Quantum Information and Computation, Vol. 12, No. 7&8 (2012) 0609–0619

ON THE EXISTENCE OF LOSS-TOLERANT QUANTUM OBLIVIOUS TRANSFER PROTOCOLS

JAMIE SIKORA

Combinatorics & Optimization, University of Waterloo
200 University Ave. W., Waterloo, ON N2L 3G1, Canada

Received November 3, 2010
Revised March 23, 2012

Oblivious transfer is the cryptographic primitive where Alice sends one of two bits to Bob but is oblivious to the bit received. Using quantum communication, we can build oblivious transfer protocols with security provably better than any protocol built using classical communication. However, with imperfect apparatus, one needs to consider other attacks. In this paper, we present an oblivious transfer protocol which is impervious to lost messages.

Keywords: quantum cryptography, oblivious transfer, loss-tolerance
Communicated by: S Braunstein & H Zbinden

1 Introduction

Quantum information allows us to perform certain cryptographic tasks which are impossible using classical information alone. In 1984, Bennett and Brassard gave a quantum key distribution scheme which is unconditionally secure against an eavesdropper [1, 2, 3]. This led to many new problems including finding quantum protocols for other cryptographic primitives such as *coin-flipping* and *oblivious transfer*.

Coin-flipping is the cryptographic primitive where Alice and Bob generate a random bit over a communication channel. We discuss two kinds of coin-flipping protocols, *weak coin-flipping* where Alice wants outcome 0 and Bob wants outcome 1, and *strong coin-flipping* where there are no assumptions on desired outcomes. We define weak coin-flipping below.

Definition 1 (Weak coin-flipping (WCF) protocol): A *weak coin-flipping* protocol, denoted WCF, with cheating probabilities $(A_{\mathrm{WCF}}, B_{\mathrm{WCF}})$ and bias $\varepsilon_{\mathrm{WCF}}$ is a protocol with no inputs and output $c \in \{0, 1\}$ satisfying:

- if Alice and Bob are honest, they output the same randomly generated bit c;
- A_{WCF} is the maximum probability dishonest Alice can force honest Bob to accept the outcome $c = 0$;
- B_{WCF} is the maximum probability dishonest Bob can force honest Alice to accept the outcome $c = 1$;
- $\varepsilon_{\mathrm{WCF}} := \max\{A_{\mathrm{WCF}}, B_{\mathrm{WCF}}\} - 1/2$.

The idea is to design protocols which protect honest parties from cheating parties and there are no security guarantees when both parties are dishonest. We can assume neither party aborts in a WCF protocol. If, for instance, Alice detects Bob has cheated then she may declare herself the winner, i.e., the outcome is $c = 0$. This is not the case in strong coin-flipping since there is no sense of "winning."

Definition 2 (Strong coin-flipping (SCF) protocol): A *strong coin-flipping* protocol, denoted SCF, with cheating probabilities $(A_{\text{SCF}}, B_{\text{SCF}})$ and bias ε_{SCF} is a protocol with no inputs and output $c \in \{0, 1, \text{abort}\}$ satisfying:

- if Alice and Bob are honest, then they never abort and they output the same randomly generated bit $c \in \{0, 1\}$;
- A_{SCF} is the maximum probability dishonest Alice can force honest Bob to accept some outcome $c = a$, over both choices of $a \in \{0, 1\}$;
- B_{SCF} is the maximum probability dishonest Bob can force honest Alice to accept some outcome $c = b$, over both choices of $b \in \{0, 1\}$;
- $\varepsilon_{\text{SCF}} := \max\{A_{\text{SCF}}, B_{\text{SCF}}\} - 1/2$.

We note here that SCF protocols can be used as WCF protocols. The only issue is if the outcome is "abort." In this case, the party who detected the cheating announces themselves the winner. Doing this, the bias in the WCF protocol is the same as in the SCF protocol.

