0368.4170 Cryptography and Game Theory	Ran Canetti, Alon Rosen
Lecture 5	
18 November 2009	Fall 2009

Scribe: D. Sotnikov

1. FAIRNESS

By fairness we mean the guarantee that all parties receive their outputs from the protocol "simultaneously", that is, the protocol should prevent the situation where one of the parties that gets the result of the computation or learns some information on it, will prevent the other party from learning absolutely it's output.

For example let us focus on the Blum's protocol to coin tossing problem

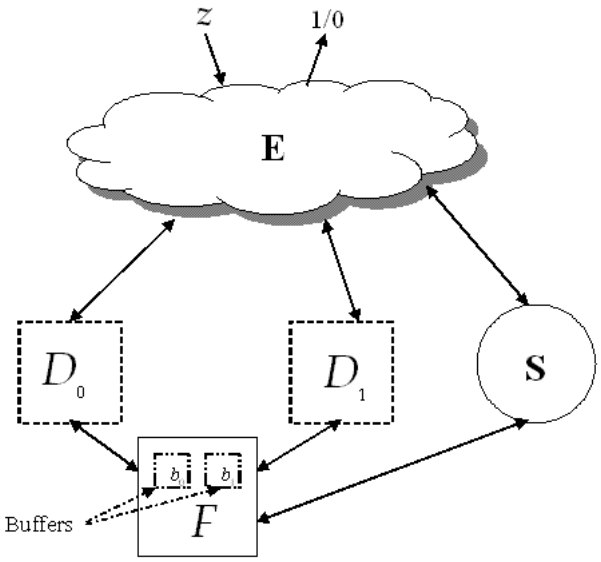
1.1. **Blum's protocol.**

- P_0 chooses randomly one bit r_0 and sends the commitment $com(r_0)$ to the P_1
- P_1 chooses randomly one bit r_1 and sends it to P_0
- P_0 opens the commitment that he sent in round 1
- Both sides output $r = r_0 \oplus r_1$

Observe that P_0 always can abort the protocol after he gets the r_1 and in this case only P_0 gets the final result of the function calculation, so P_0 can abort the protocol in the case that he didn't like the function's "output" and therefore cause the other players to output only the values that he want. (e.g. to abort if $r = 0$ in this case function will output only 1).

1.2. **Towards a definition.** We want to avoid the situation where any party can totally bias the function output, we want to build protocol that gives the output simultaneously to all the parties, we would like to simply change the definition of the ideal model, and have the trusted party give the outputs to both parties at the same time. But this does not really make sense: In an asynchronous network one can't require that both parties generate output "at the same time". And indeed the model of ITM-based computation does not allow an ITM to write to two other ITMs "at the same time" because we can't run two ITM simultaneously in parallel. So we'll have to do something else.

What we'll do is to have the trusted party F use two local tapes (buffers) to hold the outputs of the two parties and then F can write the result of the function calculation to both buffers simultaneously and each party can read from the appropriate buffer asynchronously at his tern.



To catch the new behavior we need to change the last row at the protocol definition, now S can say "go" only once.

- when S says "go" F initialize the j buffer by y_j for all $j \in [1, \#Parties]$ after that sends y_i to S, get y_i' from S and appropriates the y_i' to the buffer i.

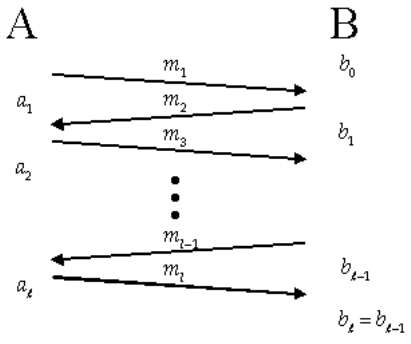
Now we can ask: Are there protocols that realize this ideal functionality? More concretely, can we build any protocol that can guaranty totally random output without any bias?

We will show a lower bound on the bias as a function of the number of rounds, for this bound we will use a weaker definition of security, which is not simulation based. The fact that we use a weaker notion of security will only strengthen the lower bound.

Theorem 1.1. ***[Cleve 86]:(informal statement)*** *For any* l-round *message passing protocol where the parties agree on a common bit* $w.p. \geq \frac{1}{2} + \epsilon$ *there exists a strategy for one of the parties that biases the output from random by at least* $\frac{\epsilon}{O(l)}$.

Remark 1.2. The theorem will hold even in the "fail stop model", where the only allowed deviation from the protocol is to halt prematurely.

