1/p-Secure Multiparty Computation without Honest Majority and the Best of Both Worlds

- Amos Beimel (**BGU**)
- Eran Omri (**BIU**)
- Yehuda Lindell (**BIU**)
- **Ilan Orlov (BGU)**

Our Results in a Glance

- We explore **1/p**-secure multiparty protocols **without** an honest majority
- Positive result:
  - **1/p**-secure protocols for **constant** number of parties for computing any function with polynomial-sized range tolerating any number of corrupt parties
- Impossibility result:
  - There is no general **1/p**-secure protocol for **non-constant** number of parties
- Best of both worlds:
  - A single protocol that
    - Honest majority → Full security
    - No honest majority → **1/p**-security

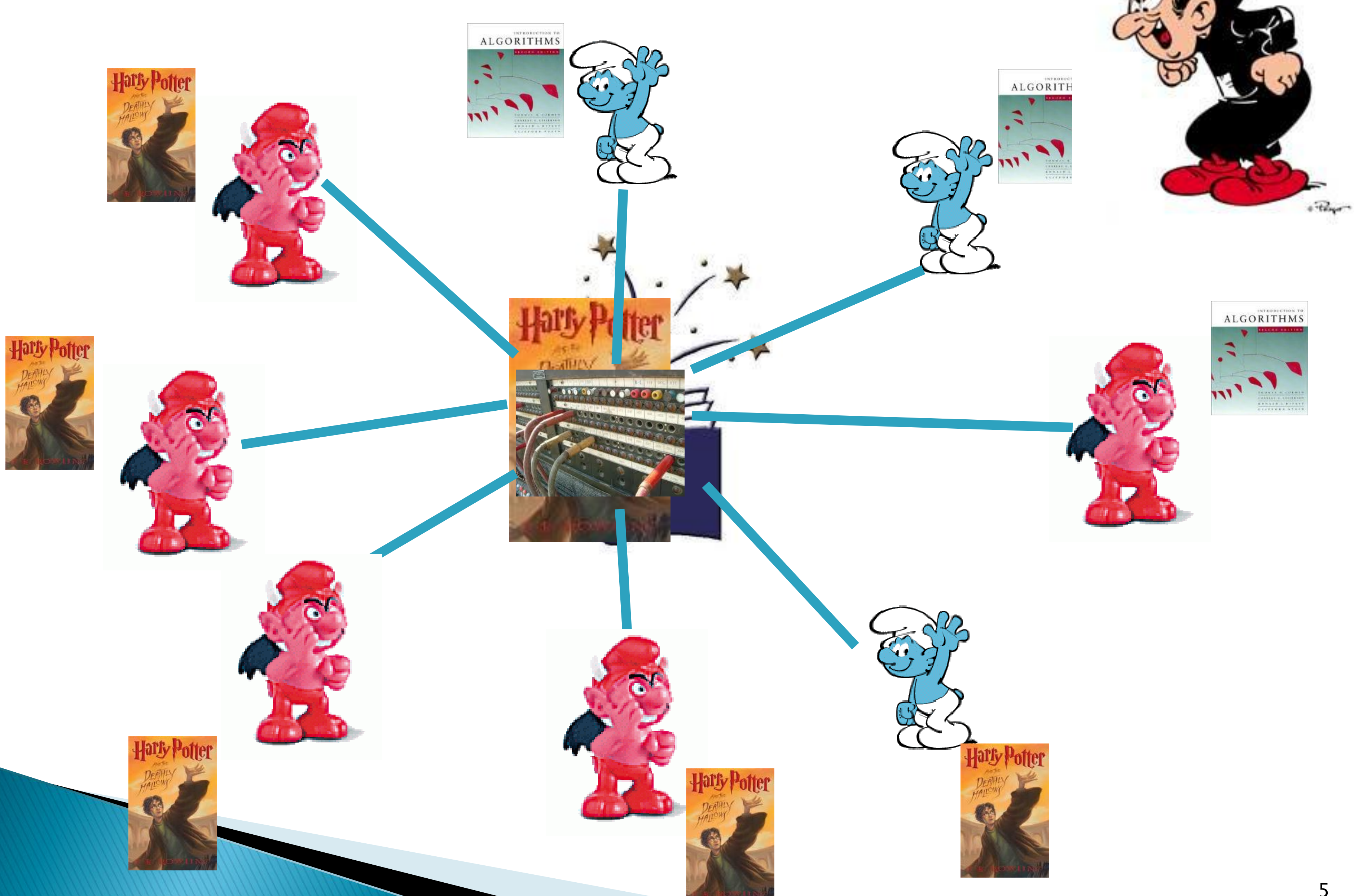
Talk Outline

- Background
- Our results
- The ideas of our protocol
- Summary and Open Problems

