Improved Device Independent Weak Coin Flipping Protocols

4th May 2021

Abstract

[OUTDATED: Needs to be rewritten]
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

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.

1.1 About Weak Coin Flipping

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

[1] which are analysed
[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.

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?).

1.2 Contributions

[TODO: fix it—this is outdated] In this work, we start with a device independent (DI) coin flipping (CF) protocol introduced[4] in [?] 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.

1.3 Proof Technique

Notation and Cheat Vectors

We introduce some notation to facilitate the discussion here. Denote the DI CF protocol introduced in [?] 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) malicious and Bob follows the protocol $\mathcal{I}$. Then, the cheat vectors for Alice constitute the set

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

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.

[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.

Compositions

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

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

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 "abort augmented" composition (of protocol $\mathcal{X}$) by $C_{\text{AAC}}(\mathcal{X})$.

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

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

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

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

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

  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 $p_X^*(C_{\text{AAC}}(\mathcal{P}))$ and $p_X^*(C_{\text{AAC}}(\mathcal{P}, \mathcal{P}, \ldots, \mathcal{P}, \mathcal{Q}))$.

aoeu

2 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 [**?**].

2.1 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]

[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 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.

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 $\square^A_1, \square^A_2 \ldots \square^A_p$ and Bob is given boxes $\square^B_1, \square^B_2, \ldots \square^B_q$.
   (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]

   $$\{M^{A_1}_{a|x}\}_a, \{M^{A_2}_{a|x}\}_a, \ldots \{M^{A_p}_{a|x}\}_a$$

   which act on them and Bob is, analogously, given quantum registers $B_1, B_2, \ldots B_q$ together with POVMs

   $$\{M^{B_1}_{b|y}\}_b, \{M^{B_2}_{b|y}\}_b, \ldots, \{M^{B_q}_{b|y}\}_b.$$

   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.

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

(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 **??**) which uses only quantum boxes (Definition **??**) can be implemented as a measure and exchange protocol (Definition **??**).

2.2 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. $\sigma_x\sigma_y + \sigma_y\sigma_x = 0$.

2.3 The Protocol

The best DI CF protocol known is the one introduced in [**?**]. 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 Claim **??**, it is clear that when both players follow Algorithm **??** using GHZ boxes (Definition **??**), Bob never aborts and they win with equal probabilities. The security of the protocol is summarised next.

[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.

Lemma 8 (Security of SCF)**.** *[?] Let $\mathcal{I}$ denote the protocol corresponding to Algorithm ??. Then, the success probability of cheating Bob, $p_B^*(\mathcal{I}) \leq \frac{3}{4}$ and that of cheating Alice, $p_A^*(\mathcal{I}) \leq \cos^2(\pi/8)$. 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.

3 First Technique: Self-testing (single shot, unbalanced)

TODO: Assumption: No honest abort.

We make two observations.

First, in Algorithm ?? 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 $P_B^* \lesssim 0.667\ldots$.

3.1 Cheat Vectors

As alluded to in Section ??, using cheat vectors, it is sometimes possible to compose protocols and obtain a lower bias compared to protocols which are composed without using cheat vectors. We describe such procedures in the next section, Section ??. Here, we simply define cheat vectors and show that self-testing allows one to express relevant optimisation problems over cheat vectors as semi definite programmes.

Definition 9 (Cheat Vectors)**.** Given a protocol $\mathcal{I}$, denote by $\mathbb{C}_B(\mathcal{I})$ the set of *cheat vectors* for Bob, which is defined as follows :

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

and analogously, denote by $\mathbb{C}_A(\mathcal{I})$ the set of cheat vectors for Alice (see Equation (??)).

3.2 Alice self-tests | Protocol $\mathcal{P}$

We begin with the case where Alice self-tests. In the honest implementation, the *trio* of boxes used in the following are characterised by the GHZ setup (see Claim ??).

Algorithm 10 (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 11. *Let $\mathcal{P}$ denote the protocol corresponding to Algorithm ??. Then Alice's cheating probability $p_A^*(\mathcal{P}) \leq \cos^2(\pi/8) \approx 0.852$. Further, let $c_0, c_1, c_\perp \in \mathbb{R}$, and $\mathbb{C}_B(\mathcal{P})$ be the set of cheat vectors for Bob. Then, as $N \to \infty$, the solution to the optimisation problem $\max(c_0\alpha + c_1\beta + c_\perp\gamma)$ over $\mathbb{C}_B(\mathcal{Q})$ approaches that of a semi definite programme. In particular, i.e. for $c_0 = c_\perp = 0$ and $c_1 = 1$, $p_B^*(\mathcal{P}) \lesssim 0.667\dots$ (in the limit).*

We defer the proof to Section ??. The value for $p_B^*(\mathcal{P})$ was obtained by numerically solving the corresponding semi definite programme while the analysis for cheating Alice is the same as that of the original protocol.

