Achieving Oblivious Transfer Using Weakened Security Assumptions

(Extended Abstract)

Claude Crépeau*
Department of Computer Science
MIT

Joe Kilian†
Mathematics Department
MIT

Abstract

A useful paradigm in studying cryptographic scenarios is that of *protocol minimalism*. That is, given a cryptographic model, one wishes to determine the simplest protocols one needs in order to be able to implement secure protocols in general. In the standard cryptographic model, this approach allows one to encapsulate ones cryptographic assumptions. In other, nonstandard scenarios, the approach can greatly simplifying the task of developing protocols without cryptographic assumptions.

Oblivious transfer protocols, first introduced by Rabin[R], are conceptually very simple, yet can be used to implement a wide variety of protocols([EGL],[BCR1],[K]). The versatility of these games amply motivates a wider study of the power of simple two-party games.

In this paper, we present some general techniques for establishing the cryptographic strength of a wide variety of games. As case studies, we analyze some "weakened" versions of the standard forms of oblivious transfer. We also consider variants of oblivious transfer which are motivated by coding theory and physics. Among our results, we show that a noisy telephone line is in fact a very sophisticated cryptographic device. We also present an application to quantum cryptography.

*Supported in part by an NSERC Postgraduate Scholarship. Some of this research was performed while visiting CWI and Bell Communication Research.

†Research supported in part by a Fannie and John Hertz foundation fellowship, and NSF grant 865727-DCR. Some of this research was performed while visiting Bell Communication Research.

1 Introduction

Our work is motivated by a recent trend in cryptographic research. Protocol problems that have previously been solved subject to intractability assumptions are now being solved without these assumptions. Examples of this trend include a new completeness theorem for multiparty protocols[BGW,CCD], and a protocol for byzantine agreement using private channels[FM]. These breakthroughs illustrate both the strengths and the weaknesses of using the cryptographic model. Devising first a protocol that uses cryptographic assumptions can give powerful intuition that later allows one to create a protocol that works without assumptions. However, there is a danger that the cryptographic assumptions one uses can become inextricably bound up in the protocol. It may take years before these assumptions can be ironed out of the final protocol.

One way to keep a firm grasp on ones cryptographic assumptions is to compartmentalize them into a small set of relatively simple primitives. One then attempts to build protocols on top of these primitives, without using any cryptographic assumptions in the high level design. The problem of eliminating cryptographic assumptions from the protocol is then reduced to that of implementing the primitives without cryptography.

In this paper, we explore a particularly useful set of primitives, known as *oblivious transfers*. First introduced by Rabin, oblivious transfer protocols are games in which one player, Sam(the sender), can impart some information to another player, Rachel(the receiver), without knowing precisely what information he has imparted. Oblivious transfers come in a wide variety of flavors, and are not obviously reducible to each other. Following the work of Brassard, Crépeau, Robert[BCR2], and Crépeau[C], we develop techniques for establishing equivalences between a wide variety of oblivious transfers.

Why study oblivious transfer?

A compelling reason to study oblivious transfer protocols is their connection to the problem of *oblivious circuit evaluation*. Oblivious circuit evaluation can be described as follows: Two players, Sam and Rachel, possess secrets i and j respectively. Sam would like to help Rachel compute some function $f(i,j)$, where f can be computed by a polynomial sized circuit. However, Sam does not want to give Rachel any more information about i than is conveyed by knowledge of $f(i,j)$, and Rachel does not want Sam to gain any information at all. The value of oblivious circuit evaluation as a primitive in the design of cryptographic protocols is clear.

Yao [Y] developed the first protocol to perform oblivious circuit evaluation. This protocol used a powerful form of oblivious transfer called *1-2 oblivious transfer*(defined in Section 2), which was implemented using cryptographic assumptions. Unfortunately, the use of cryptographic assumptions occurred elsewhere in the protocol as well. A more recent protocol, due to Kilian[K], reduces oblivious circuit evaluation to 1-2 oblivious transfer without any cryptographic assumptions.[1] Crépeau[C] has reduced 1-2 oblivious transfer to a weaker form of oblivious transfer, which we refer to as standard or ordinary oblivious transfer(defined in Section 2). Thus, one only has to implement oblivious transfer in order to have the full power of oblivious circuit evaluation at ones disposal.

A motivation from the study of ordinary circuits.

An analogy can be made between the study of oblivious circuit evaluation and ordinary circuit evaluation. One of the very trivial but extremely important properties of circuits is that they can be broken up into extremely simple computational entities, such as NOR gates. A nontrivial property of secure two-party protocols is that they can be broken up into extremely simple secure protocols.

In studying the finer points of the reduction from circuits to gates, many natural questions arise, two of which have direct bearing to this paper. These are:

- What types of gates are sufficient for circuit computation?
- Can unreliable gates still be used in circuit computation?

[1]Goldreich-Vainish ([GV]) independently discovered a reduction for the case where both parties are honest.

These two questions have been largely answered. Concerning the first question, we know that a wide assortment of gates can also be used in place of NOR gates. Concerning the second, we know that gates which give the wrong answer almost half the time may still be used to perform reliable computations, given a suitable interpretation of what constitutes the output of the circuit.

Given that we know how to reduce secure two-party protocols to a simple protocol, we can ask the following analogous questions.

- What primitive protocols are sufficient to perform oblivious circuit computation in general?
- Can protocols whose security guarentees may be unreliable still be used to achieve oblivious circuit computation?

