Oblivious Transfer from Weak Noisy Channels

Jürg Wullschleger

University of Bristol, UK
j.wullschleger@bristol.ac.uk

Abstract. Various results show that oblivious transfer can be implemented using the assumption of *noisy channels*. Unfortunately, this assumption is not as weak as one might think, because in a cryptographic setting, these noisy channels must satisfy very strong security requirements.

Unfair noisy channels, introduced by Damgård, Kilian and Salvail [Eurocrypt '99], reduce these limitations: They give the adversary an unfair advantage over the honest player, and therefore weaken the security requirements on the noisy channel. However, this model still has many shortcomings: For example, the adversary's advantage is only allowed to have a very special form, and no error is allowed in the implementation.

In this paper we generalize the idea of unfair noisy channels. We introduce two new models of cryptographic noisy channels that we call the *weak erasure channel* and the *weak binary symmetric channel*, and show how they can be used to implement oblivious transfer. Our models are more general and use much weaker assumptions than unfair noisy channels, which makes implementation a more realistic prospect. For example, these are the first models that allow the parameters to come from experimental evidence.

1 Introduction

Secure two-party computation, introduced in [23], allows two mutually distrustful players to calculate a function in a secure way. This means that both players get the correct output, but nothing more than that. Even though secure two-party computation is generally impossible without any further assumption, it has been shown in [11,14] that if a very simple primitive called *oblivious transfer* is available, then any two-party computation can be implemented in an unconditionally secure way.

Oblivious transfer was first defined in [21], however without realizing its connection to cryptography. In the cryptographic context, the two variants of oblivious transfer were defined in [19] and [9], which were shown to be equally powerful in [3]. Throughout this work, we will only consider *chosen one-out-of-two oblivious transfer*, or OT for short. Here, a sender can send two message bits x_0 and x_1, and a receiver can choose which of the two messages he wants to receive by sending a choice bit c. He receives x_c, but does not get to know the other message bit x_{1-c}, and the sender does not get to know the choice bit c. There exist various implementations of OT that are secure against computationally bounded

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

O. Reingold (Ed.): TCC 2009, LNCS 5444, pp. 332–349, 2009.

adversaries, under various hardness assumptions. Against adversaries with unbounded computational power, OT can only be implemented if the players have access to an additional (weaker) functionality.

1.1 OT from (Unfair) Noisy Channels

In [5], it has been shown that OT can be implemented from various weaker forms of OT, as well as *noisy channels*. Therefore, noise is not always a bad thing; in a cryptographic context it can become a valuable resource. These protocols have later been improved and generalized in [4], [6] and [16]. The basic idea of all these protocols is very similar: First, they construct some kind of erasure channel. Then, this erasure channel is used many times to implement OT. The correctness and the security is guaranteed using error correcting codes and privacy amplification.

These noisy channels seem to be quite weak primitives and easily implementable, but they have some rather strong requirements: The statistics of the channel must be *exactly* the same in every instance, and known to both players. And, apart from the output of the channel, a dishonest player must not get *any* additional output.

In [8], weaker forms of noisy channels called *unfair noisy channels* were introduced. Unfair noisy channels are binary symmetric noisy channels that let the dishonest player change the error-rate in the channel by a certain amount. For example, this makes the protocol secure against an adversary that might use better transmitters or detectors in order to break the protocol. In this model, OT must be implemented in a different way, using the following two steps. First, from only a few instances of the channel, a weak form of OT (called WOT) is constructed. In the second step, the security is amplified, i.e., many of these WOTs are used to get one secure instance of OT. The resulting protocol is only secure in the *semi-honest model*, i.e., under the assumption that the dishonest player follows the protocol. To make the protocol secure in the *malicious model*, where the dishonest player may deviate in an arbitrary way from the protocol, a third step is needed, which uses *bit commitments* and *zero-knowledge proofs* to force the dishonest player to follow the protocol.

The results from [8] were later improved in [7], and OT amplification was improved in [22].

1.2 Limitations of Unfair Noisy Channels

Even though unfair noisy channels are much weaker than (fair) noisy channels, they still have some very strong assumptions, which makes them hard to implement. Let us look at the following example:

A (fair) binary symmetric noisy channel with error ε lets a sender input a bit $x \in \{0,1\}$. The channel then outputs a value $Y \in \{0,1\}$ to the receiver, where $\Pr[Y \neq x] = \varepsilon$. Let us assume that neither the sender nor the receiver can influence ε, but that the dishonest receiver gets an additional value $E \in \{0,1\}$, where $\Pr[E = 1] = \mu$, and $E = 1 \Rightarrow Y = x$. Therefore, with some probability μ,

the dishonest receiver gets to know that the value Y he received is in fact equal to x. If μ is small, then this channel is very close to a fair binary symmetric noisy channel. However, even then it cannot be modeled by an unfair noisy channel, because there, the receiver can only change the error probability of the channel in a certain range, but he can never be sure that his received bit is the bit sent by the sender. Therefore, unfair noisy channels forbid the adversary to have this kind of advantage.

Now, let us assume that we are given an implementation a noisy channel, and that the statistics of the channel show that the channel behaves like a fair noisy channel. The accuracy of these statistics are only *polynomial*. For example, the channel might as well be the channel from the example above, where μ is only polynomially small. Therefore, we cannot conclude that the channel is really a implementation of a fair noisy channel, and neither can the channel be modeled by an unfair noisy channel. To be able to implement OT in this situation, we need to have a model that allows the implementation to behave *arbitrarily* with some probability.

1.3 Contribution

The goal of this work is to present new, more realistic models for noisy channels (called *weak noisy channels*) and to show that oblivious transfer can be implemented in an unconditionally secure way, assuming that such weak noisy channels exist. Opposed to the unfair noisy channel which is defined as an ideal functionality, our definitions are merely a list of conditions that a implementation of a weak noisy channels should satisfy. For a given implementation, one only needs to check these (quite simple) conditions, and does not need to show a cryptographically secure reduction of an ideal functionality to the implementation. This makes our model easier to apply.

We will introduce the following three models of weak noisy channels:

- *Weak erasure channels in the semi-honest model*[1] *(PassiveWEC).* These are weak variants of erasure channels[2] (channels that transmit a bit with some probability).
- *Weak binary symmetric channels in the semi-honest model (PassiveWBSC).* As the unfair noisy channel, these are weak variants of binary symmetric channels.
- *Weak erasure channels in the malicious model (ActiveWEC).*

We defined these channels such that they fitted well into the protocol proposed in [8], which is the only protocol we know of to implement OT from weak noisy channels.

