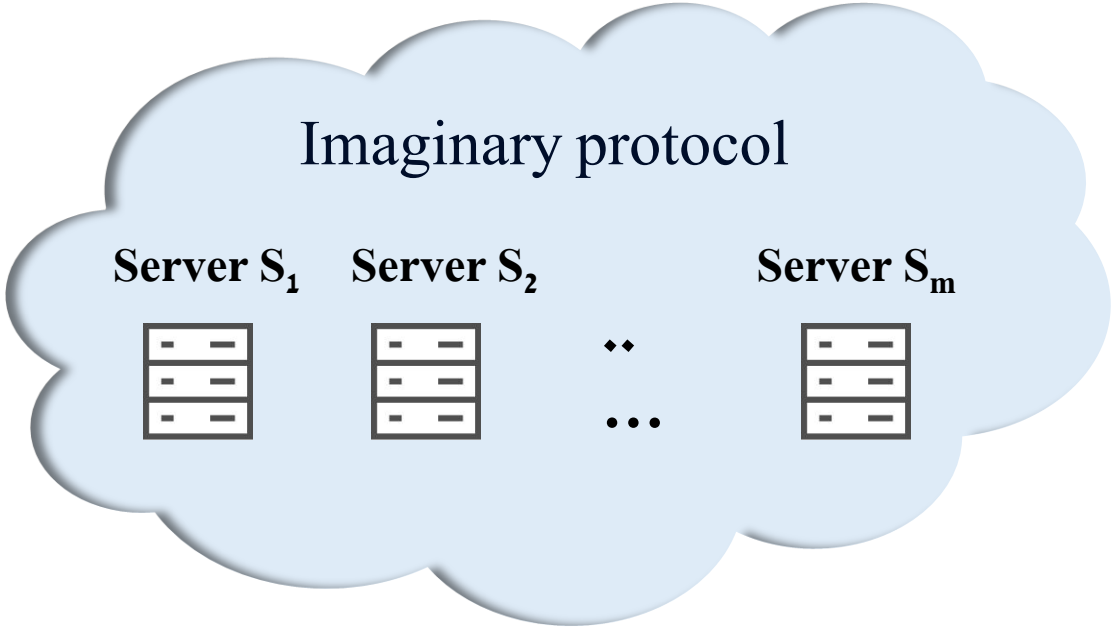
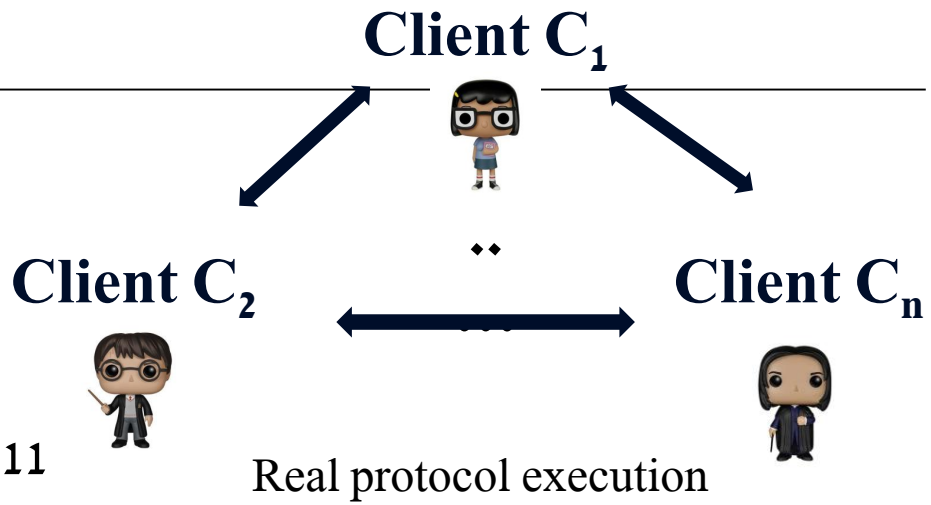