Notation 1.3. Recall that our definition requires each one of the protocol participants to have a buffer that holds the output value. Let the parties be A, B and let a_i (respectively b_i) denote the value of the buffer of party A (respectively B) after the receipt of message i.



Let define some notation that will we useful for us at the proof:

$\ell = \begin{cases} \frac{l}{2} & l \, mod \, 2 = 0 \\ \frac{l+1}{2} & else \end{cases}$

$AGREE_{A,B}(l) = Prob_{A,B}\left[a_\ell = b_\ell\right] - \frac{1}{2}$

$BIAS^0_{A,B^\star} = \left|Prob_{A,B^\star}\left[a_\ell = 1\right] - \frac{1}{2}\right|$ when $B^\star$ doesn't send the $\ell' th$ message

$BIAS^1_{A^\star,B} = \left|Prob_{A^\star,B}\left[b_\ell = 1\right] - \frac{1}{2}\right|$ when $A^\star$ doesn't send the $\ell' th$ message

As a warm-up, we will prove a weaker claim:

Claim 1.4. No finite protocol (A, B) could not obtain $AGREE_{A,B}(l) = \frac{1}{2}$ and $BIAS^0_{A,B^\star} = BIAS^1_{A^\star,B} = 0$.

In another words: There does not exist any l-round protocol that can guaranty agreement between two parties without any bias.

Proof. We will proof it by induction, the base case $l = 0$ is trivial.

Assume that we prove the $Claim 2.4$ for all protocols that have less then l rounds and now we will prove it for all l-round protocols.

Assume the by contradiction that exists some l-round protocol (A, B), that get $AGREE_{A,B}(l) \geq \frac{1}{2}$ and $BIAS^0_{A,B^\star} = BIAS^1_{A^\star,B} = 0$. Let show that it yields to existence of $(l-1)$-round protocol (A', B') that get $AGREE_{A',B'}(l) \geq \frac{1}{2}$ and $BIAS^0_{A',B^\star} = BIAS^1_{A^\star,B'} = 0$ in contradiction to the induction assumption. $AGREE_{A,B}(l) \geq \frac{1}{2}$ means that $Prob_{A,B}\left[a_\ell = b_\ell\right] = 1$. Now we need to check two separate cases:

- At the first case $Prob_{A,B}\left[a_\ell = a_{\ell-1}\right] = 1$, then A can to output the correct value without getting the last message from B, but in this case we get $(l-1)$-round protocol (A', B') that get $AGREE_{A',B'}(l) = \frac{1}{2}$, contradiction.
- At the second case $Prob_{A,B}\left[a_\ell \neq a_{\ell-1}\right] = \delta > 0$, then $B^\star$ can bias the $A's$ output. The $B^\star$ protocol is:
  - run at B till b_ℓ is known
  - if (b_ℓ= 1) then continue at B else abort

  So A will to output 1 with probability $Prob_{A,B^\star}\left[a_\ell = b_\ell\right] \cdot Prob_{A,B^\star}\left[b_\ell = 1\right] + Prob_{A,B^\star}\left[a_\ell \neq a_{\ell-1}\right] \cdot Prob_{A,B^\star}\left[b_\ell = 0\right] = \frac{1}{2} + \frac{\delta}{2}$, in contradition to assumption that the protocol work without any bias.

□

Now we are ready to rigorously state the Cleve's theorem and to prove it.

Theorem 1.5. ***[Cleve 86](formal statement):*** *Let (A, B) be protocol s.t. $AGREE_{A,B}(l) \geq \epsilon$ then either $\exists B^\star$ s.t. $BIAS^0_{A,B^\star} > \frac{\epsilon}{4 \cdot \ell + 1}$ or $\exists A^\star$ s.t. $BIAS^1_{A^\star,B} > \frac{\epsilon}{4 \cdot \ell + 1}$.*

Proof. Let us define $4 \cdot \ell + 1$ strategies:

Strategy $A^\star_{0,0}$ is : newer send the first message

Strategy $A^\star_{0,i}$ is :

- run A till a_i is known
- if (a_i= 0) then continue at A and abort at the next turn else abort now

Strategy $A^\star_{1,i}$ is :

- run A till a_i is known
- if (a_i= 1) then continue at A and abort at the next turn else abort now

Strategy $B^\star_{0,i}$ is :

- run B till b_i is known
- if (a_i= 0) then continue at B and abort at the next turn else abort now

Strategy $B^{\star}_{1,i}$ is :