To show the flexibility and generality of our models, we show that it is very easy to implement a PassiveWEC from *Gaussian channels*, and that the (passive) unfair noisy channel can be seen as an instance of a PassiveWBSC.

[1] See Section 2.2 for explanation of the semi-honest and the malicious model.

[2] Note that the original definition of OT by Rabin [19] is in fact also an erasure channel, so a WEC is also a weak form of Rabin OT.

In Sections 3 and 4, we show that PassiveWEC implies WOT, and PassiveWBSC implies PassiveWEC in the semi-honest model. Then, in Section 5, we show that ActiveWEC implies both bit commitment and a committed version of PassiveWEC in the malicious model. This implies that in a certain range of parameters, each of the three weak noisy channels allows for any secure two-party computation to be achieved. For each of the weak noisy channels, we also present a *simulation* of the channel using nothing else than noiseless communication, and in the case of ActiveWEC, shared randomness. Since it is impossible to implement bit commitment or oblivious transfer from noiseless communication and shared randomness, it is also impossible to implement them using the simulated weak noisy channels.

Full proofs are provided in the full version of this work.

2 Preliminaries

We start with some basic definitions and lemmas that we will need later.

We will use the following convention: Lower case letters will denote fixed values and upper case letters will denote random variables. Calligraphic letters will denote sets and domains of random variables. For a random variable X over $\mathcal{X}$, we denote its distribution by $P_X : \mathcal{X} \to [0,1]$ with $\sum_{x\in\mathcal{X}} P_X(x) = 1$. For a given distribution $P_{XY} : \mathcal{X} \times \mathcal{Y} \to [0,1]$, we write for the marginal distribution $P_X(x) := \sum_{y\in\mathcal{Y}} P_{XY}(x,y)$ and, if $P_Y(y) \neq 0$, $P_{X|Y}(x \mid y) := P_{XY}(x,y)/P_Y(y)$ for the conditional distribution. Let $h(x) := -x\log x - (1-x)\log(1-x)$ be the binary entropy function.

2.1 Statistical Distance and Maximal Bit-Prediction Advantage

The *statistical distance* of two distributions P_X and P_Y over the same domain $\mathcal{U}$ is defined as

$$\delta(P_X, P_Y) := \frac{1}{2}\sum_{u\in\mathcal{U}} |P_X(u) - P_Y(u)| \,.$$

For a distribution P_{XY} over $\{0,1\}\times\mathcal{Y}$, the *maximal bit-prediction advantage* of X from Y for a function f is defined as

$$\mathrm{PredAdv}(X \mid Y) := 2 \cdot \max_f \Pr[f(Y) = X] - 1 \,.$$

Lemmas 1, 2 and 3 give some intuition about these measures: The random variable B (or C) indicates that an error occurred: If $B = 0$, everything is fine. But if $B = 1$, the adversary may have complete knowledge. (See also [12] and [22].)

Lemma 1. *Let P_{BX} and P_{CY} be distributions over $\{0,1\}\times\mathcal{U}$ such that $\Pr[B = 1] = \Pr[C = 1] = \varepsilon$. Then $\delta(P_X, P_Y) \leq \varepsilon + (1-\varepsilon)\cdot\delta(P_{X|B=0}, P_{Y|C=0})$.*

Lemma 2. *Let P_{XY} be a distribution over $\{0,1\}\times\mathcal{Y}$. There exists a conditional distribution $P_{B|XY}$ over $\{0,1\}\times\{0,1\}\times\mathcal{Y}$ such that $\Pr[B = 1] \leq \mathrm{PredAdv}(X \mid Y)$ and such that for all functions $f : \mathcal{Y} \to \{0,1\}$, $\Pr[f(Y) = X \mid B = 0] = 1/2$.*

Lemma 3. *Let P_{XY} be a distribution over $\{0,1\}\times\mathcal{Y}$ with $\delta(P_{Y|X=0}, P_{Y|X=1}) \leq \varepsilon$. There exists a random variable B over $\{0,1\}$ such that $\Pr[B=1 \mid X=0] = \Pr[B=1 \mid X=1] = \varepsilon$, and $P_{Y|X=0,B=0} = P_{Y|X=1,B=0}$.*

Lemma 4. *Let P_{XY} be a distribution over $\{0,1\}\times\mathcal{Y}$ with $\delta(P_{Y|X=0}, P_{Y|X=1}) \leq b$ and $\mathrm{PredAdv}(X) \leq a$. Then $\mathrm{PredAdv}(X \mid Y) \leq 1-(1-a)(1-b)$.*

We say that X is ε-close to uniform with respect to Y, if $\delta(P_{XY}, P_U P_Y) \leq \varepsilon$, where P_U is the uniform distribution.

2.2 Adversaries

We distinguish between two different models, the *semi-honest model* and the *malicious model*. In the *semi-honest model*, the adversary is *passive*, which means that he follows the protocol, but may try to get additional knowledge from the messages received. In the malicious model, the adversary is *active*, which means that he may change his behavior in an arbitrary way.

2.3 Randomized Functionalities

All our channels are randomized, because we think the security conditions tend to be more intuitive this way. But randomized channels are usually also easier to implement (See for example Protocol ActiveToPassiveWEC in Section 5.3). The results for randomized channels immediately imply similar conditions for non-randomized channel, as they can be converted into a randomized channel simply by requiring the players to choose their inputs at random. Our definitions are weak enough that this works even in the malicious case.

2.4 Oblivious Transfer Amplification

Our work is based on *oblivious transfer amplification* from [8,22], which gives a way to implement oblivious transfer (OT) from weak oblivious transfer (WOT). We will take the definition of WOT from the full version of [22], however we use the weaker requirement of $\mathrm{PredAdv}(C \mid U) \leq p$ instead of $\mathrm{PredAdv}(C \mid U, E) \leq p$. As explained there, the reduction of OT to WOT still works for this weaker definition, as long as the error correction is always done from the sender to the receiver, which is normally the case.

