Fairness in two-party random sampling

Margus Niitsoo

May 12, 2008

Abstract

This report generalizes the results given by Cleve (1986) that bound the fairness achievable in a multyparty coin tossing protocol. By using essentially the same proof, we show similar bounds for a biased coin and then extend the result to sampling from an arbitrary distribution. We also show bounds for two-party computation of a function but do so only in the average case.

1 Introduction

Suppose Alice and Bob want to toss a fair coin and agree on what the outcome was. Suppose further that Alice and Bob live on different continents and cannot meet face to face to do so. They therefore have to agree on the value over the internet. However, neither of them particularily trusts the other and fears that the other might cheat given the opportunity. This is the problem of tossing a coin over the telephone. Using bit commitment fair protocols can be constructed. However, all of them are fair only if both parties follow the protocol all the way to the result without halting. However, this might not always be the case and in the case where one party halts, the coinflip can in fact be biased.

Cleve [1] showed that at least one of the parties can bias the coin rather significantly and this result can be weakened only by increasing the number of rounds of communication. We generalize his result to the case of any distribution and to give the proof in more modern terms.

2 The ideal model

Suppose we have a protocol for two parties Alice and Bob that should result in both of them agreeing on a value chosen from some distribution $\mathcal{D}$ if both of them are honest. Essentially, we want our protocol to guarantee that after both parties have agreed to start the protocol, the exchange of messages will always lead to both parties returning the same value that is sampled from the required distribution $\mathcal{D}$. If we have a trusted third party we can use the model presented on Figure 1.

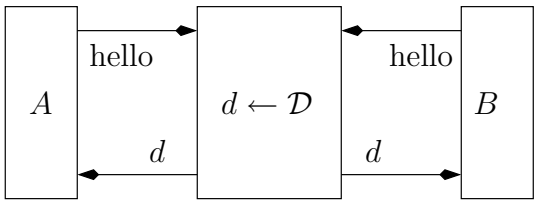

Figure 1: The ideal-world functionality

The protocol consists of the two parties, A and B interested in sampling a value and a third party trusted by both of them. A and B both send a greeting message to the third party informing that they want to begin a protocol. The third party then samples from the desired distribution and sends the value back to both A and B. The only malicious behaviour allowed for both parties is not to partake in the protocol (by not sending the first hello) and in this case neither gets the output value d. Thus either neither of them or both of them get the d.

3 The real world

However, we know that in reality a trusted and objective third party is nearly impossible to find. Quoting a passage from a Chinese philosopher Zhuangzi:

> Whom shall I ask as arbiter between us? If I ask someone who takes your view, he will side with you. How can such a one arbitrate between us? If I ask someone who takes my view, he will side with me. How can such a one arbitrate between us? If I ask someone who differs from both of us, he will be equally unable to decide between us, since he differs from both of us. And if I ask someone who agrees with both of us, he will be equally unable to decide between us, since he agrees with both of us.

Therefore, we want Alice and Bob to be able to agree on the value chosen without any outside help. To do that, we assume that they exchange messages such that first Alice sends one to Bob, then Bob sends one back and so on. The protocol structure is depicted on Figure 2.

Essentially, the protocol consists of r rounds. Each round begins with A

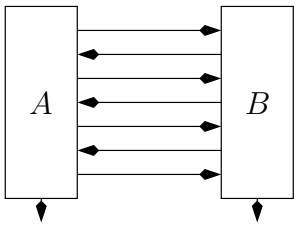

Figure 2: The real-world functionality

sending a message to B and ends with B sending a message back to A. Thus, a total of r messages are sent in both direction during the whole protocol. We assume that both A and B are allowed to do computations between recieving a message sent to them and sending one back again. In the end of the protocol both of them are expected to output a value – the sample from $\mathcal{D}$ they thought they agreed on. We want that value to be the same for both parties if the protocol was followed.