Aharonov, Ta-Shma, Vazirani, and Yao [4] first showed the existence of an SCF protocol with bias $\varepsilon_{\text{SCF}} < 1/2$ followed shortly by Ambainis [5] who showed an SCF protocol with bias $\varepsilon_{\text{SCF}} = 1/4$. As for lower bounds, Lo and Chau [6] showed that bias $\varepsilon_{\text{SCF}} = 0$ is impossible. Kitaev [7], and later Gutoski and Watrous [8], extended this result to show that the bias of *any* SCF protocol satisfies $\varepsilon_{\text{SCF}} \geq 1/\sqrt{2} - 1/2$. This bound was proven to be tight by Chailloux and Kerenidis [9] who showed the existence of protocols with bias $\varepsilon_{\text{SCF}} < 1/\sqrt{2} - 1/2 + \delta$ for any $\delta > 0$.

As for WCF protocols, it was shown that the bias could be less than Kitaev's bound. For example, the protocols in [10, 11, 12] provide biases of $\varepsilon_{\text{WCF}} = 1/\sqrt{2} - 1/2$, $\varepsilon_{\text{WCF}} = 0.239$, and $\varepsilon_{\text{WCF}} = 1/6$, respectively. The best known lower bound for WCF is by Ambainis [5] who showed that a protocol with bias ε_{WCF} must use $\Omega(\log\log(1/\varepsilon_{\text{WCF}}))$ rounds of communication. Then, in a breakthrough result, Mochon [13] showed the existence of WCF protocols with bias $\varepsilon_{\text{WCF}} < \delta$ for any $\delta > 0$.

Oblivious transfer is the cryptographic primitive where Alice sends to Bob one of two bits but is oblivious to the bit received. We define oblivious transfer and its notions of cheating below.

Definition 3 (Oblivious transfer (OT) protocol): An *oblivious transfer* protocol, denoted OT, with cheating probabilities $(A_{\text{OT}}, B_{\text{OT}})$ and bias ε_{OT} is a protocol *with inputs* satisfying:

- Alice inputs two bits (x_0, x_1) and Bob inputs an index $b \in \{0, 1\}$;

- when Alice and Bob are honest they never abort, Bob learns x_b perfectly, Bob gets no information about $x_{\bar{b}}$, and Alice gets no information about b;
- A_{OT} is the maximum probability dishonest Alice can learn b without Bob aborting the protocol;
- B_{OT} is the maximum probability dishonest Bob can learn $x_0 \oplus x_1$ without Alice aborting the protocol;
- $\varepsilon_{\text{OT}} := \max\{A_{\text{OT}}, B_{\text{OT}}\} - 1/2$.

When a party cheats, we only refer to the probability which they can learn the desired values without the other party aborting. For example, when Bob cheats, we do not require that he learns either bit with probability 1.

In the OT definition above, there can be different ways to interpret the bias. For example, we could consider worst-case choices over inputs, we could assume the inputs are chosen randomly, etc. The protocol construction given in this paper is independent of how the inputs are chosen so this is not an issue.

Like weak coin-flipping, oblivious transfer has a related primitive which is useful for the analysis in this paper.

Definition 4 (Randomized oblivious transfer (Random-OT) protocol): A *randomized oblivious transfer* protocol, which we denote as Random-OT, with cheating probabilities $(A_{\text{ROT}}, B_{\text{ROT}})$ and bias ε_{ROT} is a protocol with *no inputs* satisfying:

- Alice outputs two randomly generated bits (x_0, x_1) and Bob outputs two bits (b, x_b) where $b \in \{0, 1\}$ is independently, randomly generated;
- when Alice and Bob are honest they never abort, Bob gets no information about $x_{\bar{b}}$, and Alice gets no information about b;
- A_{ROT} is the maximum probability dishonest Alice can learn b without Bob aborting the protocol;
- B_{ROT} is the maximum probability dishonest Bob can learn $x_0 \oplus x_1$ without Alice aborting the protocol;
- $\varepsilon_{\text{ROT}} := \max\{A_{\text{ROT}}, B_{\text{ROT}}\} - 1/2$.

We note here that a protocol is considered *fair* if the cheating probabilities for Alice and Bob are equal and *unfair* otherwise.

OT is an interesting primitive since it can be used to construct secure two-party protocols [14]. See also [15], [16], [17]. It was shown by Lo [18] that $\varepsilon_{\text{OT}} = 0$ is impossible. This result was improved by Chailloux, Kerenidis, and Sikora [19] who showed that the bias of every oblivious transfer protocol satisfies $\varepsilon_{\text{OT}} \geq 0.0852$.