Definition 1 (WOT, semi-honest model). *A* weak (randomized) oblivious transfer*, denoted by (p,q,ε)-WOT, is a primitive between a sender and a receiver, that outputs (X_0, X_1) to the honest sender and (C, Y) to the honest receiver. Let U be the additional auxiliary output*[3] *to a dishonest sender and let V be the auxiliary output to a dishonest receiver. Let $E := X_C \oplus Y$. The following conditions must be satisfied:*

[3] Or the *view* of the adversary, i.e., everything he knows at the end of the protocol.

– *Correctness:* $\Pr[E = 1] \leq \varepsilon$.
– *Receiver Security:* $\mathrm{PredAdv}(C \mid U) \leq p$.
– *Sender Security:* $\mathrm{PredAdv}(X_{1-C} \mid V, E) \leq q$.

Theorem 1 ([22]). *Let p, q and ε be constants such at least one of the following conditions holds:*

$$p{+}q{+}2\varepsilon \leq 0.24\,, \quad 22q{+}44\varepsilon < 1{-}p\,, \quad 22p{+}44\varepsilon < 1{-}q\,, \quad 49p{+}49q < (1{-}2\varepsilon)^2\,,$$

$$q = 0 \ \wedge\ p < (1-2\varepsilon)^2\,, \qquad p = 0 \ \wedge\ q < (1-2\varepsilon)^2\,, \qquad \varepsilon = 0 \ \wedge\ p + q < 1\,.$$

Then there exists a protocol that efficiently implements OT from (p, q, ε)-WOT secure in the semi-honest model.

We will only use the first four bounds, because we assume that $p, q, \varepsilon > 0$.

2.5 Bit Commitment

To achieve oblivious transfer in the malicious model, we will need *bit commitments*. A bit commitment scheme is a pair of protocols, a **Commit** protocol and a **Open** protocol, executed between a committer and a receiver. The players first execute the Commit protocol, where the committer has an input b. Then, they may also execute the Open protocol. After the Open protocol, the receiver either accepts or rejects. If he accepts, he gets a value b'. The protocols are ε-secure, if they satisfy the following properties:

– *Correctness*: If both players follow the protocols, then the receiver rejects with a probability smaller than ε, and if he accepts, he outputs $b' = b$ with probability at least $1 - \varepsilon$.
– *Binding*: If the receiver is honest, then for any malicious sender, with probability $1 - \varepsilon$, there exists at most one value after the commit protocol that the receiver will accept with a probability bigger than ε in the open phase.
– *Hiding*: If the committer is honest, then no malicious receiver gets to know b with a probability bigger than ε.[4]

3 Weak Erasure Channel in the Semi-honest Model

In this section, we present a reduction of WOT to *weak erasure channels (WEC)* in the semi-honest model. A weak erasure channel lets a honest sender send a bit, which is then received by the honest receiver with a certain probability, and gets lost otherwise. Dishonest players are allowed to receive some additional information, so a dishonest receiver may get to know some information about the input even in the case where the channel lost the bit, and a dishonest sender may get information about whether the bit has been lost or not.

[4] This means that if $b \in \{0, 1\}$ and V is the receiver's view, then we require that $\delta(P_{V|B=0}, P_{V|B=0}) \leq \varepsilon$.

Definition 2 (WEC, semi-honest model). *$(d_0, d_1, p, q, \varepsilon)$-PassiveWEC is a primitive where the honest sender has output $X \in \{0,1\}$ and the honest receiver has output $Y \in \{0,1,\Delta\}$. Furthermore, the dishonest sender may receive an additional value U, and the dishonest receiver may receive an additional value V. These values must satisfy the following conditions:*

- Correctness: $\Pr[Y = \Delta] \in [d_0, d_1]$, $\Pr[Y \neq X \mid Y \neq \Delta] \leq \varepsilon$.
- Receiver Security: $\delta(P_{XU|Y\neq\Delta}, P_{XU|Y=\Delta}) \leq p$.
- Sender Security: $\mathrm{PredAdv}(X \mid V, Y = \Delta) \leq q$.

The parameters can be interpreted as follows: d_0,d_1 and ε are parameters of the honest players. The probability that the output of the channel is Δ is in the interval $[d_0, d_1]$. (Defining this as an interval gives some freedom to the implementation, which may be important, as parameters often cannot be known precisely.) ε is the probability that the output of the honest receiver is wrong, if the output is not Δ. According to Lemma 2, q is the probability that the dishonest receiver gets to know the input of the channel, given that the output of the channel is Δ, and according to Lemma 3, p is the probability that a dishonest sender gets to know whether $Y = \Delta$ or $Y \neq \Delta$.

3.1 Simulation of PassiveWEC

We start by showing for which values a PassiveWEC can be simulated by only using noiseless communication. Since OT cannot be implemented from noiseless communication, such PassiveWEC therefore cannot be used to implement OT. Note that in any simulation that only uses noiseless communication, we always have $d_0 = d_1$, as both players know all the probabilities. In the following simulation, we require that $\varepsilon \in [0, \frac{1}{2}]$, $d, g \in [0,1]$, and $g \geq (1-2\varepsilon)(1-d)$.

Protocol $\mathsf{SimWEC}(d, \varepsilon, g)$

1. The sender chooses x uniformly at random and sends the receiver $m := x$ with probability g, and $m := \Delta$ otherwise. The sender outputs x.
2. If the receiver gets $m \in \{0,1\}$, he outputs $y := m$ with probability $\frac{(1-2\varepsilon)(1-d)}{g}$, and $y := \Delta$ otherwise.
3. If the receiver gets $m = \Delta$, he outputs y chosen at random with probability $\frac{2\varepsilon(1-d)}{1-g}$, and $y := \Delta$ otherwise.

Theorem 2. *For any d, ε, p and q, where $p+q+2\varepsilon \geq 1$, $(d, d, p, q, \varepsilon)$-PassiveWEC is simulatable in the semi-honest model.*

3.2 WOT from PassiveWEC

Protocol PassiveWECtoWOT

1. The sender and the receiver execute PassiveWEC twice. The sender receives (x_0, x_1), the receiver (y_0, y_1).

2. If there exists a c, such that $y_c \neq \Delta$ and $y_{1-c} = \Delta$, then the receiver sets $y := y_c$, outputs (c, y), tells the sender to terminate the protocol and terminates.
3. If the sender receives the message to terminate the protocol, he outputs (x_0, x_1) and terminates. Otherwise, they restart the protocol.