4 Problem statement

We are interested in how closely we can approximate the ideal world functionality with the model we have in the real world. For that we use the notion of *relative resilience* introduced by Beaver [2] and extended by Goldreich [3]. To prove that the real world model can emulate the functionality of the ideal world model, we have to be able to simulate the ideal world behaviour of the trusted third party given the real world behaviour of A and B. However, unlike in the standard problem statement, we are interested in showing that this simulation cannot be too good or that there is a lower bound on the simulation of the ideal world model with any real world model of the type described above.

5 Bounds on fairness

The original proof of Cleve was about protocols tossing a fair coin. He assumed that both parties do not always get the same result in the end even if the protocol is properly followed. We also introduce the probability q of the parties agreeing on the answer if the protocol is followed without halting. The theorem as applied to the unfair cointoss protocol can thus be stated as follows:

Theorem 1. *Let π be a biased cointoss protocol with r rounds for two parties where the coin lands 1 with probability $p \geq 0.5$. Assume that the protocol terminates with both parties agreeing on the same value with probability q. Then there always exists a real world adversary that can bias the result by at least $\frac{q-p}{4r+1}$.*

Proof. Assume that the protocol π is between parties A and B and that the messages exchanged within the protocol are all bitstrings. In reality each party has the possibility to quit at each round – he can just stop sending out the messages required. We can model this behaviour as sending out $\perp$ values instead of reasonable messages. If one of the parties quits at round i then the other party still has to produce some kind of an output which we call the *default* output. Let a_i be the default output of A if B quits at round i and let a_{r+1} denote the output if the protocol is followed to the end. Similarly, let b_i be the output of B if A quits in the round $i+1$ and let b_r denote the output in the case that protocol is followed to the end (see Figure 3). The idea of the proof is to construct a number of different adversaries and to prove that at least one of them will bias the selection process quite considerably.

There are five types of adversaries. Their algorithms are given on Figure 4. In total we therefore have $4r + 1$ different adversaries. Essentially all of them play the honest party for a set number of rounds and then quit at a certain round, usually depending on what the output would be if the adversary would quit instead.

We now try to estimate how much these adversaries influence the computations of the other party which is assumed to remain honest.

We say that an adversary *biases* the output of the honest party by ϵ towards either 0 or 1 if the probability of the honest party outputing that value while playing with that adversary is at least ϵ more than it should be according to the ideal biased cointoss being performed.

We now see how much each of the adversaries biases the output. The simplest case is A_0^0 for which the output is always b_0 since A_0^0 quits at round

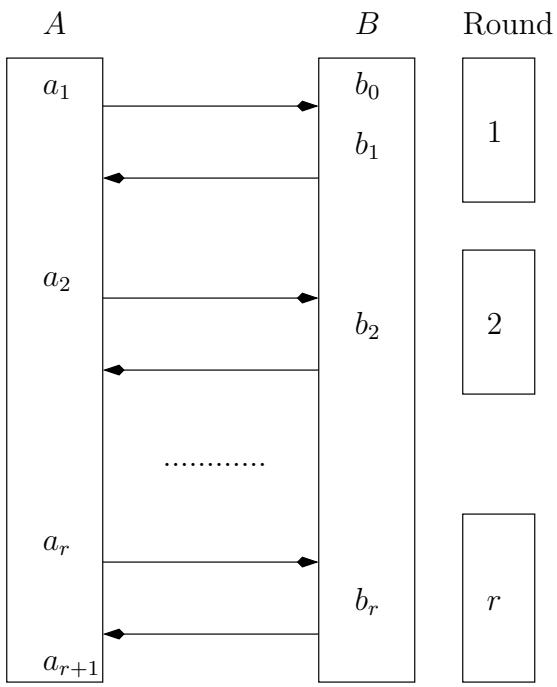

Figure 3: Schematic for default outputs

1. The bias towards 1 is therefore

$$\Pr[b_0 = 1] - p$$

since the cointoss protocol should return 1 with probability p but in the case of this adversary it returns b_0. Analogously the bias towards 0 is

$$\Pr[b_0 = 0] - (1 - p) = p - \Pr[b_0 = 1].$$

We can choose the larger of the two to get $|\Pr[b_0 = 0] - p|$.

