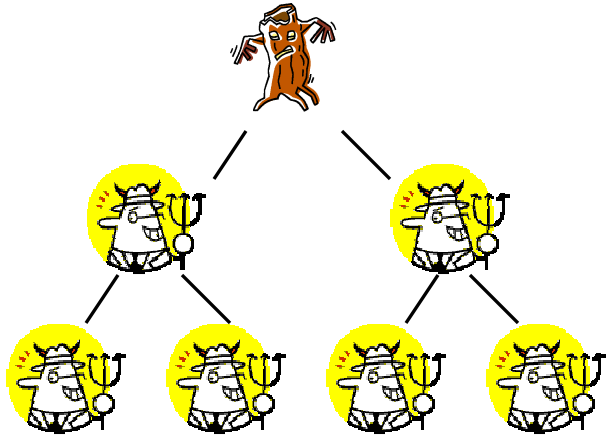
Collusion-Free Multiparty Computation in the Mediated Model

Joël Alwen (NYU)
Jonathan Katz (U. Maryland)
Yehuda Lindell (Bar-Ilan U.)
Giuseppe Persiano (U. Salerno)
abhi shelat (U. Virginia)
Ivan Visconti (U.Salerno)

Crime

Organized Crime



Standard Crypto Model:
Single adversary coordinating all corrupted parties.

Why Standard Crypto Model Assumes Organized Crime

Intuition: Protect against strongest adversary

On the other hand, unclear how to avoid it in standard communication models.

How to Coordinate

1. Security requires randomness
2. Randomness enables side channels
3. Side channels imply collusion

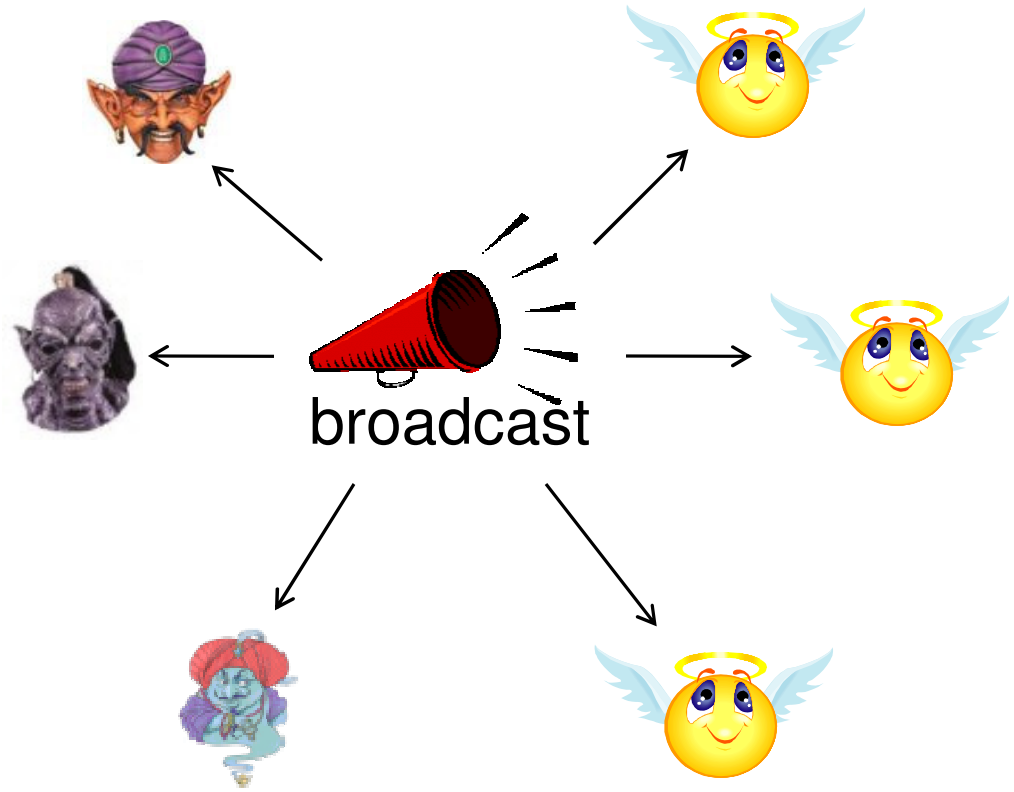
ERGO, organized crime.

Collusion-free protocol

"The protocol does not **introduce** any opportunities for parties to collude."