Theorem 3. *Protocol PassiveWECtoWOT securely implements a*

$$\left(1 - \frac{2d_0(1-d_1)}{d_1(1-d_0) + d_0(1-d_1)}(1-p)^2, q, \varepsilon\right)\text{-WOT}$$

secure against passive adversaries out of $(d_0, d_1, p, q, \varepsilon)$*-PassiveWEC. The expected number of instances used is at most* $1/\min(2d_0(1-d_0), 2d_1(1-d_1))$.

Theorem 3 is not difficult to show using Lemma 3 and Lemma 4. Corollary 1 follows now from Theorem 1, Theorem 3 and

$$1 - 2d_0(1-d_1)w(1-p)^2 \leq 2p + (d_1 - d_0)w \ .$$

Corollary 1. *Let* $d_0 \leq d_1$, p, q *and* ε *be constants, and let* $w = 1/(d_1(1-d_0) + d_0(1-d_1))$. *If at least one of the conditions*

$$2p + q + (d_1 - d_0)w + 2\varepsilon \leq 0.24 \ , \quad 11q + 22\varepsilon < d_0(1-d_1)w(1-p)^2 \ ,$$

$$44p + 22(d_1 - d_0)w + 44\varepsilon < 1 - q \ , \quad 98p + 49q + 49(d_1 - d_0)w < (1-2\varepsilon)^2$$

holds, then there exists a protocol that uses $(d_0, d_1, p, q, \varepsilon)$*-PassiveWEC and efficiently implements OT secure in the semi-honest model.*

3.3 An Example: The Gaussian Channel

The *Gaussian channel* is often used in information theory as a model of a noisy channel, because it models real physical channels quite well. It has been shown that a perfect and fair Gaussian channel implies bit commitment, see [17,18]. A Gaussian channel is a channel where the sender has input $x_g \in \mathbb{R}$ and the receiver has output $Y_g = x_g + E_g$, where $E_g \sim \mathcal{N}(0,1)$, i.e., the channel has an additive error that is normal distributed.

We can easily implement a PassiveWEC from this channel in the following way: Let $a, b \in \mathbb{R}^+$. The sender chooses $x \in \{0,1\}$ uniformly at random, sends $x_g := (2x-1)a$ and outputs x. The receiver gets y_g, and outputs $y = \Delta$ if $|y_g| \leq b$, $y = 1$ if $y_g > b$ and $y = 0$ otherwise. With an arbitrary small error, we can make the Gaussian channel discrete. In the limit, we get a $(d, d, p, q, \varepsilon)$-PassiveWEC, where $d = \Phi(b-a) - \Phi(-a-b)$, $\varepsilon = \frac{\Phi(-a-b)}{1-d}$, $p = 0$ and $q = 2\frac{\Phi(b-a)-\Phi(-a)}{d} - 1$. Choosing for example $a = 1$ and $b = 2.5$, we get $d \approx 0.93296$, $\varepsilon \leq 0.0035$, and $q \leq 0.6604$. Since $44 \cdot \varepsilon < 1 - q$, it follows from Corollary 1 that oblivious transfer can be implemented. Together with the bit commitment protocols from [17,18], this implies (using a protocol similar to ActiveToPassiveWEC) that OT can be implemented from (perfect and fair) Gaussian channels in the malicious model.

To the best of our knowledge, this has not been known before, as previous results in [6,16] rely on the fact that the channel is discrete and cannot be applied

to the Gaussian channel. Note that in contrast to the reductions from [17,18], our reduction even works for Gaussian channels that are neither perfect nor fair.

4 Weak Binary Symmetric Channel in the Semi-honest Model

Weak Binary Symmetric Channel is a weak form of a binary symmetric channel. The channel transmits the input bit of the sender to the receiver, but flips the bit with some probability. Again, the definition is randomized.

Definition 3 (WBSC, semi-honest model). $(\varepsilon, \varepsilon_0, \varepsilon_1, p, q)$-*PassiveWBSC is defined as follows: The honest sender has output* $X \in \{0,1\}$ *and the honest receiver has output* $Y \in \{0,1\}$*. Furthermore, the dishonest sender may receive an additional value* $U \in \mathcal{U}$*, and the dishonest receiver may receive an additional value* $V \in \mathcal{V}$*. These values must satisfy the following conditions:*

- Correctness: $\Pr[X = 0] \in [\frac{1-\varepsilon}{2}, \frac{1+\varepsilon}{2}]$*, and for* $x \in \{0,1\}$, $\Pr[Y \neq x] \in [\varepsilon_0, \varepsilon_1]$.
- Receiver Security: $\delta(P_{UX|Y=X}, P_{UX|Y\neq X}) \leq p$.
- Sender Security: *For all* $y \in \{0,1\}$*:* $\delta(P_{V|X=0,Y=y}, P_{V|X=1,Y=y}) \leq q$.

The parameters can be interpreted as follows: ε is the bias of X, and ε_0 and ε_1 define the error interval of the honest players. From Lemma 3 it follows that p is the probability that the sender, and q is the probability that the receiver gets to know whether $X = Y$ or not. Note that in order to make our reduction work, the sender security has a slightly different form than the receiver security. If $\varepsilon = 0$ and $\varepsilon_0 = \varepsilon_1$, the sender security implies $\delta(P_{VY|Y=X}, P_{VY|Y\neq X}) \leq q$. So in this case, the sender security is strictly stronger than the receiver security.

4.1 Simulation of PassiveWBSC

The following simulation is basically the same as in [8] for the unfair noisy channel. Let $\varepsilon_A, \varepsilon_B \in [0, \frac{1}{2}]$.

Protocol $\mathsf{SimWBSC}(\varepsilon_A, \varepsilon_B)$

1. The players toss a uniform coin $M \in \{0,1\}$.
2. The sender calculates $X := 1 - M$ with probability ε_A and $X := M$ otherwise, and outputs X.
3. The receiver calculates $Y := 1 - M$ with probability ε_B and $Y := M$ otherwise, and outputs Y.

Theorem 4. *Let* $\varepsilon := \varepsilon_A(1-\varepsilon_B) + \varepsilon_B(1-\varepsilon_A)$*,* $p := \frac{(1-\varepsilon_A)(1-\varepsilon_B)}{1-\varepsilon} - \frac{\varepsilon_A(1-\varepsilon_B)}{\varepsilon}$*, and* $q := \frac{(1-\varepsilon_A)(1-\varepsilon_B)}{1-\varepsilon} - \frac{(1-\varepsilon_A)\varepsilon_B}{\varepsilon}$*. The Protocol* $\mathsf{SimWBSC}(\varepsilon_A, \varepsilon_B)$ *securely implements a* $(0, \varepsilon, \varepsilon, p, q)$-*PassiveWBSC in the semi-honest model.*

Theorem 4 implies that $(0, \varepsilon, \varepsilon, p, q)$-PassiveWBSC is simulatable if $p + q > 1$.