In this paper, we explore both these questions. Considering the first question, for instance, we investigate the properties of an ordinary noisy channel. By a noisy channel, we mean a communication line in which a transmitted bit is flipped with a certain fixed probability. This model has been extensively studied in coding theory, but relatively little was previously known about its cryptographic capabilities. We show that a noisy channel can be used to implement two-party cryptographic protocol without any intractability assumptions. We also study a transfer mechanism we refer to as quantum transfer. This mechanism abstractly models a transfer mechanism based on quantum mechanics.

Considering the second question, we study weaker variants of two of the more standard forms of oblivious transfer. We investigate scenarios in which the security properties guarenteed by these mechanisms may be almost completely violated. We show that in many of these scenarios, it is still possible to achieve the full power of ordinary oblivious transfer.

Statement of our Main Results.

Our results may be summarized as follows. Before reading these theorems, we refer the reader to Section 2 of the paper, which provides the necessary terminology.

Theorem 1: α-1-2 slightly oblivious transfer is as powerful as 1-2 oblivious transfer.

Theorem 2: Noisy transfer is as powerful as 1-2 oblivious transfer.

Theorem 3: α-slightly oblivious transfer is as powerful as 1-2 oblivious transfer.

Theorem 4: Quantum transfer is as powerful as 1-2 oblivious transfer.

Outline of the Paper

In Section 2, we define the two most common forms of oblivious transfer. We then define two less standard forms of oblivious transfer, and two weakened forms of the standard oblivious transfers. Finally, we conclude with a discussion of what we mean by reducing one protocol to another. In Section 3, we show how to achieve oblivious transfer using α-1-2 oblivious transfer. In Section 4, we present a fairly general technique for strengthening our reductions. In Section 5, we apply this technique, showing how to achieve oblivious transfer using α oblivious transfer. In Section 6, we show that noisy transfer is equivalent to ordinary oblivious transfer. In Section 7, we show how to achieve oblivious transfer using quantum transfer as a building block.

2 Definitions.

In this section, we describe the various forms of information transfer mechanisms we will be considering. We define the two standard mechanisms, two nonstandard transfer mechanisms, and two weakened versions of the standard forms of oblivious transfer. We then define what we mean by reducing one form of oblivious transfer to another.

2.1 Standard forms of oblivious transfer.

There are two standard forms of oblivious transfer. We refer to these mechanisms as *oblivious transfer* and *1-2 oblivious transfer*.

Oblivious Transfer: In this protocol, Sam has a secret bit, b. At the end of the protocol, one of the following two events occurs, each with probability $\frac{1}{2}$.

1. Rachel learns the value of b.

2. Rachel gains no further information about the value of b (other than what Rachel knew before the protocol).

At the end of the protocol, Rachel knows which of these two events actually occurred, and Sam learns nothing.

Less formally, we can view this protocol as one in which Sam sends a letter to Rachel, which arrives exactly half the time.

1-2 Oblivious Transfer: In this protocol, Sam has two secret bits, b_0 and b_1. Rachel has a selection bit, s. At the end of the protocol, the following three conditions hold.

1. Rachel learns the value of b_s.

2. Rachel gains no further information about the value of b_{1-s}.

3. Sam learns nothing about the value of s.

Less formally, Sam has two secrets. Rachel can select exactly one of them, and Sam doesn't know which secret Rachel selected.

2.2 Two nonstandard transfer mechanisms.

We will consider two nonstandard transfer mechanisms, one motivated by coding theory, the other by quantum cryptography.

Noisy Transfer: In this protocol. Sam has a secret bit, b. Rachel has no information about b. At the end of the protocol, Rachel receives a bit b'. With probability 3/4, $b' = b$, otherwise $b' = \overline{b}$. Sam learns nothing.

This protocol may be thought of as simulating a noisy communication channel, in which a bit is flipped with probability 1/4. We can parameterize the above definition by replacing the 3/4 with a probability ρ. We call this *ρ-noisy transfer*. In this paper, we only consider the "standard" noisy transfer, where $\rho = 3/4$.

Quantum Transfer: In this protocol, Sam and Rachel have real numbers, θ and Θ. At the end of the protocol, Rachel receives a value which is 0 with probability $\cos^2(\theta - \Theta)$, and 1 otherwise. Sam learns nothing.

Dirtier Notions of Oblivious Transfer.
In describing oblivious transfers, we make two distinct specifications. First, we specify what information is being transferred. Second, we impose a set of security conditions, specifying what information each party is guaranteed *not* to know at the end of the protocol, and specifying that certain events cannot be controlled by either party. The definitions of oblivious transfer and 1-2 oblivious transfer are particularly stringent in their security conditions. In oblivious transfer, Sam has no control over whether Rachel receives b. In 1-2 oblivious transfer, Sam gains no information about Rachel's selection s. We would like to be able to handle cases in which a malicious Sam

can, thorough some form of cheating, violate these security conditions. This motivates the following definitions.

α-Slightly Oblivious Transfer: This protocol is the same as oblivious transfer, except that instead of Rachel learning bit b with probability $\frac{1}{2}$, she learns it with probability p. If Sam is nonmalicious, $p = \frac{1}{2}$. If Sam is malicious, he may choose any value of p he wishes, subject to $1 - \alpha \leq p \leq \alpha$.

α-1-2 Slightly Oblivious Transfer: This protocol is the same as $1-2$ oblivious transfer, except that at the conclusion of the protocol, a malicious Sam can guess Rachel's selection bit s with probability α.

In both these definitions, the interesting range for α is $\frac{1}{2} \leq \alpha < 1$.