Introduction to MPC-in-the-Head

Carmit Hazay
Faculty of Engineering, Bar-Ilan University

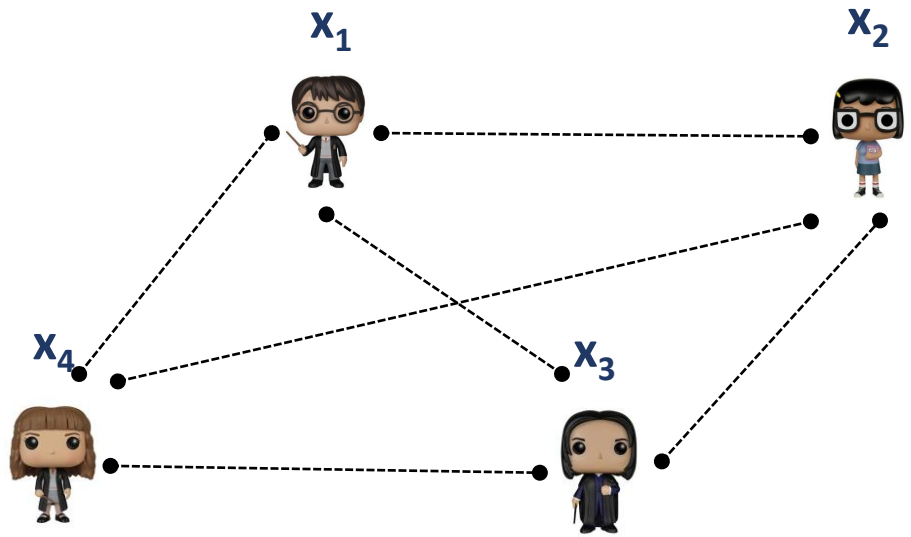


Based on Yuval Ishai's slides from Bar-Ilan 1st Winter School 2011

Back to the Classics in the 1980s

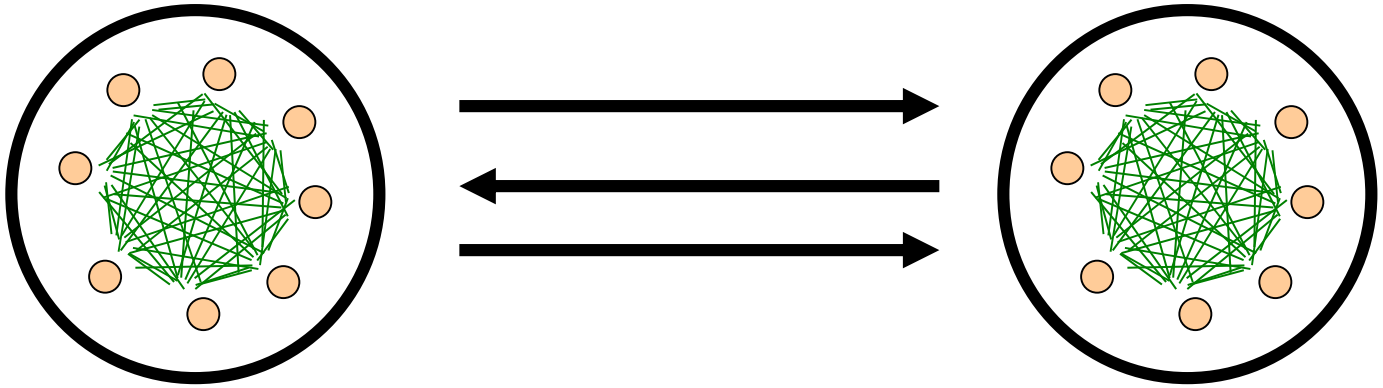
- Zero-knowledge proofs for NP [GMR85,GMW86]
- Computational MPC with no honest majority [Yao86, GMW87]
- Unconditional MPC with honest majority [BGW88, CCD88, RB89]
- Unconditional MPC with no honest majority assuming ideal OT [Kilian88]

Are these unrelated?