4.2 PassiveWEC from PassiveWBSC

We will now give a reduction of PassiveWEC to PassiveWBSC. The protocol itself has already been used in [5] and [4]. The intuition behind the following protocol is simple: The sender sends a bit twice over a binary noisy channel. If the receiver gets twice the same message, he knows (with a small error) what the sender has sent and outputs that. If he receives two different messages, he does not know the input and outputs Δ. Note that since two channels are randomized, the sender cannot choose his input, and therefore has to additionally send $x_0 \oplus x_1$.

Protocol PassiveWBSCtoWEC

1. The players execute PassiveWBSC twice. The sender gets (x_0, x_1), the receiver (y_0, y_1).
2. The sender sends $k := x_0 \oplus x_1$ to the receiver and outputs $x := x_0$.
3. If $y_0 \oplus y_1 = k$, the receiver outputs $y := y_0$. Otherwise, he outputs $y := \Delta$.

Theorem 5. *Let*

$$\begin{aligned} d_0 &:= \min(2\varepsilon_0(1-\varepsilon_0), 2\varepsilon_1(1-\varepsilon_1)) \,, \\ d_1 &:= \max(2\varepsilon_0(1-\varepsilon_0), 2\varepsilon_1(1-\varepsilon_1), \varepsilon_0(1-\varepsilon_1) + \varepsilon_1(1-\varepsilon_0)) \,. \\ \varepsilon' &:= \frac{\varepsilon_1 - \varepsilon_0}{\varepsilon_1 + \varepsilon_0 - 2\varepsilon_0\varepsilon_1} - \frac{2\varepsilon}{1+\varepsilon^2} \,. \end{aligned}$$

Protocol PassiveWBSCtoWEC securely implements a

$$\left(d_0, d_1, 1-(1-p)^2, 1-(1-\varepsilon')(1-q)^2, \frac{\varepsilon_1^2}{\varepsilon_1^2 + (1-\varepsilon_1)^2}\right)\text{-PassiveWEC}$$

in the semi-honest model out of two instances of $(\varepsilon, \varepsilon_0, \varepsilon_1, p, q)$*-PassiveWBSC.*

Proof (Sketch). It is easy to verify that

$$\Pr[Y \neq X \mid Y \neq \Delta] \leq \frac{\varepsilon_1^2}{\varepsilon_1^2 + (1-\varepsilon_1)^2}$$

and $\Pr[Y = \Delta] \in [d_0, d_1]$, and the security against a dishonest sender can be shown using Lemma 3 and Lemma 1.

Let V_0 and V_1 be the additional information a dishonest receiver gets in the two executions of the PassiveWBSC. We have $V := (K, V_0, V_1, Y_0, Y_1)$. Using Lemma 3 and Lemma 1 it can be shown that

$$\delta(P_{V_0V_1|X=0,K=k,Y_0=y_0,Y_1=y_1}, P_{V_0V_1|X=1,K=k,Y_0=y_0,Y_1=y_1}) \leq 1-(1-q)^2 \,.$$

We can bound

$$\begin{aligned} &\Pr[X = x \mid Y_0 = y_0, Y_1 = y_1, K = k, Y = \Delta] \\ &\qquad \leq \frac{(1+\varepsilon)\varepsilon_1 \cdot (1+\varepsilon)(1-\varepsilon_0)}{(1+\varepsilon)\varepsilon_1 \cdot (1+\varepsilon)(1-\varepsilon_0) + (1-\varepsilon)\varepsilon_0 \cdot (1-\varepsilon)(1-\varepsilon_1)} \,, \end{aligned}$$

from which follows

$$\mathrm{PredAdv}(X \mid Y_0 = y_0, Y_1 = y_1, K = k, Y = \Delta) \leq \frac{\varepsilon_1 - \varepsilon_0}{\varepsilon_1 + \varepsilon_0 - 2\varepsilon_0\varepsilon_1} + \frac{2\varepsilon}{1+\varepsilon^2} .$$

The statement now follows from Lemma 4. □

4.3 An Example: The Unfair Noisy Channel

The *passive unfair noisy channel* (γ, δ)-PassiveUNC from [8,7] is a special case of a PassiveWBSC, namely a $(0, \delta, \delta, p, p)$-PassiveWBSC, where

$$p := \frac{(1-\delta)\delta - (1-\gamma)\gamma}{(1-2\gamma)\delta(1-\delta)} .$$

Note, however, that the bounds that we get using our results are not as good as the bounds from [8,7].

5 WEC in the Malicious Model

The assumption that the adversary is semi-honest and therefore follows the protocol is quite strong and often too strong. As shown in [10], there exist compilers that can convert protocols which are only secure in the semi-honest model into protocols that are also secure in the malicious model. The basic idea is that at the beginning, the players are committed to all the secret data they have, and after every computation step they do, they commit to the newly computed values and show with a zero-knowledge proof that the new committed value contains indeed the correct value, according to the protocol. To implement this in our setting, we need two things: A bit commitment protocol, and a protocol that implements a committed version of the passive weak noisy channel. Hence, for any weak noisy channel in the active model, we need to show that it implies bit commitment and a committed version of either PassiveWEC or PassiveWBSC for parameters that allow us to achieve OT in the semi-honest model. (See also [7] for a more detailed discussion.)

Defining a weak noisy channel in the malicious model turns out to be much more tricky than in the semi-honest model. It is possible to define them in the same way as in the semi-honest model, however we think that this would not give a very realistic model. For example, the dishonest player probably may choose an attack where he does not get the output of the honest player. Therefore, we think that it is preferable to state the security conditions such that the malicious player does not need to get the value of the honest player. In the following we will do this for the WEC. For the WBSC, we were not able to come up with a simple definition.