Solution Concept

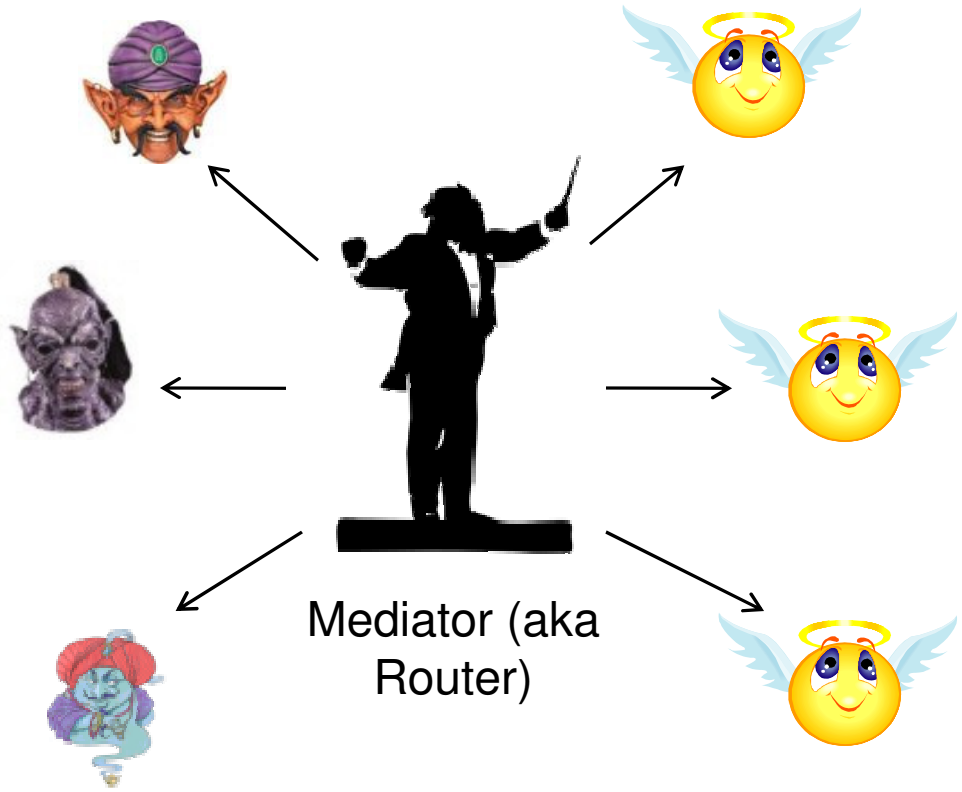
Standard Model



Problem: "Randomness enables side channels"

