_{\mathrm{B}}\big) = 0$, i.e. $\eta\big(\Upsilon_{\mathrm{A}}, \{\bar{y}, \bar{y}'\}, \Upsilon_{\mathrm{B}}\big) = 1$. Now, since $\eta\big(\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \{\bar{y}, \bar{y}'\}\big) = 1$ by assumption and neither $\bar{y}$ nor $\bar{y}'$ is redundant, one can infer rather straightforwardly that $\eta(\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, y) = \eta(\Upsilon_{\mathrm{A}}, y, \Upsilon_{\mathrm{B}})$ for all $y \in \Upsilon_{\mathrm{B}}$, as claimed.

3.2 Reduction of OT to Correlated Data

We now sketch a protocol that implements OT from the correlated data produced by an appropriate offline protocol. Within this sketch we also informally argue for the protocol's security. Given a redundancy-free 2-party function F that has some OT-core $(\tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}')$, the protocol proceeds as follows:

0. W.l.o.g. we may assume that the OT-core $(\tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}')$ is of the first or last type in Fig. 3; else we interchange the roles of Alice and Bob. W.l.o.g. we also assume that Alice's and Bob's actual input and output symbols coincide with that of Fig. 3, i.e. $\tilde{x} = \tilde{y} = 0$ and so on. Furthermore, w.l.o.g. we assume that $(\tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}')$ is a "robust" OT-core, whose existence we have shown in Section 3.1.
1. Alice and Bob execute an offline protocol (as sketched in Section 3.1), where the probability mass functions n_{A} and n_{B} that stand for Alice's and Bob's prescribed input distribution respectively, are such that $n_{\mathrm{A}}(0) \approx \frac{1}{3}$ and $n_{\mathrm{A}}(1) \approx \frac{2}{3}$ and $n_{\mathrm{B}}(0) \approx n_{\mathrm{B}}(1) \approx \frac{1}{2}$. Note that in general these will not be the exact input probabilities, as for meaningful tests in the protocol steps **Control A** and **Control B** we still need all other inputs to be used with some polynomial frequency. However, for growing security parameter the relative frequency of the other inputs may polynomially converge to zero. Further note that even if a party is corrupted, its actual input distribution in non-aborted protocol runs must be polynomially close to honest behaviour, since $(\tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}')$ was chosen to be a "robust" OT-core.
2. We want to handle all possible types of OT-cores analogously, therefore we let Alice announce where she got output "1". All corresponding input-output tuples are deleted from the recorded data by both parties. When Alice tries to delete too little, Bob aborts the protocol. He also aborts the protocol when he has to delete some input-output tuple other than $\big(1, f_{\mathrm{B}}(1,1)\big)$. Since Alice cannot distinguish between situations $(0,0)_F$ and $(0,1)_F$, this forces her to play honestly up to some polynomially small fraction of the recorded data.
3. Now most of the remaining input-output tuples belong to situations $(0,0)_F$, $(0,1)_F$, $(1,0)_F$. Since all according outputs are "0", it suffices that Alice and Bob henceforth only keep track of their recorded input strings. Note that at this stage about one quarter of the remaining recorded data belongs to situations $(0,0)_F$, one quarter to $(0,1)_F$ and one half to $(1,0)_F$.
4. Alice deletes some elements from her recorded input string, such that afterwards the string is balanced (i.e. it contains the same number of "0"s

and "1"s). She annonces the corresponding indices to Bob, who deletes the according elements from his recorded data. If Alice tries to delete too much, Bob aborts the protocol.

5. Alice randomly permutes her recorded input string, such that afterwards each element at an odd possition is different from its subsequent element. She announces the permutation to Bob, who permutes his input string accordingly. Thereby their input strings become strings of pairs (each starting at an odd position), such that a pair "01" or "10" on Bob's side indicates the respective inverted pair "10" or "01" on Alice's side and a pair "00" on Bob's side gives him no information about the pair on Alice's side. If Bob finds a pair "11" (starting at an odd position), he aborts the protocol. Note that about half of Bob's pairs are "00", one quarter is "01" and one quarter is "10".