Various settings for oblivious transfer have been studied such as the bounded-storage model [20] and the noisy-storage model [21]. In this paper, we study only information theoretic security but we allow the possibility of lost messages (more on this below). Oblivious transfer has a rich history, has various definitions, and has many names such as the *set membership problem* [22] or *private database querying* [23].

A *loss-tolerant protocol* is a quantum cryptographic protocol which is impervious to lost messages. That is, neither Alice nor Bob can cheat more by declaring that a message was lost (even if it was received) or by sending blank messages deliberately. We prefix a protocol with "LT-" to indicate that it is loss-tolerant.

The idea of loss-tolerance was first applied to strong coin-flipping by Berlin, Brassard, Bussieres, and Godbout in [24]. They showed a vulnerability in the best known coin-flipping protocol construction by Ambainis [5]. They circumvented this problem and presented an LT-SCF protocol with bias $\varepsilon_{\mathrm{SCF}} = 0.4$. Aharon, Massar, and Silman generalized this protocol to a family of LT-SCF protocols with bias slightly smaller at the cost of using more qubits in the communication [25]. Chailloux added an encryption step to the protocol in [24] to improve the bias to $\varepsilon_{\mathrm{SCF}} = 0.359$ [26]. The best known protocol for LT-SCF is by Ma, Guo, Yang, Li, and Wen [27] who use an EPR-based protocol which attains a bias of $\varepsilon_{\mathrm{SCF}} = 0.3536$. It remains an open problem to find the best possible biases for LT-WCF and LT-SCF. In fact, we do not even know if there is an LT-WCF protocol with bias less than the best possible bias for LT-SCF; they may share the same smallest possible bias.

The first approach to designing loss-tolerant oblivious transfer protocols was by Jakobi, Simon, Gisin, Bancal, Branciard, Walenta, and Zbinden [23]. They designed a loss-tolerant protocol for private database querying which is also known as "1-out-of-N oblivious transfer." The protocol is not technically an oblivious transfer protocol (using the definition in this paper) since an honest Bob may receive too much information. However, it is practical in the sense that it is secure against the most evident attacks. The backbone of their protocol is the use of a quantum key distribution scheme. This differs from the loss-tolerant protocol in this paper which is based on weak coin-flipping.

1.1 The results of this paper

We first present a protocol in Section 2 and prove it is not loss-tolerant. Then, in Section 3, we show how to build LT-OT protocols from LT-WCF and LT-Random-OT protocols. Namely, we prove the following theorem.

Theorem 5: Suppose there is an LT-WCF protocol with cheating probabilities $(A_{\mathrm{WCF}}, B_{\mathrm{WCF}})$ and bias $\varepsilon_{\mathrm{WCF}}$ and an LT-Random-OT protocol with cheating probabilities $(A_{\mathrm{ROT}}, B_{\mathrm{ROT}})$ and bias $\varepsilon_{\mathrm{ROT}}$. Then there exists an LT-OT protocol with cheating probabilities

$$A_{\mathrm{OT}} = A_{\mathrm{WCF}}\,|A_{\mathrm{ROT}} - B_{\mathrm{ROT}}| + \min\{A_{\mathrm{ROT}}, B_{\mathrm{ROT}}\}, \tag{1}$$

$$B_{\mathrm{OT}} = B_{\mathrm{WCF}}\,|A_{\mathrm{ROT}} - B_{\mathrm{ROT}}| + \min\{A_{\mathrm{ROT}}, B_{\mathrm{ROT}}\}. \tag{2}$$

This protocol has bias

$$\varepsilon_{\mathrm{OT}} \leq |A_{\mathrm{ROT}} - B_{\mathrm{ROT}}| + \min\{A_{\mathrm{ROT}}, B_{\mathrm{ROT}}\} - 1/2 = \varepsilon_{\mathrm{ROT}}. \tag{3}$$

We have $\varepsilon_{\mathrm{OT}} < \varepsilon_{\mathrm{ROT}}$ when $\varepsilon_{\mathrm{WCF}} < 1/2$ and $A_{\mathrm{ROT}} \neq B_{\mathrm{ROT}}$. Furthermore, the LT-OT protocol is fair when the LT-WCF protocol is fair.