3.3 Bob self-tests | Protocol $\mathcal{Q}$

What if we modified the protocol and had Bob self-test his boxes? Does that yield a better protocol? We address the first question now and the second in the subsequent section.

Algorithm 12 (Bob self-tests his boxes)**.** *Proceed exactly as in Algorithm ??, except for the self-testing where the rolls of Alice and Bob are reversed. More explicitly, suppose there are N trios of boxes; Alice has the first part and Bob has the remaining two parts, of each trio.*

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

Henceforth, proceed as in Algorithm ?? after the self-testing step.

As already indicated in Section ??, we don't expect the cheating probabilities to improve but we do expect an SDP characterisation of Alice's cheat vectors.

Lemma 13. *Let $\mathcal{Q}$ denote the protocol corresponding to Algorithm ??. Then, Alice's cheating probability, $p_A^*(\mathcal{Q}) \leq 3/4$ and Bob's cheating probability, $p_B^*(\mathcal{Q}) \leq \cos^2(\pi/8)$ (which are the same as those in Lemma ??). Further, let $c_0, c_1, c_\perp \in \mathbb{R}$, and $\mathbb{C}_A(\mathcal{Q})$ be the set of cheat vectors for Alice. Then, as $N \to \infty$, the solution to the optimisation problem $\max(c_0\alpha + c_1\beta + c_\perp\gamma)$ over $(\alpha, \beta, \gamma) \in \mathbb{C}_A(\mathcal{Q})$ approaches that of a semi definite programme.*

The proof is again deferred; see Section ??.

4 Second Technique: Bias Suppression

In this section, we use the convention that $\mathcal{I}, \mathcal{P}$ and $\mathcal{Q}$ correspond to the protocols described in Algorithm ??, Algorithm ?? and Algorithm ??, respectively. Notice that $p_A^*(\mathcal{X}) \geq p_B^*(\mathcal{X})$ where $\mathcal{X} \in \{\mathcal{I}, \mathcal{P}, \mathcal{Q}\}$. We call such protocols "unbalanced". In this section we start from unbalanced WCF protocols and compose them to construct balanced WCF protocols. To this end, we introduce some notation and the term "polarity", to capture which among A and B is favoured.

Definition 14 (Unbalanced protocols, Polarity)**.** Given a WCF protocol $\mathcal{X}$, we say that it is unbalanced if $p_A^*(\mathcal{X}) \neq p_B^*(\mathcal{X})$. We say that $\mathcal{X}$ has polarity A if $p_A^*(\mathcal{X}) > p_B^*(\mathcal{X})$ and polarity B if $p_A^*(\mathcal{X}) < p_B^*(\mathcal{X})$.

Finally, let $X, Y \in \{A, B\}$ be distinct and suppose that $\mathcal{R}$ is an unbalanced protocol. Then, we define $\mathcal{R}_X$ to be protocol $\mathcal{R}$ where Alice's and Bob's roles are possibly interchanged so that $\mathcal{R}_X$ has polarity X, i.e. $p_X^*(\mathcal{R}_X) > p_Y^*(\mathcal{R}_X)$. We refer to $\mathcal{R}_X$ as $\mathcal{R}$ polarised towards X.

TODO: It might help to explain "roles of Alice and Bob" are interchanged a little bit using the "flip and declare" protocol.

TODO: Winner gets Polarity

We now describe how these protocols can be composed such that the "winner gets polarity".

Algorithm 15 ($C(.,.)$ and $C(.)$)**.** *Given two unbalanced WCF protocols, $\mathcal{X}$ and $\mathcal{Y}$, let $\mathcal{X}_A, \mathcal{X}_B$ and $\mathcal{Y}_A, \mathcal{Y}_B$ be their polarisations (see Definition ??). Define $C(\mathcal{X}, \mathcal{Y})$ as follows:*

1. *Alice and Bob execute $\mathcal{X}_A$ and obtain outcome $X \in \{A, B, \perp\}$.*
2. *If*
   (a) *$X = A$, execute $\mathcal{Y}_A$ and obtain outcome $Y \in \{A, B, \perp\}$, else if*
   (b) *$X = B$, execute $\mathcal{Y}_B$ and obtain outcome $Y \in \{A, B, \perp\}$, and finally if*
   (c) *$X = \perp$, set $Y = \perp$.*

   Output Y.