Solution: Re-Randomize

Mediated Model

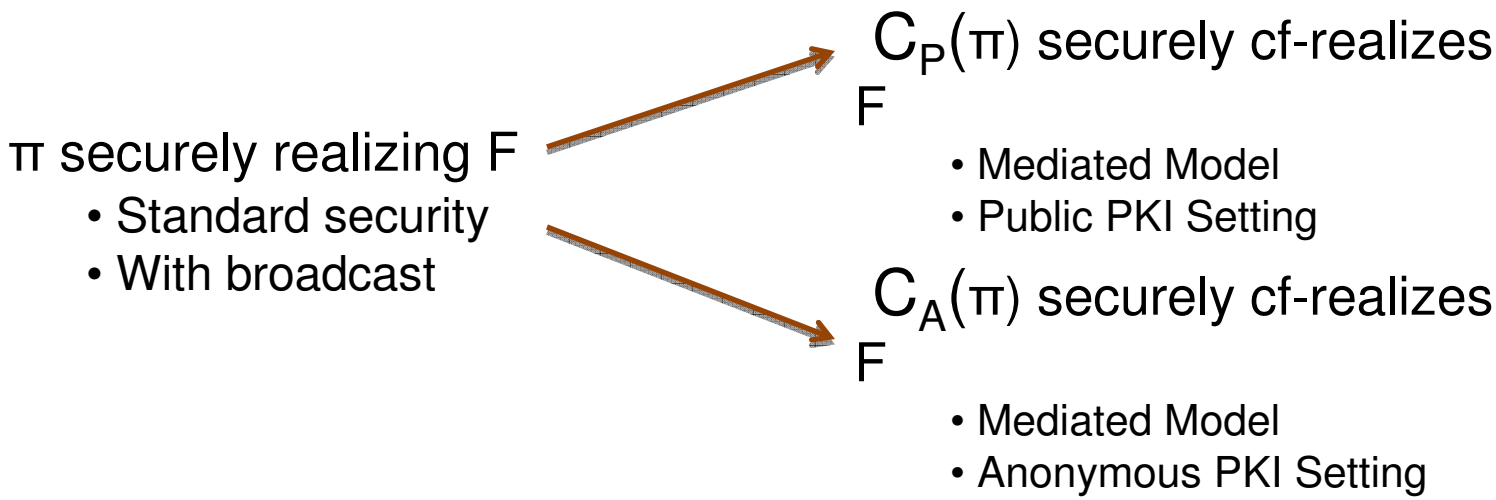


But not a TRUSTED PARTY

Main Results

1. Improved definition of Collusion-free

2. Give protocol compilers C_P and C_A:



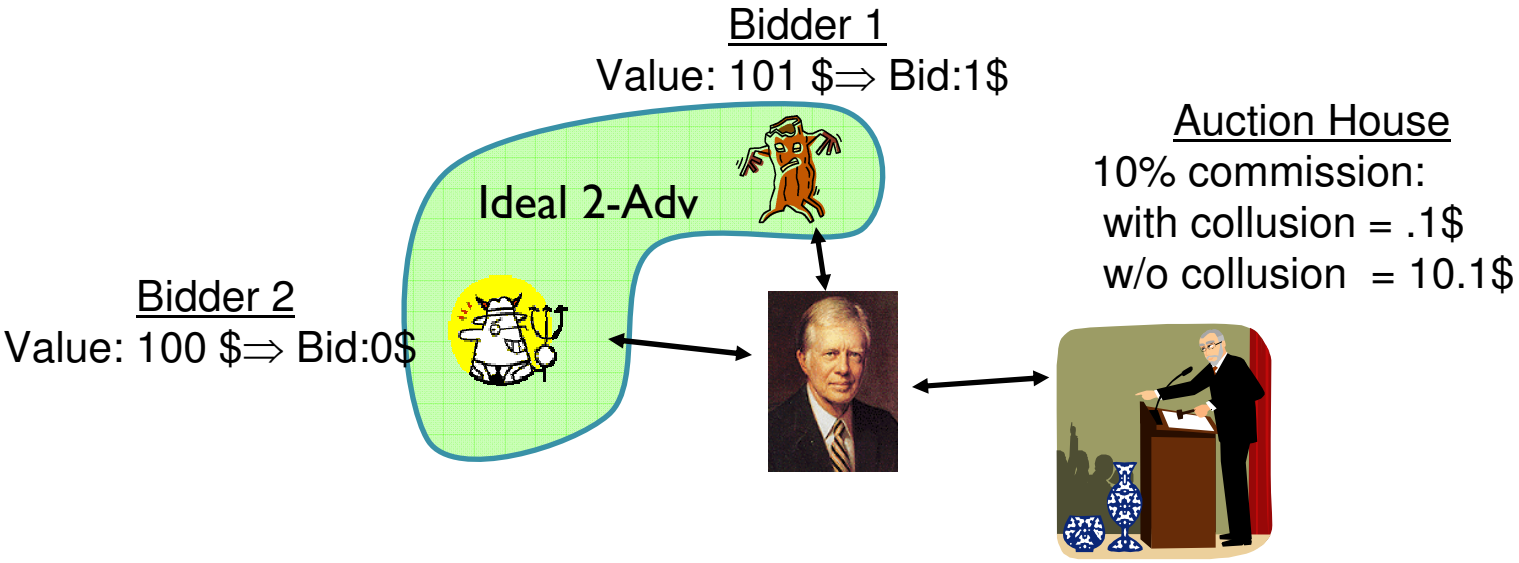
Result: Collusion-free computation for any n-party functionality.

Motivation: Auction

Parties: n bidders, auction house

Collusion: Bidders decide amongst themselves who is willing to bid the most. Winner bids 1$, rest bid 0$.

Result: auction house's commission diminished



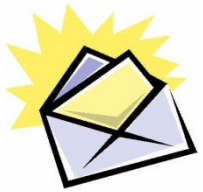
Motivation: Applications to Game Theory

- Implementing Nash Equilibria
  - Weak Stability: Unilateral deviations are irrational.
- Playing Bayesian Games
  - i.e. games with secret input
    - e.g. valuation of an item by a bidder in an auction
- Playing games of Imperfect Information
  - i.e. games in which players do have full knowledge of the current global state.
    - e.g. hidden cards in opponents hand in poker
- More generally: Playing Mediated Games
  - i.e. games with isolated players talking only to a trusted mediator

Previous Work

Main Goal: Enforce isolation. Avoid steganography.