For A_i^0, consider the bias towards the answer 0. In the ideal protocol the output should be 0 with probability $1 - p$. In the case of this adversary, however, there are two possibilities. If $a_i = 0$ then A_i^0 quits at round $i + 1$ so the output of the protocol for honest B is b_i. Otherwise (when $a_i = 1$) A_i^0 quits at round i so B returns b_{i-1}. Therefore, B returns 0 precisely when either $a_i = 0$ and $b_i = 0$ or $a_i = 1$ and $b_{i-1} = 0$. The bias is therefore

$$\Pr[a_i = 0 \wedge b_i = 0] + \Pr[a_i = 1 \wedge b_{i-1} = 0] - (1 - p).$$

The total bias which is the maximum of biases towards both 0 and 1, is therefore at least this big.

Analogously, the adversary A_i^1 biases the output of B by at least

$$\Pr[a_i = 1 \wedge b_i = 1] + \Pr[a_i = 0 \wedge b_{i-1} = 1] - p.$$

For the adversaries controlling B the biases they cause are similarly at least

$$\Pr[b_i = 0 \wedge a_{i+1} = 0] + \Pr[b_i = 1 \wedge a_i = 0] - (1 - p)$$

for B_i^0 and at least

$$\Pr[b_i = 1 \wedge a_{i+1} = 1] + \Pr[b_i = 0 \wedge a_i = 1] - p$$

for B_i^1.

Then the average of all the biases Δ can be bounded from below by

$$\begin{aligned}
\Delta \;\geq\; & \frac{1}{4r+1}\Big[|\Pr[b_0=1]-p| + \\
& + \sum_{i=1}^{r}\Big(\Pr[a_i=0\wedge b_i=0]+\Pr[a_i=1\wedge b_{i-1}=0]-(1-p)+ \\
& + \Pr[a_i=1\wedge b_i=1]+\Pr[a_i=0\wedge b_{i-1}=1]-p+ \\
& + \Pr[b_i=0\wedge a_{i+1}=0]+\Pr[b_i=1\wedge a_i=0]-(1-p)+ \\
& + \Pr[b_i=1\wedge a_{i+1}=1]+\Pr[b_i=0\wedge a_i=1]-p\Big)\Big] = \\
= \;& \frac{1}{4r+1}\Big[|\Pr[b_0=1]-p|-2r + \\
& + \sum_{i=1}^{r}\Big(\Pr[a_i=0\wedge b_i=0]+\Pr[a_i=1\wedge b_{i-1}=0]+ \\
& + \Pr[a_i=1\wedge b_i=1]+\Pr[a_i=0\wedge b_{i-1}=1]+ \\
& + \Pr[b_i=0\wedge a_{i+1}=0]+\Pr[b_i=1\wedge a_i=0]+ \\
& + \Pr[b_i=1\wedge a_{i+1}=1]+\Pr[b_i=0\wedge a_i=1]\Big)\Big] = \\
= \;& \frac{1}{4r+1}\Big[|\Pr[b_0=1]-p|-r + \\
& + \sum_{i=1}^{r}\Big(\Pr[a_i=1\wedge b_{i-1}=0]+\Pr[a_i=0\wedge b_{i-1}=1]+ \\
& + \Pr[b_i=0\wedge a_{i+1}=0]+\Pr[b_i=1\wedge a_{i+1}=1]\Big)\Big] = \\
= \;& \frac{1}{4r+1}\Big[|\Pr[b_0=1]-p|-r+\Pr[a_1=1\wedge b_0=0]+\Pr[a_1=0\wedge b_0=1]+ \\
& + \Pr[b_r=0\wedge a_{r+1}=0]+\Pr[b_r=1\wedge a_{r+1}=1]+ \\
& + \sum_{i=1}^{r-1}\Big(\Pr[a_{i+1}=1\wedge b_i=0]+\Pr[a_{i+1}=0\wedge b_i=1]+ \\
& + \Pr[b_i=0\wedge a_{i+1}=0]+\Pr[b_i=1\wedge a_{i+1}=1]\Big)\Big] = \\
= \;& \frac{1}{4r+1}\Big[|\Pr[b_0=1]-p|-1+\Pr[a_1=1\wedge b_0=0]+\Pr[a_1=0\wedge b_0=1]+ \\
& + \Pr[b_r=0\wedge a_{r+1}=0]+\Pr[b_r=1\wedge a_{r+1}=1]\Big].
\end{aligned}$$

