ROUND COMPLEXITY LOWER BOUND OF ISC PROTOCOL IN THE PARALLELIZABLE MODEL

Huijing Gong

CMSC 858F

Overview

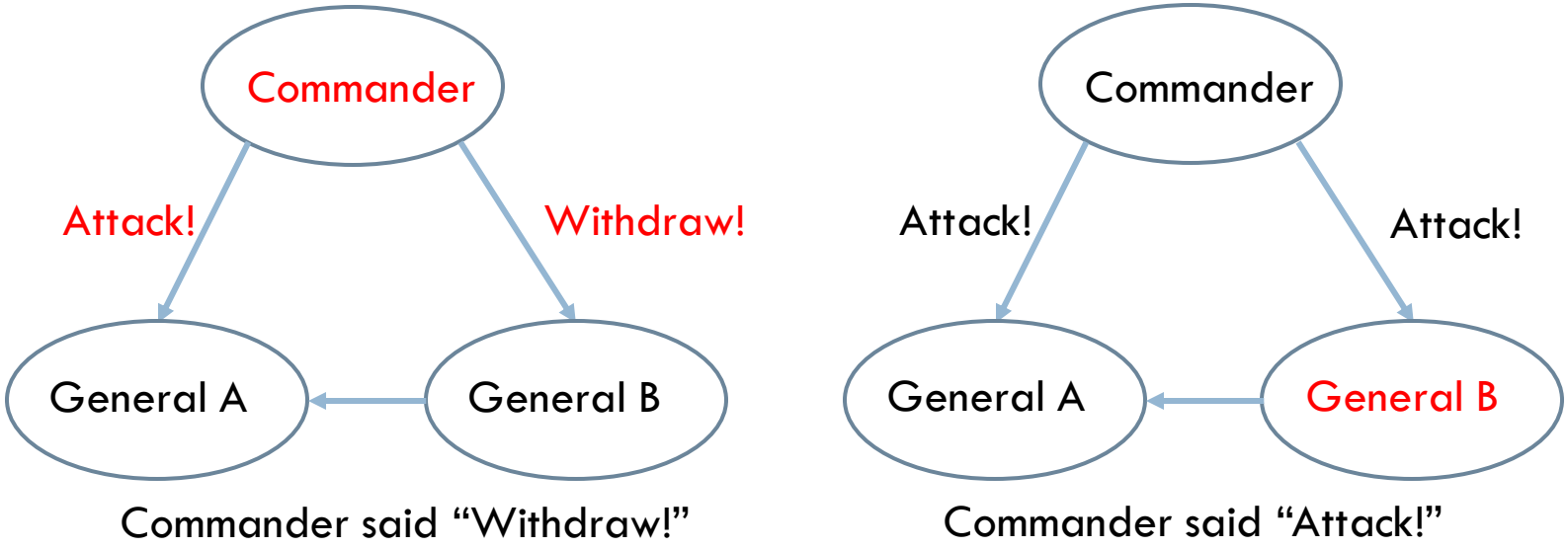
- Background
  - Byzantine Generals Problem
  - Network Model w/o Pre-existing Setup
- ISC Protocol in Parallelizable Model
  - ISC, Parallelizable Model
  - Intuition of Protocol
- Round Complexity Lower Bound
  - Theorem
  - Proof

Background

- Byzantine Generals Problem
  - Commanding general and generals camped outside an enemy city
  - Commanding general sends the order to all
  - The generals exchange messages to agree on a battle plan: withdraw or attack
  - Traitor(s): confuse others