- Steg.-free Signatures: [S83,D96,S96,BDI+96,BS05]
- Collusion Free MPC: Verifiable Determinism
  - Initiated by Lepinski, Micali, shelat at STOC'05
  - Other works [LMS05b, ILM05, ILM08]
  - Make use of strong physical assumptions


- New Approach: Rerandomization [ASV08]
  - In the Mediated Model
    - Network model still strong assumption
    - But allows for computation with Turing Machines
  - Commitments and Zero Knowledge

Definitions

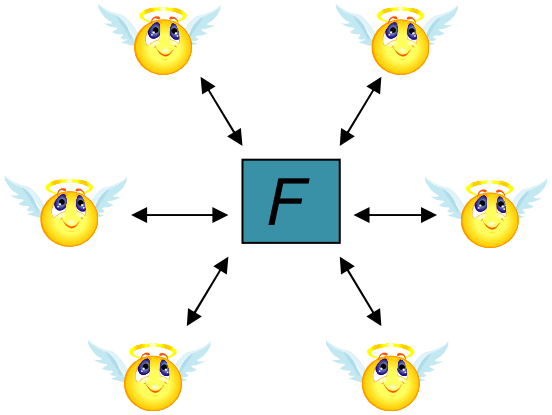
Multiparty Computation

"Protocol Π realizes functionality F"

Ideal

Players

1) Get Private Input
2) Send it to "Ideal Functionality" F



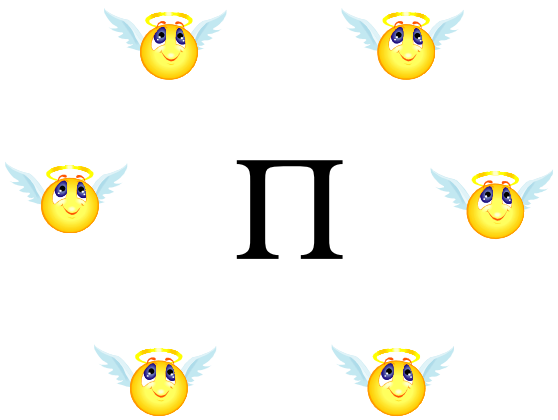
3) Receive Private Output

$\approx$

Real

Players

1) Get Private Input
2) Interact (run protocol Π)

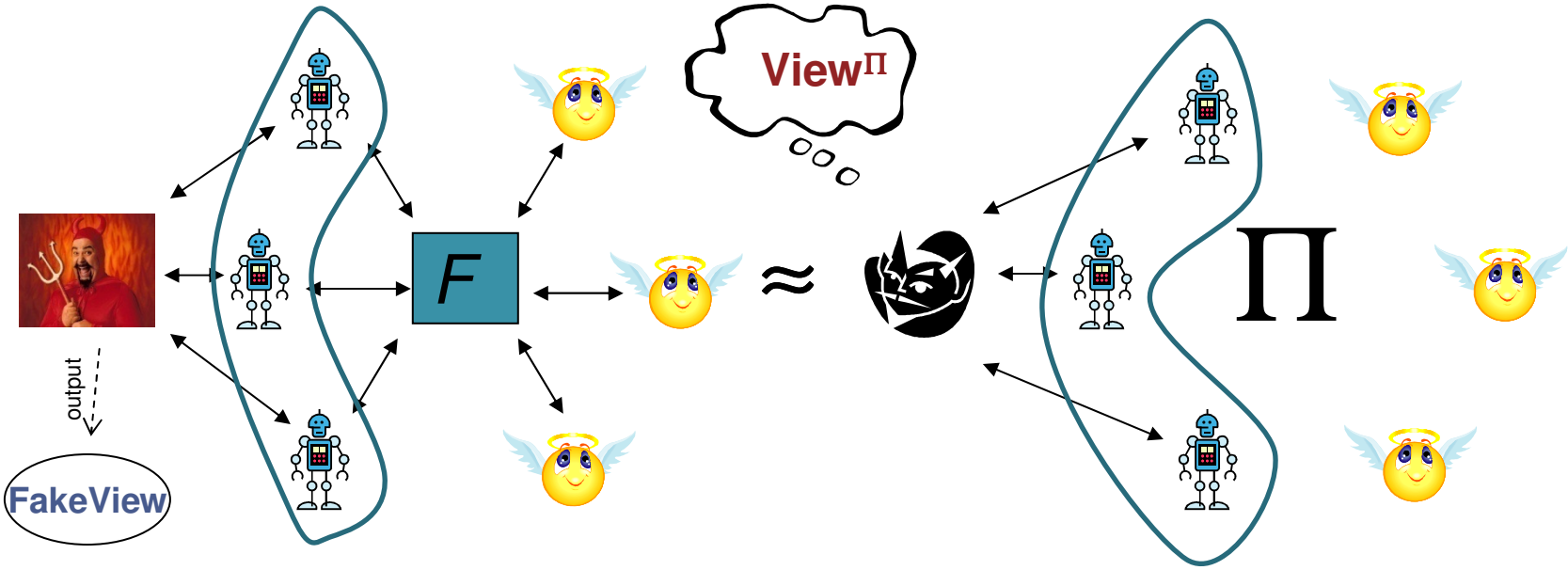


3) Compute Private Output

F can be probabilistic, and/or reactive with a secret persistent internal state.