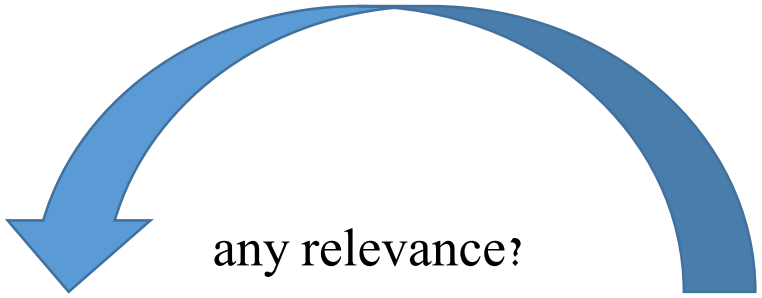
Message of this Talk

- Honest-majority MPC is useful even when there is no honest majority



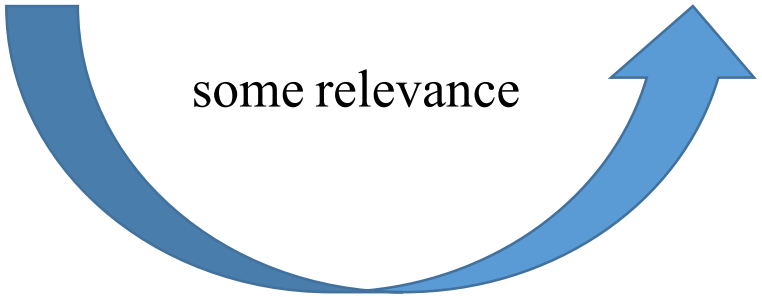
- Establishes unexpected relations between classical results
- New results for MPC with no honest majority

Bridging the Two Settings



The computational setting:
- zero-knowledge proofs
- efficient two-party protocols

The information theoretic setting:
- information-theoretic cryptography
- honest-majority MPC



Bridging the Two Settings

- Add to computational setting a simple ideal functionality
  - Ideal commitment oracle for ZK (Com-hybrid model)
  - Ideal OT oracle for general protocols (OT-hybrid model)
- Makes unconditional (and UC) security possible
  - Analogous to secure channels in information theoretic world
- Why OT and Com?
  - Generality: Com or OT can be realized in a variety of models, under a variety of assumptions
  - Efficiency: Com or OT can be realized with little overhead
    - Essentially free given preprocessing [BG89]
    - Cheap preprocessing: fast OT […,PVW08], faster OT extension [Bea96,IKNP03,…]

- Still: Why should information theoretic research be relevant?

Bridging the Two Settings

- Add to computational setting a simple ideal functionality
  - Ideal commitment oracle for ZK (Com-hybrid model)
  - Ideal OT oracle for general protocols (OT-hybrid model)
- M
- W

A high-level idea:

- Run MPC "in the head"
- Commit to generated views
- Use consistency checks to ensure honest majority

nptions

faster OT extension [Bea96,IKNP03,…]

- Still: Why should information theoretic research be relevant?

Zero-Knowledge Proofs

- Goal: ZK proof for an NP-relation R(x,w)

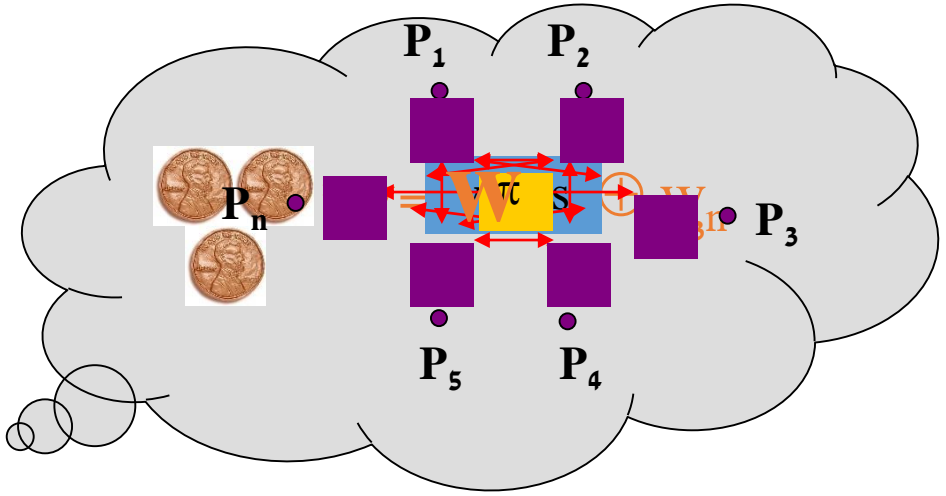
- Towards using MPC:
  - Define n-party functionality

    $g(x; w_1,\ldots,w_n) = R(x, w_1 \oplus \ldots \oplus w_n)$
  - Use any 2-secure, perfectly correct protocol for g
    - Security in semi-honest model
    - Honest majority when n>4

Zero-Knowledge from MPC [IKOS07]

Given MPC protocol π for
$g(x; w_1,\ldots,w_n) = R(x, w_1 \oplus \ldots \oplus w_n)$