We telescoped the sum by noting that $-p-(1-p)=-1$ and

$$\begin{aligned}
&\Pr[a_i=1\wedge b_i=1]+\Pr[a_i=0\wedge b_i=1]+ \\
&+\Pr[a_i=1\wedge b_i=0]+\Pr[a_i=0\wedge b_i=0] \;=\; 1
\end{aligned}$$

After also seeing that

$$\begin{aligned} |\Pr[b_0 = 1] - p| &= \max\{\Pr[b_0 = 1] - p, p - \Pr[b_0 = 1]\} = \\ &= \max\{\Pr[b_0 = 1] - p, -(1 - \Pr[b_0 = 0]) + p)\} = \\ &= \max\{\Pr[b_0 = 1] - p, \Pr[b_0 = 0] - (1 - p))\} \geq \\ &\geq \max\{\Pr[b_0 = 1], \Pr[b_0 = 0]\} - p \end{aligned}$$

(because $p \geq 1 - p$ since $p \geq 0.5$) and that

$$\begin{aligned} & \Pr[a_1 = 1 \wedge b_0 = 0] + \Pr[a_1 = 0 \wedge b_0 = 1] = \\ = \; & 1 - (\Pr[a_1 = 1] \Pr[b_0 = 1] + \Pr[a_1 = 0] \Pr[b_0 = 0]) \geq \\ \geq \; & 1 - \max\{\Pr[b_0 = 1], \Pr[b_0 = 0]\}, \end{aligned}$$

(because maximum is always larger than the weighed average) we can simplify the inequality even further to get

$$\Delta \geq \frac{1}{4r+1}\Big[\Pr[b_r = 0 \wedge a_{r+1} = 0] + \Pr[b_r = 1 \wedge a_{r+1} = 1] - p\Big]$$

. Because we assumed that if the protocol is followed until the end then the outputs of two parties agree with a probability at least q, we finally have that

$$\Delta \geq \frac{q-p}{4r+1}.$$

Since Δ is the average of the biases, at least one of the $4r + 1$ adversaries has to do be able to achieve a bias that is at least this good. □

6 Distance between the real and ideal world models

After a little consideration it should be clear that the theorem is actually about the real and ideal world differences and that $\frac{q-p}{4r+1}$ is clearly a lower bound on the closeness of the real and ideal models for the two party coin tossing protocols. However, this bound can easily be neglected by choosing q to be equal to p. This would mean that the protocol only gives the same result on both processors with the probability of only q. This is a stark contrast to the ideal world where the two parties always recieve the same element sampled from $\mathcal{D}$ and this difference between models allows us to obtain the following bound on the difference between the two models

Corollary 2. *The statistical difference between the real and ideal world models for two party cointoss protocols is bounded from below by* $\max\{1-q, \frac{q-p}{4r-1}\}$.

Proof. Let π be any protocol for two-party unfair cointoss. Then even if both parties follow the protocol, its functionality differs from the intended ideal world by at least $1 - q$ because in these cases the two parties do not agree on the same output. We also know from the previous theorem that there exists an adversary that can bias the coin by at least $\frac{q-p}{4r-1}$. Therefore the statistical difference between real and ideal worlds is at least the maximum of the two. □

7 Sampling from an arbitrary distribution

The previous result can be generalized to the case of sampling from an arbitrary distribution $\mathcal{D}$. The trick is to note that we can use a two party protocol that samples from $\mathcal{D}$ to emulate a biased cointoss.

This can be done by choosing a subset H of $\operatorname{supp}\mathcal{D}$ and then coding the outputs of the protocol as either 0 or 1 based on wether $x \in H$ or not. If the probability that $x \leftarrow \mathcal{D}$ belongs to H is p then this protocol amounts to tossing a biased coin with the probability p of getting a 1. As we know that such protocols can be biased and have to be at least $\max\{1-q, \frac{q-p}{4r-1}\}$ distant from the ideal world, so are all the protocols sampling from $\mathcal{D}$.

