Ideas | Device Independent Weak Coin Flipping

Jamie Sikora, Thomas Van Himbeeck, Atul Singh Arora

March–... 2020

Abstract

(in progress)

Contents

1 Notes from the meetings/thoughts

We follow two approaches with the common goal of constructing better protocols than the ones currently known.

1.1 Meeting/afterthoughts 1 | Monday

Jamie cooked up a bunch of protocols. Here's the rough list of things to pursue.

- Explore the viability of the J0 protocol, proposed by Jamie.
- We also had to determine if the "fully distrustful" article had a better bias for device independent (strong) coin flipping (EDIT: it did) than the device independent weak coin flipping protocol, by essentially the same authors.
- My task was to try to formalise the "formalism".

1.2 Meeting/afterthoughts 2 | Tuesday

There were a few ideas which were floated around by Jamie.

- F1 $\overset{?}{\iff}$ F2 where F1 is the first framework and F2 is this one: since in DI protocols, we can start with any arbitrary state, consider these, followed by only classical information, then one last step where a part of the system is sent to the other.
- Understand the GHZ based protocols and re-express them using the F1 framework.

1.3 Meeting/afterthoughts 3 | Thursday [with Tom]

I read and presented the PRL protocol (without the security analysis). Then I tried presenting the equivalence of the two frameworks, intuitively.

- It looks like we can use Jamie's idea; call it the J3 protocol. It basically adds an additional verification step and since it is for weak CF, it doesn't affect the bias for Bob.
- Tom said that allowing quantum communication can be used to break the security because the quantum device can encode and send the input it was fed and since we allow the adversary to control the quantum channels, this makes them too powerful.
  - True but does learning the other player's input (maybe not always but sometimes) necessarily mean they have too much power? The only sane conclusion I can draw from this is that one has to necessarily allow classical information to be communicated but perhaps allowing quantum communication is also relevant.
  - Why isn't teleportation enough? In the honest case it is clearly enough. We want to show that in the dishonest case, the two models are equivalent. The two models being, one with classical + quantum communication the other with only classical communication. In both, we allow an arbitrary number of devices to be shared (which may be shielded to enforce the required structure for non-locality).
    Question: I have to show that a strategy in one can be equivalently expressed in the other.

1.4 Meeting/afterthoughts 4 | Monday

- I explained the framework; with things much better formalised and with higher clarity about how to impose the constraints for a cheating Bob.
  - Jamie pointed out how some "self-testing of unitaries" like scheme wouldn't quite work.
  - I forgot to ask him, thereafter, how to treat post measurement states of boxes (if at all this makes sense).
- Jamie then explained his abort based protocol which seemingly improves the bias; intuitively, this helps when you compose protocols: if you catch a player cheating, you just abort, instead of playing further and getting cheated further.

1.5 Meeting/afterthoughts 5 | Tuesday [with Tom]

- Jamie presented his new abort based protocols. Tom seemed to be ok with the ideas.
- I presented the proofs of Alice's and Bob's security. We compared it with Jamie's direct analysis and saw where they differed. We worked towards characterising the set $\mathcal{A}_c$, where in addition to the probability that Alice succeeds in convincing Bob, we keep track of the probability of her getting caught cheating.
- Tom also pointed out an oversight about testing Bob; we were sending both s_A and r_A to Bob, before testing him. We must delay the sending of r_A else he can pass the test with certainty.

1.6 aoeu

- Current Approach (improve bias):
  - Continuity; Self-testing (rigorous proof of security)
- NPA based; if
- Quantum Messages exchange; see if this is sorted
- Dip Dip Boom | edge of point games; Certify Unitaries
  - Dip Dip Boom with entanglement + classical communication.
- Entanglement based version; MBQC; have some graph states and use some operations
  - Interactive protocol; like a measurement based approach; could be equivalent
  - may be more natural to translate into device independent approach
    * Issue; can you share it between Alice and Bob

1.7 July 23–25, 2020

- [numerics]
  - Results:
    * Post Jamie Magic: 0.42 (or some such);
      - The issue was that I had not implemented the measurement of A (I was just measuring in the standard Z basis always);
    * Post Jamie Magic, Debug 1: 0.499 (or similar):
      - I realised that after the measurement and computation of S, I wasn't actually simulating the sending of S. Did this by entangling it with S' and tracing it out (so the result is a mixed state and Bob can hold it).
    * Post Jamie Magic, Debug 2; 0.625 (or similar):
      - This seemed to make a lot more sense; decided to do a sanity check to recover 0.75 when we don't do a GHZ test (i.e. the extra test for Bob).
    * Post Jamie Magic, Debug 3; 0.999 (or similar):
      - [*] don't know what is happening;
  - Two issues
    * When I remove the GHZ test, I expect the result to be 0.75 (the older value); instead I get 0.999.
    * Isn't it crucial to send Z? Because how else will Bob know which basis to measure in (e.g. when he's honest). Sure, Alice has both X and Y so for her finding Z is not an issue but Bob has only Y. [this still doesn't explain why the previous step is failing; this should give more power to Bob but he is already able to maximally cheat (without the GHZ test)].

1.8 Meeting notes | Thursday, October 8, 2020

I'm only writing the salient features

- Tom pointed out that the composition techniques, both tree like and recursive, are essentially equivalent; they're a ladder
  - Issue: There's a discrepancy with the numerics
    * [EDIT: Oct 14] I checked, there was a mistake with the numerics; both logical and conceptual (discussed later)
  - Question: it seems that the ladder is unique for most compositions; is this true in general (Jamie seemed to suggest that if the path to a node is relevant, such as majority to win, it may no longer be unique; this is another direction altogether though)
- Jamie pointed out that one could use better protocols in terms of $P^*_{A/B}$ at the leaf and the ones with abort on higher layers.

