Advanced Tools from Modern Cryptography

Lecture 15

MPC: Miscellaneous (and Revision)

Output Delivery

- 3 levels:
  - Unfair (a.k.a., selective abort)
    - Adversary can see its output and decide which set of honest parties receive theirs
  - Fair
    - Adversary can cause abort for all parties or none, before seeing its output
  - Guaranteed output delivery
    - Adversary cannot prevent honest parties from producing an output. (Adversary will have well-defined inputs no matter what it does.)

Fair Coin-Tossing

- Consider two parties trying to toss a coin using any arbitrary unfair setup
  - Can implement an unfair coin tossing functionality
  - How about using another setup like commitment?
    - Alice commits to a random bit a, Bob sends a bit b, Alice opens and they output $a \oplus b$
    - Still unfair: Alice can abort after learning the outcome
  - Two parties can never obtain a fair coin, given only unfair setups, even under computational assumptions, even for standalone security

Recall

Broadcast

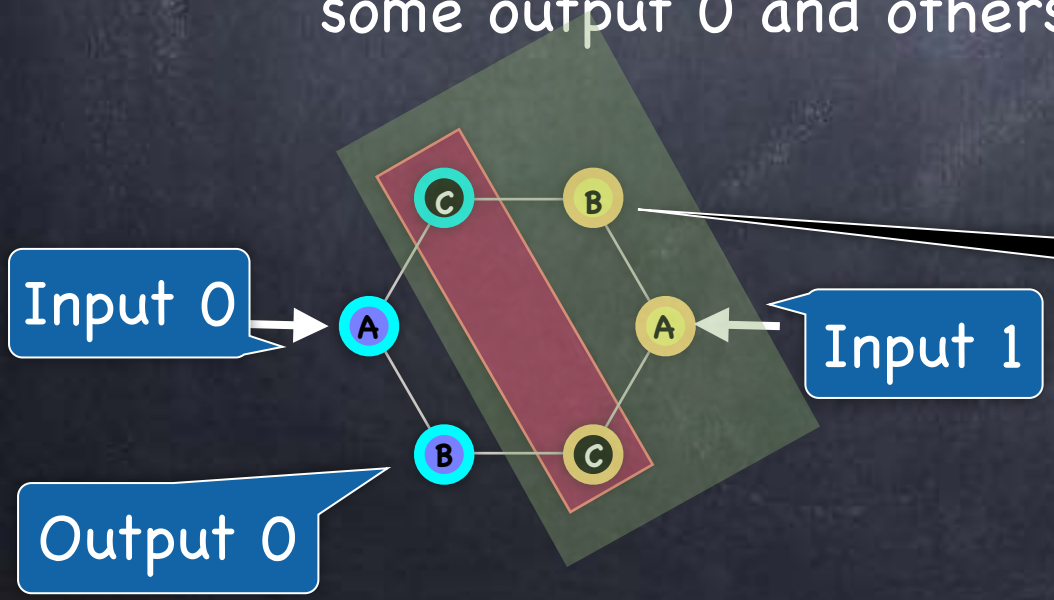
- BGW protocol relied on broadcast to ensure all honest parties have the same view of disputes, resolution etc.
- Concern addressed by broadcast: a corrupt sender can send different values to different honest parties
- Broadcast with selective abort can be implemented easily, even without honest majority
  - Sender sends message to everyone. Every party cross-checks with everyone else, and aborts if there is any inconsistency.
- If corruption threshold $t < n/3$, then it turns out that broadcast with guaranteed output delivery can be implemented

Otherwise not!

- If broadcast given as a setup, can do MPC with guaranteed output delivery for up to $t < n/2$

No Broadcast with Guaranteed Output if 1/3 Corrupt

- Broadcast requirements (message being a single bit):
  - If sender honest, all honest parties should output the bit it sends (can't abort)
  - All honest parties should agree on the outcome (can't have some output 0 and others 1)