Note that in these definitions, there is a careful distinction made between the powers of a malicious Sam verses the powers of a nonmalicious Sam. Since a malicious Sam is always more powerful than a nonmalicious Sam, it would at first seem natural to simply assume that Sam is malicious. However, we require that the protocols we build on top of these primitives meet the following two requirements: They must work when Sam is nonmalicious, and they must maintain their security conditions when Sam is malicious. So, for example, if one is building a protocol using a 3/4-slightly oblivious transfer subprotocol, one *cannot* require Sam to send 1000 bits, having at least 600 get through to Rachel. A malicious Sam could easily do this, but a nonmalicious Sam could not.

2.3 Our notion of reductions.

In the remainder of this paper, we will consider reductions from the standard oblivious transfers to the four nonstandard transfer mechanisms described above. We now briefly describe what we mean by a cryptographic reduction.

Essentially, our reduction is a black-box reduction. If we are reducing transfer mechanism Q to transfer mechanism R, we assume that we have a black box which implements protocol R. For instance, if we reduce to noisy transfer, we assume the existence of a black box which takes a bit from Sam, complements it with a certain probability, and sends the resulting answer to Rachel. In order to reduce Q to R, we specify a protocol which uses the black box for R, and takes an additional input, k, written in unary. We refer to k as the *security parameter* for the protocol. We require that the number of steps taken by each party during the execution of the protocol be polynomial in k.

Unlike computational notions of cryptographic reduction, we require our reductions to be information theoretic. Thus, in analyzing the security of our protocols, we assume that both Sam and Rachel are infinitely powerful, and thus able to break any form of encryption. For the execution of our protocols, however, we assume that Sam and Rachel are probabilistic polynomial-time interactive Turing machines.

However, we do adopt a common notion from cryptography, that of *statistical indistinguishability.* By this, we mean that we are willing to slightly relax our information transfer and security conditions. For instance, in the ideal form of 1-2 oblivious transfer, Rachel always recovers b_s, and gets no information about b_{1-s}. For the purposes of our reductions, we allow Rachel to fail to recover the correct value of b_s, but this event must occur with probability less than k^{-c} for any constant c. Likewise, we allow Rachel to have some edge over chance at guessing b_{1-s}, but this advantage must also be small. Whereas in the ideal model, Rachel could guess b_{1-s} with probability no greater than $1/2$, we merely require that Rachel not be able to guess b_{1-s} with probability greater than $1/2 + k^{-c}$, for any constant c.

If we make k sufficiently large, then the simulated 1-2 oblivious transfer will be effectively indistinguishable from the ideal transfer. Thus, for example, we can use the simulated transfer mechanism in place of the ideal one without compromising security.

3 Reducing 1-2 oblivious transfer to α-1-2 oblivious transfer.

In this section, we sketch the reduction from 1-2 oblivious transfer to α-1-2 oblivious transfer, proving Theorem 1. In other words, we are given a protocol in which Sam has two secrets, b_0 and b_1, and the following conditions hold.

1. Rachel learns b_s.
2. Rachel learns nothing about b_{1-s}.
3. Sam can predict s with probability at most α, assuming that he had no *a priori* knowledge of s.

Using this protocol as a subroutine, we construct a protocol where Sam cannot predict s with probability significantly greater than $\frac{1}{2}$.

Suppose Sam has two secrets, b_0, b_1. Consider the following protocol:

1: Sam chooses $(r_{0,1}, r_{0,2}, ..., r_{0,n-1})$ a list of $n-1$ random bits, and chooses $r_{0,n} = b_0 \bigoplus \bigoplus_{i=1}^{n-1} r_{0,i}$.

2: Sam selects a second list of bits defined by $r_{1,i} = r_{0,i} \bigoplus b_0 \bigoplus b_1$ for $1 \leq i \leq n$.

3: Sam transfers one bit out of each pair $(r_{0,i}, r_{1,i})$ to Rachel for $1 \leq i \leq n$ using their α-1-2 oblivious transfer.

4: Let c_i be the random variable specifying that Rachel received the bit $r_{c_i,i}$ at round i.

5: Rachel uses the bits she received to compute:

$$b_{C_n} = B_n$$

where

$$C_n = \bigoplus_{i=1}^{n} c_i \text{ and } B_n = \bigoplus_{i=1}^{n} r_{c_i,i}$$

to get one of b_0, b_1.

Using a simple argument, we can show that if Sam can predict c_i with probability no better than α, he can predict C_n with probability no better than $\frac{1}{2} + (2\alpha - 1)^n/2$. Thus, we can achieve, with exponential closeness, pure 1-2 oblivious transfer using only n calls to an α-1-2 slightly oblivious transfer protocol. This simplification of our earlier protocol is due to Brassard.

Note that Sam can choose some $r_{i,j}$ which do not obey the constraints given. However, this possible strategy gives him no further information about s, and may only serve to randomize Rachel's final answer. However, we cannot force Sam to give away his secrets in the first place, so this does not give him any more power than if he abided by the protocol.

4 Making honest reductions more robust.

In this section we develop a simple technique for strengthening some of our reductions. Using this technique, we can write simple reductions which depend on the receiver being honest, and in a fairly routine fashion, convert them to protocols which are robust against cheating by the receiver. This technique will be crucial in our reductions from 1-2 oblivious transfer to α-oblivious transfer and noisy transfer.

4.1 The general scenario.

We consider transfer mechanisms with the *verifiable obliteration* property. By this we mean that the transfer mechanism occasionally gives the receiver a value which is uncorrelated with the bit sent, and for which the verifier knows this fact. Two examples of such mechanisms are ordinary oblivious channel and α-oblivious transfer. Our intermediate goal is to implement some form or another of 1-2 oblivious transfer. Having accomplished this, we then try to apply the techniques of Section 3 to implement standard 1-2 oblivious transfer.