   Further note that primarily there is only one way Alice may get some additional information about where Bob has "00"-pairs: She chooses the permutation adversarially, so that some "11"-pairs are produced on her side. However, since her input string is roughly balanced since the beginning of step 3, she must produce roughly as much "00"-pairs as "11"-pairs on her side and for each "00"-pair she is caught cheating by Bob with probability $\frac{1}{2}$. So even a corrupted Alice may know at most polynomially few positions where Bob has "00"-pairs.
6. Since Bob now can reconstruct about half of Alice's input string and Alice has only few information about where exactly Bob can do that, we can treat the recorded data like the result of Rabin-OT calls and adapt standard reduction techniques[3]. To that effect we rename Alices input string into a string of half length over the alphabet $\{0,1\}$ and accordingly for Bob over the alphabet $\{0,1,\perp\}$; in particular the renaming is "01"$\mapsto$"0", "10"$\mapsto$"1" on Alice's side and "10"$\mapsto$"0", "01"$\mapsto$"1", "00"$\mapsto$"$\perp$" on Bob's side. When a party cheated, we can represent that by a special symbol "$\top$" in that party's string. However, the symbol "$\top$" may occur only with some polynomial relative frequency, say less than $k^{-\gamma}$. Let $\kappa := \lceil k^{1-\gamma} \rceil$.
7. Now, let $b_0, b_1 \in \{0,1\}$ be Alice's $\binom{2}{1}$-OT input and let $c \in \{0,1\}$ be Bob's choice bit. Alice chooses two random bit strings $\tilde{b}_0, \tilde{b}_1 \in \{0,1\}^\kappa$ with $\bigoplus_{j=1}^{\kappa} \tilde{b}_0[j] = b_0$ and $\tilde{b}_0[j] \oplus \tilde{b}_1[j] = b_0 \oplus b_1$ for $j = 1, \ldots, \kappa$. Bob chooses a random bit string $\tilde{c} \in \{0,1\}^\kappa$ with $\bigoplus_{j=1}^{\kappa} \tilde{c}[j] = c$.
8. Alice and Bob respectively partition their recorded input strings into κ consecutive substrings of equal length l with l as large as possible; remaining elements are just discarded. Let $\tilde{s}_{\mathrm{A}}^{(j)}$ denote Alices j-th substring and $\tilde{s}_{\mathrm{B}}^{(j)}$ Bob's j-th substring. Note that by our choice of κ at least one of the $\tilde{s}_{\mathrm{A}}^{(j)}$ does not contain the symbol "$\top$". Further note that for each $\tilde{s}_{\mathrm{B}}^{(j)}$ about half of the

[3] Note that due to a subtle issue we cannot directly apply the results of [5,9,29] for reduction of OT to weak OT; e.g. in our case a corrupted Alice can choose to learn some prefix of Bob's string. In contrast, weak OT does not allow the adversary to influence when exactly additional information is leaked.

contained elements equal "$\perp$", because of the permutation at the beginning of step 3.

For $j = 1, \ldots, \kappa$ now the following subprotocol is executed:

(a) Bob chooses some disjoint random sets $K_0^{(j)}, K_1^{(j)} \subseteq \{1, \ldots, l\}$ of equal cardinality $\lceil \frac{l}{3} \rceil$, such that no element of $\tilde{s}_{\mathrm{B}}^{(j)}$ indexed by $K_{\tilde{c}[j]}^{(j)}$ is "$\perp$". He announces $\left(K_0^{(j)}, K_1^{(j)}\right)$ to Alice. Note that Alice does not get any information about at least one of the $\tilde{c}[j]$, since the corresponding $\tilde{s}_{\mathrm{A}}^{(j)}$ does not contain the symbol "$\top$". Hence she stays ignorant of Bob's choice bit c.