1.9 Thoughts + Results | Thursday, October 14, 2020

- Notation
  - Protocols
    * Original: $\mathcal{I}$; $p^*_A(\mathcal{I}) \approx 0.853\ldots$, $p^*_B(\mathcal{I}) \approx 0.75$
    * Intermediate Result | Alice self-tests + extra test: $\mathcal{P}$; $p^*_A(\mathcal{P}) \lesssim 0.853\ldots$, $p^*_B(\mathcal{P}) \lesssim 0.667\ldots$; $C_B(\mathcal{P})$ as an SDP (but low abort probability; heavy computationally)
    * Intermediate Result | Bob self-tests: $p^*_{A/B}(\mathcal{Q}) = p^*_{A/B}(\mathcal{I})$; $C_A(\mathcal{Q})$ as an SDP

- Compositions:
  - k compositions of $\mathcal{P}$, for instance, are denote by $C(\mathcal{P}, k)$. When k is suppressed, it is assumed that we pick a large enough k such that $p^*_A(C(\mathcal{P}, k)) \approx p^*_B(C(\mathcal{P}, k))$.
  - Standard Composition: $C_{\text{stand.}}$
  - Sikora Composition: C_{Sikora} (for $\mathcal{P}$, uses the cheat vectors for Bob and for $\mathcal{Q}$ uses the cheat vectors for Alice).
- State of the art: $\mathcal{I}$ and $C_{\text{stand.}}$:
  $p^*_{A/B}(C_{\text{stand.}}(\mathcal{I})) \approx \frac{1}{2} + 0.336\ldots$
- Cumulative Result 1: $\mathcal{P}$ and $C_{\text{stand.}}$:
  $p^*_{A/B}(C_{\text{stand.}}(\mathcal{P})) \approx \frac{1}{2} + 0.3199\ldots$
  - TODO: prove that under standard composition, if $p^*_A(\mathcal{P}) \geq p^*_B(\mathcal{P})$ then $p^*_A(C_{\text{stand.}}(\mathcal{P}, 1)) \geq p^*_B(C_{\text{stand.}}(\mathcal{P}, 1))$.
- Cumulative Result 2: $\mathcal{Q}$ and C_{Sikora}:
  $p^*_{A/B}(C_{\text{Sikora}}(\mathcal{Q})) \approx \frac{1}{2} + 0.317\ldots$
- Cumulative Result 3: $\mathcal{P}$ at the leaves, $\mathcal{Q}$ otherwise and C_{Sikora}:
  $p^*_{A/B}(C_{\text{Sikora}}(\mathcal{Q}, \mathcal{Q}\ldots\mathcal{P})) \approx \frac{1}{2} + 0.2908\ldots$
  where I abused the definition of C_{Sikora} slightly.
  TODO: perhaps output how many times I change the "polarity" of the protocol?

- Issues and their resolution:
  - I was essentially making two (related) mistakes; one was that in the code, I had p^*_A and p^*_B flipped; but this I didn't realise at the time, was because of the following:
    - While
      $$0.853\cdots \approx p^*_A(Q) > p^*_B(Q) = 0.75,$$
      we have
      $$0.8123\cdots \approx p^*_A(C_{\text{stand.}}(\mathcal{Q}, 1)) < p^*_B(C_{\text{stand.}}(\mathcal{Q}, 1)) \approx 0.827.$$
    - So effectively, I had to solve the sub-tree and flip it as needed before joining it to the upper branch for constructing the ones higher up in the hierarchy.

- Questions:
  - So how does this relate to Tom's ladder?
  - What tree must one consider for the majority ladder? I think there, each path becomes a node somehow because it corresponds to a different strategy...

2 F1 | A framework for Distrustful DI protocols

For the moment, consider referring to Collaborations->DI_WCF->F1 (and its subsections). TODO: Add the description here.

3 Sikora's Suppression Technique

We consider a somewhat restricted and simultaneous slightly more general class of weak coin flipping protocols.
EDIT: I got myself thoroughly confused with the whole discussion.

3.1 WCF$^\perp$: Abort-phobic Weak Coin Flipping

Assumption 1. *In the usual definition of WCF, we assume that the players can output A and B. The sole preference of Alice is to convince Bob that the outcome was A and, analogously, the sole preference of Bob is to convince Alice that the outcome was B.*

This is why we never care about what happens when the players disagree. This is because an optimal cheating player, say Alice, will always output A and so if there is a disagreement, it means that Alice failed to convince Bob that the outcome was A. This case is not penalised in the standard construction; this follows directly from the assumption, e.g. Alice doesn't distinguish between the remaining outcomes.

3.1.1 Preference formalised using cones

Distinguishing between the remaining outcomes, i.e. disagreement and abort, can become important (as we shall see). This in turn must be formalised by defining what the players prefer. Taking the most operational perspective, any form of weak coin flipping protocol must end with three possible outcomes—both players agree that the output is either A, B or $\perp$ (abort). For concreteness, suppose Alice is cheating and Bob is honest. Let the elements of, what we call a cheat vector, $(\alpha_A, \alpha_B, \alpha_\perp = 1 - \alpha_A - \alpha_B)$, define the probabilities of the protocol resulting in the outputs $A, B, \perp$ respectively. Often, we drop the last component of the cheat vector for clarity. We formalise Alice's preferences by defining an order among the vectors (with respect to a cone K) as follows:

$$\begin{aligned} (\alpha_A, \alpha_B) \succ_K (\alpha'_A, \alpha_B) &\iff \alpha_A > \alpha'_A, \\ (\alpha_A, \alpha_B) \succ_K (\alpha_A, \alpha'_B) &\iff \alpha_B > \alpha'_B. \end{aligned}$$

The first relation captures the assumption that Alice wishes to increase the probability of obtaining the outcome A, which is the same as that of standard weak coin flipping. The second captures the assumption that Alice prefers to increase the probability of Bob winning compared to that of aborting. This ordering is equivalent to defining the generalised inequality with respect to the cone K where $K := \{(a, b) : a, b \in \mathbb{R}_{\geq 0}\}$. We thus have the following.