If the receiver is honest, the total obliteration property of the channel makes it very easy to design protocols for some form of 1-2 oblivious transfer. For instance, to reduce 1-2 oblivious transfer to ordinary oblivious transfer, we can use the following protocol.

Protocol Honest-Reduce(k)

1: Let b_0, b_1 be Sam's secret bits, let s be the bit Rachel wishes to see, and let k be the security parameter. Sam uniformly selects bits $c[1], \ldots, c[k]$, and transfers them through the oblivious transfer channel.

2: Rachel randomly picks i_0, i_1 such that she received $c[i_0]$, and didn't receive $c[i_1]$(for k large, i_0, i_1 will exist with high probability). She sends i_s, i_{1-s} to Sam.

3: Sam sends $b_0 \oplus c[i_s], b_1 \oplus c[i_{1-s}]$ to Rachel.

Remark: When we talk about the players sending messages to each other, we mean through a clear channel. By using trivial redundancy schemes (e.g. sending the same bit many times in succession), it is possible to simulate a clear channel using any of our noisy channels.

If Rachel is being honest, she will be able to reconstruct b_s and have no information about b_{1-s}. Here is where the total obliteration property is so useful: Rachel can arrange to have the secret she doesn't wish to receive encrypted with bits she has no information about.

Of course, in the above example, Rachel can easily learn both of Sam's secrets by picking i_0, i_1 such that she received both $c[i_0]$ and $c[i_1]$. In order to make our protocol more robust, we need a mechanism by which Rachel can verify that she really doesn't know one of these bits. As a first step, we modify Steps 2 and 3 of the above example, and include a checking phase which Sam may run instead of Step 3. Our modified phases are as follows,

2': Rachel randomly picks i_0, i_1 such that she received $c[i_0]$, and didn't receive $c[i_1]$(for k large, i_0, i_1 will exist with high probability). She picks a random bit, r and sends i_r, i_{1-r} to Sam.

3': Rachel sends $r \oplus s$ to Sam, who sends $b_0 \oplus c[i_s], b_1 \oplus c[i_{1-s}]$ to Rachel.

Our checking step is as follows: Rachel sends r to Sam, and tries to predict $c[j]$ for all $j \neq i_1$. Sam computes the number of correct guesses by Rachel. We note that going through the checking step gives Sam absolutely no information about s.

The intuition behind this checking step is that Rachel is allowed to select one of the two indices she gave to Sam, and say, "I didn't receive this bit." She then gives evidence to support this claim by trying to guess the values of the other bits. Now if Rachel excludes a bit she didn't receive, she will score better than if she excludes a bit she did receive. But if both of the indices she sent to Sam refer to bits she received, she will have to exclude one of these bits, and thus lower her expected score by 1/2.

Since the expected scores of an honest Rachel and a cheating Rachel are only slightly different, the above test may not seem very useful. However, we can amplify its effectiveness with the following trick. We have Sam and Rachel run Steps 1 and 2 of the protocol Q times in parallel, where for concreteness we set $Q = k^{10}$. Sam and Rachel then uniformly select a number $M \in [1, Q]$. This is easily accomplished using commital protocols. They run the checking step on all the games but the Mth, and Sam aborts if Rachel's combined score is less than

$$\frac{3}{4}k^{11} - k^6,$$

Otherwise, they play out the third step of game M. the k^6 term is a fairly arbitrary quantity which is much more than the standard deviation for the number of bits received, and much less than the number of rounds. The final protocol is as follows.

Protocol Reduce(k)

1: Let b_0, b_1 be Sam's secret bits, let s be the bit Rachel wishes to see, and let k be the security parameter. Set $Q = k^{10}$. For $j \in [1, Q]$, Sam uniformly selects bits $c^j[1], \ldots, c^j[k]$. He transfers thse bits to Rachel.

2: For $j \in [1, Q]$, Rachel randomly picks i_0^j, i_1^j such that she received $c^j[i_0]$, and didn't receive $c^j[i_1]$(for k large, i_0^j, i_1^j will exist with high probability). She picks a random bit, r^j and sends i_r^j, i_{1-r}^j to Sam.

2.5: (Checking Part) Sam and Rachel choose a random $M \in [1, Q]$. For $j \in [1, Q], j \neq M$, Rachel sends r^j and, for $i \in [1, k]$, sends her guesses for $c^j[i]$. If she didn't receive $c^j[i]$ she just guesses, and otherwise she sends the value she received. If the total number of correct answers is less than

$$\frac{3}{4}k^{11} - k^6,$$

then Sam aborts the protocol.

3: Rachel sends $r^M \oplus s$ to Sam, who sends $b_0 \oplus c^M[i_s^M], b_1 \oplus c^M[i_{1-s}^M]$ to Rachel.

If Rachel obeys the protocol, she will recover one bit, and learn nothing about the other. We now argue that even if she cheats, the probability that she gets information about two bits is small (though not neglible).

Lemma 4.1: In protocol **reduce** the probability that Rachel receives information about both b_0 and b_1 is at most $O(1/k^3)$.