In Subsection 3.4, we construct an unfair LT-Random-OT protocol with cheating probabilities $(A_{\mathrm{ROT}}, B_{\mathrm{ROT}}) = (1, 1/2)$. Combining this with the fact that there is a fair LT-WCF protocol with bias $\varepsilon_{\text{WCF}} = 0.3536$ [27], we get the following corollary.

Corollary 6: There exists a fair LT-OT protocol with bias $\varepsilon_{\text{OT}} = 0.4268$.

2 An Example of a Random-OT Protocol that is Not Loss-Tolerant

In this section, we examine a protocol for Random-OT and show it is not loss-tolerant. This protocol has the same vulnerability as the best known coin-flipping protocol constructions based on bit-commitment, see [24] for details.

Protocol 7 (A Random-OT protocol [19]):

(i) Bob randomly chooses $b \in \{0,1\}$ and sends Alice half of the two-qutrit state

$$|\phi_b\rangle := \frac{1}{\sqrt{2}}|bb\rangle + \frac{1}{\sqrt{2}}|22\rangle. \tag{4}$$

(ii) Alice randomly chooses $x_0, x_1 \in \{0,1\}$ and applies the following unitary to the qutrit

$$|0\rangle \to (-1)^{x_0}|0\rangle, \quad |1\rangle \to (-1)^{x_1}|1\rangle, \quad |2\rangle \to |2\rangle. \tag{5}$$

(iii) Alice returns the qutrit to Bob. Bob now has the two-qutrit state

$$\frac{(-1)^{x_b}}{\sqrt{2}}|bb\rangle + \frac{1}{\sqrt{2}}|22\rangle. \tag{6}$$

(iv) Bob performs the measurement $\{\Pi_0 := |\phi_b\rangle\langle\phi_b|,\ \Pi_1 := \mathrm{I} - \Pi_0\}$ on the state.

(v) If the outcome is Π_0 then $x_b = 0$. If the outcome is Π_1 then $x_b = 1$.

(vi) Any lost messages are declared and the protocol is restarted from the beginning.

It has been shown in [19] that Bob can learn $x_0 \oplus x_1$ with probability 1 and Alice can learn b with maximum probability 3/4. However, this does not take into account "lost-message strategies." We now show such a strategy and how Alice can learn b perfectly. Suppose Alice measures the first message in the computational basis. If she sees outcome "0" or "1" then she knows Bob's index b with certainty. If the outcome is "2" then she replies to Bob, "Sorry, your message was lost." Then they restart the protocol and Alice can measure again. Eventually, Alice will learn b perfectly proving this protocol is not loss-tolerant.

This protocol illustrates another interesting point about the design of OT protocols. One may not be able to simply change the amplitudes in the starting states to balance the cheating probabilities. For example, if we were to change the amplitudes in $|\phi_b\rangle$, then Bob would have a nonzero probability of getting the wrong value for x_b. Thus, balancing an unfair OT protocol is not as straightforward as it can be in coin-flipping.

3 Constructing Loss-Tolerant Oblivious Transfer Protocols

In this section, we prove Theorem 5 by constructing an LT-OT protocol from an LT-WCF protocol and a (possibly unfair) LT-Random-OT protocol. In doing so, we have to overcome some issues that are not present when designing LT-SCF protocols. These issues include:

- it is not always possible to simply reset a protocol with inputs;
- balancing the cheating probabilities can be difficult;
- it is not possible to switch the roles of Alice and Bob since Bob must be the receiver;
- an honest party must not learn extra information about the other party's inputs (or outputs in the case of Random-OT).

We deal with these issues by reducing the problem one step at a time. First we reduce the task of finding LT-OT protocols to finding LT-Random-OT protocols in Subsection 3.1. Then we build an LT-Random-OT protocol from an LT-WCF protocol and two (possibly unfair) LT-Random-OT protocols in Subsection 3.2. In Subsection 3.3, we show how to create the two LT-Random-OT protocols from a single LT-Random-OT protocol. Finally, we show an unfair LT-Random-OT protocol in Subsection 3.4 to prove Corollary 6.

3.1 *Equivalence between* LT-OT *protocols and* LT-Random-OT *protocols with respect to bias*

Having a protocol with inputs is an issue when building protocols loss-tolerantly. In recent LT-SCF protocols, if messages were lost for any reason, then the protocol is simply restarted at some point, but this is not always an option with OT because the inputs could have context, e.g., Alice's bits could be database entries. For this reason, we cannot simply "reset" them and repeat the protocol. To remedy this issue, we use Random-OT.

