Out of the Box | Weak Coin Flipping and friends

23rd March 2021

Abstract

We report a device independent weak coin flipping protocol[1] with $P_A^* \leq \cos^2(\pi/8)$ and $P_B^* \leq 0.667...$, by making seemingly minor changes to the best known protocol due to SCAKPM'11 [10.1103/PhysRevLett.106.220501], with $P_A^* \leq \cos^2(\pi/8) \approx 0.85$ and $P_B^* \leq 3/4 = 0.75$. In terms of bias, we improve the SCAKPM'11 result from ≈ 0.336 to ≈ 0.3199. This improvement is due to two ingredients: a self-testing (of GHZ) step and an extra cheat detection step for Bob. We also introduce a new bias suppression technique that ekes out further security from the abort probability to obtain ... Note that the SCAKPM'11 result held for both strong and weak coin flipping; ours holds only for the latter. TODO: Fix me!

Contents

1 Introduction **2**
- 1.1 Proof Technique . . . 2

2 An illustration of a Device Dependent Weak Coin Flipping protocol **4**

3 Device Independent Weak Coin Flipping protocols | State Of The Art **4**
- 3.1 Device Independence and the Box Paradigm . . . 4
- 3.2 The GHZ Test . . . 6
- 3.3 The Protocol . . . 6

4 First Technique: Self-testing and its limitations **6**
- 4.1 Idea . . . 6
  - 4.1.1 Protocols (single shot, unbalanced) . . . 7

5 Second Technique: Suppression Technique **8**
- 5.1 Idea . . . 9
- 5.2 Limitation . . . 9
- 5.3 Protocol . . . 9

6 Beyond WCF—SCF and OT **9**
- 6.1 SCF . . . 9
- 6.2 OT . . . 9

7 Background (self testing bounds) **9**

8 Security Proofs using Semidefinite Programming **9**
- 8.1 SDP when Alice self-tests . . . 9
- 8.2 SDP when Bob self-tests . . . 9

9 Jamie's **11**

[1] which are analysed

1 Introduction

INTERNAL/Atul: Colour coding—Purple is for informal discussions, black is for formal statements and blue is for proofs. We can remove these from the final version; I put it to minimise verbiage.

Secure two-party computation is a cryptographic setting where two parties, conventionally called Alice and Bob, receive inputs x and y and their goal is to compute some function $f_A(x, y)$ and $f_B(x, y)$ respectively which depends on both their inputs. However, they do not wish to reveal their inputs. Coin flipping (CF) is a cryptographic primitive in this setting, i.e. a building block for constructing more applicable secure two-party cryptographic schemes, where Alice and Bob wish to exchange messages and agree on a random bit, without trusting each other. A protocol that implements coin flipping must protect an honest player from a malicious[2] player.

A weaker primitive, unsurprisingly, known as *weak coin flipping* (WCF) is where a zero corresponds to Alice winning and one corresponds to Bob winning. It is weaker because now the protocol has to protect Alice from a malicious Bob who tries to bias the outcome towards one (and not towards zero) and conversely, it must protect Bob from a malicious Alice who tries to bias the outcome towards zero (and not towards one). To emphasise the distinction, the former primitive is often termed *strong coin flipping* (SCF).

We primarily focus on WCF in this article and begin with introducing some notation. We denote by P_A^* the highest probability of a malicious Alice convincing an honest Bob that she won (i.e. in the WCF protocol, Alice uses her best cheating strategy against Bob who in turn is following the protocol as described, to convince him that the outcome is zero). Analogously, P_B^* is the highest probability of a malicious Bob convincing an honest Alice that he won. The bias of a WCF protocol is defined as $\epsilon := \max\left\{P_A^*, P_B^*\right\} - \frac{1}{2}$. A protocol that is completely secure, has $\epsilon = 0$ and one that is completely insecure has $\epsilon = \frac{1}{2}$.

Using a classical channel of communication between Alice and Bob, unless one makes further assumptions such as computational hardness of certain problems or relativistic assumptions,[3] coin flipping (even weak) is impossible to implement with any security, to wit: for all classical protocols at least one of the parties, viz. a malicious Alice or a malicious Bob, can win with certainty because one can show $\epsilon = \frac{1}{2}$ (viz. $\max\{P_A^*, P_B^*\} = 1$). Using a quantum channel of communication, it was shown that WCF can be implemented with vanishing bias. These works, however, do not account for noise in their implementation. One path towards more robust security is device independence wherein the players do not even trust their devices (recall, they already do not trust the other party). This is in contrast to the device independent setting considered in key distribution where the two parties trust each other but neither their devices nor the communication channel (TODO: is the classical communication channel trusted?).