Let $\mathcal{Z}^{i+1} := C_{\text{AAC}}(\mathcal{X}, \mathcal{Z}^i)$ for $i \geq 1$, and $\mathcal{Z}^1 := \mathcal{X}$. Then, formally, define $C(\mathcal{X}) := \lim_{i\to\infty} \mathcal{Z}^i$.[9]

Example 16. TODO: Perhaps show how $\mathcal{Z}_B$ is constructed from $\mathcal{Z} = C_{\text{stand}}(\mathcal{X}, \mathcal{Y})$. Perhaps have this in the appendix.

In the following sections, we work out some examples and analyse their security and see where having a neat characterisation of cheat vectors helps.

4.1 Standard Analysis

In the standard analysis, we only use bounds on the cheating probabilities, p_A^* and p_B^*, of the protocols being composed to obtain the corresponding bounds for the resultant protocol. Let us take an example. Consider protocol $\mathcal{P}$ (see Algorithm ??) and recall (see Lemma ??)

$$p_A^*(\mathcal{P}_A) \leq \alpha \approx 0.852\ldots,$$
$$p_B^*(\mathcal{P}_A) \leq \beta \approx 0.667\ldots.$$

Note that therefore $p_A^*(\mathcal{P}_B) \leq \beta$ and $p_B^*(\mathcal{P}_B) \leq \alpha$. Further, let $\mathcal{P}' := C(\mathcal{P}, \mathcal{P})$, i.e. Alice and Bob first execute $\mathcal{P}_A$ and if the outcome is A, they execute $\mathcal{P}_A$, otherwise they execute $\mathcal{P}_B$ (see Figure ??). Then,

$$p_A^*(\mathcal{P}'_A) \leq \alpha\alpha + (1-\alpha)\beta =: \alpha^{(1)}, \tag{3}$$
$$p_B^*(\mathcal{P}'_B) \leq \beta\alpha + (1-\beta)\beta =: \beta^{(1)}.$$

[9] This is just to facilitate notation. This way the cheating probabilities p_A^* and p_B^* converge and numerically this only takes a few compositions to reach in our case.

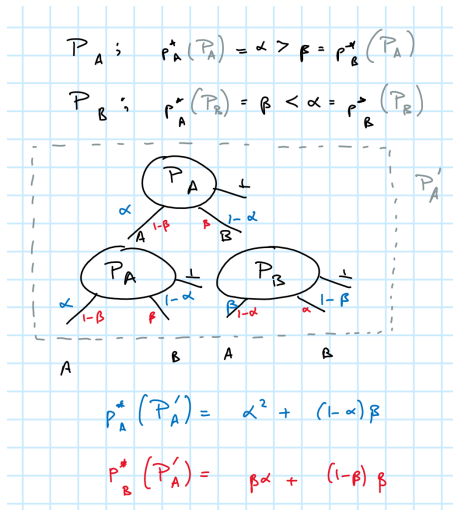

Figure 1: Standard analysis (TODO: improve the caption)

To see this, focus on Equation (??). Alice knows that if she wins the first round, her probability of winning is $\alpha > \beta$. She knows that in the first round, she can force the outcome A with probability at most α. Assuming that with all the remaining probability, the outcome is B (which it is not because Bob will abort with some probability) gives us a bound on the best that Alice can do. Evidently, this bound may not be tight and we address that in the following subsection.

A side remark: one consequence of this simplified analysis is that[10] $\alpha^{(1)} > \beta^{(1)}$. Proceeding similarly, i.e. defining $\mathcal{P}'' := C_{\text{stand}}(\mathcal{P}, \mathcal{P}')$, and repeating $k+1$ times overall, one obtains[11]

$$\alpha^{(k+1)} = \alpha\alpha^{(k)} + (1-\alpha)\beta^{(k)}$$
$$\beta^{(k+1)} = \beta\alpha^{(k)} + (1-\beta)\beta^{(k)}.$$

In the limit of $k \to \infty$, one obtains

$$p^*_A(C_{\text{stand}}(\mathcal{P})) = p^*_B(C_{\text{stand}}(\mathcal{P})) \leq \lim_{k\to\infty} \alpha^{(k)} = \lim_{k\to\infty} \beta^{(k)} \approx 0.8199\ldots.$$

Proceeding similarly, one obtains for $X \in \{A, B\}$ and $\mathcal{X} \in \{\mathcal{I}, \mathcal{Q}\}$,

$$p^*_X(C_{\text{stand}}(\mathcal{X})) \lesssim 0.836\ldots$$

We thus have the following.