- run B till b_i is known
- if (b_i= 1) then continue at B and abort at the next turn else abort now

bias of strategy $A^{\star}_{0,0}$is : $max\left\{Pr\left[b_0=0\right],Pr\left[b_0=1\right]\right\}-\frac{1}{2}$

bias of strategy $A^{\star}_{0,i}$is : $Pr\left[a_i=0\wedge b_i=0\right]+Pr\left[a_i=1\wedge b_{i-1}=0\right]-\frac{1}{2}$

bias of strategy $A^{\star}_{1,i}$is : $Pr\left[a_i=1\wedge b_i=1\right]+Pr\left[a_i=0\wedge b_{i-1}=1\right]-\frac{1}{2}$

bias of strategy $B^{\star}_{0,i}$is : $Pr\left[b_i=0\wedge a_{i+1}=0\right]+Pr\left[b_i=1\wedge a_i=0\right]-\frac{1}{2}$

bias of strategy $B^{\star}_{1,i}$is : $Pr\left[b_i=1\wedge a_{i+1}=1\right]+Pr\left[b_i=0\wedge a_i=1\right]-\frac{1}{2}$

We want to calculate $\frac{1}{4\cdot\ell+1}\left[A^{\star}_{0,0}+\sum_{i=1}^{\ell}(A^{\star}_{0,i}+A^{\star}_{1,i}+B^{\star}_{0,i}+B^{\star}_{1,i})\right]$

Observe that the sum is telescopic so after the reordering we will get

(remainder:

$Pr[a_i=0\wedge b_i=0]+Pr\left[a_i=1\wedge b_i=1\right]+Pr\left[b_i=1\wedge a_i=0\right]+Pr\left[b_i=0\wedge a_i=1\right]-1=0)$

$A^{\star}_{0,0}+\sum_{i=1}^{\ell}(A^{\star}_{0,i}+A^{\star}_{1,i}+B^{\star}_{0,i}+B^{\star}_{1,i})=$

$max\left\{Pr\left[b_0=0\right],Pr\left[b_0=1\right]\right\}-\frac{1}{2}+$

$Pr[a_1=1\wedge b_0=0]-\frac{1}{2}+Pr[a_1=0\wedge b_0=1]-\frac{1}{2}+$

$Pr[a_{\ell+1}=0\wedge b_\ell=0]+Pr[a_{\ell+1}=1\wedge b_\ell=1]$

(reminder: $Pr[a_{\ell+1}=0\wedge b_\ell=0]+Pr[a_{\ell+1}=1\wedge b_\ell=1]=\frac{1}{2}+\epsilon$)

$\frac{1}{4\cdot\ell+1}\left[A^{\star}_{0,0}+\sum_{i=1}^{\ell}(A^{\star}_{0,i}+A^{\star}_{1,i}+B^{\star}_{0,i}+B^{\star}_{1,i})\right]=$

$\frac{1}{4\cdot\ell+1}\left[\epsilon+Pr[a_1\neq b_0]+max\left\{Pr\left[b_0=0\right],Pr\left[b_0=1\right]\right\}-1\right]$

let define by $j\in\{0,1\}$to be one bit that success the $Pr\left[b_0=j\right]=max\left\{Pr\left[b_0=0\right],Pr\left[b_0=1\right]\right\}$ and let set a_1 to be j, ($Pr[a_1=j]=1$), in this case $Pr[a_1=b_0]=max\left\{Pr\left[b_0=0\right],Pr\left[b_0=1\right]\right\}$ so $Pr[a_1\neq b_0]+max\left\{Pr\left[b_0=0\right],Pr\left[b_0=1\right]\right\}-1=0$ so the average of bias of all the strategist that we define is $\frac{\epsilon}{4\ell+1}$ so at least one of them has bias greater or equal to $\frac{\epsilon}{4\ell+1}$.

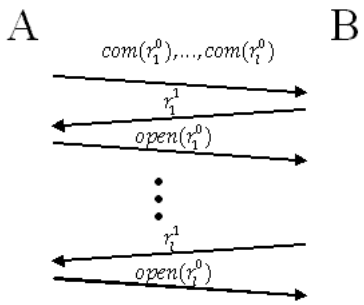
□

2. Fair coin tossing protocols

In this section we'll present two protocols - one that gives $\Theta(\frac{1}{\sqrt{\ell}})$ bias and one that gives $\Theta(\frac{1}{\sqrt{\ell}})$ bias. We will not prove the security of either one.