It is well known that OT and Random-OT share the same cheating probabilities, i.e., if there exists an OT protocol with cheating probabilities $(A_{\mathrm{OT}}, B_{\mathrm{OT}}) = (x, y)$ then there exists a Random-OT protocol with cheating probabilities $(A_{\mathrm{ROT}}, B_{\mathrm{ROT}}) = (x, y)$, and vice versa. For completeness, we show these reductions and prove they preserve loss-tolerance.

Protocol 8 (LT-Random-OT from LT-OT):

(i) Alice randomly chooses $x_0, x_1 \in \{0, 1\}$ and Bob randomly chooses $b \in \{0, 1\}$.

(ii) Alice and Bob input their choices of bits above into the LT-OT protocol so that Bob learns x_b.

(iii) Alice outputs (x_0, x_1) and Bob outputs (b, x_b).

It is straightforward to see that this reduction preserves the loss-tolerance of the LT-OT protocol since we are only restricting how the inputs are chosen. More interesting is the reduction from LT-Random-OT to LT-OT.

Protocol 9 (LT-OT from LT-Random-OT):

(i) Alice and Bob decide on their desired choices of inputs to the LT-OT protocol.

(ii) Alice and Bob use an LT-Random-OT protocol to generate the output (x_0, x_1) for Alice and (b, x_b) for Bob.

(iii) Bob tells Alice if his output bit b is equal to his desired index. If it is not equal, Bob changes it and Alice switches her two bits.

(iv) Alice tells Bob which of her two bits (x_0, x_1) are equal to her desired inputs. Alice and Bob flip their outcome bits accordingly.

This reduction is a way to derandomize the outputs of the LT-Random-OT protocol. We see that this also preserves the loss-tolerance of the LT-Random-OT protocol since classical information can simply be resent if lost in transmission.

Using the reductions above, we have reduced the task of finding LT-OT protocols to finding LT-Random-OT protocols.

3.2 Creating LT-Random-OT *protocols*

There is a simple construction of an SCF protocol with bias $\varepsilon \approx 3/4$ and it proceeds as follows. Alice and Bob first use a WCF protocol with bias $\varepsilon \approx 0$. The "winner" gets to flip a coin to determine the outcome of the SCF protocol. Of course, a dishonest player would like to "win" the WCF protocol since then they have total control of the SCF outcome.

We mimic this idea to create a protocol prototype for LT-Random-OT and discuss why it does not work.

Protocol 10 (A protocol prototype):

(i) Alice randomly chooses two bits (x_0, x_1) and Bob randomly chooses an index $b \in \{0, 1\}$.

(ii) Alice and Bob perform an LT-WCF protocol with bias ε_{WCF} to create random $c \in \{0, 1\}$.

(iii) If $c = 0$, then Bob sends b to Alice. Alice then replies with x_b.

(iv) If $c = 1$, then Alice sends (x_0, x_1) to Bob.

This protocol has bias $\varepsilon_{\text{ROT}} < 1/2$ if $\varepsilon_{\text{WCF}} < 1/2$. However, the problem is that honest Alice learns b with probability $3/4$ when Bob is honest. This is simply not allowed in a Random-OT protocol because honest Alice should never obtain any information about b. Honest Bob learns $x_0 \oplus x_1$ with probability $3/4$, which is also not allowed since he should only learn x_0 or x_1.

To remedy this problem, instead of Alice and Bob revealing their bits entirely, they can use (possibly unfair) LT-Random-OT protocols. We present a modified version of the protocol below.

Protocol 11 (An LT-Random-OT protocol):

(i) Alice and Bob perform an LT-WCF protocol with cheating probabilities $(A_{\text{WCF}}, B_{\text{WCF}})$ and bias ε_{WCF} to create random $c \in \{0, 1\}$.

(ii) If $c = 0$, then Alice and Bob generate their outputs using an LT-Random-OT protocol with cheating probabilities $(A_{\text{ROT}}, B_{\text{ROT}}) = (x, y)$, where $x \geq y$.