Theorem 17. *Let $X \in \{A, B\}$ and $\mathcal{X} \in \{\mathcal{I}, \mathcal{Q}\}$. Then $p^*_X(C(\mathcal{P})) \lesssim 0.8199\ldots$ and $p^*_X(C(\mathcal{X})) \lesssim 0.836\ldots$. [TODO: remove approx eq]*

4.2 Cheat Vector Analysis

Remark 18. Again, even though it seems that Alice might be favoured, C may result in a protocol with the opposite polarity—let $\mathcal{Z} := C(\mathcal{X}, \mathcal{Y})$, then $\mathcal{Z}_A$ is not necessarily the same as $\mathcal{Z}$.

As before, let us workout an example. Consider protocol Q this time (see Algorithm ??) and let $(v_A, v_B, v_\perp) \in \mathbb{C}_A(\mathcal{Q}_A)$ be a cheat vector for $\mathcal{Q}$. Further, recall that

$$p^*_A(\mathcal{Q}_A) \leq \alpha = 3/4 = 0.75$$
$$p^*_B(\mathcal{Q}_A) \leq \beta = \cos^2(\pi/8) = 0.8535\ldots$$

[10] $\alpha^{(1)} - \beta^{(1)} = (\alpha-\beta)\alpha - (\alpha-\beta)\beta = (\alpha-\beta)^2 > 0$

[11] Note that $\alpha^{(k+1)} - \beta^{(k+1)} = (\alpha^{(k)} - \beta^{(k)})(\alpha - \beta) > 0$, which proves Claim ??.

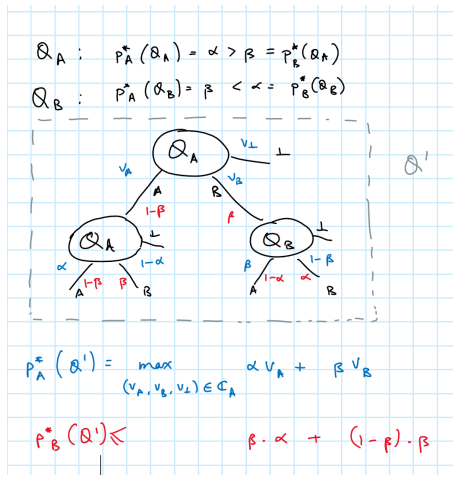

Figure 2: Cheat vector analysis. (TODO: improve the caption)
$(v_A, v_B, v_\perp) \in \mathbb{C}_A$; because we don't have a good characterisation of $\mathbb{C}_B$, we use (red) $1-\beta$ and β for obtaining a bound on $p_B^*(Q')$.

and note that, therefore, $p_A^*(Q_B) \leq \beta$ and $p_B^*(Q_B) \leq \alpha$. Let $Q' := C(Q, Q)$, i.e. Alice and Bob first execute Q_A and if the outcome is A, they execute Q_A, otherwise they execute Q_B (see Figure ??). Our goal is again to obtain an upper bound on $p_A^*(Q')$ and $p_B^*(Q')$, but this time, using cheat vectors. We start from the bottom of Figure ??.

Suppose Alice is malicious. Conditioned on the event that the protocol reaches the bottom left step, it is clear that Alice's highest probability of winning is α. In this case, the abort probability doesn't play a role. Similarly, conditioned on the event that the protocol reaches the bottom right step, it is clear that Alice's highest probability of winning is β. At the first step then, it is clear that the highest probability of Alice winning is

$$p_A^*(Q') \leq \max_{(v_A, v_B, v_\perp) \in \mathbb{C}_A} \alpha v_A + \beta v_B \tag{4}$$

$$\leq \alpha^2 + \beta(1-\alpha). \tag{5}$$

The first inequality is saturated if α and β are tight bounds. The second inequality holds because $\alpha > \beta$ so one cannot do better than putting all the weight on α (i.e. α) and the remaining on β (i.e. $1-\alpha$), as we did earlier. [DISCUSS/TODO: Remove repetition; I already explained this.] Operationally, this means that Alice is forcing the outcome A and treating the outcomes B and $\perp$ as proceeding to the Q_B step (i.e. the bottom right step). However, Bob only agrees to proceed to the bottom right step if the outcome is B and hence the maximization, Equation (??), potentially yields a better bound.

The case for malicious Bob would have been similar if we had a good characterisation for $\mathbb{C}_B(Q)$. Since we don't, we content ourselves with a weaker bound, the one analogous to Equation (??):

$$p_B^*(Q') \leq \beta\alpha + (1-\beta)\beta.$$

TODO: put the numbers (they seem to be off).

Thus to obtain $p_A^*(Q')$, one solves an SDP (numerically) and $p_B^*(Q')$ is obtained readily. Using this, one can consider $Q'' := \mathbb{C}(Q, Q')$ and evaluate the cheating probabilities $p_A^*(Q'')$ again as an SDP and $p_B^*(Q'')$ directly. This leads us to the following result.