Assumption 2. *In Abort-phobic Weak Coin Flipping (written as WCF$^\perp$), the preferences of Alice are given by $(\alpha_A, \alpha_B) \succ_K (\alpha'_A, \alpha'_B)$ while those of Bob are given by $(\beta_A, \beta_B) \succ_K (\beta'_A, \beta'_B)$ where we used the notation of cheat vectors and the definition of K as described above.*

Remark 3. WCF$^\perp$ is relevant in situations exemplified by Figure 2. Situations may arise where a cheating Alice may prefer to abort than have Bob win with a higher probability. This would mean that for Alice, $K = \{(a, -b) : a, b \in \mathbb{R}_{\geq 0}\}$ while for Bob, we must use $-K$. We do not discuss these further in this work.

3.1.2 Disagreement vs Abort

In the preceding discussion, we took the most operational perspective, i.e. any weak coin flipping must end with both players agreeing to either the outcome A, B or $\perp$. We now consider two classes of protocols.

1. Each player outputs[1] either A, B or $\perp$. If they disagree, the protocol is aborted[2].
2. Each player outputs either A or B. If they disagree, the protocol is aborted.

A priori, we note that the first class is more general because a player can abort the protocol with certainty by outputting $\perp$ while in the second class no such mechanism exists, i.e., abort is always subject to the other (read cheating) player's (a priori unknown) choice. The distinction matters if **??** holds. If **??** holds, then the distinction disappears because the outcome of a cheating player is known and therefore the honest player can abort with certainty by declaring themselves the winner.

One drawback of this approach is that noise will break the relation between abort and agreement; we defer these issues to a later stage of investigation.

3.2 Formal Definitions of WCF and WCF$^\perp$

NB. I am using two names for the same concept. WCF$^\perp$=Abort-phobic WCF=WCF with cheating abort.

Definition 4 (WCF with cheating abort)**.** Weak Coin Flipping is a two player (Alice and Bob) interactive protocol wherein they wish to generate a random bit.

- After the interactions, each player outputs either A, B or $\perp$ (for abort). The two players, together, can output $\{AA, BB, AB, BA, * \perp, \perp *\}$ where $*$ is either A or B. Whenever the outcomes differ, the players agree to abort the protocol.
  (This is needed because to proceed, they must agree on what to do next; thus even though after the interaction their outputs may vary but they must, still, agree on how to proceed).
  - *Both Honest.* If the outputs are AA (or BB), Alice and Bob both conclude that Alice (or Bob) won. These two outputs, AA and BB, appear with probability $\frac{1}{2}$.
  - *Nomenclature for Security.* Suppose Alice is cheating (i.e. deviating from the protocol in some way). Let $\vec{q}_A = (\alpha_A, \alpha_B, \alpha_\perp = 1 - \alpha_A - \alpha_B)$, what we call a *cheat vector,* where
    - α_A denotes the probability of the outcome AA
    - α_B denotes the probability of the outcome BB
    - $\alpha_\perp$ denotes the probability of any other outcome, i.e. $\alpha_\perp = 1 - \alpha_A - \alpha_B$.

    Let *cheat vector set* for Alice be $\mathcal{A}_c := \{\vec{q}_A : \text{The probabilities correspond to some cheating strategy of Alice}\}$. Analogously, define $\vec{q}_B = (\beta_A, \beta_B, \beta_\perp = 1 - \beta_A - \beta_B)$ and the *cheat vector set* for Bob be $\mathcal{B}_c$.

[1] In the analysis, we assume the cheating player learns the output of the honest player, before broadcasting his outcome. This is analogous to allowing the cheating player to create and distribute the boxes.

[2] (if they agree, then the outcome is the agreed outcome)

– *Standard WCF Security.* A WCF protocol has bias $\max\{\epsilon_A, \epsilon_B\}$ if $P_A^* = \frac{1}{2} + \epsilon_A$ where $P_A^* = \max_{(\alpha_A,\alpha_B,\alpha_\perp)\in\mathcal{A}_c} \alpha_A$ and $P_B^* = \frac{1}{2} + \epsilon_B$ where $P_B^* = \max_{(\beta_A,\beta_B,\beta_\perp)\in\mathcal{B}_c} \beta_B$.
(This formalises the fact that the players have preferences; i.e. we only care about protecting Bob from Alice trying to bias towards outcome A and analogously, about protecting Alice from Bob trying to bias towards outcome B).

For clarity, we will be slightly redundant.

Definition 5 ((Standard) WCF)**.** (Standard) Weak Coin Flipping is a two player (Alice and Bob) interactive protocol wherein they wish to generate a random bit.

- After the interactions, each player outputs either A or B. (If they catch the other player cheating, they declare themselves the winner). The two players, together, can output $\{AA, BB, AB, BA\}$. Whenever the outcomes differ, the honest player is declared the winner (assuming at least one is honest; when both are cheating, we can't say anything).
  - *Both Honest.* If the outputs are AA then Alice wins and if they are BB then Bob wins. These outputs, AA and BB, appear with probability $\frac{1}{2}$.
  - *Security.* Suppose Alice is cheating and Bob is honest. Then the highest probability (over all of Alice's cheating strategies) of Bob outputting A is given by P_A^*; Alice is assumed to output A. Analogously, the highest probability (over all of Bob's cheating strategies) of Alice outputting B (and thus winning) is given by P_B^*; Bob is assumed to output B.