Proof: (Sketch) We say that Rachel *cheats in game* i if she received both $c^M[i_0^M]$ or $c^M[i_1^M]$. Clearly, Rachel gets no information at all about b_0, b_1 until Step 3 of the protocol, so if Sam aborts in Step 2.5, she will get no useful information whatsoever. Furthermore, if Rachel did not cheat in round M, and hence did not receive $c^M[i_0^M]$ or $c^M[i_1^M]$, she will get information about at most one of b_0, b_1. Let *cheat* be equal to the number of games in which Rachel cheats. Therefore, the probability that Rachel successfully cheats is bounded above by

$$\frac{cheat}{k^{10}} \cdot prob(\text{Sam aborts in Step 2.5}).$$

Now if *cheat* is less than k^7, the lemma follows immediately, so we need only consider the case where $cheat = \Omega(k^7)$. With probability exponentially close to 1, Rachel will receive at most

$$\frac{1}{2}k^{11} + k^6,$$

bits. Therefore, with high probability, Rachel will be have received

$$\frac{1}{2}k^{11} - \Omega(k^7)$$

bits which she did not "disown" in Step 2.5. By standard techniques, we have that with probability exponentially close to 1, her final score will be at most

$$\frac{3}{4}k^{11} - \Omega(k^7).$$

However, this will cause Sam to abort. The lemma follows. ▮

It should be noted that this $1/k^3$ chance of Rachel successfully cheating is not insignificant. However, it is not hard to use this channel to implement a channel in which Rachel has only a neglible chance of successfully cheating.

5 Reducing 1-2 oblivious transfer to slightly oblivious transfer.

Using the machinery of the previous section, we can sketch our technique for proving Theorem 3. We essentially perform a three step reduction from 1-2 oblivious transfer to α-oblivious transfer. We show that if we run our **reduce** protocol, using α-oblivious transfer as a subprotocol, we implement a slightly mutated version of α-1-2 slightly oblivious transfer. As mentioned in the previous section, an additional security condition is weakened in that Rachel may have a slight chance of learning both secrets. Also, Sam may receive additional information about how good his prediction of Rachel's selection bit s is. However, despite these differences, we can use the same protocol that converts α-1-2 slightly oblivious transfer to 1-2 oblivious transfer, to get a 1-2 oblivious transfer protocol in which Rachel has a slight chance of learning both secrets. Finally, it is then straightforward to eliminate Rachel's chance of learning both secrets. For this extended abstract, we outline only the first result.

Lemma 5.1: If we run protocol **reduce**(k), using α-oblivious transfer as a subprotocol, we get a protocol with the following properties.

1. Rachel learns both b_0, b_1 with probability at most $O(1/k^3)$.
2. Sam can predict Rachel's selection, s, with probability at most

$$\frac{\alpha^2}{\alpha^2 + (1-\alpha)^2}.$$

Proof: (Sketch) Property 1 follows from Lemma 4.1. To prove property 2, we first note that Sam only receives information about s from game M. In the very best case for Sam, one of the two bits, $c^M[i_j^M]$, was sent with probability α, and the other bit, $c^M[i_{1-j}^M]$, was sent with probability $1-\alpha$. In this case, Sam should guess that $s = j$. Using elementary probability theory, the conditional probability that $s = j$ is given by

$$prob(s = j) = \frac{\alpha^2}{\alpha^2 + (1-\alpha)^2}.$$

The lemma follows. ▮

6 The power of noise.

In this section we consider the cryptographic power of an ordinary noisy communication channel, i.e. one which inverts a transmitted bit with some fixed probability. In this section we sketch portions of the proof that this family of transfer mechanisms can be used to implement 1-2 oblivious transfer, and hence a wide variety of secure two-party protocols.

6.1 A philosophical remark.

Noisy channels have been extensively studied in the field of coding theory, and it is interesting to see how our perspective differs from the more traditional one. Coding theory adopts the viewpoint that noise is a bad thing, to be eliminated as efficiently as possible. Given a noisy channel, a coding theorist tries to simulate a pristine, noiseless communication line.

From our point of view, an ideal communication line is a sterile, cryptographically uninteresting entity. Noise, on the other hand, breeds disorder, uncertainty, and confusion. Thus, it is the cryptographer's natural ally. The question we consider is whether this primordial uncertainty can be sculpted into the more sophisticated uncertainty found in secure two-party protocols. The result outlined in this section answers this question in the affirmative.

6.2 An outline of our reduction.

Our reduction consists of four main parts. We first show how to use a noisy transfer channel to simulate a very dirty transfer channel which has the total obliteration property. This allows us to start applying the techniques of Section 4. Using these techniques, we can show how to implement a version of 1-2 oblivious transfer similar to the one developed in Section 5. We can then use the proof of Theorem 1 to get an almost pure 1-2 oblivious transfer channel. This channel may be used to simulate a pure 1-2 oblivious transfer channel.

6.3 Using noise to implement a very dirty oblivious transfer.

The main difficulty of using a noisy channel to implement oblivious transfer is that whenever Sam sends Rachel some bits, Rachel has no idea which bits were probably correct, and which ones were not. To give Rachel a fighting chance, we can adopt the convention that Sam sends each bit twice in succession. If Rachel receives two bits, bb, that are the same, she considers this a "good" transmission, and that she received bit b. If she receives two different bits, then she treats this as a "bad " transmission. Bad bits convey no information about the bits which were originally sent, hence the total obliteration property holds for this simulated channel.

Of course, this convention is not ideal for a number of reasons. First, Sam can cheat, by sending illegal bit sequences, i.e. 01 an 10. These bits will be interpreted by Rachel as "bad," with probability 10/16(we assume that bits are flipped with probability 1/4), as "good" 1's, with probability 3/16, and as "good" 0's, with probability 3/16.