Theorem 19. *Let $X \in \{A, B\}$. Then*

$$p_X^*(C(Q)) \lesssim 0.8175\ldots$$

and

$$p_X^*(C(Q)) \lesssim$$

(TODO: calculate).

To suppress the bias further, we note that at the very last step, only the cheating probabilities $p_A^*(\mathcal{Q}) \leq \alpha$ and $p_B^*(\mathcal{Q}) \leq \beta$ played a role (i.e. the fact that the cheating vectors $\mathbb{C}_A$ for $\mathcal{Q}$ had an SDP characterisation was not used). Further, we know that $p_A^*(\mathcal{P}) = p_A^*(\mathcal{Q})$ but $p_B^*(\mathcal{P}) < p_B^*(\mathcal{Q})$, i.e. using $\mathcal{P}$ at the very last step will result in a strictly better protocol.

Theorem 20. *Let $X \in \{A, B\}$,*

$$\mathcal{Z}^1 := C(\mathcal{Q}, \mathcal{P}), \quad \text{and}$$
$$\mathcal{Z}^{i+1} := C(\mathcal{Q}, \mathcal{Z}^i) \quad i > 1.$$

Then

$$\lim_{i \to \infty} p_X^*(\mathcal{Z}^i) \lesssim 0.7908\ldots$$

TODO: Argue why $C(\mathcal{P})$ was not such a great idea? I think this was because the cheat vector with the first component, e.g., maxed out, the abort probaibility was tiny for $\mathcal{P}$. Perhaps do the simulation and add the results.

5 Application: Strong Coin Flipping

Using ...

6 Security Proofs

6.1 Cheat vectors optimisation using Semi Definite Programming

6.1.1 SDP when Alice self-tests

Proof of Lemma **??**. The analysis for cheating Alice is the same as the original protocol. We consider the analysis for cheating Bob. 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 $\rho_0 \in \mathcal{D}(HXRAS)$ be Alice's state after measuring to get a 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.[12]
2. Similarly, observe that Bob sending the variable g can be modelled as $\mathrm{tr}_g(\rho_1) = \mathrm{tr}_s(\rho_0)$ where $\rho_1 \in \mathcal{D}(HXRAG)$.
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.

[12] This doesn't mean that Alice forgets s. She can always construct s again because she has all the remaining registers

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.[13]

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 $a \oplus b \oplus c = xyz \oplus 1$. We model this as $\mathrm{tr}_d(\rho_2) = \rho_1 \otimes \frac{1}{2}\mathbb{I}_y$ where $\rho_2 \in \mathcal{D}(HXRAGYD)$.

Let (α, β, γ) be a *cheat vector* (see Definition **??**), i.e. for some cheating strategy of Bob,

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

Clearly, for the strategy encoded by the variables ρ_1 and ρ_2, one can write

$$\begin{aligned}\alpha &= \mathrm{tr}(\Pi_0 \rho_2)\\ \beta &= \mathrm{tr}(\Pi_1 \Pi_{\mathrm{GHZ}} \rho_2)\\ \gamma &= \mathrm{tr}(\Pi_1 \Pi_{\mathrm{abort}} \rho_2)\end{aligned}$$

where Π_i checks if $x \oplus g = i$ holds, Π_{GHZ} checks if the GHZ test passed and $\Pi_{\mathrm{abort}} := \mathbb{I} - \Pi_{\mathrm{GHZ}}$. The SDP is therefore

$$\max \quad c_0\alpha + c_1\beta + c_\perp\gamma$$

subject to

$$\begin{aligned}\mathrm{tr}_D(\rho_2) &= \rho_1 \otimes \frac{1}{2}\mathbb{I}_y\\ \mathrm{tr}_G(\rho_1) &= \mathrm{tr}_S(\rho_0)\end{aligned}$$

where

$$\begin{aligned}\rho_0 &\in \mathcal{D}(HXRAS)\\ \rho_1 &\in \mathcal{D}(HXRAG)\\ \rho_2 &\in \mathcal{D}(HXRAGYD)\end{aligned}$$

and $\rho_0 := \ldots$ (TODO: from the code). To obtain $p_B^*(\mathcal{P})$, this SDP was numerically solved for $c_1 = 1$, $c_0 = c_\perp = 0$.

[TODO: explain if really needed] This can be seen graphically if we evaluate $\beta_{\max}(\alpha) := \max_{(\alpha,\beta,\gamma)\in C} \beta$ and plot it as a function of α. Then □

6.1.2 SDP when Bob self-tests

Proof of Algorithm **??**. Denote by $\mathcal{I}$ the protocol corresponding to Algorithm **??**.