TODO: I should be able to handle abort in standard WCF as well, just like we do in standard SCF; because if not, any non-trivial noise will break the protocol.

Remark 6. A WCF protocol with cheating abort can be converted into a (standard) WCF protocol by mapping the outputs (of the protocol) A to A, B to B and $\perp$ to the honest player winning.

Definition 7 ((Standard) SCF)**.** Strong Coin Flipping is a two player (Alice and Bob) interactive protocol wherein they wish to generate a random bit.

- After the interactions, each player outputs either 0, 1 or $\perp$ (for abort). The two players, together, can output $\{00, 11, 01, 10, *\perp, \perp *\}$ where $*$ is either 0 or 1. Whenever the outcomes differ, the players agree to abort the protocol.
  - *Both Honest.* If the outputs are 00 (or 11), Alice and Bob both conclude that Alice (or Bob) won. These two outputs, 00 and 11, appear with probability $\frac{1}{2}$.
  - *(Standard SCF) Security.* Suppose Alice is cheating and Bob is honest. Then the maximum (over all possible cheating strategies of Alice) probability that Bob outputs x is given by p_{*x}. Analogously, when Bob is cheating and Alice is honest, then the maximum (over all possible cheating strategies of Bob) probability that Alice outputs y is given by p_{y*}.
  (E.g. p_{*x} indicates the probability that the protocol ends with both players agreeing to the outcome x because Alice can convince Bob of the outcome x and she, being the cheating player, can simply lie about her outcome being x).

Remark 8. A (standard) SCF protocol with security p_{*x}, p_{y*} can be converted into a (standard) WCF protocol with $P_A^* = p_{*0}$ and $P_B^* = p_{1*}$, by mapping 0 to Alice winning, 1 to Bob winning and $\perp$ to the honest player winning.

Lemma 9 (Single Suppression)**.** *Consider a SCF protocol (see Definition 7) with cheating probabilities (p_{*x}, p_{y*}) for both x and y in $\{0,1\}$ (assume the honest probabilities are $\frac{1}{2}$ and $\frac{1}{2}$). Suppose that $p_{*0} = p_{*1} > p_{0*} = p_{1*}$, i.e. the SCF protocol is unbalanced. Consider a* WCF with cheating abort *(see Definition 4) and the* cheat vector sets *$\mathcal{A}_c$ and $\mathcal{B}_c$. Using these, one can compose them to construct a SCF protocol with*

$$p'_{*x} = \max_{(\alpha_A,\alpha_B,\alpha_\perp)\in\mathcal{A}_c} \alpha_A p_{*x} + \alpha_B p_{x*}$$

and

$$p'_{y*} = \max_{(\beta_A,\beta_B,\beta_\perp)\in\mathcal{B}_c} \beta_A p_{y*} + \beta_B p_{*y}.$$

Further, $p'_{*x} \leq P_A^* p_{*x} + (1 - P_A^*) p_{x*}$ and $p'_{y*} \leq P_B^* p_{*y} + (1 - P_B^*) p_{y*}$, i.e. the new SCF protocol necessarily performs at least as good as the one obtained using a standard composition technique (which neglects aborts in WCF).

Proof. First, we construct a SCF protocol using the standard composition technique of a (standard) WCF protocol and a (standard) SCF protocol as in Figure 1. Suppose Alice cheats (Bob is honest). For the WCF protocol, she will try to bias towards A because this increases her chance of running the SCF protocol where she has the advantage (we assume $p_{*x} > p_{y*}$). Her success probability (i.e. of convincing Bob that the output is x), is thus, $p''_{*x} := P_A^* p_{*x} + (1 - P_A^*) p_{x*}$ where we used $\bar{p}_{*x} = p_{x*}$. Suppose Bob cheats (Alice is honest). Bob, similarly, would bias towards B to gain an advantage. His success probability (i.e. of convincing Alice that the output is y), is thus, $p''_{y*} := P_B^* p_{*y} + (1 - P_B^*) p_{y*}$ where we used $\bar{p}_{y*} = p_{*y}$.

We now compose a WCF protocol (with cheat abort) with a (standard) SCF protocol as in Figure 2. Suppose Alice cheats (Bob is honest). Further, suppose she uses some strategy corresponding to the cheat vector $(\alpha_A, \alpha_B, \alpha_\perp) \in \mathcal{A}_c$. Then, her success probability (i.e. of convincing Bob that the output is x) is $p'_{*x} = \alpha_A p_{*x} + \alpha_B p_{x*}$. The corresponding expression of success probability for a cheating Bob is $p'_{y*} = \beta_B p_{*y} + \beta_A p_{y*}$.

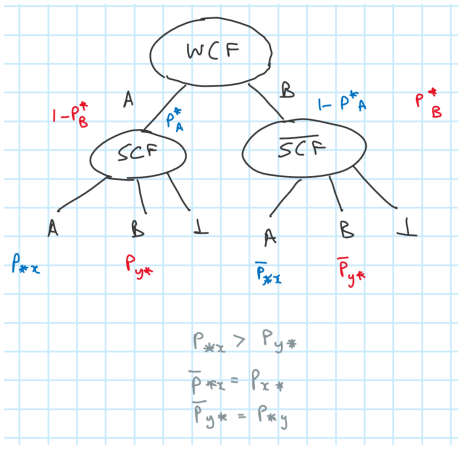

Figure 1: Standard composition of a WCF protocol and unbalanced SCF protocol produces a more balanced SCF protocol.