In this work, we start with a device independent (DI) coin flipping (CF) protocol introduced[4] in [SCA+11] which has $P_A^* = \cos^2(\pi/8) \approx 0.854$ and $P_B^* = 3/4 = 0.75$. They then compose these protocols to give a balanced protocol, i.e. with $P_A^* = P_B^* \approx \frac{1}{2} + 0.336$. To the best of our knowledge, this DI CF protocol has the best security guarantee. While Kitaev's bound for CF rules out perfect DI CF, no lower bounds on the bias are known for DI WCF. In this work, however, we focus on improving the upper bound on the bias, viz. we give DI WCF protocols with biases ≈ 0.319.

We introduce two key new ideas which result in better protocols. The first, is the use of self-testing by one party before initiating the protocol and the second, is a more general technique to convert unbalanced protocols (i.e. ones in which the probability of maliciously winning for Alice and Bob are unequal) into balanced ones.

1.1 Proof Technique

Notation and Cheat Vectors

We introduce some notation to facilitate the discussion here. Denote the DI CF protocol introduced in [SCA+11] by $\mathcal{I}$ and let $p_A^*(\mathcal{I}) \approx 0.853\ldots$ denote the maximum probability with which a malicious Alice can win against honest Bob who is following the protocol $\mathcal{I}$ and similarly, let $p_B^*(\mathcal{I}) \approx 0.75$ denote the maximum probability with which a malicious Bob can win against an honest Alice who is following the protocol $\mathcal{I}$.

One of the key observations we make in this work is the use of what we call "cheat vectors"—it is any tuple of probabilities which can arise in a CF protocol when one player is honest. More precisely, suppose Alice is (possibly)

[2] (or cheating, we use these adjectives interchangeably)

[3] in terms of the spatial locations of the observers; not to be confused with the term *relativising* from computational complexity.

[4] In fact, they introduced a device independent bit commitment protocol which they in turn use to construct a strong coin flipping protocol with the same cheating probabilities for Alice and Bob, ≈ 0.854 and 0.75 respectively.

malicious and Bob follows the protocol $\mathcal{I}$. Then, the cheat vectors for Alice constitute the set

$$\mathbb{C}_A(\mathcal{I}) := \{(\alpha, \beta, \gamma) : \exists \text{ a strategy for } A \text{ s.t. an honest } B \text{ outputs 1,0, and } \perp \text{ with probabilities } \alpha, \beta \text{ and } \gamma\}.$$

We analogously define $\mathbb{C}_B(\mathcal{I})$. Cheat vectors become useful when we try to compose protocols. The observation then, is that the abort event can be taken to abort the full protocol instead of being treated as the honest player winning. The latter gives the malicious player further opportunity to cheat and so preventing it improves the security.

Protocols

We introduce two variants of protocol $\mathcal{I}$, which we call $\mathcal{P}$ and $\mathcal{Q}$.

- $\mathcal{P}$ is essentially the same as $\mathcal{I}$ except that Alice self-tests her boxes before starting the protocol and performs an additional test to ensure Bob doesn't cheat. We show that $p_A^*(\mathcal{P}) \lesssim 0.853\ldots$ and $p_B^*(\mathcal{P}) \lesssim 0.667\ldots$. We also show that $\mathbb{C}_B(\mathcal{P})$ can be cast as an SDP.
- $\mathcal{Q}$ is also essentially the same as $\mathcal{I}$ except that Bob self-tests his boxes before starting the protocol. In this case, $p_X^*(\mathcal{Q}) = p_X^*(\mathcal{I})$ for both values of $X \in \{A, B\}$ so the advantage isn't manifest. However, now $\mathbb{C}_A(\mathcal{Q})$ can be cast as an SDP which, as we shall see, yields an advantage when $\mathcal{Q}$ is composed.

Compositions

aoeu

As the protocols $\mathcal{X} \in \{\mathcal{I}, \mathcal{P}, \mathcal{Q}\}$ all have skewed security—either $p_A^*(\mathcal{X}) > p_B^*(\mathcal{X})$ or the other way—and therefore the bias is determined by $p_{\max}^*(\mathcal{X}) := \max\{p_A^*(\mathcal{X}), p_B^*(\mathcal{X})\}$. Note that, $p_{\max}^*(\mathcal{X}) = p_{\max}^*(\mathcal{Y})$ for all $\mathcal{X}, \mathcal{Y} \in \{\mathcal{I}, \mathcal{P}, \mathcal{Q}\}$, which means that we don't immediately obtain an advantage. However, the most obvious method of composing these protocols to obtain a new protocol, which we describe later, "balances" the advantage. After this composition procedure is applied to some protocol $\mathcal{X}$, we denote the resulting protocol by $C_{\text{stand}}(\mathcal{X})$. Applying this technique to $\mathcal{P}$, we already obtain a more secure protocol.