Even if Sam follows the convention, other problems will arise. A transmission of 11 will be correctly interpreted with probability only 9/16. With probability 6/16 it will be received as "bad," and with probability 1/16 it will be interpreted as a "good" 0. Undaunted by these problems, we call the channel simulated by this convention "very dirty oblivious transfer." In very dirty oblivious transfer, Sam may send one of three values, 0(equivalent to sending 00), 1 (equivalent to sending 11), or "bad"(equivalent to sending 01 or 10). Bits received are classified as $0, 1$ or "bad."

6.4 The reduction for the case of honest parties.

For motivational purposes, we first sketch a protocol which works in the case where both parties are honest. They will obey the protocol, but afterward will try to glean extra knowledge from their record of the conversation. We denote Sam's two secrets as b_0 and b_1, and assume that Rachel wishes to know the value of b_s.

Protocol Honest-Noise-Reduce(k)

1: Sam picks bits $C = c_1, \ldots, c_{k^5}$ uniformly, and sends them to Rachel via a very dirty oblivious channel. We denote the bits Rachel actually received as $C' = c'_1, \ldots, c'_{k^5}$.

2: Rachel picks a set of indices I_s subject to the constraints,

1. $|I_s| = k$
2. For all $m \in I_s$, $c'_m \in \{0, 1\}$.

She then picks a set of indices, I_{1-s}, subject to

1. $|I_{1-s}| = k$
2. For all $m \in I_{1-s}$, c'_m is "bad."

Rachel sends I_0, I_1 to Sam, over a clear channel.

3: We write $I_m = \{i^1_m, \ldots, i^k_m\}$. For $m \in \{0, 1\}$, and $1 \leq j \leq k$, Sam sends Rachel the values

$$b^j_m = b_m \oplus c_{i^j_m}.$$

Rachel computes her guess for b_s, b'_s, according to the formula

$$b'_s = majority(\{b^j_s \oplus c_{i^j_s} | j \in [1, k]\}).$$

Remark: It should be noted that this protocol is essentialy the same as the protocol given in Section 4. the main difference is that sets of indices are used instead of single indices. This slight modification is necessary because Rachel does not know with complete certainty the value of any of the bits sent her.

If k is chosen reasonably large, then b'_s will equal b_s with high probability. The values of b^j_{1-s} yield no information to Rachel, since she has no information about $c_{i^j_{1-s}}$. Sam gets absolutely no information about s, since he has no information about which bits were properly transmitted.

6.5 Making the honest protocol robust against cheating.

The protocol described above does not work against active attacks by either Sam or Rachel. The protocol counts on Sam to never send "bad" in any of his transmissions. Whereas he has no a priori reason to believe that some bit $c_j \in \{0, 1\}$ was more likely to be received as "bad" than any other bit sent, a bit

c_j ="bad" is more likely to be received as "bad." Thus, if he sends even one "bad" bit, and it shows up in set I_j, then $s = 1 - j$ with probability greater than $\frac{1}{2}$. By sending a large number of "bad" bits, Sam can determine s with very high probability.

Likewise, the protocol counts on Rachel to insure that $c'_{i^j_{1-s}}$ ="bad." If Rachel violates this condition even once, she can gain information about b_{1-s}.

6.5.1 Constraining Sam's information.

If we could ensure that Sam doesn't transmit any 'bad" bits, then he could not get any information about s. We can't meet such a rigid requirement, but we can meet a somewhat weaker one. Specifically, we can ensure that the fraction of "bad" bits sent by Sam is manageably small. This is done by simply having Rachel check that no more than a certain number of the bits she receives are "bad." If Sam sends k^5 bits, all either 0 or 1, then with high probability, only $(6/16)k^5 + o(k^3)$ of them will be received as "bad." However, if k^3 of the bits initially sent are bad, then with high probability, $(6/16)k^5 + (4/16)k^3 + o(k^3)$ of the bits received will be "bad." Hence, we can have Rachel reject whenever sufficiently many of her bits are bad.

Once Sam is constrained to making only k^3 of his bits "bad," then we can bound his chance of predicting s away from 1. This assertion follows from the fact that with probability bounded away from 1, none of the bits indexed in I_0, I_1 will have been sent as "bad." This suffices to implement α-1-2 oblivious transfer, for some $\alpha < 1$.

6.5.2 Dealing with a malicious Rachel.

Dealing with a malicious Rachel is somewhat trickier. Essentially, we want some way of testing that Rachel doesn't know anything about the bits indexed in ones of her I_j's. This is essentially the same problem we dealt with in Section robust. We briefly describe below the modifications we make to our protocol.

1. Stages 1 and 2 of the algorithm are run N times in parallel, where N is a suficiently large power of k. We denote the bit sets sent and received as $C^i = c^i_1, \ldots, c^i_n$, and $C^{i'} = {c^{i}}'_1, \ldots, {c^{i}}'_n$ respectively, for $1 \le i \le N$. Likewise, we denote the index sets as I^i_0, I^i_1.

2. Rachel doesn't send I^i_0, I^i_1 in Step 2 of the protocol, but rather $I^i_{r_i}, I^i_{1-r_i}$, where $r_i \in \{0, 1\}$ is chosen at random.

3. We add a checking stage between Steps 2 and 3. First, Sam and Rachel choose some random M between 1 and N. Then, for $j \ne M$, Rachel reveals $r_j \oplus s$. Given $r_j \oplus s$, Sam knows which of the index sets sent to it corresponds to I^j_{1-s}, without getting any information about s. Finally, Rachel gives her values of ${c^j}'_a$, for all $a \notin I^j_{1-s}$. Sam checks that ${c^j}'_a = c^j_a$ sufficiently often, and aborts otherwise.