It remains to establish that $p'_{*x} \leq p''_{*x}$ and that $p'_{y*} \leq p''_{*y}$. To this end, from Remark 6 we know that $P^*_A = \max_{(\alpha_A,\alpha_B,\alpha_\perp)\in\mathcal{A}_c} \alpha_A$ and similarly $P^*_B = \max_{(\beta_A,\beta_B,\beta_\perp)\in\mathcal{B}_c} \beta_B$. We therefore have

$$\begin{aligned} p''_{*x} = P^*_A p_{*x} + (1-P^*_A)p_{x*} &= \left(P^*_A - (P^*_A - \alpha_A) + (P^*_A - \alpha_A)\right) p_{*x} + (1-P^*_A)p_{x*} \\ &\geq \alpha_A p_{*x} + \left((P^*_A - \alpha_A) + (1-P^*_A)\right) p_{x*} && \because p_{*x} \geq p_{x*} \\ &= \alpha_A p_{*x} + (1-\alpha_A)p_{*x} \\ &\geq \alpha_A p_{*x} + \alpha_B p_{*x} = p'_{*x}. \end{aligned}$$

□

4 Sikora's DI coin flipping protocols

Remark 10. TODO: Explain the standard box exchange description. | something like this: Alice and Bob shield each box so that it can't communicate with any other box.

Algorithm 11 (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.*

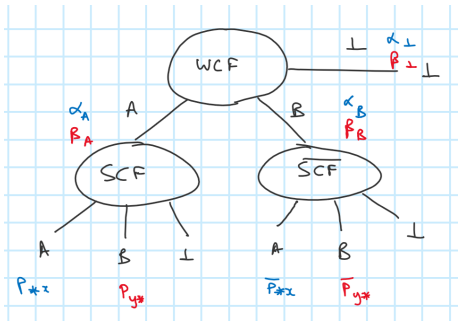

Figure 2: Composition of a WCF protocol with abort and an unbalanced SCF protocol produces a (possibly better and) more balanced SCF protocol.

Lemma 12 (Security of SCF). *(Silman et al. 2011) For Algorithm 11, $P_B^* \leq \frac{3}{4}$ 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.*

We now consider three different WCF protocols derived from this SCF protocol which should be at least as secure as simply using the SCF protocol as a WCF protocol. We use a combination of two ideas and systematically try to see which works. The first idea, *Extra Test*: test Bob as well. The second idea: use the *rigidity* of GHZ; enforce, for instance, that when Alice is honest, the state in the boxes is GHZ and that Alice's box will measure along σ_x/σ_y. Our first objective is to see if the bias is decreased by any combination of these and the second is to find the (minimal elements of the) set of cheat vectors (see Table 1).

If the bias does indeed reduce, then using the standard composition technique, we should also obtain a better SCF protocol. With some more work, if the cheating vectors can be characterised, then one should be able to use Sikora's composition technique to suppress the bias further. If the bias is not reduced, then the latter might still yield improvements.

4.1 Bidirectional rigidity

The following may seem reasonable. Suppose there are N boxes. Alice requests Bob to send αN of these (where α is say 0.5), chosen at random, for testing. Of the $N' = (1-\alpha)N$ boxes left, Bob requests Alice to send all but one of these to be tested. The has a CHSH state and measurements. This conclusion is false (at least for games that only succeed probabilistically, i.e. there is a finite probability of losing the game even with the ideal boxes).

The conclusion holds if Alice is cheating. Intuitively, this is because the first stage becomes irrelevant and in the second stage, Bob randomly asks for all but 1 box which doesn't leave Alice with much room for deviating from CHSH in the large N' limit. However, Bob can almost always get away with using an arbitrary box. Intuitively, his strategy is the following. Suppose αN is large. Bob constructs N CHSH boxes. He replaces a β fraction (where β is close to zero) of these boxes with his cheating boxes so that the likelihood of having at least one box in a randomly picked $(1-\alpha)N$ sized subset of N is close to 1. He sends these to Alice. This way, when Alice tests him, he passes with a high probability. However, when it is his turn, with high probability, he has a cheating box still remaining. He requests Alice to send all the other boxes so that the only remaining box is the cheating box.

The reasoning above works if the test was like a CHSH test where it is hard to claim a violation if a small finite number of boxes diverge from the CHSH behaviour. However, for the GHZ case, one has to be more careful in describing Bob's cheating strategy and it is not a priori clear that the same strategy works. Suppose Bob has exactly one cheating box, in the N boxes which he sends to Alice. Alice would catch him with probability α, because she can randomly ask for any αN of the boxes and the probability that the cheating box is among those is α. So with probability $1-\alpha$, Alice doesn't ask for the cheating box to be tested and in that case, Bob can simply eliminate all the other boxes. So he can succeed with probability $1-\alpha$ using this simple strategy. Clearly, by choosing α sufficiently close to 1, the protocol can be made secure against such attacks.

Consequences. Being able to bidirectionally test GHZ doesn't seem to lead to any obvious contradiction [EDIT: It does! It leads to CF; perhaps other states don't]. However, if one could make bidirectional rigidity work in general, it would mean that a CHSH state could be shared and this in turn would mean that SCF can be done perfectly which we know is impossible (Kitaev's proof).

Speculation. Could it be that the states which don't correspond to perfectly winnable games, could in fact be tested bidirectionally and could be used to give CF protocols but never with perfect bias. Perhaps one way of proceeding is just to bound the best bias possible, given a certain state is shared (and only the honest player makes fixed measurements on it, followed by a classical protocol) and then showing what class of states can be bidirectionally self-tested. Perhaps bidirectional self-testing could be studied independently and then used as a subroutine in this. *Question.* Can one give a general result which relates bidirectional self-testing with the condition that a CF protocol has the same performance in both the device dependent and device independent case?