Prover



Verifier

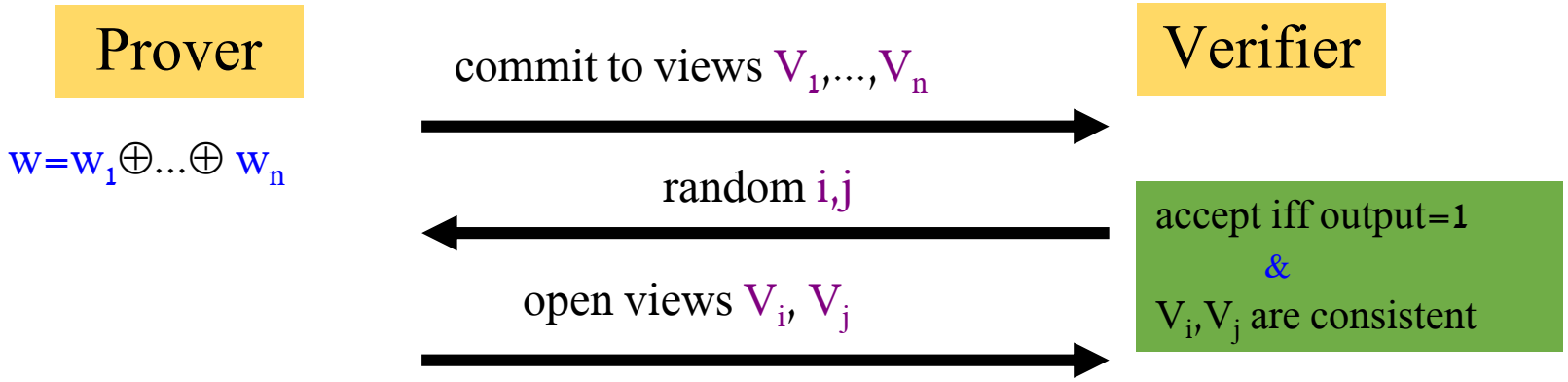
commit to views $V_1,\ldots,V_n$

random i,j

open views V_i, V_j

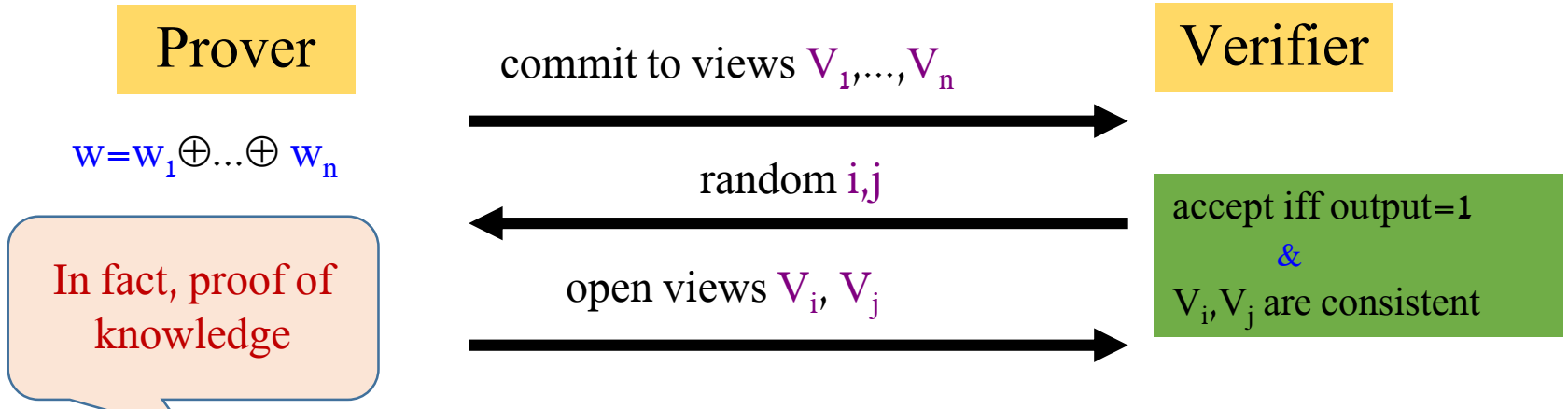
accept iff output=1
&
V_i, V_j are consistent

Analysis



- Completeness: √
- Zero-knowledge: by 2-security of π and randomness of w_i, w_j
  (Note: enough to use w_1,w_2,w_3)