Background

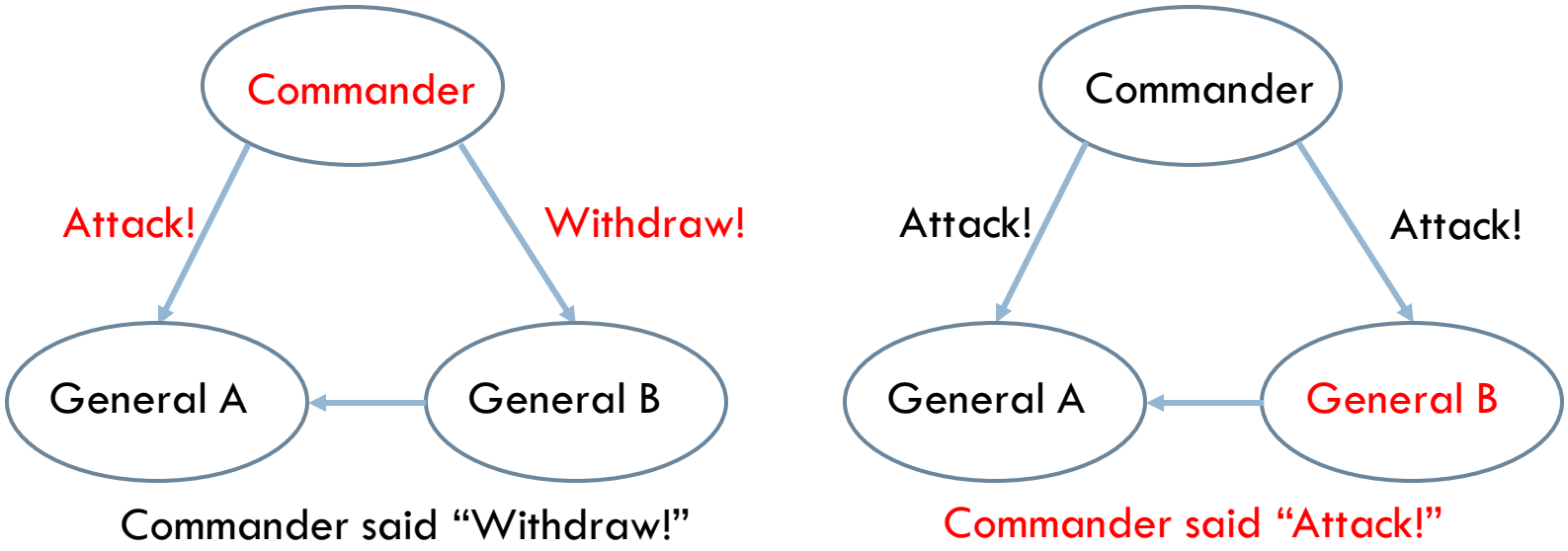
- Byzantine Generals Problem



  - Traitor(s): confuse others

Background

- Byzantine Generals Problem



  - Goal of Byzantine Agreement Protocols:
    - Generals reach agreement on whether attack or withdraw
    - Not obey Commander's order if Commander is a traitor

Background

- Network Model w/o Pre-Existing Setup
  - N Parties: cannot be authenticated by pre-existing means
    - E.g. Public-Key Infrastructure (PKI)
  - Difference:
    - No idea where a receive message sent from
    - No idea if two message received from different rounds are sent from one party
    - But, a message sent by an honest party in some run received by all other parties at the end of that run

Background

- Network Model w/o Pre-Existing Setup
  - Adversary:
    - Corrupt parties to behave arbitrarily
    - Inject message into the network (> n -1)
    - Change messages they relay
    - Send message to subset of the honest parties (< n - 1)

ISC Protocol in Parallelizable Model

- Protocol (by J. Katz, A. Miller, and E. Shi [2014]):
  - N Parties: cannot be authenticated by pre-existing means
  - Goal: Establish a PKI
  - No bound on the number of corruption
  - Adversary cannot drop or modify honest parties' message
- Time-Lock Puzzle (Proof-of-Parallelizable Work Model)
  - Take role of trusted setup assumption
  - Each honest party has equal computational power
  - Adversary(f parties) runs sequentially faster by factor f
  - f correct parties cannot solve any faster taking as whole.