4.2 ET, no Rigidity

Algorithm 13 (WCF (ET, no rigidity)). *Alice has one box and Bob has two boxes (in the security analysis, we let the cheating player distribute the boxes). All the boxes take one input and give one 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 $g \in_R \{0,1\}$ and sends it to Alice.*
3. *Alice sends x to Bob. They both compute the output $x \oplus g$.*

	No ET	ET
No Rigidity	$P_B^* \leq \frac{3}{4}$, $P_A^* \leq \cos^2(\pi/8)$	Jamie: $P_B^* \leq \frac{3}{4}$, $P_A^* \leq \cos^2(\pi/8)$; cheat vectors?
Bob tests Rigidity	$P_B^* \leq \frac{3}{4}$, $P_A^* \leq \cos^2(\pi/8)$; cheat vectors may be easier to evaluate	?
Alice tests Rigidity	$P_B^* \leq \frac{3}{4}$, $P_A^* \leq \cos^2(\pi/8)$; cheat vectors may be easier to evaluate	?

Table 1: Variants of the various WCF protocols.

4. *Test Rounds.*

   (a) *If $x \oplus g = 0$ (i.e. the outcome for Alice winning) Alice sends a to Bob (to convince him that she did not cheat).*

      i. *Bob checks if $s = a$ or $s = a \oplus x$. He aborts if the test fails.*

      ii. *Bob chooses $y, z \in_R \{0, 1\}$ such that $x \oplus y \oplus z = 1$ and then performs a GHZ using x, y, z as the inputs and a, b, c as the output from the three boxes (he measures his two boxes to obtain the two outcomes). He aborts if the test fails.*

   (b) *If $x \oplus g = 1$ (i.e. the outcome for Bob winning) then Bob sends his boxes (Alice tries to ensure he didn't tamper with the boxes).*

      i. *Alice chooses $y, z \in_R \{0, 1\}$ such that $a \oplus b \oplus c = 1$ and then performs a GHZ (like Bob did in the previous case). She aborts if the test fails.*

   (b') *Alice sends $y, z \in_R \{0, 1\}$ to Bob and Bob returns (b, c). Alice tests GHZ and aborts if it fails.*

Remark 14. Recall that the GHZ test, is that $a \oplus b \oplus c = xyz \oplus 1$, given the inputs satisfy $x \oplus y \oplus z = 1$.

4.2.1 Observations

Clearly, $P_A^* \leq \cos^2(\pi/8)$ carries over from 12. As for Bob, note that he is only tested if he correctly guesses Alice's input, x. We consider three special cases.

New boxes just for passing the test. Suppose he already used his boxes to guess Alice's input from s. We give a possible strategy for him whereby he sends new boxes to Alice.

When $x = 0$, he knows Alice's output, $a = s$, as well. He also knows that Alice will either input $(y, z) = (1, 0)$ or $(y, z) = (0, 1)$ to satisfy $x \oplus y \oplus z = 1$. He can therefore simply send deterministic boxes that satisfy $a \oplus b \oplus c = 1$ regardless of the input.

When $x = 1$, he knows Alice's output a was either s or $s \oplus x = s \oplus 1$ with equal probability. He assumes Alice's output is s (and is thus right half the times). Suppose he is right. Then, he can create deterministic boxes as follows. When the input is $(y, z) = (0, 0)$ the boxes output $(1, 1)$ if $a = 1$ and $(1, 0)$ if $a = 0$, so that $a \oplus b \oplus c = xyz \oplus 1 = 1$. When the input $(y, z) = (1, 1)$ the boxes output $(1, 0)$ if $a = 0$ and $(0, 0)$ if $a = 1$, so that $a \oplus b \oplus c = xyz \oplus 1 = 0$. If he is wrong, worst case, he gets caught.

This strategy gives him at least a $3/4$ probability of passing the test, supposing he won in the first place.

Optimal Cheating Strategy when Bob wasn't tested. We now analyse Bob's optimal cheating strategy from the case where he wasn't tested. This uses GHZ boxes. Bob begins by assuming that $s = a$ and he is right $3/4$ of the times (assuming Alice is honest; as we discussed above). He measures his boxes with $(y, z) = (1, 0)$. If $a \oplus b \oplus c = 1$ he guesses that $x = 0$, otherwise he guesses $x = 1$. This makes sense because if Alice inputted 0, the GHZ test passes with certainty (see Table 2). However, if Alice inputted 1, $a \oplus b \oplus c$ might still equal 1 but with only $1/2$ probability (I skipped the calculation but seems right). So, overall then, he has a probability of $3/4$ for correctly guessing Alice's input. This was already known.

Assuming he does guess correctly, can he pass Alice's test with certainty? With all the information he has, it might not not seem too hard. We know that either (x, y, z) was $(0, 1, 0)$ or it was $(1, 1, 0)$ with equal probability. Since Bob is getting tested, he guessed x correctly. If $x = 0$, then from Table 2, it is clear that Alice would input $(y, z) = (1, 0)$ or $(0, 1)$ but regardless, the outputs (a, b, c) satisfy the GHZ test. He can thus send these boxes (assuming they show the same output regardless of the inputs) and pass the test with certainty. Now, if $x = 1$ and he guessed correctly, it must mean that his outputs did not satisfy the GHZ test. However, since he has no information about Alice's outcome (he received $s = a \oplus r$) and as he has already measured his boxes along σ_y and σ_x (corresponding to $(y, z) = (1, 0)$), it seems that he can't succeed in passing the test with probability more than $1/2$. Thus, his overall probability of passing the test seems to be $3/4$, assuming he guessed correctly in the first place.