Analysis

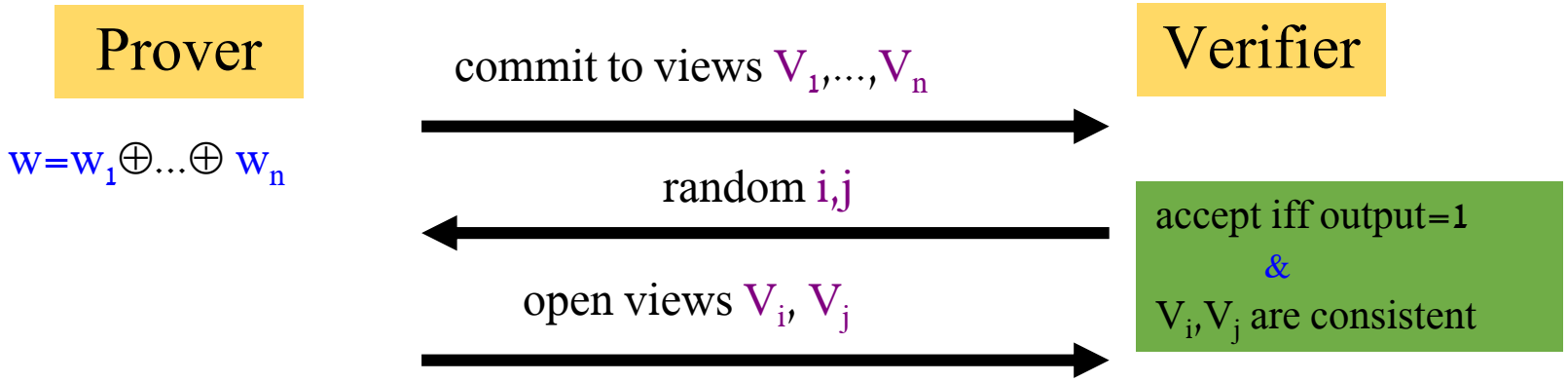


- Soundness: Suppose R(x, w)=0 for all w.
  either (1) $V_1,\ldots,V_n$ consistent with protocol π
  or (2) $V_1,\ldots,V_n$ not consistent with π

(1) → outputs=0 (perfect correctness)
→ verifier rejects

(2) → for some (i,j), V_i,V_j are inconsistent.
→ verifier rejects with prob. $\geq 1/n^2$.

Analysis



Communication complexity:

≈ (comm. complexity + rand. complexity + input size) of π

Extensions

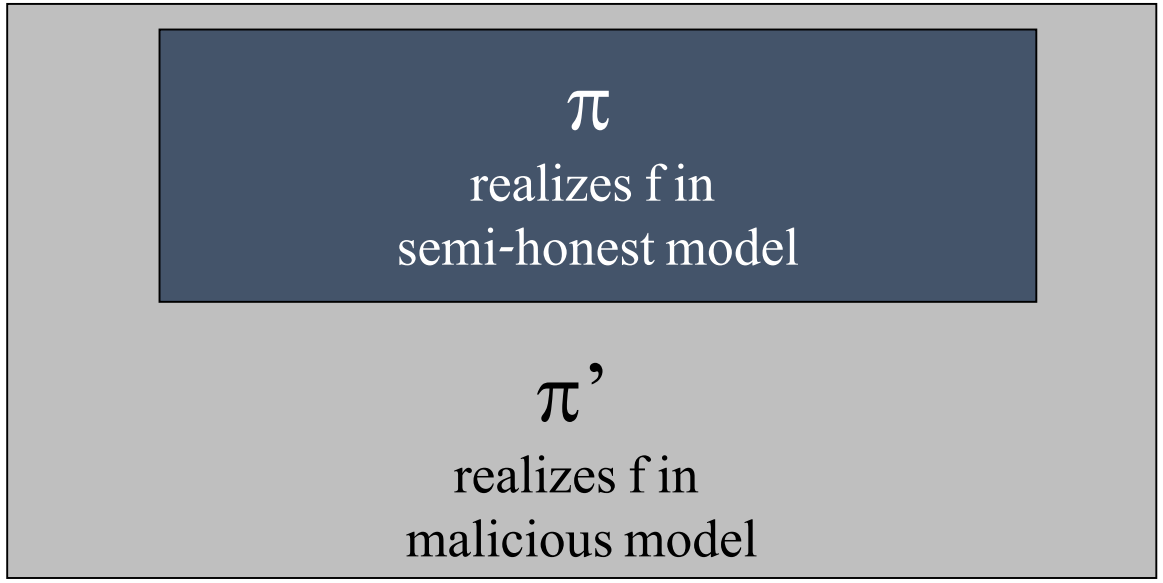
- Works also with OT-based MPC
  - Simple consistency check
- Variant: Use 1-secure MPC
  - Commit to views of parties + channels
  - Open one view and incident channels
- Handle MPC with error via coin-flipping
- Variant: Directly get 2^{-k} soundness error via security in malicious model
  - n=O(k) parties
  - Ω(n)-security with abort
  - Broadcast is “free”
- Realize Com using OWF

Stay tuned for the second talk!