(Traditional) Monolithic Adversary

- Model Real: All corrupt real parties controlled by a single malicious adversary.
- Model Ideal: All corrupt ideal parties controlled by a single simulator.

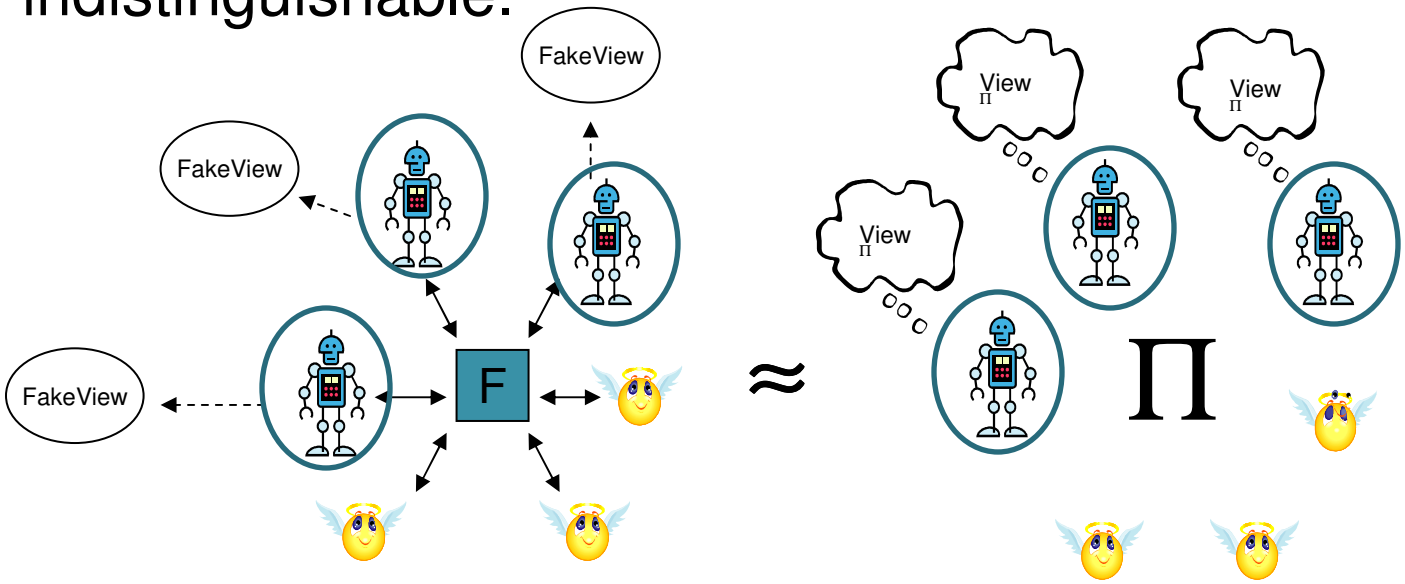


- Π is *secure* (power preservation) if for any malicious adversary there exists a simulator that outputs a (fake) view such that:

$$\{\text{FakeView}, \text{Ideal-I/O}\} \approx \{\text{View}^{\Pi}, \text{Real-I/O}\}$$

Modeling Collusion Free MPC

- Idea: Corrupt players act independently. Each has its own simulator. Joint "fake views" still remain indistinguishable.