2.1. **Protocol[Cleve86]:**

- A chooses l random bits $(r^0_1,...,r^0_l)$ and sends to B $(com(r^0_1),...,com(r^0_l))$
- While $i<l$: B chooses random bit r^1_i and sends it to A, then A sends to B the $open(r^0_i)$
- At the end both sides output the majority of $r_j=r^0_j\oplus r^1_j$.



If at some round j, B didn't get the $open(r_j^0)$ message from A or the message is corrupted then B stop the protocol iterations and chose randomly $l-j+1$ bits $(r_j,...,r_l)$ and outputs the majority of $r_1,...,r_{j-1}$ that he get from the running protocol and $r_j,...,r_l$ that he choose randomly. In the case that B abort the protocol at round j, A chose randomly $l-j+1$ bits $(r_j,...,r_l)$ and outputs the majority of $r_1,...,r_{j-1}$ that he gets from the protocol and the $r_j,...,r_l$ chosen randomly.

So A can bias the $B's$ output only if the case that all bits excepts the r_i derived to two equal size sets of ones and zeroes, so if we also give to A the total power to decide the r_i the probability to separate the output to two equal size sets of ones and zeroes is $\begin{pmatrix} 2n \\ n \end{pmatrix} \cdot \left(\frac{1}{2}\right)^{2n} = \frac{2n!}{(n!)^2} \cdot \left(\frac{1}{2}\right)^{2n}$ by Stirling we get $n! \approx \sqrt{2\pi n} \cdot (\frac{n}{e})^n$ so $\frac{2n!}{(n!)^2} \cdot \left(\frac{1}{2}\right)^{2n} \approx \frac{2\sqrt{\pi n} \cdot 2^{2n} \cdot (\frac{n}{e})^{2n}}{2\pi n \cdot (\frac{n}{e})^{2n}} \cdot \left(\frac{1}{2}\right)^{2n} = \frac{1}{\sqrt{\pi n}}$ and it yields to $Pr_{A^\star,B}\left[A\ success\ to\ bias\ B's\ output\right] = \frac{1}{\sqrt{\pi l}}$

Next we present the protocol that gets more accurate bound $\frac{1}{l}$.

2.2. **Protocol [Moran Naor Segev - Crypto 08]:** Define function f that does not take any input and outputs l pairs of bits $a_1,b_1;a_2,b_2;...;a_l,b_l$ such that for all the pairs $a_j,\ b_j$ we have: if $j < i^\star$ then a_j and b_j are independent and if $j \geq i^\star$ $a_j = b_j = r$. Where r is a random bit and $i^\star$ chosen by f randomly and uniformly distributed in $(1..l)$.

The parties will first compute f securely, using the protocol of **[GMW]**, the fact that **[GMW]** is not fair does not matter because we stop the calculation when all the parties get their fractions of the output, the first one get $a_1^1,b_1^1;a_2^1,b_2^1;...;a_l^1,b_l^1$ and the second one get $a_1^2,b_1^2;a_2^2,b_2^2;...;a_l^2,b_l^2$ such that $a_i^1 \oplus a_i^2 = a_i$ and $b_i^1 \oplus b_i^2 = b_i$. And cause we don't continue to the stage there the parties start to exchange the outputs that they get from **[GMW]** the unfairness of the **[GMW]** does not matter.

We will not prove the correctness of the protocol in the formal way, we only give some intuition of the prove.

2.2.1. *Proof Intuition:* We will denote the parties by A and B such that A gets from the **[GMW]** $a_1^1,b_1^1;a_2^1,b_2^1;...;a_l^1,b_l^1$ and B gets $a_1^2,b_1^2;a_2^2,b_2^2;...;a_l^2,b_l^2$. Through the protocol A and B exchange the outputs that they get from the **[GMW]:**

In the round i the party A sends to the B: b_i^1, and gets from B: a_i^2 , so in each round A and B can to calculate a_i and b_i.

If we run the protocol till end the parties will get $a_l = b_l$.

To bias the output one need to know the value of bit r and if the value does not satisfies him to abort, before the round $i^\star$ because after that the buffer of the second party permanently contains the value of r and the abort does not matter, but before the round $i^\star$ nobody knows the bit r otherwise it will not be random so he need to abort exactly at round $i^\star$. But $i^\star$ is chosen by f randomly so no one knows it from the beginning, so the best thing that anybody can do is to guess it and the right guess will be done with the probability $\frac{1}{l}$.

3. THE EXCHANGE PROTOCOLS:

So far we discussed fairness only in the context of coin-tossing. But fairness is important also in many other scenarios and applications for instance: Secret Exchange protocols.