Now, if many sets I^j_{1-s} indexed bits that were not "bad," then this would lower the average number of bits not referred to by the sets I^j_{1-s} which were properly received. We then employ esentially the same argument as in Section 5. We can show that if, for more than a relatively small set of i's, Rachel has I^i_{1-s} refer to any bit she received, she will be caught with high probability.

6.6 Properties of the final protocol.

Combining the fixes discussed above, we have something very close to α-1-2 slightly oblivious transfer. Our protocol has the properties that

1. Sam learns s with probability bounded away from 1.

2. Rachel can predict b_{1-s} with probability at most $\frac{1}{2} + 1/k^c$, for any fixed c we wish. This c depends on the value of N chosen.

Using the technique for reducing 1-2 oblivious transfer to 1-2 slightly oblivious transfer, we get a protocol in which Sam has no information about s, and Rachel can predict b_{1-s} with probability at most $\frac{1}{2} + 1/k^{c'}$, for some other constant c' depending on c. By choosing our parameters correctly, we can make c' as large as we wish. Finally, it is not difficult to reduce this probability to something exponentially close to $\frac{1}{2}$.

7 Reducing oblivious transfer to quantum transfer.

In this section we describe how the basic oblivious transfer protocol can be achieved using the properties of polarized light. This work relies on some ideas first described in [BBBW], a paper about quantum cryptography. We consider an ideal model where the parties can transmit single polarized photons. We also assume that the polarization can be set to some exact real angle θ determined by its creator. The same holds for the angle of the "reading" device. Also we assume that the behavior of the photons correspond exactly to the equations of quantum mechanics.

Let us abstract the properties achieved by the photons. Sam can choose any real value of an angle θ. The quantum channel is a device through which the value of θ can be transfered with the properties that:

- The only thing Rachel can learn about θ is the output of the random predicate $\gamma(\theta, \Theta)$ such that $Prob(\gamma(\theta, \Theta) = 0) = \cos^2(\theta - \Theta)$ where Θ is chosen by her.
- She can get the value of this predicate only once.

The details of the physics involved in the actual implementation of this channel can be found in [BBBW] and in the forthcoming [BB].

The main thing to notice is that $\gamma(\theta, \Theta) = 0$ when $\theta = \Theta$ and $Prob(\gamma(\theta, \Theta) = 0) = \frac{1}{2}$ when $|\theta - \Theta| = \frac{\pi}{4}$.

7.1 Protocol with honest participants.

Assume Sam wants to send a bit b. From the facts above we can deduce a very simple oblivious transfer protocol when both parties are honest:

1: Sam chooses a random θ in $\{0, \frac{\pi}{4}\}$

2: Rachel chooses a random Θ in $\{0, \frac{\pi}{4}\}$

3: Rachel gets a bit $b' = \gamma(\theta + b\frac{\pi}{2}, \Theta)$ using the quantum channel with Sam.

4: Sam tells Rachel what θ was.

The fundamental property used in this protocol is that for all angle θ and bit b we have that

$$Prob(b' = b | b = 1)$$
$$= Prob(\gamma(\theta + \frac{\pi}{2}, \Theta) = 1)$$
$$= 1 - \cos^2(\theta + \frac{\pi}{2} - \Theta)$$
$$= \sin^2(\theta + \frac{\pi}{2} - \Theta)$$
$$= \cos^2(\theta - \Theta)$$

and

$$Prob(b' = b | b = 0)$$
$$= Prob(\gamma(\theta, \Theta) = 0)$$
$$= \cos^2(\theta - \Theta)$$

Therefore

$$Prob(b' = b) = \cos^2(\theta - \Theta)$$

Note that $Prob(\theta = \Theta) = \frac{1}{2}$, so with probability $\frac{1}{2}$ Rachel will get $b' = b$ and know that she did since she learned θ in the last step. Also, $Prob(|\theta - \Theta| = \frac{\pi}{4}) = \frac{1}{2}$. When this case occures, Rachel learns nothing about b.

Now the trouble is, we don't know what Rachel is doing. She can use any angle Θ to get the bit b'. At the end of the protocol she can make sure that she gets "something" about b in each case. By using $\Theta = \frac{\pi}{8}$, for example, she gets that $Prob(b' = b) = \cos^2(\frac{\pi}{8}) \approx 0.8536$ whether $\theta = 0$ or $\theta = \frac{\pi}{4}$.

7.2 Dealing with a bad Rachel.

The key insight is that for whatever angle Θ Rachel uses, the following holds:

$$Prob(1 - \cos^2(\frac{\pi}{8}) \leq \cos^2(\theta - \Theta) \leq \cos^2(\frac{\pi}{8})) = \frac{1}{2}$$

This is because θ is chosen at random among $\{0, \frac{\pi}{4}\}$ (Notice that $1 - \cos^2(\frac{\pi}{8}) = \cos^2(\frac{3\pi}{8})$). Therefore

$$Prob(1 - \cos^2(\frac{\pi}{8}) \leq Prob(b = b') \leq \cos^2(\frac{\pi}{8})) = \frac{1}{2},$$

which means that for some constant α, $0 < \alpha < 1$, we have

$$Prob(1 - \alpha \leq Prob(b = b') \leq \alpha) = \frac{1}{2}$$

Assume Sam wants to send bit b. Our better oblivious transfer protocol is the following:

1: Sam picks r_1, r_2, r_3,... r_n for some large n to be determined.

2: Sam and Rachel use the above protocol to transfer r_1, r_2, r_3,..., r_n . Rachel gets r'_1, r'_2, r'_3,..., r'_n .