ISC Protocol in Parallelizable Model

- Interactive Set Consistency (ISC):
  - Each party has an input and output a (multi)set of size n, s.t.
    - All the honest parties agree(output) on the same (multi)set S
    - S contains all the honest parties' inputs
  - Can be used to establish PKI among parties,
    - PKI later can provide authenticated communication

ISC Protocol in Parallelizable Model

- $\mathcal{F}_{parpuz}$ Oracle
  - Modeling the Time-Lock Puzzle
  - Each party can produce a puzzle solution independently in each round
  - An adversary who corrupts f processes can solve f puzzles per round in total
  - Scheme
    - Solve a cryptographic puzzle upon request
    - Check solutions upon request
    - Polynomial Time

ISC Protocol in Parallelizable Model

- $\mathcal{F}_{parpuz}$ Oracle
  - Solve:
    - $\mathcal{F}_{parpuz}$ oracle maintains a table T.
    - Each party P_i sends (solve, x_i) to $\mathcal{F}_{parpuz}$ oracle: For I = 1, …, n, $\mathcal{F}_{parpuz}$ first check if (x_i, h_i) has been stored in T.
      - Yes: return h_i to P_i;
      - Otherwise, generate $h_i \in \{0,1\}^{\lambda}$, return h_i to P_i and store (x_i, h_i) in T.

ISC Protocol in Parallelizable Model

- $\mathcal{F}_{parpuz}$ Oracle
  - Solve:
    - Each honest party is allowed to call $\mathcal{F}_{parpuz}$ only once per round
    - Each round of honest party: All the solve request must be sent before any honest party receives its solution.
    - Each round of corrupted parties: they can call $\mathcal{F}_{parpuz}$ one after another in sequence up to f times.

ISC Protocol in Parallelizable Model

- $\mathcal{F}_{parpuz}$ Oracle
  - Check:
    - Each party P_i sends (check, (x_i^1, h_i^1), (x_i^2, h_i^2), …) to $\mathcal{F}_{parpuz}$ oracle:
    - $\mathcal{F}_{parpuz}$ oracle returns $(b_i^1, b_i^2, \ldots)$:
      - $b_i^j = 1$ if $(x_i^2, h_i^2) \in T$
      - $b_i^j = 0$, otherwise.

ISC Protocol in Parallelizable-Work Model

- Orders in rounds (honest parties)
  - Each party sends (at most) one solve-request to $\mathcal{F}_{parpuz}$ and receive the solution
  - Each party computes a message to send
  - Message are delivered to each party
  - Each party sends a list of puzzle solution to $\mathcal{F}_{parpuz}$ for verification

ISC Protocol in Parallelizable-Work Model

- Intuition of the Protocol:
  - Mining Phase:
    - Each correct party generate a chain of $O(f^2)$ puzzle solutions:
      - E.g. Solve(pk_i, Solve(pk_i, Solve(…Solve(pk_i, ϕ)…)))
    - Each correct party can create a valid puzzle chain for its own key,
    - Corrupt party only can create at most f puzzle chains before the protocol terminate

ISC Protocol in Parallelizable-Work Model

- Intuition of the Protocol:
  - Communication Phase:
    - Each party publishes their chains and propagate the puzzle chain they received from others
    - In each round r: Each party accepts a value if it has received a collection of r signatures on that value, the process then add its own signature to the collection and relay it to the other processes.
    - Signatures without associated puzzle chains are ignored
    - A correct party consider a public key “valid” if it comes along with a puzzle chain containing the public key long enough

Reference

- Aguilera, Marcos Kawazoe, and Sam Toueg. "A simple bivalency proof that t-resilient consensus requires t+ 1 rounds." *Information Processing Letters* 71.3 (1999): 155-158.
- Dolev, Danny, and H. Raymond Strong. "Authenticated algorithms for Byzantine agreement." *SIAM Journal on Computing* 12.4 (1983): 656-666.
- Lamport, Leslie, Robert Shostak, and Marshall Pease. "The Byzantine generals problem." *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4.3 (1982): 382-401.
- Katz, Jonathan, Andrew Miller and Elaine Shi. "Pseudonymous Secure Computation from Time-Lock Puzzles." *Cryptology ePrint Archive* (2014):857.