(b) For $i = 0, 1$ Alice uses the XOR of the elements in $\tilde{s}_{\mathrm{A}}^{(j)}$ indexed by $K_i^{(j)}$ as a one-time pad for $\tilde{b}_i[j]$. She sends the according cyphertexts to Bob, who learns $\tilde{b}_{\tilde{c}[j]}[j]$ by reconstructing the needed one-time pad from $\tilde{s}_{\mathrm{B}}^{(j)}$. Note that for each j Bob cannot get some information about both bits $\tilde{b}_0[j], \tilde{b}_1[j]$ at the same time, since more than one third of the elements in $\tilde{s}_{\mathrm{B}}^{(j)}$ equals "$\perp$". Hence he may learn at most one of Alice's $\binom{2}{1}$-OT inputs b_0, b_1.

9. Alice outputs the nothing symbol "$\perp$" and Bob computes and outputs $b_c = \bigoplus_{j=1}^{\kappa} \tilde{b}_{\tilde{c}[j]}[j]$. Correctness of Bob's output can be shown by induction on the Hamming weight of $\tilde{c}$.

We conclude our work with some remarks about how one can prove universal composability of this protocol, i.e. that it is simulatable in the ideal model (q.v. Section 2.1). Access to the underlying 2-party function F is in the ideal model only simulated, so the simulator can compute all the $\tilde{s}_{\mathrm{A}}^{(j)}$ or $\tilde{s}_{\mathrm{B}}^{(j)}$ respectively and hence extract the OT input of a corrupted Alice or Bob. Moreover, when Bob is corrupted, the simulator can fake a real protocol run that matches the ideal Alice's inputs b_0, b_1 as follows: Just before step 8b is entered the κ-th time, the simulator inputs the extracted choice bit c into the ideal functionality $\mathcal{F}_{\mathrm{OT}}$, thus learning b_c, and then revises $\tilde{b}_0[\kappa]$ and $\tilde{b}_1[\kappa]$ accordingly.

4 Conclusion

In this paper we showed that there is a wide class of primitives that have not been covered by existing completeness criteria, namely all 2-party functions that are essentially neither symmetric nor asymmetric. We solved this open problem by presenting simple but comprehensive criteria that combinatorially classify all complete deterministic 2-party functions with finite input and output alphabets. We proved constructively that our criteria are sufficient in the UC framework, which is the most restrictive common notion of security we know. Our criteria also turn out necessary even with respect to very weak notions of security. Therefore we consider them valid for virtually all reasonable security notions.

A remarkable corollary of our work is that every non-complete deterministic 2-party function with finite input and output alphabets is essentially symmetric. Thereby we extended the results of [21,22,20] to non-symmetric 2-party functions. The questions treated there become trivial for complete primitives and

we have shown that every essentially non-symmetric 2-party function actually is complete.

Acknowledgements. We want to thank Mike Rosulek and the anonymous reviewers of TCC 2011 for helpful comments.