Definition 4 (WEC, malicious model). *$(d_0, d_1, p, g, \varepsilon)$-ActiveWEC is a primitive with the following properties.*

- Correctness: *If both players are honest, then the sender has output $X \in \{0,1\}$ and the receiver has output $Y \in \{0,1,\Delta\}$, where $\Pr[Y = \Delta] \in [d_0, d_1]$ and $\Pr[Y \neq X \mid Y \neq \Delta] \leq \varepsilon$.*

- Receiver Security: *If the receiver is honest, then for all dishonest sender with auxiliary input z and output U, the receiver has output $Y \in \{0, 1, \Delta\}$ where* $\Pr[Y = \Delta] \in [d_0, d_1]$ *and* $\delta(P_{U|Z=z,Y\neq\Delta}, P_{U|Z=z,Y=\Delta}) \leq p$.
- Sender Security: *If the sender is honest, then for all dishonest receiver with auxiliary input z and output V, the sender has output $X \in \{0, 1\}$ and* $\mathrm{PredAdv}(X \mid V, Z = z) \leq g$.

Note that the parameter g is different from the parameter q in the semi-honest case, because we do not condition on the event $Y = \Delta$. The honest receiver can guess X using $f(Y) := Y$ if $Y \neq \Delta$, and either 0 or 1 if $Y = \Delta$. We get $\mathrm{PredAdv}(X \mid Y) \geq (1 - 2\varepsilon)(1 - d_1)$. Therefore, an ActiveWEC can only be implemented if $g \geq (1 - 2\varepsilon)(1 - d_1)$.

5.1 Simulation

Using the same simulation as for the semi-honest case, we get

Theorem 6. *For any d, ε, p and g, where*

$$dp + g + 2\varepsilon \geq 1 \qquad \wedge \qquad g \geq (1 - 2\varepsilon)(1 - d) \ ,$$

$(d, d, p, g, \varepsilon)$-ActiveWEC is simulatable in the malicious model, given that the players have access to a source of trusted shared randomness.

5.2 Bit Commitment

Our commitment protocol takes parameters n, c, m, ℓ and κ, where n is the number of instances used, c the error-tolerance of the protocol, ℓ the number of bits committed to, and κ the error. Let $c := n^{-1/3}$, and

$$\kappa := \exp(-2(1 - d_1 - c)nc^2) \ .$$

Let a be the maximum value that satisfies

$$(1 - d) \cdot a - \sqrt{\frac{a}{2} \cdot \ln \frac{1}{\kappa}} \leq (\varepsilon + c)(1 - d)n$$

for all $d \in [d_0 - c, d_1 + c]$. Let

$$m := (d_1 p + c)n + 2a + 1$$

and let $\mathcal{C} \subset \{0, 1\}^n$ be a (n, k, m)-linear code[5], i.e., with 2^k elements and minimal distance m. Let

$$\ell := k - (g + c) \cdot n - 3 \log(1/\kappa)$$

and n be big enough such that $\ell > 0$. Let H be the parity-check matrix of $\mathcal{C}$ and $g : \mathcal{R} \times \{0, 1\}^n \to \{0, 1\}^\ell$ be a 2-universal hash function. In the following protocol, the sender is the committer.

Protocol ActiveWECtoBC

Commit(b).

- The parties execute ActiveWEC n times. The sender gets $x = (x_0, \ldots, x_{n-1})$, and the receiver gets $y = (y_0, \ldots, y_{n-1})$.

[5] Since we do not have to decode $\mathcal{C}$, this could be a random linear code.

– The committer chooses $r \in \mathcal{R}$ uniformly at random and sends it to the receiver.
– The committer sends $s := (H(x), b \oplus g(r, x))$ to the receiver.

Open.

– The committer sends (b, x) to the receiver.
– Let n_Δ be the number of y_i equal to Δ. The receiver checks that $n_\Delta/n \in [d_0 - c, d_1 + c]$ and that the number i where $y_i \neq x_i$ and $y_i \neq \Delta$ is smaller than $(n - n_\Delta)(\varepsilon + c)$. He also checks that $s = (H(x), b \oplus g(r, x))$. If this is the case, he accepts, and rejects otherwise.

In the protocol, the committer has to send the receiver the parity-check of a code, because then the committer cannot guess with probability more than ε more than one value x that passes the test of the receiver in the open phase. The committer extracts a string of size ℓ from x, where ℓ is chosen small enough such that the receiver has almost no information about it.

Theorem 7. *Protocol ActiveWECtoBC implements a commitment with an error of 4κ, out of n instances of $(d_0, d_1, p, g, \varepsilon)$-ActiveWEC.*

The correctness of the protocol follows from the Chernoff/Hoeffding bound. It remains to proof that the protocol is also binding and hiding.

Lemma 5. *Protocol ActiveWECtoBC is binding with probability $1 - 4\kappa$.*

Proof. Let $d := n_\Delta/n$. Let B_i be defined as in Lemma 3. If $Y_i \neq \Delta$, let $Y_i' = Y_i$, and let Y_i' be chosen randomly from $\{0, 1\}$ otherwise, such that $\Pr[Y_i' = 1 \mid Y_i = \Delta] = \Pr[Y_i = 1 \mid Y_i \neq \Delta]$. ($Y_i'$ is therefore independent of the event $Y_i = \Delta$.) Let us assume that the sender additionally receives the values B_i and Y_i'.

We divide the n instances into 3 sets. Let S_0 be the set of values where $B_i = 1 \wedge Y_i = \Delta$, S_1 the set of values where $B_i = 1 \wedge Y_i \neq \Delta$, and S_2 the set of values where $B_i = 0$. The sender may choose a subset of S_1 of size a' and a subset of S_2 of size a, where $x_i \neq y_i'$. It follows from the Chernoff/Hoeffding bound that with probability at least κ, the receiver will notice at least $a \cdot (1 - d) - \sqrt{\frac{a}{2} \cdot \ln \frac{1}{\kappa}}$ of these errors in S_2. Therefore, the receiver will only accept with probability at least κ, if

$$a' + a \cdot (1 - d) - \sqrt{\frac{a}{2} \cdot \ln \frac{1}{\kappa}} \leq (\varepsilon + c)(1 - d)n \; .$$

The sender would only be able to find two values with the same parity-check if

$$(dp + c)n + 2(a' + a) \geq m \; .$$

The best strategy for the sender is to choose $a' = 0$, and to make a maximal. It follows from the definition of m that the sender cannot find two such values. The statement follows. □

To proof that the protocol is hiding we need some additional lemmas. The *conditional smooth min-entropy of X given Y* [20] is defined as

$$H_{\min}^{\varepsilon}(X \mid Y) := \max_{\Omega:\Pr[\Omega]\geq 1-\varepsilon} \min_{xy}(-\log P_{X\Omega|Y=y}(x)) \ .$$

Lemma 6 ([2,15]). $H_{\min}^{\varepsilon+\varepsilon'}(X \mid YZ) \geq H_{\min}^{\varepsilon}(XY \mid Z) - \log|\mathcal{Y}| - \log(1/\varepsilon')$.