General Two-Party Protocols [IPS08]

- Life is easier when everyone follows instructions…
- GMW paradigm [GMW87]:
  - Semi-honest-secure π → malicious-secure π'
  - Use ZK proofs to prove "sticking to protocol"
- Non-black-box: ZK proofs in π' involve code of π
  - Typically considered "impractical"
  - Not applicable at all when π uses an oracle
    - Functionality oracle: OT-hybrid model
    - Crypto primitive oracle: black-box PRG
    - Arithmetic oracle: black-box field or ring
- Is there a "black-box alternative" to GMW?

A Dream Goal



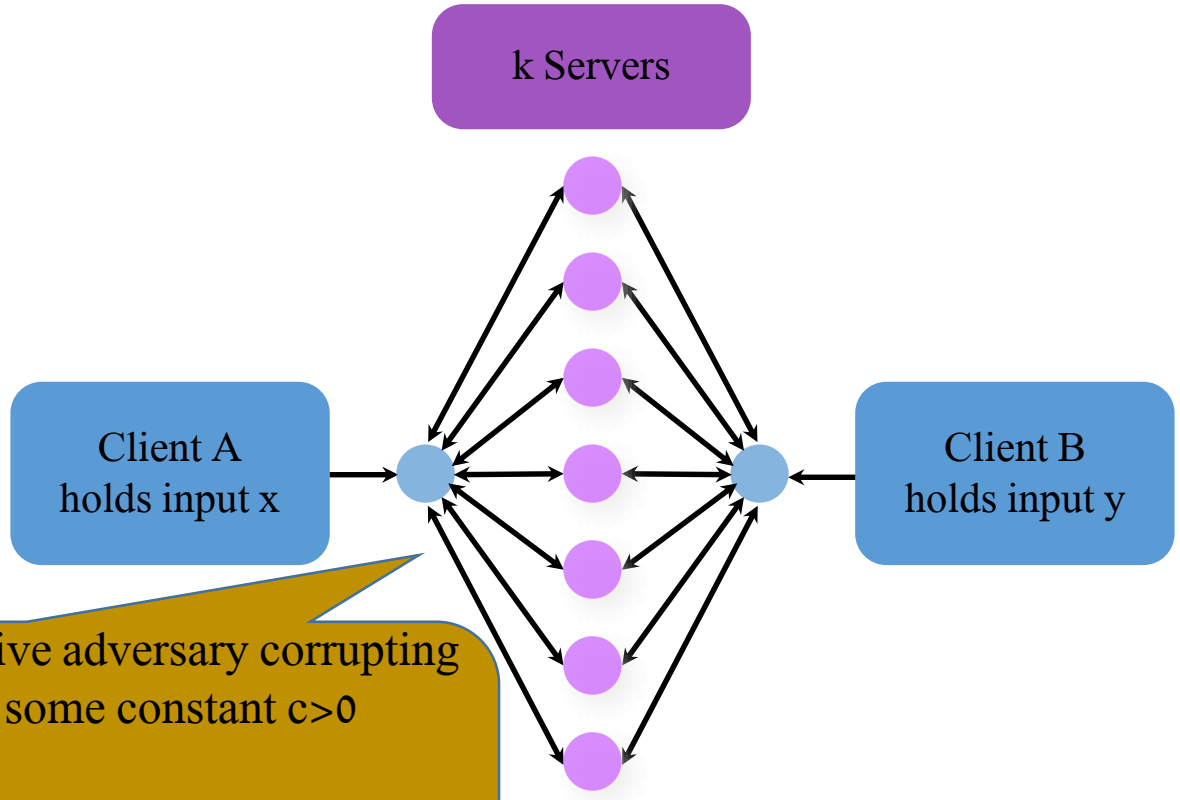
- Possible for some fixed f
  - e.g., OT [IKLP06,Hai08]
- Impossible for general f
  - e.g., ZK functionalities