- For all $X \in \{A, B\}$ the cheating probabilities for protocol $\mathcal{I}$ under the standard composition is given by

  $$p_X^*(C_{\text{stand}}(\mathcal{I})) \approx \frac{1}{2} + 0.336\ldots$$

  while for the improved protocol $\mathcal{P}$, these are given by

  $$p_X^*(C_{\text{Sikora}}(\mathcal{P})) \approx \frac{1}{2} + 0.3199\ldots. \tag{1}$$

The standard composition technique doesn't yield any improvement for $\mathcal{Q}$ because the cheating probabilities are identical to those of $\mathcal{I}$. We can extract an advantage by using a composition technique that uses "cheat vectors" and the abort event. We describe it in detail later but for now, we simply denote the new protocol obtained using this improved composition (of protocol $\mathcal{X}$) by $C_{\text{Sikora}}(\mathcal{X})$.

- Using this technique on $\mathcal{Q}$, the cheating probabilities become

  $$p_X^*(C_{\text{Sikora}}(\mathcal{Q})) \approx \frac{1}{2} + 0.317\ldots$$

  for all $X \in \{A, B\}$, which is even better than Equation (1).

- Finally, we combine both these protocols to obtain (again, for all $X \in \{A, B\}$)

  $$p_X^*(C_{\text{Sikora}}(\mathcal{Q}, \mathcal{Q}, \ldots, \mathcal{Q}, \mathcal{P})) \approx \frac{1}{2} + 0.2908\ldots$$

  where we use the same composition technique except that at the last "level" we use $\mathcal{P}$ instead of $\mathcal{Q}$.

TODO: Obvious questions (answers to which I don't have anymore; stupid memory): what about $p_X^*(C_{\text{Sikora}}(\mathcal{P}))$ and $p_X^*(C_{\text{Sikora}}(\mathcal{P}, \mathcal{P}, \ldots, \mathcal{P}, \mathcal{Q}))$.

aoeu

2 An illustration of a Device Dependent Weak Coin Flipping protocol

As an illustration of an interesting WCF protocol, we present the one due to [SR]. For $|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \left(1 - \frac{1}{\sqrt{2}}\right)|11\rangle$ and $E_0 = \frac{1}{\sqrt{2}}|0\rangle\langle 0|$, the following yields $P_A^* = P_B^* = \frac{1}{\sqrt{2}}$.

The protocol:

Round 1. Alice prepares a pair of systems in a (typically entangled) state $|\psi\rangle \in \mathcal{H}^A \otimes \mathcal{H}^B$, and sends system B to Bob.

Round 2. Bob performs the measurement associated with the positive operator-valued measure (POVM) $\{E_0, E_1\}$ on system B, and sends a classical bit b indicating the result to Alice.

Round 3. If $b = 0$ then Bob sends system B back to Alice, while if $b = 1$ then Alice sends system A to Bob. The party that receives the system then performs the measurement associated with the projection valued measure $\{|\psi_b\rangle\langle\psi_b|, I - |\psi_b\rangle\langle\psi_b|\}$, where $|\psi_b\rangle = I \otimes \sqrt{E_b}|\psi\rangle / \sqrt{\langle\psi| I \otimes E_b |\psi\rangle}$. The different possible outcomes are:
(i) $b = 0$, Alice finds $|\psi_0\rangle\langle\psi_0|$; Bob wins.
(ii) $b = 0$, Alice finds $I - |\psi_0\rangle\langle\psi_0|$; Alice catches Bob cheating.
(iii) $b = 1$, Bob finds $|\psi_1\rangle\langle\psi_1|$; Alice wins.
(iv) $b = 1$, Bob finds $I - |\psi_1\rangle\langle\psi_1|$; Bob catches Alice cheating.

3 Device Independent Weak Coin Flipping protocols | State Of The Art

In the following, we first discuss how one can describe DI WCF protocols in terms of the players exchanging "boxes"—devices which take classical inputs and give classical outputs. Subsequently we recall the GHZ test and finally we use these to delineate the DI-CF due to [SCA+11].