References

1. Beimel, A., Malkin, T., Micali, S.: The All-or-Nothing Nature of Two-Party Secure Computation. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 80–97. Springer, Heidelberg (1999)
2. Cachin, C., Crépeau, C., Marcil, J.: Oblivious transfer with a memory-bounded receiver. In: Proceedings of FOCS 2001, pp. 493–502 (1998)
3. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: Proceedings of FOCS 2001, pp. 136–145 (2001), revised version online `http://eprint.iacr.org/2000/067`
4. Crépeau, C.: Equivalence between Two Flavours of Oblivious Transfers. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 350–354. Springer, Heidelberg (1988)
5. Crépeau, C., Kilian, J.: Weakening security assumptions and oblivious transfer (abstract). In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 2–7. Springer, Heidelberg (1990)
6. Crépeau, C., Morozov, K., Wolf, S.: Efficient Unconditional Oblivious Transfer from Almost Any Noisy Channel. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 47–59. Springer, Heidelberg (2005)
7. Crépeau, C., van de Graaf, J., Tapp, A.: Committed Oblivious Transfer and Private Multi-party Computation. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 110–123. Springer, Heidelberg (1995)
8. Damgård, I., Fehr, S., Renner, R., Salvail, L., Schaffner, C.: A Tight High-Order Entropic Quantum Uncertainty Relation with Applications. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 360–378. Springer, Heidelberg (2007)
9. Damgård, I., Kilian, J., Salvail, L.: On the (Im)possibility of Basing Oblivious Transfer and Bit Commitment on Weakened Security Assumptions. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 56–73. Springer, Heidelberg (1999)
10. Goldwasser, S., Levin, L.A.: Fair Computation of General Functions in Presence of Immoral Majority. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 77–93. Springer, Heidelberg (1991)
11. Harnik, D., Ishai, Y., Kushilevitz, E., Nielsen, J.B.: OT-Combiners via Secure Computation. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 393–411. Springer, Heidelberg (2008)
12. Harnik, D., Naor, M., Reingold, O., Rosen, A.: Completeness in two-party secure computation: A computational view. Journal of Cryptology 19(4), 521–552 (2006)
13. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding Cryptography on Oblivious Transfer – Efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (2008)
14. Kilian, J.: Founding cryptography on oblivious transfer. In: Proceedings of STOC 1988, pp. 20–31. ACM, New York (1988)
15. Kilian, J.: A general completeness theorem for two-party games. In: Proceedings of STOC 1991, pp. 553–560. ACM, New York (1991)

16. Kilian, J.: More general completeness theorems for secure two-party computation. In: Proceedings of STOC 2000, pp. 316–324. ACM, New York (2000)
17. Kraschewski, D.: Vollständigkeitskriterien von kryptographischen Primitiven. Diploma thesis, Institut für Algorithmen und Kognitive Systeme, Universität Karlsruhe (2006)
18. Kraschewski, D., Müller-Quade, J.: Completeness theorems with constructive proofs for symmetric, asymmetric and general 2-party-functions. Unpublished manuscript of the present work with different and more complicated proof techniques, based on the first author's diploma thesis [17] (2008)
19. Kraschewski, D., Müller-Quade, J.: Completeness theorems with constructive proofs for finite deterministic 2-party functions (full version). Cryptology ePrint Archive, Report 2010/654 (2010), Full version of the present work, online available at `http://eprint.iacr.org/2010/654`
20. Künzler, R., Müller-Quade, J., Raub, D.: Secure Computability of Functions in the IT Setting with Dishonest Majority and Applications to Long-Term Security. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 238–255. Springer, Heidelberg (2009)
21. Kushilevitz, E.: Privacy and communication complexity. SIAM Journal on Discrete Mathematics 5(2), 273–284 (1992)
22. Maji, H.K., Prabhakaran, M., Rosulek, M.: Complexity of Multi-party Computation Problems: The Case of 2-Party Symmetric Secure Function Evaluation. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 256–273. Springer, Heidelberg (2009)
23. Maji, H.K., Prabhakaran, M., Rosulek, M.: A Zero-One Law for Cryptographic Complexity with Respect to Computational UC Security. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 595–612. Springer, Heidelberg (2010)
24. Mayers, D.: On the Security of the Quantum Oblivious Transfer and Key Distribution Protocols. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 124–135. Springer, Heidelberg (1995)
25. Mayers, D.: Quantum Key Distribution and String Oblivious Transfer in Noisy Channels. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 343–357. Springer, Heidelberg (1996)
26. Meier, R., Przydatek, B., Wullschleger, J.: Robuster Combiners for Oblivious Transfer. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 404–418. Springer, Heidelberg (2007)
27. Rabin, M.O.: How to exchange secrets by oblivious transfer. Technical report, Aiken Computation Laboratory, Harvard University (1981)
28. Wolf, S., Wullschleger, J.: Oblivious Transfer Is Symmetric. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 222–232. Springer, Heidelberg (2006)
29. Wullschleger, J.: Oblivious-transfer amplification. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 555–572. Springer, Heidelberg (2007)
30. Wullschleger, J.: Oblivious Transfer from Weak Noisy Channels. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 332–349. Springer, Heidelberg (2009)