(iii) If $c = 1$, then Alice and Bob generate their outputs using an LT-Random-OT protocol with cheating probabilities $(A_{\text{ROT}}, B_{\text{ROT}}) = (y, x)$.

(iv) Alice and Bob abort if and only if either LT-Random-OT protocol is aborted.

We now prove that this LT-Random-OT protocol has cheating probabilities equal to those in Theorem 5. We show it for cheating Alice as the case for cheating Bob is almost identical. Since $x \geq y$, Alice would prefer the outcome of the WCF protocol to be $c = 0$. She can force $c = 0$ with probability A_{WCF} and in this case she can learn b with probability x. If $c = 1$, she can learn b with probability y. Letting A'_{ROT} be the amount she can learn b in the protocol above, we have

$$A'_{\text{ROT}} = A_{\text{WCF}}\, x + (1 - A_{\text{WCF}})\, y = A_{\text{WCF}}\, (x - y) + y. \tag{7}$$

All that remains to prove Theorem 5 is to show that an LT-Random-OT protocol with cheating probabilities $(A_{\text{ROT}}, B_{\text{ROT}}) = (\alpha, \beta)$ implies the existence of an LT-Random-OT protocol with cheating probabilities $(A_{\text{ROT}}, B_{\text{ROT}}) = (\beta, \alpha)$, for any $\alpha, \beta \in [1/2, 1]$. This way, we set $x = \max\{\alpha, \beta\}$ and $y = \min\{\alpha, \beta\}$.

3.3 Symmetry in **LT-Random-OT** *protocols*

Suppose we have an LT-Random-OT protocol and this protocol has cheating probabilities $(A_{\text{ROT}}, B_{\text{ROT}}) = (\alpha, \beta)$, for some $\alpha, \beta \in [1/2, 1]$. We now show how to create an LT-Random-OT protocol with cheating probabilities $(A_{\text{ROT}}, B_{\text{ROT}}) = (\beta, \alpha)$. The trick is to switch the roles of Alice and Bob.

Protocol 12 (Randomized version of a protocol in [28]):

(i) Alice and Bob use an LT-Random-OT protocol with cheating probabilities equal to $(A_{\text{ROT}}, B_{\text{ROT}}) = (\alpha, \beta)$ except that Bob is the sender and Alice is the receiver. Let Alice's output be (b, x_b) and let Bob's output be (x_0, x_1).

(ii) Alice randomly chooses $d \in \{0, 1\}$ and sends $d \oplus x_b$ to Bob.

(iii) Alice outputs $(x'_0, x'_1) = (d, d \oplus b)$ and Bob outputs $(b', m) = (x_0 \oplus x_1, d \oplus x_b \oplus x_0)$.

(iv) Alice and Bob abort if and only if the LT-Random-OT protocol is aborted.

Notice this protocol is loss-tolerant since classical messages can be resent if lost in transmission. We can write Bob's output m as $d \oplus x_b \oplus x_0 = d \oplus bb'$. Thus, if $b' = 0$ then $m = d = x'_0$ and if $b' = 1$ then $m = d \oplus b = x'_1$. Therefore, Bob gets the correct value for $x'_{b'}$. Since $x'_0 \oplus x'_1 = d \oplus (d \oplus b) = b$, honest Bob gets no information about Alice's other bit and cheating Bob can learn $x'_0 \oplus x'_1$ with maximum probability α. Since $b' = x_0 \oplus x_1$, honest Alice gets no information about b' and cheating Alice can learn b' with maximum probability β. Therefore, $(A_{\text{ROT}}, B_{\text{ROT}}) = (\beta, \alpha)$, as desired. Since b, x_0, x_1, and d are all randomly generated, so are x'_0, x'_1, and b' making this a valid LT-Random-OT protocol.

This completes the proof of Theorem 5.

3.4 An unfair **LT-Random-OT** *protocol*

We present here an LT-Random-OT protocol and show that it has cheating probabilities $(A_{\text{ROT}}, B_{\text{ROT}}) = (1/2, 1)$. Note that even though this protocol has bias $\varepsilon_{\text{ROT}} = 1/2$, it can be used to create a protocol with smaller bias using recent LT-WCF protocols and Theorem 5.

Protocol 13 (An unfair LT-Random-OT protocol):

(i) Bob randomly chooses an index $b \in \{0,1\}$ and another random bit $d \in \{0,1\}$.