3.1 Device Independence and the Box Paradigm

We describe device independent protocols as classical protocols with the one modification: we assume that the two parties can exchange boxes and that the parties can shield their boxes (from the other boxes i.e. the boxes can't communicate with each other once shielded).[5]

Definition 1 (Box)**.** A *box* is a device that takes an input $x \in \mathcal{X}$ and yields an outputs $a \in \mathcal{A}$ where $\mathcal{X}$ and $\mathcal{A}$ are finite sets. Typically, a set of n boxes, taking inputs $x_1, x_2, \ldots x_n$ and yielding outputs $a_1, a_2 \ldots a_n$ are *characterised* by a joint conditional probability distribution, denoted by

$$p(a_1, a_2 \ldots a_n | x_1, x_2 \ldots x_n).$$

Further, if $p(a_1, a_2 \ldots a_n | x_1, x_2 \ldots x_n) = \mathrm{tr}\left[M^1_{a_1|x_1} \otimes M^2_{a_2|x_2} \cdots \otimes M^n_{a_n|x_n} \rho\right]$ then we call the set of boxes, *quantum boxes*, where $\{M^i_{a'|x'}\}_{a' \in \mathcal{A}_i}$ constitute a POVM for a fixed i and x', ρ is a density matrix and their dimensions are mutually consistent.

Henceforth, we restrict ourselves to quantum boxes.

[5] TODO: Verify if this notion is in fact correct; I hope I'm not making a major mistake somehow. I should be able to take the POVMs as tensor products right, because I can change them at will, independent of the others (and ensuring that there's no communication between them; could they be somehow entangled, i.e. could it be that somehow the measurement operators are themselves quantum correlated?); I would like to reach the conclusion starting from the locality assumption.

Definition 2 (Protocol in the box formalism)**.** A generic two-party protocol in the box formalism has the following form:

1. Inputs:
   (a) Alice is given boxes $\square_1^A, \square_2^A \ldots \square_p^A$ and Bob is given boxes $\square_1^B, \square_2^B, \ldots \square_q^B$.
   (b) Alice is given a random string r^A and Bob is given a random string r^B (of arbitrary but finite length).
2. Structure: At each round of the protocol, the following is allowed.
   (a) Alice and Bob can locally perform arbitrary but finite time computations on a Turing Machine.
   (b) They can exchange classical strings/messages and boxes.

A protocol in the box formalism is readily expressed as a protocol which uses a (trusted) classical channel (i.e. they trust their classical devices to reliably send/receive messages), untrusted quantum devices and an untrusted quantum channel (i.e. a channel that can carry quantum states but may be controlled by the adversary).

Assumption 3 (Setup of Device Independent Two-Party Protocols)**.** *Alice and Bob*

1. *both have private sources of randomness,*
2. *can send and receive classical messages over a (trusted) classical channel,*
3. *can prevent parts of their untrusted quantum devices from communicating with each other, and*
4. *have access to an untrusted quantum channel.*

We restrict ourselves to a "measure and exchange" class of protocols—protocols where Alice and Bob start with some pre-prepared states and subsequently, only perform classical computation and quantum measurements locally in conjunction with exchanging classical and quantum messages. More precisely, we consider the following (likely restricted) class of device independent protocols.

Definition 4 (Measure and Exchange (Device Independent Two-Party) Protocols)**.** A *measure and exchange (device independent two-party) protocol* has the following form:

1. Inputs:
   (a) Alice is given quantum registers $A_1, A_2, \ldots A_p$ together with POVMs[6]
   $$\{M_{a|x}^{A_1}\}_a, \{M_{a|x}^{A_2}\}_a, \ldots \{M_{a|x}^{A_p}\}_a$$
   which act on them and Bob is, analogously, given quantum registers $B_1, B_2, \ldots B_q$ together with POVMs
   $$\{M_{b|y}^{B_1}\}_b, \{M_{b|y}^{B_2}\}_b, \ldots, \{M_{b|y}^{B_q}\}_b.$$
   Alice shields $A_1, A_2, \ldots A_p$ (and the POVMs) from each other and from Bob's lab. Bob similarly shields $B_1, B_2 \ldots B_q$ (and the POVMs) from each other and from Alice's lab.
   (b) Alice is given a random string r^A and Bob is given a random string r^B (of arbitrary but finite length).
2. Structure: At each round of the protocol, the following is allowed.
   (a) Alice and Bob can locally perform arbitrary but finite time computations on a Turing Machine.
   (b) They can exchange classical strings/messages.
   (c) Alice (for instance) can
      i. send a register A_l and the encoding of her POVMs $\{M_i^{A_l}\}_i$ to Bob, or
      ii. receive a register B_m and the encoding of the POVMs $\{M_i^{B_m}\}_i$.

   Analogously for Bob.

It is clear that a protocol in the box formalism (Definition 2) which uses only quantum boxes (Definition 1) can be implemented as a measure and exchange protocol (Definition 4).

[6]For concreteness, take the case of binary measurements. By $\{M_{a|x}^{A_1}\}_a$, for instance, we mean $\{M_{0|x}^{A_1}, M_{1|x}^{A_1}\}$ is a POVM for $x \in \{0, 1\}$.

3.2 The GHZ Test

Before we define the current best DI CF protocol, we briefly remind the reader of the GHZ test, upon which the aforementioned protocol depends, and set up some conventions.