Lemma 7 (Leftover hash lemma [1,13]). *Let X be a random variable over $\mathcal{X}$ and let $m > 0$. Let $h : \mathcal{R} \times \mathcal{X} \to \{0,1\}^m$ be a 2-universal hash function. If $m \leq H_{\min}^{\varepsilon}(X \mid Y) - 2\log(1/\varepsilon')$, then for R uniform over $\mathcal{R}$, $h(R,X)$ is $(\varepsilon+\varepsilon')$-close to uniform with respect to (R,Y).*

Lemma 8. *Protocol ActiveWECtoBC is hiding with probability $1-3\kappa$.*

Proof. The sender holds $X = (X_1, \ldots, X_n)$, and the receiver $V = (V_1, \ldots V_n)$, S and the auxiliary input z. Using Lemma 2, for every pair (X_i, V_i), there exists a random variable B_i, such that $\Pr[B_i = 1] = g$ and X_i is uniform, given $(V_i, B_i = 0, Z = z)$. From the Chernoff/Hoeffding bound follows that with probability $1-\kappa$, the number of $B_i = 0$ is at least $n(1-g-c)$ and therefore $H_{\infty}^{\kappa}(X \mid V, Z = z) \geq n(1-g-c)$. Using Lemma 6, we get $H_{\infty}^{2\kappa}(X \mid V, S, Z = z) \geq n(1-g-c) - (n-k) - \log(1/\kappa)$. Finally, we can apply Lemma 7, and get that $g(X,R)$ is 3κ-close to uniform, since $\ell \leq H_{\infty}^{2\kappa}(X \mid V, S, Z = z) - 2\log(1/\kappa)$. This implies that the protocol is hiding with probability $1-3\kappa$. □

Note that for any $e > 0$, and $k \leq (1 - h(m/n))\, n - e$, a random linear (n,k)-code has a minimal distance of at least m with probability at least $1 - 2^{-e}$. If we choose a random linear code and let $n \to \infty$, then $b/n \to \varepsilon$, and hence $m/n \to d_1 p + 2\varepsilon$. From the property of the random linear code, we get $k/n \to 1 - h(d_1 p + 2\varepsilon)$. We need $\ell > 0$, which is equivalent to $g < k/n$. We get the following corollary.

Corollary 2. *For any d_1, d_1, ε, p and q where*

$$d_1 p + 2\varepsilon < \frac{1}{2}, \qquad \text{and} \qquad g + h(d_1 p + 2\varepsilon) < 1 \ ,$$

$(d_0, d_1, p, g, \varepsilon)$-ActiveWEC implies bit commitment.

Our bound is optimal for $p = 0 \wedge \varepsilon = 0$. Otherwise, it does not reach the simulation bound, since $h(x) > x$ for all $0 < x < \frac{1}{2}$. It would be interesting to know whether this bound can be improved. Note that it is also possible to implement bit commitment in the other direction. We will leave this to the full version of this work.

5.3 Committed PassiveWEC from ActiveWEC

In the following, we present the protocol to implement a committed version of PassiveWEC in the malicious model, using ActiveWEC. It uses a similar idea already used in [7]: The players execute ActiveWEC n times and commit to their

output values. Then, they open all except one that is chosen at random, and check if the statistics are fine. If they are, then with high probability, also the statistics of the remaining instance is fine.

The following lemma is essential to the proof, because it can be used to bound the parameter p for any committed value Y produced by the dishonest receiver, if he passes the test by the honest sender. It is easy to verify that the lemma is tight if V is equal to X with probability p and Δ otherwise.

Lemma 9. *Let P_{XV} be a distribution over $\{0,1\} \times \mathcal{V}$. If $\mathrm{PredAdv}(X \mid V) \leq g$, then for any function $Y = f(V) \in \{0,1,\Delta\}$ where $\Pr[Y = \Delta] \in [d_0, d_1]$ and $\Pr[Y \neq X \mid Y \neq \Delta] \leq \varepsilon$, we have*

$$\delta(P_{V|X=0,Y=\Delta}, P_{V|X=1,Y=\Delta}) \leq \frac{g - (1-2\varepsilon)(1-d_1)}{d_1} .$$

Proof. Let B be the random variable defined by Lemma 2. We have $\Pr[B = 1] = g$ and $P_{V|X=0,B=0} = P_{V|X=1,B=0}$. Given $B = 0$, V does not have any information about X. Hence, for any $Y = f(V)$, we have

$$\Pr[Y \neq X \mid Y \neq \Delta] \geq \frac{1}{2} \cdot \frac{\Pr[Y \neq \Delta \wedge B = 0]}{\Pr[Y \neq \Delta]} .$$

Therefore, it must hold that $2\varepsilon \Pr[Y \neq \Delta] \geq \Pr[Y \neq \Delta \wedge B = 0]$. We get

$$\begin{aligned} \Pr[B = 1 \mid Y = \Delta] &= \frac{g - \Pr[Y \neq \Delta] + \Pr[Y \neq \Delta \wedge B = 0]}{\Pr[Y = \Delta]} \\ &\leq \frac{g - (1-2\varepsilon)(1 - \Pr[Y = \Delta])}{\Pr[Y = \Delta]} \\ &\leq \frac{g - (1-2\varepsilon)(1-d_1)}{d_1} . \end{aligned}$$

The statement follows now by applying Lemma 1. □

In addition to ActiveWEC, our protocol needs bit commitments and coin-tosses. Coin-toss can easily be implemented using bit commitments.

Again, c is the error-tolerance, and κ is the error in the protocol. We choose $c := n^{-1/3}$ and $\kappa := \exp(-2(1-d_1-c)nc^2)$. Furthermore, let n be big enough such that $c \geq 1/((1-d_1-c)n)$.

Protocol ActiveToPassiveWEC

1. The sender and the receiver execute ActiveWEC n times. The sender gets $(x_0, \ldots, x_{n-1})$, and the receiver $(y_0, \ldots, y_{n-1})$.
2. Both players commit to their values.
3. Using coin-toss, they randomly select one instance s of the n instances.
4. They open all commitments, except for instance s. If any of the players does not accept one opening of a commitment, they abort.