It is evident that $p_B^*(\mathcal{Q}) \leq p_B^*(\mathcal{I})$ because compared to $\mathcal{I}$, in $\mathcal{Q}$ Alice performs an extra test. However, it is not hard to see that the inequality is saturated, i.e. $p_B^*(\mathcal{Q}) = p_B^*(\mathcal{I})$. Consider ... (TODO: recall/re-construct the cheating strategy for Bob that lets him win with the same 3/4 probability).

From Lemma **??**, it is also clear that $p_A^*(\mathcal{Q}) = p_A^*(\mathcal{I})$ because the only difference between Bob's actions in $\mathcal{Q}$ and $\mathcal{I}$ is that Bob self-tests to ensure his boxes are indeed GHZ. However, the optimal cheating strategy for $\mathcal{I}$ can be implemented using GHZ boxes.

This establishes the first part of the lemma. For the second part, i.e. establishing that optimising $c_0\alpha + c_1\beta + c_\perp\gamma$ over $(\alpha, \beta, \gamma) \in \mathbb{C}_A$ is an SDP, we proceed as follows. We know that Bob starts with the state $\sigma_0 = \mathrm{tr}_H(|\mathrm{GHZ}\rangle\langle\mathrm{GHZ}|_{HBC})$ due to the self-testing step.

1. Alice sending s is modelled as $\mathrm{tr}_S(\sigma_1) = \sigma_0$ where $\sigma_1 \in \mathcal{D}(SBC)$.

[13] This would mean that Bob can hold a purification, thereby y.

2. Bob sending $g \in_R \{0,1\}$ is modelled as $\sigma_1 \otimes \frac{1}{2}\mathbb{I}_G$.
3. As in the previous proof, since Bob is testing, he can deduce x from $x \oplus g = 0$. So formally we do nothing.
4. Alice sending a is modelled as $\mathrm{tr}_A(\sigma_2) = \sigma_1 \otimes \frac{1}{2}\mathbb{I}_G$ where $\sigma_2 \in \mathcal{D}(SBCGA)$.

As in the previous proof, let

$$\begin{aligned}\alpha &= \mathrm{tr}(\Pi_0 \Pi_{\mathrm{GHZ}}, \sigma_2)\\ \beta &= \mathrm{tr}(\Pi_1 \Pi_{\mathrm{GHZ}}, \sigma_2)\\ \gamma &= \mathrm{tr}(\Pi_{\mathrm{Abort}}, \sigma_2)\end{aligned}$$

(TODO: See why I have an asymmetry between abort and GHZ in the two proofs) [TODO: complete these] where Π_i are ... and Π_{GHZ} is ... . Then the final SDP can be expressed as

$$\max \quad (c_0\alpha + c_1\beta + c_\perp\gamma)$$

subject to

$$\begin{aligned}\mathrm{tr}_A(\sigma_2) &= \sigma_1 \otimes \frac{1}{2}\mathbb{I}_G\\ \mathrm{tr}_S(\sigma_1) &= \mathrm{tr}_H(|\mathrm{GHZ}\rangle\langle\mathrm{GHZ}|_{HBC})\end{aligned}$$

where

$$\begin{aligned}\sigma_1 &\in \mathcal{D}(SBC)\\ \sigma_2 &\in \mathcal{D}(SBCGA).\end{aligned}$$

□

7 Self-testing: asymptotic case

8 Self-testing: finite testing

We assume that the 3n boxes are described by some joint quantum state and local measurement operators. After playing the GHZ game with $3(n-1)$ of them, and verifying that they all pass, we want to make a statement about the remaining box, whose state $\tilde{\rho}$ is conditioned on the passing of all the other test.

8.1 Estimation of GHZ winning probability

Algorithm 1 Estimation of the GHZ value

1: Pick a box $J \in [n]$ uniformly at random.
2: For $i \in [n]\backslash J$, play the GHZ game with box i, denote outcome of game as $X_i \in \{0,1\}$
3: If

$$\Omega: \quad X_i = 1, \text{ for all } i \in [n]\backslash J \tag{6}$$

4: Then conclude that the remaining box satisfies

$$T: \quad E[X_J|J,\Omega] \geq 1-\delta \tag{7}$$

The expectation value of $E[X_J|J,\Omega]$ accurately describes the expected GHZ value associated to the state of the remaining boxes J, conditioned on having measuring some outcome sequence in the other boxes which passes all the GHZ tests. Note that the conditioning in J is important because otherwise we would get a bound on the GHZ averaged over all boxes, but we are only interested in the remaining box.