Definition 5. Suppose we are given three boxes, $\square^A, \square^B$ and $\square^C$, which accept binary inputs $a, b, c \in \{0,1\}$ and produces binary output $x, y, z \in \{0,1\}$ respectively. The boxes pass the GHZ test if $a \oplus b \oplus c = xyz \oplus 1$, given the inputs satisfy $x \oplus y \oplus z = 1$.

Claim 6. Quantum boxes pass the GHZ test with certainty (even if they cannot communicate), for the state $|\psi\rangle_{ABC} = \frac{|000\rangle_{ABC}+|111\rangle_{ABC}}{\sqrt{2}}$, and measurement[7] $\frac{\sigma_x+\mathbb{I}}{2}$ for input 0 and $\frac{\sigma_y+\mathbb{I}}{2}$ for input 1 (in the notation introduced earlier, $M^A_{0|0} = |+\rangle\langle+|, M^A_{1|0} = |-\rangle\langle-|$ and so on, where $|\pm\rangle = \frac{|0\rangle\pm|1\rangle}{\sqrt{2}}$).[8]

The proof is easier to see in the case where the outcomes are ± 1; it follows from the observations that $\sigma_y \otimes \sigma_y \otimes \sigma_y |\psi\rangle = -|\psi\rangle$, $\sigma_x \otimes \sigma_x \otimes \sigma_x |\psi\rangle = |\psi\rangle$ and the anti-commutation of σ_x and σ_y matrices, i.e. $\sigma_x\sigma_y + \sigma_y\sigma_x = 0$.

3.3 The Protocol

The best DI CF protocol known is the one introduced in [SCA+11]. While this is a protocol for SCF, and so also works as a WCF protocol, we do not know of any better protocol for the latter.

Algorithm 7 (SCF, original)**.** *Alice has one box and Bob has two boxes (in the security analysis, we let the cheating player distribute the boxes). Each box takes one binary input and gives one binary output.*

1. *Alice chooses $x \in_R \{0,1\}$ and inputs it into her box to obtain a. She chooses $r \in_R \{0,1\}$ to compute $s = a \oplus x.r$ and sends s to Bob.*
2. *Bob chooses $g \in_R \{0,1\}$ (for "guess") and sends it to Alice.*
3. *Alice sends x and a to Bob. They both compute the output $x \oplus g$.*
4. *Test round*
   (a) *Bob tests if $s = a$ or $s = a \oplus x$. If the test fails, he aborts. Bob chooses $b, c \in_R \{0,1\}$ such that $a \oplus b \oplus c = 1$ and then performs a GHZ using a, b, c as the inputs and x, y, z as the output from the three boxes. He aborts if this test fails.*

From **??**, it is clear that when both players follow Algorithm 7, then Bob never aborts and they win with equal probabilities. The security of the protocol is summarised next.

Lemma 8 (Security of SCF)**.** *[SCA+11] For Algorithm 7, $P^*_B \leq \frac{3}{4}$ and $P^*_A \leq \cos^2(\pi/8)$. Further, both bounds are saturated by a quantum strategy which uses a GHZ state and the honest player measures along the σ_x/σ_y basis corresponding to input $0/1$ into the box. Cheating Alice measures along $\sigma_{\hat{n}}$ for $\hat{n} = \frac{1}{\sqrt{2}}(\hat{x} + \hat{y})$ while cheating Bob measures his first box along σ_x and second along σ_y.*

Note that both players can cheat maximally assuming they share a GHZ state and the honest player measures along the associated basis. This entails that even though the cheating player could potentially tamper with the boxes before handing them to the honest player, surprisingly, exploiting this freedom does not offer any advantage to the cheating player.

4 First Technique: Self-testing and its limitations

4.1 Idea

We make two observations.

[7] we added the identity so that the eigenvalues associated become $0, 1$ instead of $-1, 1$.

[8] TODO: Think: Should I add the classical value? This would require me to add what it means to have a classical box.

First, in Algorithm 7 only Bob performs the test round. In WCF, there is a notion of Alice winning and Bob winning. Thus, if $x \oplus g = 0$, i.e. the outcome corresponding to "Alice wins", we can imagine that Bob continues to perform the test to ensure (at least to some extend) that Alice did not cheat. However, if $x \oplus g = 1$, i.e. the outcome corresponding to "Bob wins", we can require Alice to now complete the GHZ test to ensure that Bob did not cheat. It turns out that this does not lower P_B^*. Interestingly, the best cheating strategy deviates from the GHZ state and measurements for the honest player. We omit the details here (see TODO: write this down somewhere) but mention this to motivate the following.