3: Rachel picks two disjoint subsets U, V of size $\frac{n}{3}$ from the set $\{1, 2, ..., n\}$ such that she received each of the bits r_u for each u∈U.

4: Rachel tells Sam what U and V are in a random order.

5: Sam chooses one of these sets (call it X) and tells Rachel which one he selected.

6: Sam computes $b \bigoplus m$ and sends it to Rachel, where $m = \bigoplus_{x \in X} r_x$

7: If $X = U$ then Rachel can compute m and get b, otherwise $X = V$ and Rachel cannot learn anything significant about b.

7.3 Why does this work?

The reason why this protocol works is that, with very high probability, Rachel will not be able to compute with certainty $\bigoplus_{x \in X} r_x$ for at least one of $X = U$ or $X = V$. This is because U and V together include $\frac{2n}{3}$ elements, whereas the expected number of $i \in \{1, 2, ..., n\}$ such that $1 - \alpha \leq Prob(r_i = r'_i) \leq \alpha$ is only $\frac{n}{2}$. In fact, with very high probability, at least $\frac{n}{6}$ of the elements j in $U \bigcup V$ are such that $1 - \alpha \leq Prob(r_j = r'_j) \leq \alpha$. Therefore one of U or V most include at least $\frac{n}{12}$ of them. The XOR of these $\frac{n}{12}$ bits is almost completely unpredictable and thus wipes out any information that might be obtained from the global XOR for that subset. A detailed analysis is ommitted from this extended abstract; we refer the reader to [C] for a detailed exposition of a very similar analysis.

This result can be easily extended to a more general case as long as the probability that $Prob(b = b')$ is bounded by a constant less than 1, for some constant fraction of the time.

Another thing that Rachel could do is wait until she learns the value of θ before evaluating the predicate γ, making sure that she makes her measurement with the right angle. The solution to this problem is for Sam to challenge Rachel's honesty in her reading step. Instead of step 4, in the original protocol, Sam asks Rachel to "commit" to the values of b' and Θ she used and randomly decides to go on as before revealing θ or asks her to reveal the values of b' and Θ she used. If Rachel is saving photons without reading them she has probability $\frac{1}{4}$ of being clearly wrong. The details of this argument are ommitted from this extended abstract.

7.4 What about Sam?

But what about if Sam tries to cheat? We argue that in this case there is nothing he can get. Basically he does not get anything useful back from Rachel during the protocol. His judgment about whether Rachel got the bit or not is equivalent to the knowledge of the value Θ which he never learns for the bits that are actually used in the protocol.

8 Acknowledgments.

We would like to acknowledge Gilles Brassard, Ernie Brickell, Ivan Damgård, Cynthia Dwork, Joan Feigenbaum, Shafi Goldwasser, and Silvio Micali for their valuable comments, ideas, and encouragement.

9 References

[**BB**] Bennett, Charles, and Gilles Brassard, "Quantum Cryptography," *To appear.*

[**BBBW**] Bennett, Charles, Gilles Brassard, Seth Breidbart and Stephen Weisner. "Quantum Cryptography, or Unforgeable Subway Tokens," *Proceedings Crypto '82*, Plenum Press.

[**BCR1**] Brassard, Gilles, Claude Crépeau, and Jean-Marc Robert. "All-or-Nothing Disclosure of Secrets," *Proceedings Crypto 86*, Springer-Verlag, 1987.

[**BCR2**] Brassard, Gilles, Claude Crépeau, and Jean-Marc Robert. "Information Theoritic Reductions Among Disclosure Problems," *Proceedings of the* 27^{th} *FOCS*, IEEE, 1986, 168–173.

[**BFM**] Blum, Manuel, Paul Feldman, and Silvio Micali, "Noninteractive Zero-Knowledge Proofs and their Applications," *Proceedings of the* 20^{th} *STOC*, ACM, 1988.

[**BGW**] Ben-Or, Michael, Shafi Goldwasser, and Avi Wigderson, "Completeness Theorems for Noncryptographic Fault-tolerant Distributed Computation," *Proceedings of the* 20^{th} *STOC*, ACM, 1988.

[**C**] Crépeau Claude, "Equivalence Between Two Flavours of Oblivious Transfer", *Proceedings of Crypto 87*, 1988,Springer-Verlag.

[**CCD**] D. Chaum, C. Crépeau and I. Damgård, "Multiparty unconditionally secure protocols," *Proceedings of the* 20^{th} *STOC*, ACM, 1988.

[**EGL**] Even S., Goldreich O., and A. Lempel, "A Randomized Protocol for Signing Contracts," *CACM*, vol. 28, no. 6, 1985, pp. 637-647.

[**FM**] Feldman, Paul, Silvio Micali. "Byzantine Agreement from Scratch," *Proceedings of the* 20^{th} *STOC*, ACM, 1988.

[**GV**] Goldreich O., Vainish, "Multi-Party Protocols, an Efficiency Improvement," *Proceedings, Advances in Cryptology- Crypto 87.*

[**IY**] Implazziago, Russell, and Moti Yubg, "Direct Minimum-Knowledge Computations," *Proceedings, Advances in Cryptology- Crypto 87.*

[**K**] Kilian, Joe, "On The Power of Oblivious Transfer," *Proceedings of the* 20^{th} *STOC*, ACM, 1988.

[**R**] Rabin, M., "How to exchange secrets by oblivious transfer," Tech. Memo TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[**Y**] Yao, Andrew C., "How to Generate and Exchange Secrets," *Proceedings of the* 27^{th} *FOCS*, IEEE, 1986, 162–167.