New boxes for guessing and passing the test. This seems even easier than the optimal strategy that uses GHZ. Since Bob distributes the boxes, he programs Alice's box to always output $a = 0$. Bob always guesses that $x = s$. If $x = 0$ then he is indeed correct because $s = a = 0$. If $x = 1$ then $s = a \oplus r = r$ so his guess is right half the times. Overall, he guesses correctly $3/4$ of the times. How can he pass Alice's test? Since Bob is tested when he guesses correctly, he knows x and that $a = 0$ by construction.

- Suppose $x = 1$.
  - Note that Alice will either input $y = z = 0$ or $y = z = 1$ to satisfy $x \oplus y \oplus z = 1$.
  - Bob has to ensure, then, that $b \oplus c = 1$ when $y = z = 0$ and $b \oplus c = 0$ when $y = z = 1$.
  - He sets $c = 1$ (the last box always outputs one). He sets $b = y$.
    Or equivalently, he could set $c = 0$ and $b = \bar{y}$.
- Suppose $x = 0$.
  - He creates deterministic boxes satisfying $b \oplus c = 1$ regardless of the input.

Clearly, he passes Alice's test with certainty.

Conclusion. Evidently, Bob can cheat optimally and pass Alice's test by using the strategy described above (optimal because for a possibly weaker protocol, $P_B^* \leq 3/4$ and we found a strategy which saturates the bound in an a priori stronger protocol). However, if Bob is forced to use a GHZ state (which can be enforced by rigidity) then one might be able to lower P_B^*.

x	y	z	$xyz \oplus 1$
1	1	1	0
1	0	0	1
0	1	0	1
0	0	1	1

Table 2: The values for which $x \oplus y \oplus z = 1$, $a \oplus b \oplus c$ must equal the last column for a GHZ test to pass.

4.2.2 Security Analysis

We defer the complete security analysis for this case to the end. This is because, as suggested by the discussion above, self-testing Bob and performing the extra test, might yield a smaller P_B^* and then one can construct a better protocol by using the standard composition technique. This saves us the effort of finding the cheat vectors (assuming of course, it works).

Lemma 15 (Security of WCF)**.** *Standard Security: Need to do Bob's part again; Alice's should carry over. Need to do both Alice and Bob to characterise the sets $\mathcal{A}_c$ and $\mathcal{B}_c$ in addition to the usual things.*

4.3 ET and Alice tests Rigidity (i.e. Bob is self-tested)

Algorithm 16 (WCF (ET, Alice tests rigidity))**.** *Before the protocol begins, Bob distributes N boxes and Alice does a rigidity test on $N-1$ randomly chosen boxes. If any test fails, Alice aborts. They then proceed as in Algorithm 13.*

4.3.1 Using SDPs

The analysis for a malicious Alice stays unchanged. For a malicious Bob, however, the primary simplification is that we can assume the state in the boxes is $|\psi\rangle = \frac{|000\rangle+|111\rangle}{\sqrt{2}}$ and that Alice measures along σ_z/σ_y. We simulate the classical communication using noiseless quantum communication (TODO: discuss these in more detail later). We suppose that the initial state is

$$|\psi_1\rangle := |\psi\rangle_{ABC} \left(\frac{|001\rangle|01\rangle + |010\rangle|10\rangle + |100\rangle|00\rangle + |111\rangle|11\rangle}{\sqrt{4}} \right)_{X'Y'Z'Y''Z''} |0\rangle_{A'} |00\rangle_{S'S''} \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right)_{R'} \tag{1}$$

where the distribution of the systems/registers, initially, is as follows. Alice possesses register A and Bob possesses the registers BC. All the remaining registers are assumed to be with Alice. The primes are used to indicate that they correspond to classical communication (and randomness). In the SDP formulation, then, we note that

$$\rho_1 = \mathrm{tr}_{BC} |\psi_1\rangle \langle\psi_1|$$

(strictly speaking, it should really by $\rho_1 = \mathrm{tr}_{BC}(|\psi_1\rangle \langle\psi_1|)$ because Alice doesn't have the systems B and C but because of the preceding self-testing step, she knows what the state must be). According to the protocol, since Alice is honest (remember, we are analysing for a malicious Bob), she is supposed

- to choose a random number x and input this into the box to obtain the outcome a.
- She then sends $s = a \oplus x \cdot r$ to Bob.

Since, later, she also has to choose y, z consistent with $x \oplus y \oplus z = 1$, without loss of generality, we assume she already chose these from the beginning. Further, we assume she chose them quantumly (because if we start using mixed states to simulate classical randomness, then the SDP formalism allows Bob to hold a purification which yields a trivial bias). We thus write the random choice as the state present in the $X'Y'Z'$ registers (one could imagine the register being $X'Y'Z'X''Y''Z''$ and the state as $|001\rangle|001\rangle + |010\rangle|010\rangle \ldots$ so that tracing out one part yields the classical mixture we want while also ensuring Bob doesn't have access to it; this increases the dimension of the problem and so to keep it low, we only retain $X'Y'Z'$; TODO: rewrite this part because now I am using Y', Z' but this only changes the notation, the idea remains the same). We simulate the measurement using a unitary U_1 (the details constitute the next section) which does the following two things: First it uses X' as control and encodes the outcome of the measurement in A'. Then, it uses X', R' and A' as controls and encodes the output of the function $s = a \oplus x \cdot r$ into $S'S''$, i.e. it stores the same value in both S' and S'' as $|ss\rangle_{S'S''}$. The state at the end of this is denoted by

$$\rho_2 = U_1 \rho_1 U_1^\dagger.$$

At this point, Alice sends the register S' to Bob and Bob sends the register G' (which contains the "guess" if he is cheating, a random bit if he is honest). This is because the protocol at this point is that

- Bob sends a random number g to Alice.