Second, Alice (say) can harness the self-testing property of GHZ states and measurements to ensure that Bob has not tampered with her boxes. One way of proceeding is that N copies of the supposedly correct boxes are distributed. Alice now picks one out of these N boxes at random and asks Bob to send the associated two boxes to each $N-1$ box that Alice posses. Alice runs the GHZ test on each box and if even one test fails, she declares that Bob cheated. This way, for a large N, Alice can ensure with near certainty, that she has a box containing the correct state and (which performs the correct) measurements. Note that no such scheme can be concocted which simultaneously self-tests Alice and Bob's boxes. More precisely, no such procedure can ensure that Alice and Bob share a GHZ state (Alice one part, Bob the other two, for instance) because this would mean perfect (or near perfect) SCF is possible which is forbidden even in the device dependent case. Kitaev showed that for any SCF protocol, $\epsilon \geq \frac{1}{\sqrt{2}} - \frac{1}{2}$.

Combining these two observations, results in an improvement in the security for Alice. We obtain a protocol with $P_A^* \leq 3/4$, which is the same as before, but $P_B^* \lesssim 0.667....$

4.1.1 Protocols (single shot, unbalanced)

In the honest implementation, the *trio* of boxes used in the following are characterised by the GHZ setup (see **??**).

Algorithm 9 (Alice self-tests her boxes). *There are N trios of boxes; Alice has the first part and Bob has the remaining two parts, of each trio.*

1. *Alice selects a number $i \in_R \{1, 2 \dots N\}$ and sends it to Bob.*
2. *Bob sends his part of the trio of boxes corresponding to $\{1, 2 \dots N\} \backslash i$, i.e. he sends all the boxes, except the ones corresponding to the trio i.*
3. *Alice performs a GHZ test on all the trios labelled $\{1, 2 \dots N\} \backslash i$, i.e. all the trios except the ith.*

We restrict ourselves to the ith trio. Alice has one box and Bob has two boxes. Each box takes one binary input and gives one binary output.

1. *Alice chooses $x \in_R \{0, 1\}$ and inputs it into her box to obtain a. She chooses $r \in_R \{0, 1\}$ to compute $s = a \oplus x.r$ and sends s to Bob.*
2. *Bob chooses $g \in_R \{0, 1\}$ (for "guess") and sends it to Alice.*
3. *Alice sends x [EDIT: maybe not send a] and a to Bob. They both compute the output $x \oplus g$.*
4. *Test rounds:*
   (a) *If $x \oplus g = 0$:*
   [EDIT: Send a]
   Bob tests if $s = a$ or $s = a \oplus x$. If the test fails, he aborts. Bob chooses $b, c \in_R \{0, 1\}$ such that $a \oplus b \oplus c = 1$ and then performs a GHZ using a, b, c as the inputs and x, y, z as the output from the three boxes. He aborts if this test fails.
   (b) *Else, if $x \oplus g = 1$:*
      i. *Alice chooses $y, z \in_R \{0, 1\}$ s.t. $x \oplus y \oplus z = 1$ and sends them to Bob.*
      ii. *Bob inputs y, z into his boxes, obtains and sends b, c to Alice.*

   Alice tests if x, y, z as inputs and a, b, c as outputs, satisfy the GHZ test. She aborts if this test fails.

Lemma 10. *For Algorithm 9, Alice's cheating probability $P_A^* \lesssim \cos^2(\pi/8) \approx 0.852$ and Bob's cheating probability $P_B^* \lesssim 0.667....$*

Proof. The analysis for cheating Alice is the same as the original protocol. We consider the analysis for cheating Bob. We call the tuple (α, β, γ) a *cheat vector* if for some cheating strategy of Bob,

$$\alpha = \Pr[\text{Alice outputs } 0]$$
$$\beta = \Pr[\text{Alice outputs } 1]$$
$$\gamma = \Pr[\text{Alice aborts}].$$

We model Alice's steps as follows: We use

- $|+\rangle_x$ for her choice of input x,
- $|+\rangle_r$ for her choice of r,
- $\frac{1}{2}\mathbb{I}_H$ for her part of the GHZ state
- $|0\rangle_s$ a qubit to be used to send s to Bob
- $|0\rangle_a$ the outcome of Alice's GHZ state measurement

The argument proceeds as follows. Let ρ_0 be Alice's state after measuring to get and creating s (TODO: complete this part; likely in the code already).