At the Secret Exchange protocols two or more parties have some secret information (e.g. authentication signature) and they need to exchange the secret (e.g. to sign on document), in this protocols we have the same problems that we see at the case of coin-tossing.

Claim 3.1. If we did not allow any additional assumption we can't get the perfect Exchange protocol.

Proof. Otherwise if we have some protocol π that have perfect fairness, that allows to exchange between to parties some secret information without any assumption on this secret information, we can choose the secret information to be random bits and in this way to exchange the random bits and to get perfect Coin Tossing protocol. □

Claim 3.2. The **[MNS]** protocol can be extended to the case of the Exchange protocols and to give same lower bound of bias between the parties outputs.

Proof. Let change the output of the function f from pairs of bits to pairs of strings and we will get the same proof as before. □

4. FAIR FUNCTION EVALUATION

From $Claim$ 4.1 it follows that we can't solve the problem of exchange secret information fairly in general case, still we also can get perfect exchange protocols for specific functions:

- Gordon, Hazay, Kats and Lindell show that exists perfect-fair protocol for the calculation of function that gets two numbers x and y and return value is bit that represents the correctness of the predicate $x < y$.

or by using additional assumption on the information that we need to exchange:

- **Gradual release:** in this case parties divide there message s to many biased pieces of information (e.g if we want to send bit 0 we will build the random generator that returns 0 with probability $\frac{1}{2} + \frac{1}{l}$, and to send it's outputs, so after l round we expect to sent one 0 bit more that 1). So if the adversary terminates the protocol at any time the amount of work required by the honest party to complete the output computation is polynomially related to the amount of work spent by the adversary to learn the output because the number of samples that each party have differed by 1. This approach works only in the case that where is only one possible deviation from the protocol: termination of the protocol's run.
- **Resource Fairness:** in this approach the parties can verify the correctness of the information that they get. At the beginning both parties define exponentially big set of "possible secrets" and prove by ZK to each other that this set contains the real secret. At each round parties could divide the size of the set by 2. So if one side terminates the second one need to do no more that twice amount of work that the first did to get the secret, because he only need to check all the object in the remained set and it is no more that twice bigger.

or by using additional assumption on the model (e.g. some additional party with some specific properties):

- **Optimistic Model:** In this model we have some trusted party, Judge, that gets at the beginning the some information from both parties and at the case of deviation can open to the honest party the secret of the deviated party. In this case both parties understand that the second party could get the secret also in the case that they abort, there is no incentive to abort. This Judge doesn't take part at the protocol only the existence of Judge already turn the protocol to be perfect exchange. In this case we also assume Resource Fairness that allow to the parties to give to the Judge some partially information of the secret (e.g. the first party can give to the Judge his secret

encrypted by public key of the second party) but he also need to proof to the Judge by ZK proof that there is realy his secret and not some random sequence of bits.

5. Multiparty Fair Protocols

In the multiparty case we can do standard perfect fairness protocol and also guaranty that all the honest parties will get the output at the end of the protocol, but only in the case that the honest parties in majority.

For the summary : In the field of the multiparty fair protocols very little is known and there are lots of intriguing open questions.

References

[1] D. Beaver, S. Goldwasser. Multi Party Fault Tolerant Computation with Faulty Majority Based on Oblivious Transfer. Proceedings of the Annual IEEE Symposium on Foundations of Computer Science, 1989, pp. 468- 473.

[2] M. Blum. Coin flipping by telephone - A protocol for solving impossible problems. In Proceedings of the 25th IEEE Computer Society International Conference, pages 133-137, 1982.

[3] R. Cleve. Limits on the security of coin flips when half the processors are faulty. In Proceedings of the 18th Annual ACM Symposium on Theory of Computing, pages 364-369, 1986.

[4] R. Cleve and R. Impagliazzo. Martingales, collective coin flipping and discrete control processes. http://www.cpsc.ucalgary.ca/~cleve/pubs/martingales.ps, 1993.

[5] Oded Goldreich, Silvio Micali, Avi Wigderson: How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority STOC 1987: 218-229

[6] Shafi Goldwasser, Leonid A. Levin. Fair Computation of General Functions in Presence of Immoral Majority. Annual CRYPTO Conference (IEEE/LACR), Santa Barbara, August 1990 (Proceedings 1991).

[7] Juan A. Garay, Philip D. MacKenzie, Manoj Prabhakaran, Ke Yang: Resource Fairness and Composability of Cryptographic Protocols. TCC 2006: 404-428

[8] T. Moran, M. Naor and G. Segev. An Optimally Fair Coin Toss. Crypto 08.