(ii) Bob sends Alice the qubit $H^b|d\rangle$.

(iii) Alice randomly chooses $x_0, x_1 \in \{0,1\}$ and applies the unitary $X^{x_0}Z^{x_1}$ to the qubit.

(iv) Alice returns the qubit to Bob which is in the state $X^{x_0}Z^{x_1}H^b|d\rangle = H^b|x_b \oplus d\rangle$ (up to global phase).

(v) Bob has a two-outcome measurement (depending on b and d) to learn x_b perfectly.

(vi) If any messages are lost the protocol is restarted from the beginning.

We see that this is a valid Random-OT protocol. Firstly, because honest Bob learns x_b and gets no information about $x_{\bar{b}}$ (since $H^b|x_b \oplus d\rangle$ does not involve $x_{\bar{b}}$). Secondly, Alice cannot learn any information about b, even if she is dishonest, since the density matrices for $b = 0$ and $b = 1$ are identical. Therefore, $A_{\text{ROT}} = 1/2$. This protocol is loss-tolerant concerning cheating Alice since b and d are reset if any messages are lost so Alice cannot accumulate useful information. It is also loss-tolerant concerning cheating Bob since he can already learn both of Alice's bits perfectly. He can do this by first sending Alice half of

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle. \tag{8}$$

Each choice of (x_0, x_1) corresponds to Bob having a different Bell state at the end of the protocol. From this, x_0 and x_1 can be perfectly inferred, yielding $B_{\text{ROT}} = 1$.

4 Conclusions and Open Questions

We have designed a way to build LT-OT protocols by using an LT-WCF protocol to help balance the cheating probabilities in a (possibly unfair) LT-Random-OT protocol. This protocol uses well known reductions between OT and Random-OT and the reduction to switch the roles of Alice and Bob.

The construction in this paper is robust enough to design OT protocols with other definitions of cheating Bob. Suppose that Bob wishes to learn $f(x_0, x_1)$ where $f \neq \text{XOR}$ is some functionality. In this case, we may not be able to switch the roles of Alice and Bob in a way that switches the cheating probabilities as in Subsection 3.3. However, instead of just using one LT-Random-OT protocol and creating another from it, we could have just as easily used two different LT-Random-OT protocols (with a consistent notion of cheating Bob).

A limitation of this protocol design is that it uses LT-Random-OT protocols as subroutines. Even if LT-WCF protocols with bias $\varepsilon_{\text{WCF}} \approx 0$ are constructed, using the protocol in Subsection 3.4 can reduce the bias to only $\varepsilon_{\text{OT}} \approx 1/4$. It would be interesting to see if there exists an LT-OT protocol with cheating probabilities $(A_{\text{OT}}, B_{\text{OT}}) = (\alpha, \beta)$ where $\alpha + \beta < 3/2$.

An open question is to show if using more LT-WCF subroutines can help improve the bias. In [9], many WCF protocols were used to drive the bias of a SCF protocol down towards the optimal value of $1/\sqrt{2} - 1/2$. Can something similar be done for OT or LT-OT?

Acknowledgements

I would like to thank Ashwin Nayak and Levent Tunçel for helpful discussions. I acknowledge support from NSERC, MITACS, and ERA (Ontario).