1. Observe that $\mathrm{tr}_s(\rho_0)$ can be used to denote Alice's state after sending s to Bob. This is because Bob can hold a purification of $\mathrm{tr}_s(\rho_0)$ thereby holding s.
2. Similarly, observe that Bob sending a the variable g can be modelled as $\mathrm{tr}_g(\rho_1) = \mathrm{tr}_s(\rho_0)$.
3. While on the third step, Alice sends x to Bob, we need not formally include this because if Alice is testing Bob, he knows $x \oplus g = 1$.
4. Test rounds
   (a) $x \oplus g = 0$ is the case where Alice outputs 0; this means Bob already failed; not the event we are interested in.
   (b) $x \oplus g = 1$ is the case where Alice tests Bob.
      i. While Alice sends y and z to Bob, z can be determined from the condition $x \oplus y \oplus z = 1$ and the values of x and y. Therefore, formally, it suffices that Alice appends $\frac{1}{2}\mathbb{I}_y$ to ρ_1 to account for this step.
      ii. Bob is supposed to respond with b, c but formally, it suffices to send $d = b \oplus c$ because the GHZ condition only requires Alice to test $a \oplus b \oplus c = xyz \oplus 1$.

The SDP is therefore
Note: □

We defer the proof to 8.1. The analysis for a cheating Bob, P_B^*, is easily cast as an SDP thanks to the self-testing step. The analysis for cheating Alice, P_A^*, is the same as the one for the original protocol. Using a standard composition technique, which we outline in the following section, the cheating probabilities can be balanced. The key idea is to repeat the unbalanced protocol and appropriately alternate between giving the advantage to Alice and Bob. The resulting bias is 0.319....

5 Second Technique: Suppression Technique

Consider a WCF protocol, $\mathcal{P}_A$, where the success probability of cheating Alice is greater than that of cheating Bob. We denote this by

$$p_A^*(\mathcal{P}_A) = \alpha > \beta = p_B^*(\mathcal{P}_A).$$

The subscript A in $\mathcal{P}_A$ simply represents that Alice has the advantage. By $\mathcal{P}_B$ we denote the same protocol is run with Alice's and Bob's roles swapped. Thus,

$$p_A^*(\mathcal{P}_B) = \beta < \alpha = p_B^*(\mathcal{P}_B).$$

One can compose these protocols, into a new protocol $\mathcal{P}'_A$ as follows (see Figure 1): Alice and Bob run $\mathcal{P}_A$ (to decide whether they run $\mathcal{P}_A$ in the subsequent round or $\mathcal{P}_B$). If the outcome is A, they run $\mathcal{P}_A$ and if the outcome is B, they run $\mathcal{P}_B$. The outcome of this round, determines the outcome of the full protocol. It is easy to see that

$$p_A^*(\mathcal{P}'_A) = \alpha^2 + (1-\alpha)\beta =: \alpha^{(1)}, \quad \text{and}$$
$$p_B^*(\mathcal{P}'_A) = \beta\alpha + (1-\beta)\beta =: \beta^{(1)}.$$

One can repeat this procedure with $\mathcal{P}'_A$ and $\mathcal{P}'_B$ (which is defined by switching the roles of Alice and Bob in $\mathcal{P}'_A$) to obtain $\mathcal{P}''_A$ and $\mathcal{P}''_B$, together with $\alpha^{(2)}$ and $\beta^{(2)}$. More explicitly, one can write,

$$\alpha^{(i+1)} = (\alpha^{(i)})^2 + (1-\alpha^{(i)})\beta^{(i)}$$
$$\beta^{(i+1)} = \beta^{(i)}\alpha^{(i)} + (1-\beta^{(i)})\beta^{(i)}$$

Applying this to $\alpha^{(0)} = \cos^2(\pi/8) \approx 0.852$ and $\beta^{(0)} \lesssim 0.667$, we obtain

$$\lim_{n\to\infty} \alpha^{(n)} = \lim_{n\to\infty} \beta^{(n)} \approx 0.8199\ldots$$

which yields the biases stated in the previous subsection.

5.1 Idea

A key observation here is that here, we treated the probability of aborting as the honest player winning. We did this, for instance when Alice cheats, by assuming that the probability of Bob winning is $1 - p_A^*$. However, this probability actually constitutes two events: first, that Bob deduces he wins and second, that Bob detects Alice cheating (for instance, when the GHZ test fails). In usual treatments, we simply assume that if Bob detects Alice cheating, he declares himself the winner. However, when we compose protocols, instead of allowing

5.2 Limitation

5.3 Protocol

6 Beyond WCF—SCF and OT

6.1 SCF

6.2 OT

7 Background (self testing bounds)

8 Security Proofs using Semidefinite Programming

8.1 SDP when Alice self-tests

8.2 SDP when Bob self-tests

References

[SCA+11] J. Silman, A. Chailloux, N. Aharon, I. Kerenidis, S. Pironio, and S. Massar, *Fully distrustful quantum bit commitment and coin flipping*, Physical Review Letters **106** (2011), no. 22.