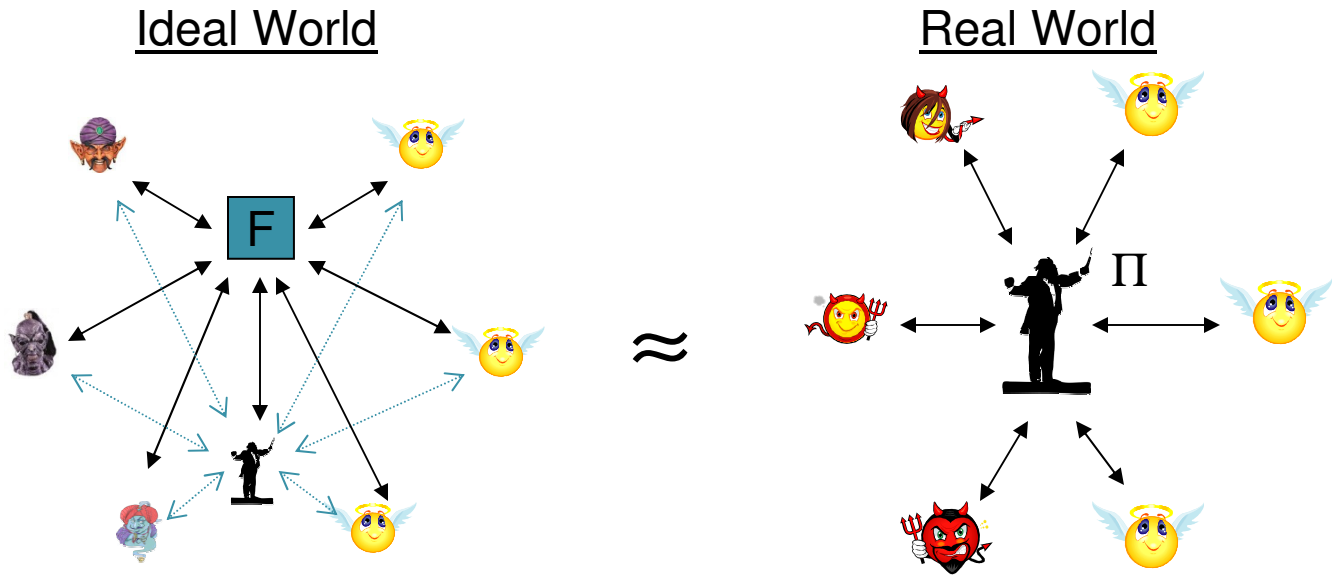


{ {FakeView}, Ideal-I/O} ≈ { {View$^{\Pi}$}, Real-I/O}

Anything they can compute together with Π they can also compute with F.

The Mediated Model

- New Communication Model
  - Communication channel modeled as turing machine (called *mediator*)
  - The mediator can also have input to F



F : Uncorruptable (ideal) functionality

: Honest parties do not use blue communication lines (corrupted ones can)

: Mediator honest ⇒ ideal players separate

Mediator corrupt ⇒ standard security (monolithic adversary)

Establishing Identities

We explore two settings:

- Anonymous Setting: Identities setup after inputs determined
  - Achieves stronger notion of collusion-freeness.
  - Requires more trust in mediator
  - Implementation:
    1. Parties generate key pairs and send their public key to mediator.
    2. For each player the Mediator sends a vector of fresh independent commitment to all public keys.

- Public PKI Setting: PKI setup before inputs determined
  - Each player knows the identity (public keys) of all other payers involved in the execution.
  - More practical (realistic).
  - Implementation:
    1. Parties generate keys and send public keys to trusted setup TTP.
    2. TTP redistributes all public keys consistently.

Note: Neither setting requires honest key generation or proof

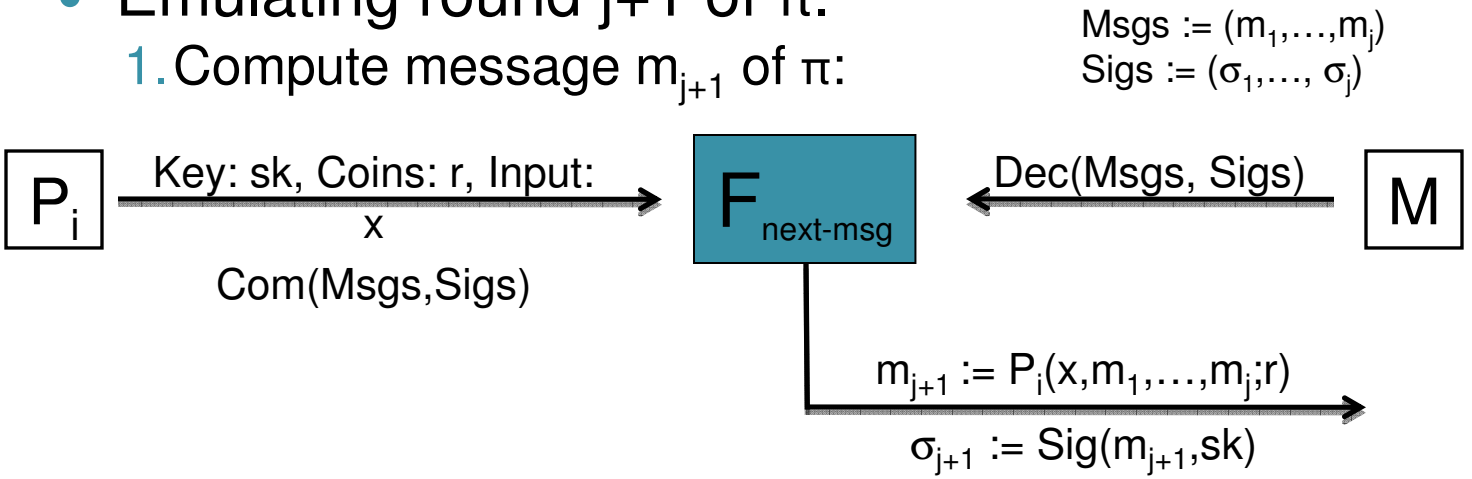
Assumptions and Tools

- π is n-party protocol
  - Securely computes F.
  - Plain model with broadcast channel
    - W.l.o.g. assume all messages sent via broadcast.
- Primitives
  - Signatures.
  - Perfectly binding Commitments.
- 2-party (bounded) concurrently self-composable protocols.
  - SFE.
  - ZK protocol.