5. Let n_Δ be the number of y_i that is equal to Δ. They check if n_Δ is in the interval $[(d_0 - c) \cdot n - 1, (d_1 + c) \cdot n]$, and the number of y_i that is not equal to Δ nor x_i is smaller than $(\varepsilon + c) \cdot (n - n_\Delta)$. If not, they abort.
6. The sender outputs $x := x_s$, the receiver $y := y_s$.

Theorem 8. *Protocol ActiveToPassiveWEC implements a committed version of*

$$\left(d_0 - 2c, d_1 + 2c, p, \frac{g - (1-2\varepsilon)(1-d_1)}{d_1} + \frac{6}{d_1^2}c, \varepsilon + 2c\right)\text{-PassiveWEC}$$

with an error of at most 3κ in the malicious model. It uses coin-toss, bit commitment and n independent instances of $(d_0, d_1, p, g, \varepsilon)$-ActiveWEC.

Theorem 8 can be shown using the Chernoff/Hoeffding bound and Lemma 9. Note that c is only polynomially small and cannot be made negligible. Here we see an advantage of our definition compared to the PassiveUNC in [8,7]: We do not have to introduce the additional error parameter $p(k)$ as it has to be done for the committed version of the PassiveUNC, nor do we have to add an additional amplification step to the reduction to make this additional error negligible. The following corollary follows from Corollary 1 and Theorem 8.

Corollary 3. *Let $d_0 \leq d_1$, p, g and ε be constants, and let $w := 1/(d_1(1-d_0) + d_0(1-d_1))$ and $q := \frac{g-(1-2\varepsilon)(1-d_1)}{d_1}$. If at least one of the conditions*

$$p + q + w(d_1 - d_0) + 2\varepsilon < 0.24\,, \quad 11q + 22\varepsilon < d_0(1-d_1)w(1-p)^2\,,$$

$$44p + 22w(d_1 - d_0) + 44\varepsilon < 1 - q\,, \quad 98p + 49q + 49w(d_1 - d_0) < (1-2\varepsilon)^2$$

holds, then there exists a protocol that uses $(d_0, d_1, p, g, \varepsilon)$-ActiveWEC and bit commitments and efficiently implements OT secure in the malicious model.

To achieve any two party computation from a $(d_0, d_1, p, g, \varepsilon)$-ActiveWEC, the conditions of Corollary 3 and Corollary 2 must be satisfied simultaneously.

6 Conclusions and Open Problems

We gave new, weaker security definitions for the erasure channel and the binary symmetric channel, and showed that they imply oblivious transfer. The advantage of our new definitions is that they allow the use of channels from which the statistics are not known with arbitrary precision, which make it possible to use channels where the parameters come from experimental evidence. Note that together with the computational WOT amplification from [22], our results can also be used in a computational setting.

It seems to be difficult to close the gap between the possibility and the impossibility bounds for OT. But maybe it is possible to get a tight bound for bit commitment. Still missing is a definition of the weak binary symmetric channel in the malicious model. Furthermore, it would be nice to have a bit commitment protocol that works for a weak form of the Gaussian channels.

Acknowledgment

I thank the anonymous referees for many helpful comments. I was supported by the U.K. EPSRC, grant EP/E04297X/1.

References

1. Bennett, C.H., Brassard, G., Robert, J.-M.: Privacy amplification by public discussion. SIAM Journal on Computing 17(2), 210–229 (1988)
2. Cachin, C.: Smooth entropy and rényi entropy. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 193–208. Springer, Heidelberg (1997)
3. Crépeau, C.: Equivalence between two flavours of oblivious transfers. In: Price, W.L., Chaum, D. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 350–354. Springer, Heidelberg (1988)
4. Crépeau, C.: Efficient cryptographic protocols based on noisy channels. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 306–317. Springer, Heidelberg (1997)
5. Crépeau, C., Kilian, J.: Achieving oblivious transfer using weakened security assumptions (extended abstract). In: Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1988), pp. 42–52 (1988)
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. Damgård, I.B., Fehr, S., Morozov, K., Salvail, L.: Unfair noisy channels and oblivious transfer. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 355–373. Springer, Heidelberg (2004)
8. 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)
9. Even, S., Goldreich, O., Lempel, A.: A randomized protocol for signing contracts. Commun. ACM 28(6), 637–647 (1985)
10. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC 1987), pp. 218–229. ACM Press, New York (1987)
11. Goldreich, O., Vainish, R.: How to solve any protocol probleman efficiency improvement. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 73–86. Springer, Heidelberg (1988)
12. Holenstein, T.: Strengthening key agreement using hard-core sets. PhD thesis, ETH Zurich, Switzerland, Reprint as vol. 7 of ETH Series in Information Security and Cryptography, Hartung-Gorre Verlag (2006)
13. Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC 1989), pp. 12–24. ACM Press, New York (1989)
14. Kilian, J.: Founding cryptography on oblivious transfer. In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC 1988), pp. 20–31. ACM Press, New York (1988)
15. Maurer, U., Wolf, S.: Privacy amplification secure against active adversaries. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 307–321. Springer, Heidelberg (1997)

16. Nascimento, A., Winter, A.: On the oblivious transfer capacity of noisy correlations. IEEE Trans. on Information Theory 54(6) (2008)
17. Nascimento, A.C.A., Skludarek, S., Barros, J., Imai, H.: The commitment capacity of the gaussian channel is infinite. IEEE Trans. on Information Theory, Special Issue on Information Security (2007)
18. Oggier, F., Morozov, K.: A practical scheme for string commitment based on the gaussian channel. In: Proceedings of 2006 IEEE Information Theory Workshop (ITW 2008) (2008)
19. Rabin, M.O.: How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory (1981)
20. Renner, R., Wolf, S.: Simple and tight bounds for information reconciliation and privacy amplification. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 199–216. Springer, Heidelberg (2005)
21. Wiesner, S.: Conjugate coding. SIGACT News 15(1), 78–88 (1983)
22. Wullschleger, J.: Oblivious-transfer amplification. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 555–572. Springer, Heidelberg (2007); Full version (PhD Thesis, ETH Zurich), http://arxiv.org/abs/cs.CR/0608076
23. Yao, A.C.: Protocols for secure computations. In: Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 1982), pp. 160–164 (1982)