[SR] R. W. Spekkens and Terry Rudolph, *A quantum protocol for cheat-sensitive weak coin flipping*.

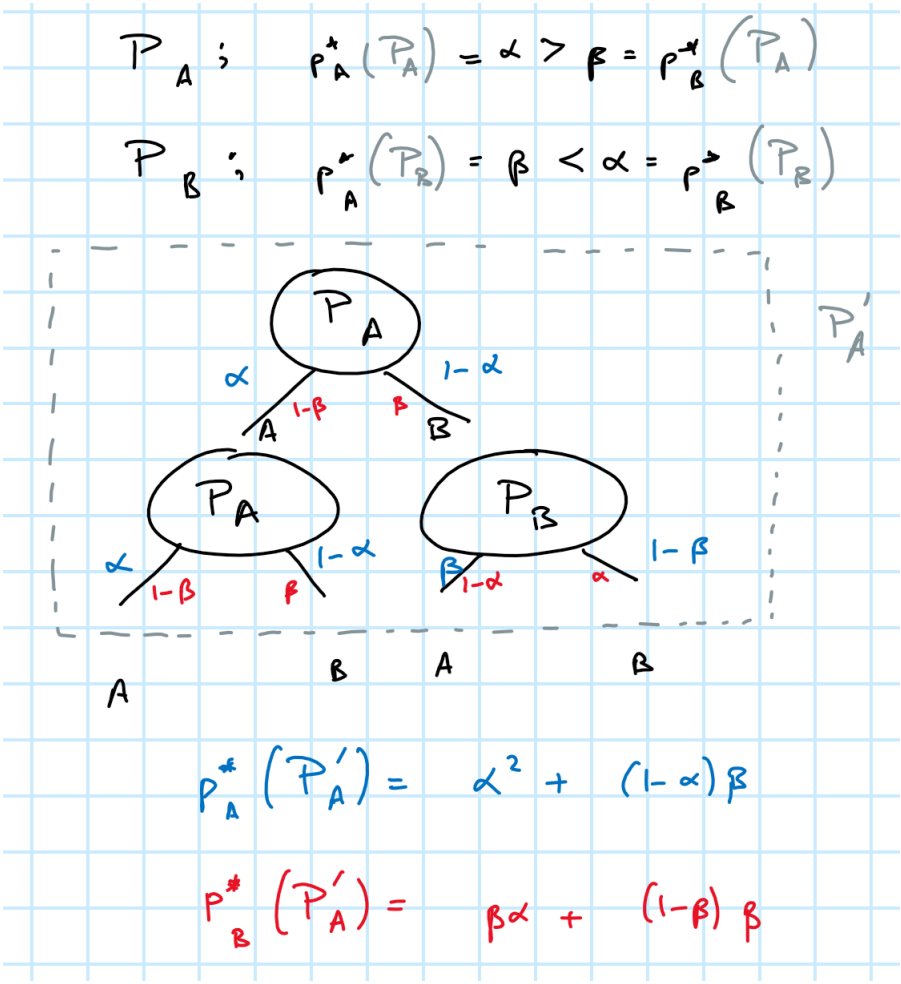

Figure 1: Standard composition technique. (TODO: improve the caption)

9 Jamie's

Section

1) Introduction

1.1) Intro, blah blah.

1.2) WCF protocol with no DI security. Spekkens-Rudolph?

1.3) 2011 protocol, why it is DI (idea) and its bias.

1.4) New idea: self-testing. Also, no self-testing Alice AND Bob (due to Kitaev).

1.5) Our NEW protocol(s). Self-test Alice, Self-test Bob (these are two protocols). Their DD security, and why DD security equals DI security (approximately, conceptually).

1.6) Suppression technique. Final bias.

1.7) Applications: WCF new bias. SCF new bias. OT new bias.

Section

2) Background

2.1) Self-testing bounds.

Section 3) Semidefinite programming

3.1) SDP for self-testing Alice. SDP for self-testing Bob. (single-shot cases). Cheat vector graphs here as well.

3.2) SDP for suppression technique for WCF, SCF.

3.3) SDP continuity bounds (I will do this).

Section 4) Conclusions, and why we're awesome.

--

Here is a crappy DI-OT protocol.

Fix two OT protocols, one with P_A,OT* = 1, P_B,OT* = 1/2 and the other with P_A,OT* = 1/2, P_B,OT* = 1.

DI-OT protocol:

Step 0) Do DI-WCF from our paper. Step 1) If no abort, and Alice wins, they use the P_A,OT* = 1 protocol. Step 2) If no abort, and Alice loses, they use the P_A,OT* = 1/2 protocol.

This yields a bias of < 1/2. Not pretty, but it works. There is likely a way to make this better, but it's a first step.