High Level Idea

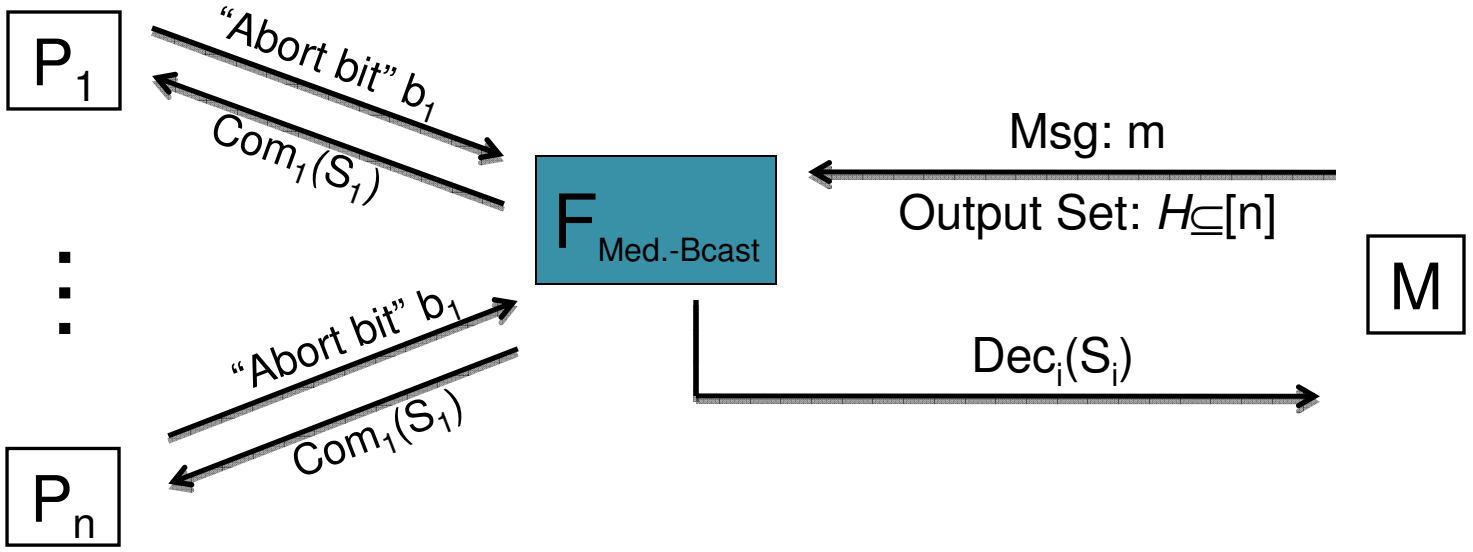
- Jointly emulate an execution of π.
  - Mediator maintains list of π-messages received by each player.
  - Players maintain only their random tapes, signing keys, and inputs to π.
  - Emulation proceeds as a sequence of two party computations between a player and the mediator.

- Emulating round j+1 of π.
  1. Compute message m_{j+1} of π:



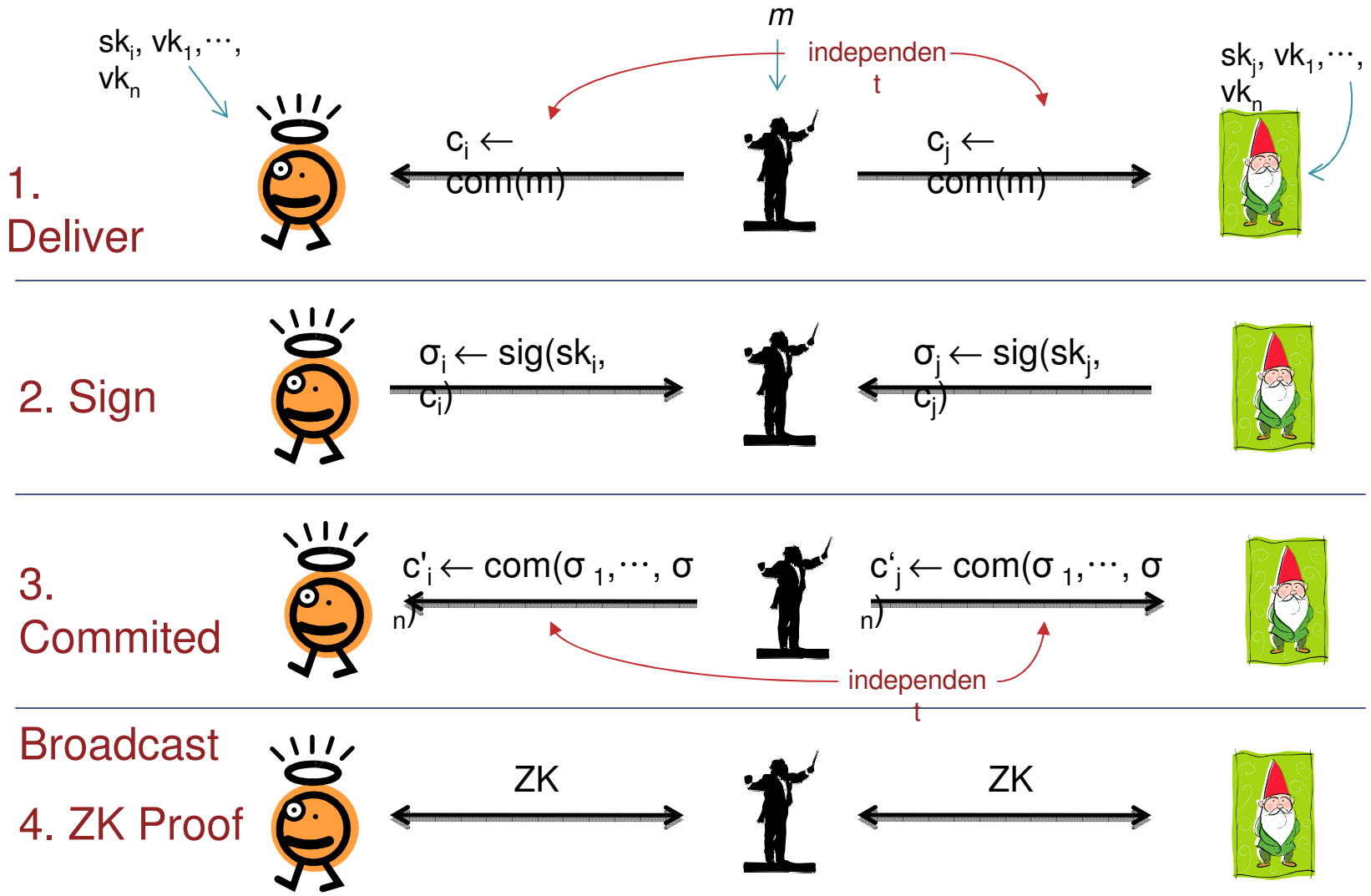
  2. Emulate broadcast of $m'_{i+1} := (m_{i+1},\sigma_{i+1})$.

Mediated Broadcast Functionality



1. If at least one P_i set $b_i = 1$ then all $S_i := \perp$
2. If $i \notin H$ then $S_i := \perp$
3. Else $S_i := m$

Mediated Broadcast



Statement: c' is com of (valid) sig of com of same message

Side-channels

- SFE input privacy, Com hiding and ZK properties imply π-messages (nor sigs) ever seen by players.

  $\Rightarrow$ Players views remain independent of each other until output is delivered.

- Using aborts to communicate
  - [ASV08] allows log(# rounds) bits of communication via aborts.
  - This work: 1 bit at end of computation.
    - How: Mediator uses default messages for aborting party and emulation of π continues until output delivery.
    - Result: Round # of abort remains hidden. Only bit communicated is that an abort occurred at some point.

Honest but Curious Mediator

- π secure against passive (eves dropping) adversary & 2-party SFE's input privacy
  ⇒ Mediator learns nothing about I/O of players.

- Mediator removes side channels.
  ⇒ Corrupt players can not communicate or coordinate.

- Result: Compiled protocol is a collusion-free secure realization of F.

Corrupt Mediators

- Mediator controls scheduling

  $\Rightarrow$ Require bounded (by n) concurrent security for 2-party SFEs and for ZK.

- π secure against active adversary

  $\Rightarrow$ F realized faithfully. (Correctness)

  $\Rightarrow$ Privacy of honest players maintained.

- Corrupt players can communicate via corrupt mediator.

  $\Rightarrow$ Security falls back to standard monolithic adversary security.

Open Problems

- Efficient constructions (esp. for specific functionalities such as auctions).
- Alternative (yet more realistic) models where similar results are possible.
- Security & Collusion-Freeness under stronger composition.
- Anonymous settings with reduced trust in mediator for setup phase.