A Motivating Story

16/8

The Goal



The Model

- **m** parties
- **r**-round protocol
  - **r=poly**(security parameter)
- Adversary:
  - Polynomial time
  - Malicious – corrupts and controls some of the parties
  - Rushing adversary
    - In each round:
      - Sees all messages of honest parties
      - Chooses and sends messages on behalf of malicious parties
        - Can depend on the messages of honest parties
  - More realistic than simulations channels
- Broadcast channel

Security Definitions

- The security definitions involve a comparison between two worlds:

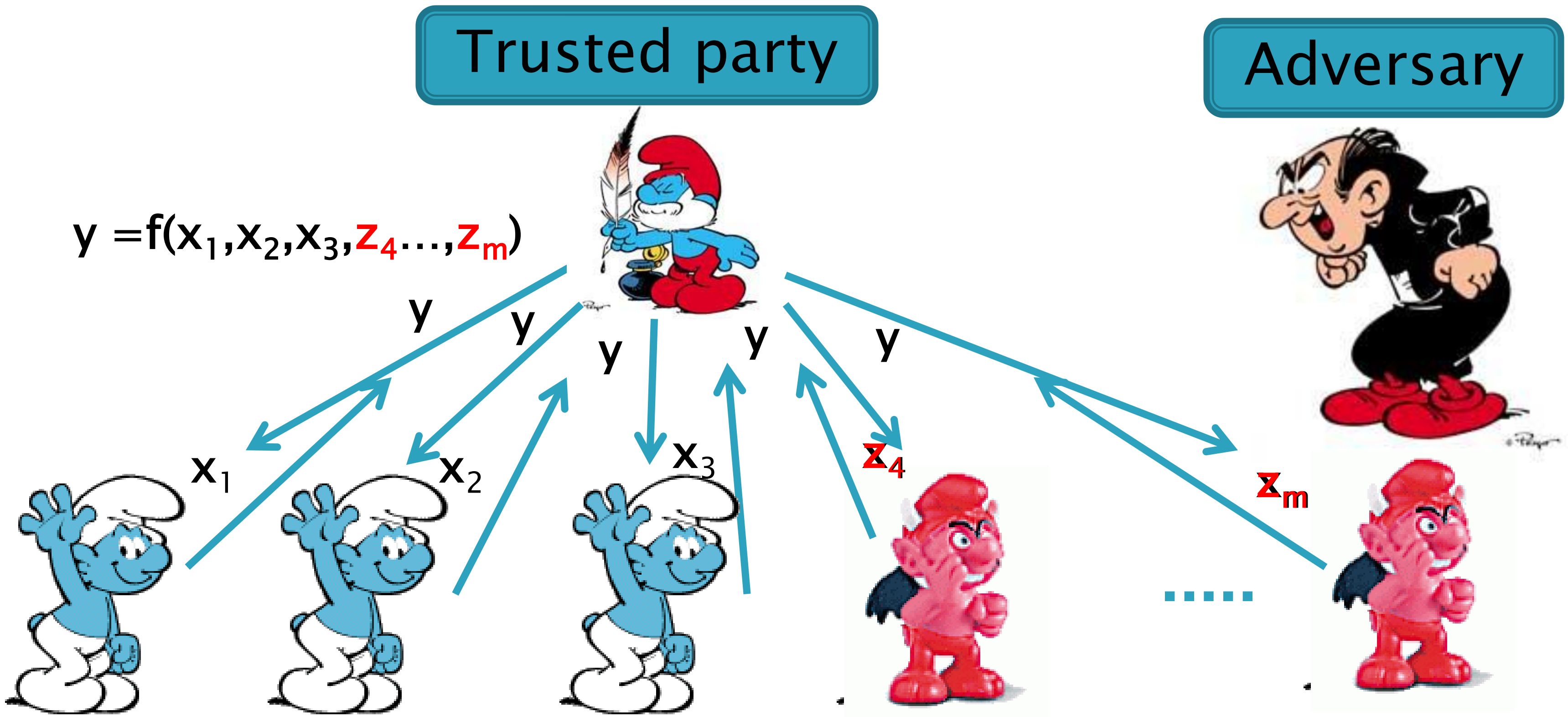
Ideal World

There is a trusted party that helps with the computation

Real World

The protocol

Ideal Computation of a Function



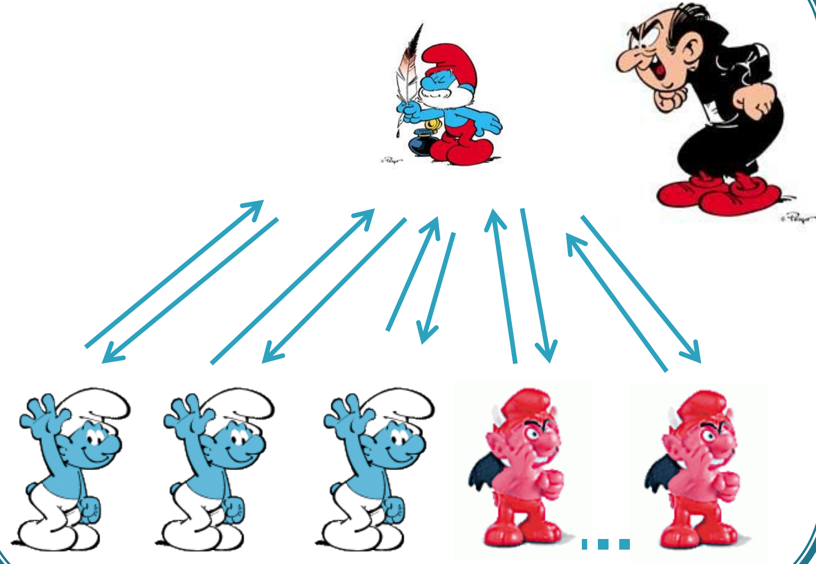
- Guarantees many nice properties:

  Privacy, correctness, and **Fairness**

  (fairness = corrupt parties get the output ⇨ the honest parties get the output)

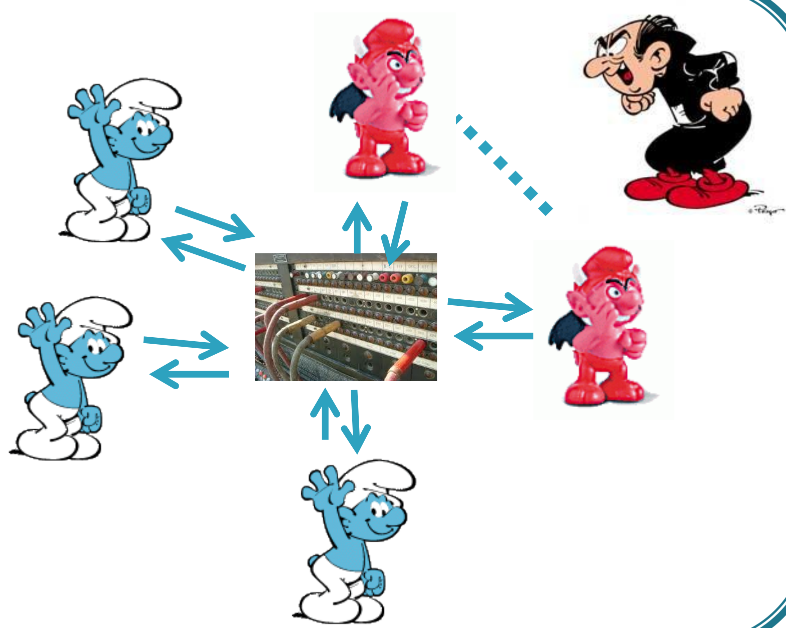
Secure Computation – Full Security

Ideal World



$\approx$

Real World



Security Requirement:

No REAL world adversary can do more harm than IDEAL world adversary

Is Full Security Achievable?