Idea

- Combine two types of "easy" protocols:
  - Outer protocol:
    honest-majority MPC
  - Inner protocol:
    semi-honest 2-party protocol
    - possibly in OT-hybrid model

- Both are easier than our goal
- Both exist unconditionally

Outer Protocol



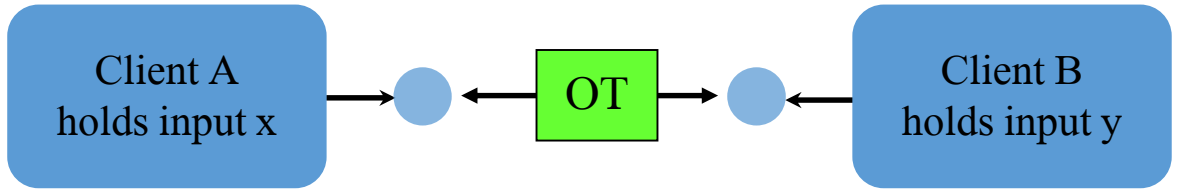
Secure against malicious adaptive adversary corrupting one client and t=ck servers, for some constant c>0

Security with abort suffices

Straight-line simulation

Example: "BGW-lite"

Inner Protocol

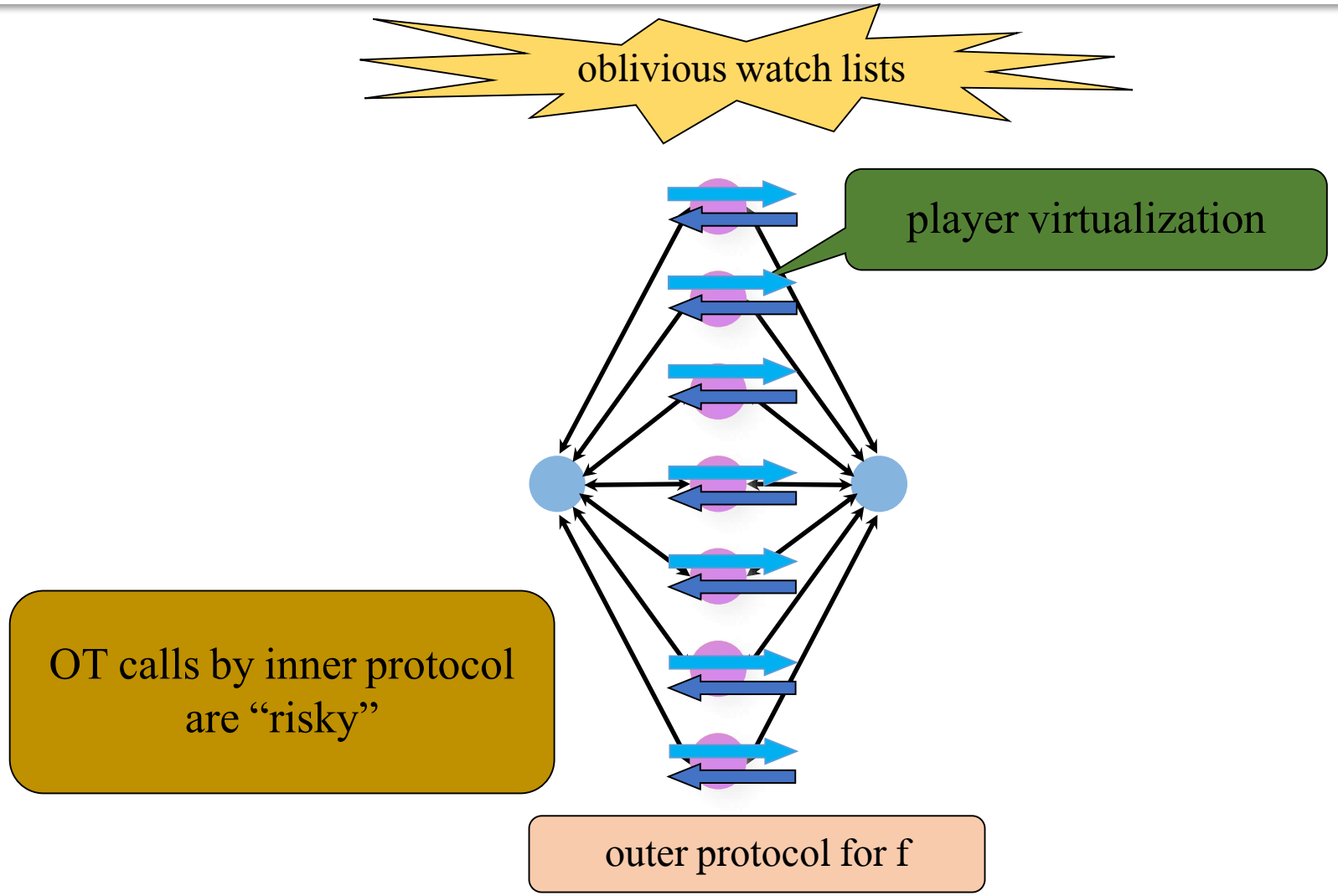


Secure against semi-honest adversary

(Adaptive security w/erasures)

Example: "GMW-lite"

Combining the Two Protocols



A Closer Look at Server Emulation

- Assume servers are deterministic
  - This is already the case for natural protocols
  - Can be ensured in general with small overhead
- In outer protocol, server i
  - Gets messages from A and B
  - Sends messages to A and B
  - May update a secret state
- Captured by reactive 2-party functionality F_i
  - Inputs = incoming messages
  - Outputs = outgoing messages
- Use semi-honest protocol for F_i
  - Distribute server's computation between clients
  - "Local" computations do not need to be distributed

A Closer Look at Watchlists

- Inner protocol can't prevent clients from cheating by sending "bad messages"
  - Bad randomness handled via simple coin-tossing
- Watchlist mechanism ensures that cheating does not occur too often
  - Client doesn't know which instances of inner protocol are watched
  - Client cheats in $\leq t/2$ instances
    → cheating tolerated by t-security of outer protocol
  - Client cheats in $> t/2$ instances
    → will be caught with overwhelming probability
- "Cut-and-choose gone live"

Setting up the Watchlists

- Each client picks n long one-time pads R_i
- $|R_i|$ = length of messages + randomness in execution of i-th inner protocol
  - Short PRG seed suffices for computational security
- Each client uses OT to select ~ t/2 of the other client's pads R_i
- Implemented via Rabin-OT for each server
  - Reduces to a constant number of 1-out-of-2 string-OTs per server
  - With overwhelming probability, p fraction of R_i are received