Proposition 1 (Security statement)**.** *For any implementation of the boxes, the joint probability that that the test Ω passes and that the conclusion T is false is small* $\Pr[\Omega \cap \bar{T}] \leq \frac{1}{1-\delta+n\delta} \leq \frac{1}{n\delta}$*, where the first upper-bound is tight.*

This is the correct form of the security statement. It is important to bound the joint distribution of Ω and $\overline{T}$, and not $\Pr[\overline{T}|\Omega]$, conditioning on passing the test Ω. Indeed in the latter case, it would not be possible to conclude anything of value about the remaining box J, as there could be some implementation of the boxes which has a very low expectation value of GHZ, but which passes the test with small but non-zero probability. The present security definition has a nice interpretation in the composable security framework of [ref]. Consider an hypothetical ideal protocol, which after having chosen J, only passes when T is true. In that case, $\Pr[\Omega \cap \overline{T}] = 0$. Then the actual protocol is equivalent the ideal one, except that if fails with probability ϵ, and so it is ϵ-close to the ideal algorithm.

Proof. For a given implementation of the boxes, let $p(x_1, \cdots x_n)$ denote the joint probability distribution of passing the GHZ games. Let $S = \{j|\ E[X_j|j, \Omega] < 1 - \delta\} \subset [n]$ be the set of boxes that have an expectation value for GHZ (conditioned on passing in the other boxes) below our target threshold and let $m = |S|$ be the number of such boxes. The value of m is unknown, so we will need to maximise over it in the end.

Let $\alpha = \Pr(\{X_i\}_i = 1)$ and $\beta_i = \Pr(\{X_i\}_{i\neq j} = 1 \cap X_j = 0)$ be respectively the probabilities of the events where all the tests pass, or they all pass except for the jth test. This allows us to rewrite $E[X_j|J = j, \Omega] = \Pr(\{X_i\} = 1)/\Pr(\{X_i\}_{i\neq j} = 1) = \alpha/(\alpha + \beta_j)$, and so, by definition of S, we have $\alpha/(\alpha + \beta_j) < (1 - \delta)$, for $i \in S$, which is equivalent to $\beta_j > \frac{\delta}{1-\delta}\alpha$.

The aim of the proof is to bound the probability $\Pr[\Omega \cap \overline{T}]$, which we can rewrite as

$$\Pr(\Omega \cap \overline{T}) = \sum_j \frac{1}{n} \Pr(\Omega \cap \overline{T}|J = j) = \sum_{j\in S} \frac{1}{n} \Pr(\{X_i\}_{i\neq j} = 1) = \frac{1}{n} \sum_{j\in S} (\alpha + \beta_i)\,, \tag{8}$$

where we have conditioned and summed over the different values of J and kept the ones, conditioned on which T is false. We are thus left with the optimisation problem

$$\max_{\alpha>0,(\beta_i)_i>0} \quad \frac{1}{n}\left(\sum_{j\in S} \alpha + \beta_j\right) \tag{9}$$

$$\text{subject to} \quad \alpha + \sum_{j\in S} \beta_j \leq 1 \tag{10}$$

$$\beta_j \geq \frac{\delta}{1-\delta}\alpha, \text{ for } j \in S \tag{11}$$

This is a linear problem. Simplifying it by defining $\Sigma = \sum_{j\in S} \beta_j$, gives

$$\max_{\alpha>0,\Sigma>0} \quad \frac{1}{n}(m\alpha + \Sigma) \tag{12}$$

$$\text{subject to} \quad \alpha + \Sigma \leq 1 \tag{13}$$

$$\Sigma \geq m\frac{\delta}{1-\delta}\alpha \tag{14}$$

It is easily shown that the maximum is attained for $(\alpha, \Sigma) = \left(\frac{1-\delta}{1-\delta+m\delta}, \frac{m\delta}{1-\delta+m}\right)$ which gives the upper-bound

$$\Pr[\Omega \cap \overline{T}] \leq \frac{1}{n} \max_m \frac{m}{1-\delta+m\delta} = \frac{1}{1-\delta+n\delta} \tag{15}$$

We note that the upper-bound is an increasing function of m and so the maximum is attained for $m = n$. This yield the desired upper-bound. From the converse statement, we note that from the present proof we can construct a probability distribution $p(x_1, \cdots x_n)$, which saturates all inequalities, and so the upper-bound $\frac{1}{1-\delta+n\delta}$ is tight. □

8.2 Robust self-testing and continuity of SDP

8.3 The self-testing step [Discuss with Tom before writing]

Theorem 21. *Let $\{X_1, X_2 \dots X_n\}$ be n random variables which are possibly correlated and take values in $\{0, 1\}$. Let $J \in \{1, 2 \dots n\}$ be a random variable, sampled from a uniformly random distribution. Then*

$$\Pr\left\{\left|\frac{1}{n-1} \cdot \sum_{i \in \{1,2\dots n\} \setminus J} X_i - \mathbb{E}(X_J|J)\right| > r\right\} \leq \exp(-cnr^2)$$

for some $c > 0$.