In this case, however, there is usually more than one choice for H. We are interested in the one that gives the best bounds. For that we choose as H the subset for which the probability of $x \leftarrow \mathcal{D}$ such that $x \in H$ is as close to one half as possible and denote this probability as p. Since both H and its complement $\bar{H}$ are equally close, we assume that H is the one for which $p \geq 0.5$ as the theorem in the previous section is stated for such p. Formally, all this can be written out as

$$p := \min_{H \subset \operatorname{supp}\mathcal{D}} \{\Pr[x = 1H] \geq 0.5\}.$$

It is easy to see that this maximizes the bound in the theorem and thus also in the ideal-real world difference result. We can formalize the result in the following way:

Corollary 3. *Let π be a two-party protocol for sampling from a finite distribution $\mathcal{D}$. Then difference between the output of π and the ideal world model is at least $\max\{1-q, \frac{q-p}{4r-1}\}$ where p is as defined above and q is the probability of outputs of two fair parties agreeing.*

8 Two party computation of a function

The result can be generalized to the case of two-party computations of functions if we know the expected input distribution of the parties. Let π be a protocol intended for two-party computation of a function $f(a, b)$ where the first party A knows a and the second party B knows b. Assume that the inputs of A are from a distribution $\mathcal{D}_A$ and inputs of B from $\mathcal{D}_B$ and let $\mathcal{D}$ be the distribution of the outputs of $f(a, b)$ if $a \leftarrow \mathcal{D}_A$ and $b \leftarrow \mathcal{D}_B$.

Assume that the inputs for protocol π are chosen from the said distributions. We can then view the protocol as one with no inputs that samples from the distribution $\mathcal{D}$. We know from before that in this case there exists an adversary that can bias the output by $\frac{q-p}{4r-1}$. This essentially means that if we know the input distributions of both parties, we can guarantee that a bias of at least that size will occur on average.

We note that there are some cases where a bias can be proven to exist regardless of the distributions – for instance, two party computation of XOR function can be reduced to tossing a fair coin. However, there seem to be no immediate and general ways of proving that bias always exists based on the theorem presented in this paper. We therefore leave it as an open problem.

9 Conclusion

We have showed that there exists a bound on fairness of two-party protocols intended for tossing a biased coin and then generalized that result to the case for sampling from any distribution. We also explored the possibility of extending the result to cover two-party computation of a deterministic function, and although on average the same bound still holds, we leave the existance of bounds in the general case as an open problem.

References

[1] Cleve, R. Limits on the Security of Coin Flips When Half the Processors are Faulty. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pp. 364-369, 1986

[2] Beaver, D. Secure multiparty protocols and zero-knowledge proof systems tolerating a faulty minority. In *Journal of Computing*, Vol 4, pp. 75-122, 1991

[3] Goldreich, O. The Foundations of Cryptography, Vol 2, ISBN 0-521-83084-2, 2004

Algorithm 1: A_0^0

quit at round 1

Algorithm 2: A_i^0

Follow the protocol A for rounds $1, 2, \ldots, i-1$
Compute a_i
if $a_i = 0$ then follow the protocol for one more round and quit at round $i+1$.
otherwise quit at round i.

Algorithm 3: A_i^1

Follow the protocol A for rounds $1, 2, \ldots, i-1$
Compute a_i
if $a_i = 1$ then follow the protocol for one more round and quit at round $i+1$.
otherwise quit at round i.

Algorithm 4: B_i^0

Follow the protocol B for rounds $1, 2, \ldots, i-1$
Compute b_i
if $b_i = 0$ then follow the protocol for one more round and quit at round $i+1$.
otherwise quit at round i.

Algorithm 5: B_i^1

Follow the protocol B for rounds $1, 2, \ldots, i-1$
Compute b_i
if $b_i = 1$ then follow the protocol for one more round and quit at round $i+1$.
otherwise quit at round i.

Figure 4: Algorithms of adversaries