Using the Watchlists

- **Consider here B watching A**
  - A watches B symmetrically
- A uses sequential parts of each R_i to mask her (progressive) view of the i-th inner protocol
  - If B obtained R_i, it has full view of i-th inner protocol
  - Can detect (and abort) as soon as A cheats
  - What about ideal OT calls in inner protocol?
    - Cheating caught w/prob ½ if OT inputs are random
    - Use OT to random-OT reduction

Stay tuned for the second talk!

Simulation (High Level)

- Suppose A is corrupted in final protocol
- Main simulator runs outer simulator to
  - Extract input of A
  - Generate outer protocol messages from B
  - Generate full view of inner protocols watched by A (requires corrupting ~ t/2 servers)
  - Generates A's inputs and outputs in other inner protocols (communication of A with servers)
    - Feed to inner simulator to generate inner protocol view
    - Valid as long as A does not deviate from inner protocol
- Main simulator can observe deviation from inner protocol
  - When A cheats on i-th inner protocol, outer simulator corrupts i-th server and main simulator aborts w/prob. p

A New Protocol Compiler

- **Given a 2-party functionality F**
  - Get an honest-majority-secure outer protocol Π for the functionality F (with 2 clients and k servers)
  - Get a semi-honest-secure inner protocol ρ^{OT} for a 2-party functionality G^{Π} corresponding to the servers' program in Π

    (G^{Π} is a reactive functionality defined black-box w.r.t Π)

- **Our (2-party) protocol Φ^{OT}, with black-box access to Π and ρ, is a malicious-secure protocol for F**

Applications

- Revisiting the classics
  - BGW-lite + GMW-lite → Kilian
- Efficient MPC with no honest majority
  - O(1) bits per gate in OT-hybrid model (+ additive term)
  - All crypto can be pushed to preprocessing
- Constant-round MPC^{OT} (t<n) using black-box PRG
  - Extending 2-party "cut-and-choose" Yao
- Efficient OT extension
- Constant-rate b.b. reduction of OT to semi-honest OT
- Constant-rate OT combiners
- Secure arithmetic computation over black-box fields/rings
- Protocols making black-box use of (fully) homomorphic encryption