Proof. Let

$$Z_i := \mathbb{E}\left[\frac{1}{n-1}\left(\sum_{l \in [n] \setminus J} X_l\right) - X_J | F^i\right]$$

for $i \in \{0, 1, \dots n+1\}$ where $\{F^i\}$ is a filteration. From Fact **??**, $Z_0 \dots Z_n$ is a martingale. Let F^0 be the trivial sigma algebra. Suppose F^1 specifies J and F^{i+1} specifies $\{J, X_1 \dots X_i\}$ for $i \in \{1, 2 \dots n-1\}$. Note that

$$\begin{aligned} Z_0 &= \frac{1}{n-1}\mathbb{E}\left(\sum_{l \in [n] \setminus J} X_l\right) - \frac{1}{n}\sum_{j \in [n]} \mathbb{E}(X_j) \\ &= \frac{1}{n-1}\frac{1}{n}\sum_{j \in [n]}\sum_{l \in [n] \setminus j} \mathbb{E}(X_l) - \frac{1}{n}\sum_{j \in [n]} \mathbb{E}(X_j) \\ &= 0. \end{aligned}$$

Also note that

$$Z_n = \frac{1}{n-1}\sum_{l \in [n] \setminus J} X_i - \mathbb{E}(X_J|J).$$

Finally, since $|Z_i - Z_{i-1}| \leq d/n$ for some fixed $d > 0$ (the filteration only fixes one X at a time so the differences can change by at most $1/n$; TODO argue for the 0 case), we can apply Theorem **??** to deduce

$$\mathbb{P}\left(\left|\frac{1}{n-1}\sum_{l \in [n] \setminus J} X_i - \mathbb{E}(X_J|J)\right| > r\right) \leq e^{-cnr^2}$$

where $c > 0$ is some constant. □

8.3.1 Notes on Martingales

Definition 22 (Martingale)**.** Let $(\Omega, F, \mathbb{P})$ be a probability space. Consider random variables $X_0, \dots X_n$ and a filteration $F_0, F_1 \dots F_n$ i.e. sigma algebras satisfying $F_0 \subseteq F_1 \cdots \subseteq F_n \subseteq F$. Then the sequence $(X_i, F_i)_{i=0}^n$ is a *martingale* if

1. $X_i \in \mathbb{L}^1(\Omega, F_i, \mathbb{P})$, i.e. $\mathbb{E}(|X_i|) < \infty$
2. $X_{i-1} = \mathbb{E}(X_i|F_{i-1})$ (almost surely) for all $i \in \{1, 2 \dots n\}$.

Fact 23. *Let $X \in \mathbb{L}^1(\Omega, F, \mathbb{P})$ be a random variable and suppose $\{F^i\}_{i=1}^n$ is an arbitrary filteration. Define $Z_i := \mathbb{E}(X|F^i)$ for $i \in \{1, 2 \dots n\}$. Then $Z_1, Z_2 \dots Z_n$ forms a martingale with respect to the filteration.*

The main property we need is

$$\begin{aligned}\mathbb{E}(Z_i|F_{i-1}) &= \mathbb{E}(\mathbb{E}(X|F_i)|F_{i-1}) \\ &= \mathbb{E}(X|F_{i-1}) && \because F_{i-1} \subseteq F_i \\ &= Z_{i-1}.\end{aligned}$$

In particular, when $F_0 = \{\emptyset, \Omega\}$ and $F_n = F$ we have

$$Z_0 = \mathbb{E}(X|F_0) = \mathbb{E}(X), \quad Z_n = \mathbb{E}(X|F_n) = X.$$

Theorem 24 (The Azuma-Hoeffding inequality)**.** *Let $(Z_k, F_k)_{k=0}^n$ be a real-valued martingale sequence. Suppose that there exist nonnegative reals $d_1, \dots d_n$ such that $|Z_k - Z_{k-1}| \leq d_k$ (almost surely) for all $k \in \{1, \dots n\}$. Then, for every $r > 0$,*

$$\mathbb{P}\left(|Z_n - Z_0| \geq r\right) \leq 2 \exp\left(-\frac{r^2}{2\sum_{k=1}^n d_k^2}\right).$$

8.4 The continuity argument [Enter Jamie]