References

1. D. Mayers, "Unconditional security in quantum cryptography," *Journal of the ACM*, vol. 48, no. 3, pp. 351–406, 2001.
2. H.-K. Lo and H. Chau, "Unconditional security of quantum key distribution over arbitrarily long distances," *SCIENCE*, vol. 283, pp. 2050–2056, 1999.
3. J. Preskill and P. Shor, "Simple proof of security of the BB84 quantum key distribution protocol," *Phys. Rev. Lett.*, vol. 85, no. 2, pp. 441–444, 2000.
4. D. Aharonov, A. Ta-Shma, U. Vazirani, and A. Yao, "Quantum bit escrow," in *Proceedings of 32nd Annual ACM Symposium on the Theory of Computing*, pp. 705–714, 2000.
5. A. Ambainis, "A new protocol and lower bounds for quantum coin flipping," in *Proceedings of 33rd Annual ACM Symposium on the Theory of Computing*, pp. 134 – 142, 2001.
6. H.-K. Lo and H. Chau, "Is quantum bit commitment really possible?," *Phys. Rev. Lett.*, vol. 78, no. 17, pp. 3410–3413, 1997.
7. A. Kitaev, "Quantum coin-flipping." Presentation at the 6th Workshop on Quantum Information Processing (QIP 2003), 2002.
8. G. Gutoski and J. Watrous, "Toward a general theory of quantum games," in *Proceedings of the 39th ACM Symposium on Theory of Computing*, pp. 565–574, 2007.
9. A. Chailloux and I. Kerenidis, "Optimal quantum strong coin flipping," in *Proceedings of 50th IEEE Symposium on Foundations of Computer Science*, pp. 527–533, 2009.
10. R. Spekkens and T. Rudolph, "Quantum protocol for cheat-sensitive weak coin flipping," *Phys. Rev. Lett.*, vol. 89, no. 22, pp. 1–4, 2002.
11. I. Kerenidis and A. Nayak, "Weak coin flipping with small bias," *Inf. Process. Lett.*, vol. 89, no. 3, pp. 131–135, 2004.
12. C. Mochon, "A large family of quantum weak coin-flipping protocols," *Phys. Rev. A*, vol. 72, no. 2, p. 022341, 2005.
13. C. Mochon, "Quantum weak coin flipping with arbitrarily small bias." Available as arXiv.org e-Print quant-ph/0711.4114, 2007.
14. J. Kilian, "Founding cryptography on oblivious transfer," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pp. 20–31, 1988.
15. S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Communications of the ACM*, vol. 28, no. 6, pp. 637–647, 1985.
16. C. Crépeau, "Equivalence between two flavours of oblivious transfers," in *Advances in Cryptology CRYPTO 1987*, pp. 350–354, 1987.
17. M. Rabin, "How to exchange secrets by oblivious transfer," in *Technical Report TR-81, Aiken Computation Laboratory, Harvard University*, 1981.
18. H.-K. Lo, "Insecurity of quantum secure computations," *Phys. Rev. A*, vol. 56, no. 2, pp. 1154–1162, 1997.
19. A. Chailloux, I. Kerenidis, and J. Sikora, "Lower bounds for quantum oblivious transfer," in *Proceedings of IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, vol. 8, pp. 157–168, 2010.
20. I. Damgård, S. Fehr, L. Salvail, and C. Schaffner, "Cryptography in the bounded quantum-storage model," *SIAM Journal of Computing*, vol. 37, no. 6, pp. 1865–1890, 2008.
21. C. Schaffner, "Simple protocols for oblivious transfer and secure identification in the noisy-quantum-storage model," *Phys. Rev. A*, vol. 82, p. 032308, 2010.
22. R. Jain, J. Radhakrishnan, and P. Sen, "A new information-theoretic property about quantum states with an application to privacy in quantum communication," *Journal of ACM*, vol. 56, no. 6,

Article no. 33, 2009.

23. M. Jakobi, C. Simon, N. Gisin, J.-D. Bancal, C. Branciard, N. Walenta, and H. Zbinden, "Practical private database queries based on a quantum key distribution protocol," *Phys. Rev. A*, vol. 83, p. 022301, 2011.
24. G. Berlin, G. Brassard, F. Bussieres, and N. Godbout, "Loss-tolerant quantum coin flipping," in *Proceedings of the Second International Conference on Quantum, Nano and Micro Technologies (ICQNM 2008)*, pp. 1–9, 2008.
25. N. Aharon, S. Massar, and J. Silman, "A family of loss-tolerant quantum coin flipping protocols," *Phys. Rev. A*, vol. 82, p. 052307, 2010.
26. A. Chailloux, "Improved loss tolerant quantum coin flipping," in *AQIS '10: Asian Conference on Quantum Information*, 2010.
27. J.-J. Ma, F.-Z. Guo, Q. Yang, Y.-B. Li, and Q.-Y. Wen, "Semi-loss-tolerant strong coin flipping protocol using EPR pairs," *Journal of Quantum Information and Computation*, vol. 12, no. 5 & 6, pp. 489–500, 2012.
28. S. Wolf and J. Wullschleger, "Oblivious transfer is symmetric," in *Advances in Cryptology - EUROCRYPT '06*, pp. 222–232, 2006.