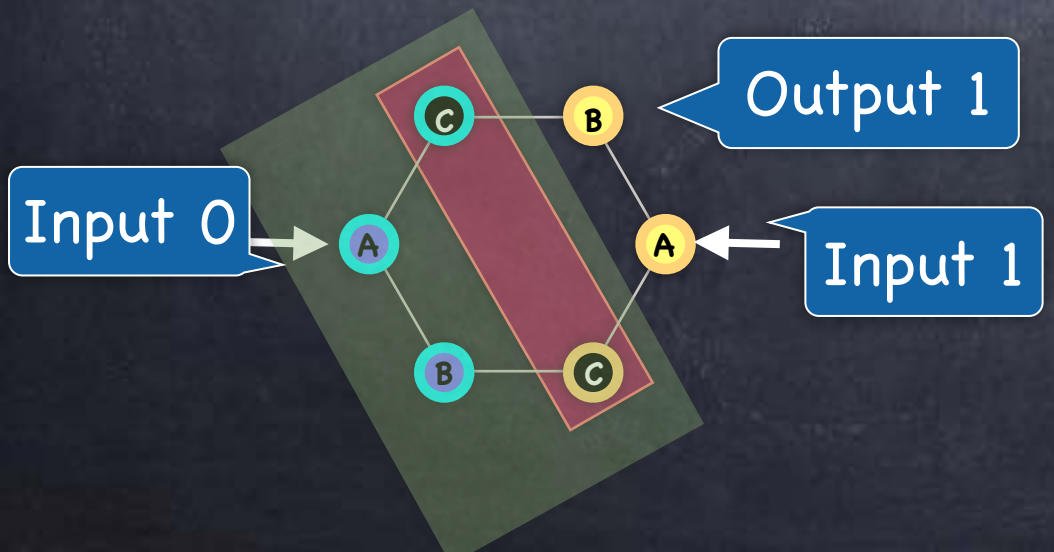
Consider 6 parties running the code for A, B, C (A is the sender)

Adversary corrupting C

Note: can't do this if A, B allowed to have a priori shared secrets (say message authentication keys)

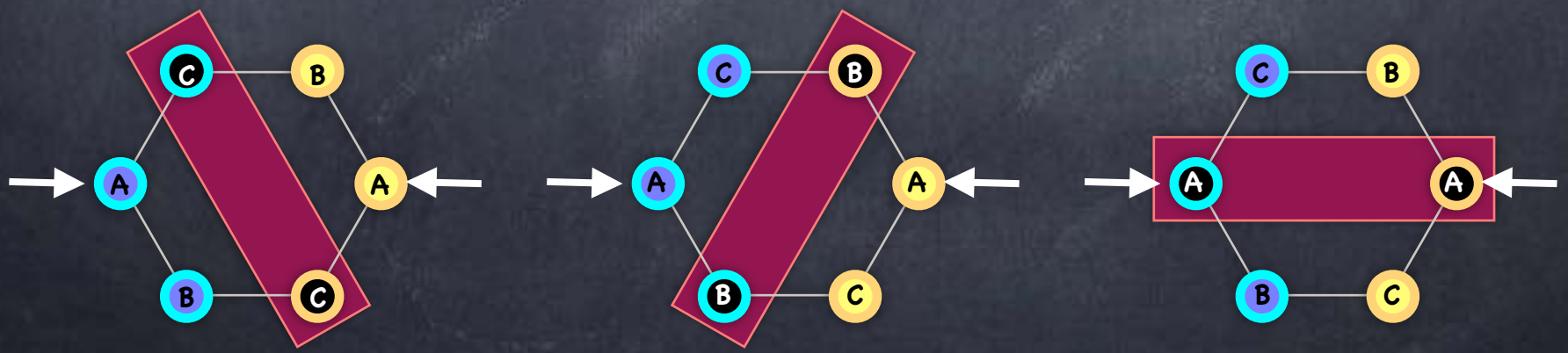
No Broadcast with Guaranteed Output if 1/3 Corrupt

- Broadcast requirements (message being a single bit):
  - If sender honest, all honest parties should output the bit it sends (can't abort)
  - All honest parties should agree on the outcome (can't have some output 0 and others 1)



No Broadcast with Guaranteed Output if 1/3 Corrupt

- Broadcast requirements (message being a single bit):
  - If sender honest, all honest parties should output the bit it sends (can't abort)
  - All honest parties should agree on the outcome (can't have some output 0 and others 1)



No Broadcast with Guaranteed Output if 1/3 Corrupt

- Broadcast requirements (message being a single bit):
  - If sender honest, all honest parties should output the bit it sends (can't abort)
  - All honest parties should agree on the outcome (can't have some output 0 and others 1)
- Impossible to satisfy both constraints simultaneously, if 1/3 can be corrupt
  - Irrespective of what computational assumptions are used!
  - But a priori shared keys can give broadcast with guaranteed output delivery against unrestricted corruption

Revision