The most general strategy Bob can play, at this point, is to give any arbitrary state ρ_3 over the registers $AA'X'Y'Z'Y''Z''S''G'$, constrained only by the fact that he can't touch Alice's laboratory, i.e.

$$\rho_3, \text{ s.t. } \mathrm{tr}_{S'}(\rho_2) = \mathrm{tr}_{G'}(\rho_3).$$

Since we are interested in Bob's probability of success, we note

- if $x \oplus g = 0$ then Alice sends y, z to Bob and he responds with b'', c''. Alice accepts that Bob won if the GHZ test succeeds with x, y, z as the input and a', b', c' as the outputs.

Thus, Alice sends Y' and Z' to Bob while Bob returns registers B' and C' with his responses. Again, this means he sends an arbitrary state ρ_4 over the registers $AA'B'C'X'Y''Z''S''G'$, constrained only by the fact that he, again, can't influence Alice's laboratory, i.e.

$$\rho_4, \text{ s.t. } \mathrm{tr}_{Y'Z'}(\rho_3) = \mathrm{tr}_{B'C'}(\rho_4)$$

and Alice declares Bob a winner with probability $\mathrm{tr}(\Pi\rho_4)$, where Π just represents the winning condition, namely $x \oplus g = 1$ (1 is Bob wins, 0 is Alice wins, by convention) and that $a \oplus b \oplus c = xyz \oplus 1$. We discuss this in detail, soon enough. In summary then, we have

$$\max \mathrm{tr}(\Pi\rho_4)$$

subject to

$$\begin{aligned}
\mathrm{tr}_{B'C'}(\rho_4) &= \mathrm{tr}_{Y'Z'}(\rho_3), \\
\mathrm{tr}_{G'}(\rho_3) &= \mathrm{tr}_{S'}(\rho_2), \\
\rho_2 &= U_1\rho_1 U_1^\dagger \\
\rho_1 &= \mathrm{tr}_{BC}(|\psi_1\rangle\langle\psi_1|)
\end{aligned}$$

where $|\psi_1\rangle$ is as defined in Equation (1), U_1 is as defined in Equation (2) and Π is as defined in Equation (3). For clarity,

$$\begin{aligned}
\rho_1, \rho_2 &\in \mathrm{Pos}(\mathcal{H}_{AX'Y'Z'Y''Z''A'S'S''R'}) \\
\rho_3 &\in \mathrm{Pos}(\mathcal{H}_{AX'Y'Z'Y''Z''A'S''R'G'}) \\
\rho_4 &\in \mathrm{Pos}(\mathcal{H}_{AX'Y''Z''A'S''R'B'C'}).
\end{aligned}$$

Clearly, we can simplify this further. We have

$$\max \mathrm{tr}(\Pi\sigma_2)$$

subject to

$$\begin{aligned}
\mathrm{tr}_{B'C'}(\sigma_2) &= \mathrm{tr}_{Y'Z'}(\sigma_1) \\
\mathrm{tr}_{G'}(\sigma_1) &= \mathrm{tr}_{S'BC}(U_1|\psi_1\rangle\langle\psi_1|U_1^\dagger)
\end{aligned}$$

where

$$\begin{aligned}
\sigma_1 &\in \mathrm{Pos}(\mathcal{H}_{AX'Y'Z'Y''Z''A'S''R'G'}) \\
\sigma_2 &\in \mathrm{Pos}(\mathcal{H}_{AX'Y''Z''A'S''R'B'C'}).
\end{aligned}$$

4.3.2 The Unitary and the measurement

We write

$$U_1 = U_{II}U_I \tag{2}$$

where U_I acts on AA' with X' as control while U_{II} acts on SS' with $X'R'A'$ as control. We define

$$\begin{aligned}
U_I := \mathbb{I}_{\text{rest}} \otimes \Bigg(&|0\rangle\langle 0|_{X'} \otimes \left(\Pi_{|0\rangle_A} \otimes \mathbb{I}_{A'} + \Pi_{|1\rangle_A} \otimes \hat{X}_{A'}\right) \\
&+ |1\rangle\langle 1|_{X'} \otimes \left(\Pi_{\frac{|0\rangle+i|1\rangle}{\sqrt{2}}A} \otimes \mathbb{I}_{A'} + \Pi_{\frac{|0\rangle-i|1\rangle}{\sqrt{2}}A} \otimes \hat{X}_{A'}\right)\Bigg)
\end{aligned}$$

and

$$U_{II} := \mathbb{I}_{\text{rest}} \otimes \sum_{a,x,r \in \{0,1\}} \hat{X}_S^{a \oplus x.r} \otimes \hat{X}_{S'}^{a \oplus x.r} \otimes |axr\rangle\langle axr|_{A'X'R'}.$$

This brings us to the final measurement. We have

$$\Pi := \sum_{a,b,c,x,y,z \in S_{\text{GHZ}}} |abcxyzg\rangle\langle abcxyzg|_{A'B'C'X'Y''Z''G'} \tag{3}$$

where $S_{\text{GHZ}} = \{(a, b, c, x, y, z, g) : x \oplus g = 1 \text{ and } a \oplus b \oplus c = xyz \oplus 1\}$.

5 Acknowledgements

This research was supported in part by Perimeter Institute for Theoretical Physics. Research at Perimeter Institute is supported by the Government of Canada through the Department of Innovation, Science and Economic Development and by the Province of Ontario through the Ministry of Research, Innovation and Science.

You further agree that following your visit, you will keep PI apprised of any manuscripts related to your visit by writing to pipublications@perimeterinstitute.ca.

References

Silman, J. et al. (June 2011). "Fully Distrustful Quantum Bit Commitment and Coin Flipping". In: *Physical Review Letters* 106.22. DOI: 10.1103/physrevlett.106.220501.