- [GoldreichMicaliWigderson87]: Any polynomial-time **F** can be computed with full security **with** an honest majority

- [Cleave86]: Any **r**-round **m**-party coin-tossing protocol has bias $\Omega(1/\mathbf{r})$ **without** an honest majority

- **Conclusion**: impossible to achieve full security **without** an honest majority for general functionalities

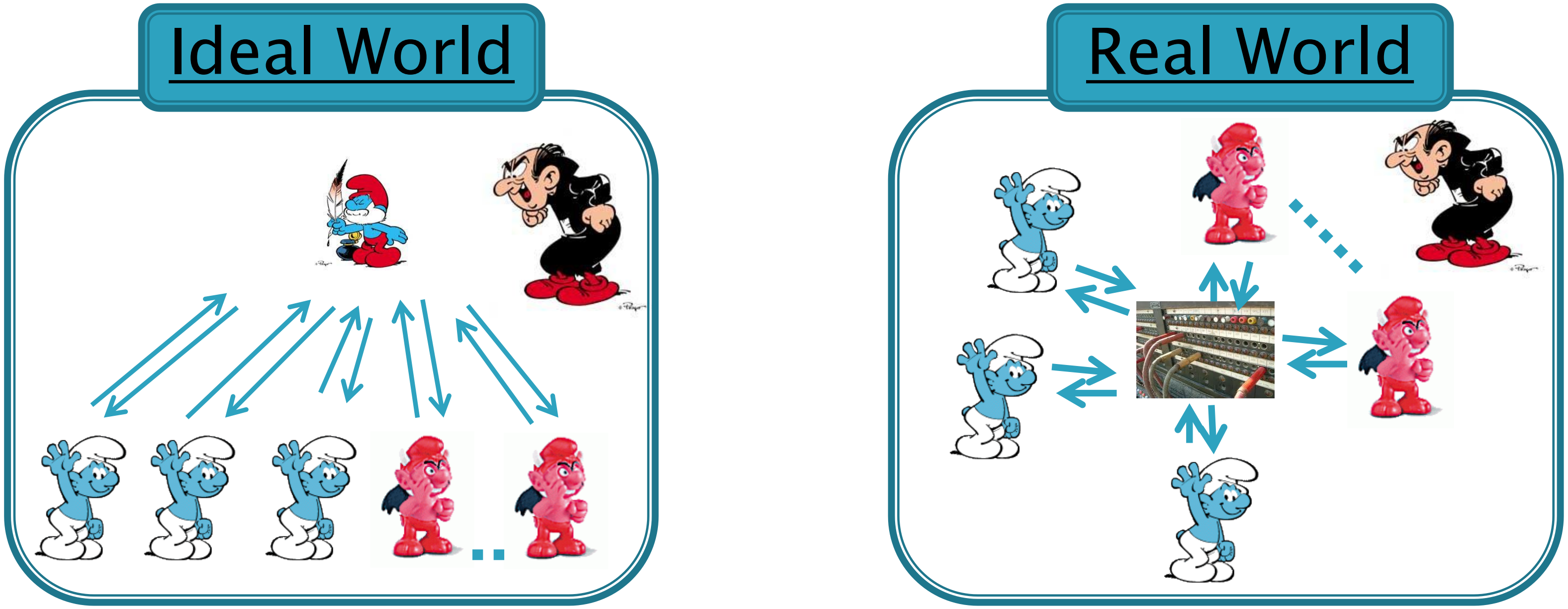
What Can Be Achieved Without an Honest Majority ?

- [GMW87]: Security-with-abort
  - Achieved without an honest majority
  - Does not provide **ANY** fairness!!
    - The adversary can learn the output, while the honest parties learn noting

Can we get reasonable fairness without honest majority?

1/p-Security [Gordon, Katz 2010]

- Compare the previous two worlds:



- Full security – REAL fully emulates IDEAL
- 1/**p**-security – REAL emulates IDEAL within "computational distance" of at most 1/**p**

1/p-Secure 2-Party Computation [GK10]

- For every function **F**, where the size of domain or range is polynomial, there exists a **1/p**-secure 2-party protocol
  - For every polynomial **p**

- <u>Impossibility</u>: Domain or range **have to be** polynomial

GK: Can this result be extended to the multiparty case?

Talk Outline

- Background
- Our results
- The ideas of our protocol
- Summary and Open Problems

Our Main Result

Theorem: For every function **F**, where

1. Number of parties **m** is constant
2. Size of range of **F** is polynomial

there exists a **1/p**-secure protocol that tolerates up to **m**−1 corrupt parties

- For every polynomial **p**

Informally: We constructed 1/p-secure protocols for constant number of parties

Also when

1. No. of corrupt parties < 2**m**/3
2. **F** is deterministic & size of domain of **F** is constant
3. **m=O(log log n)**

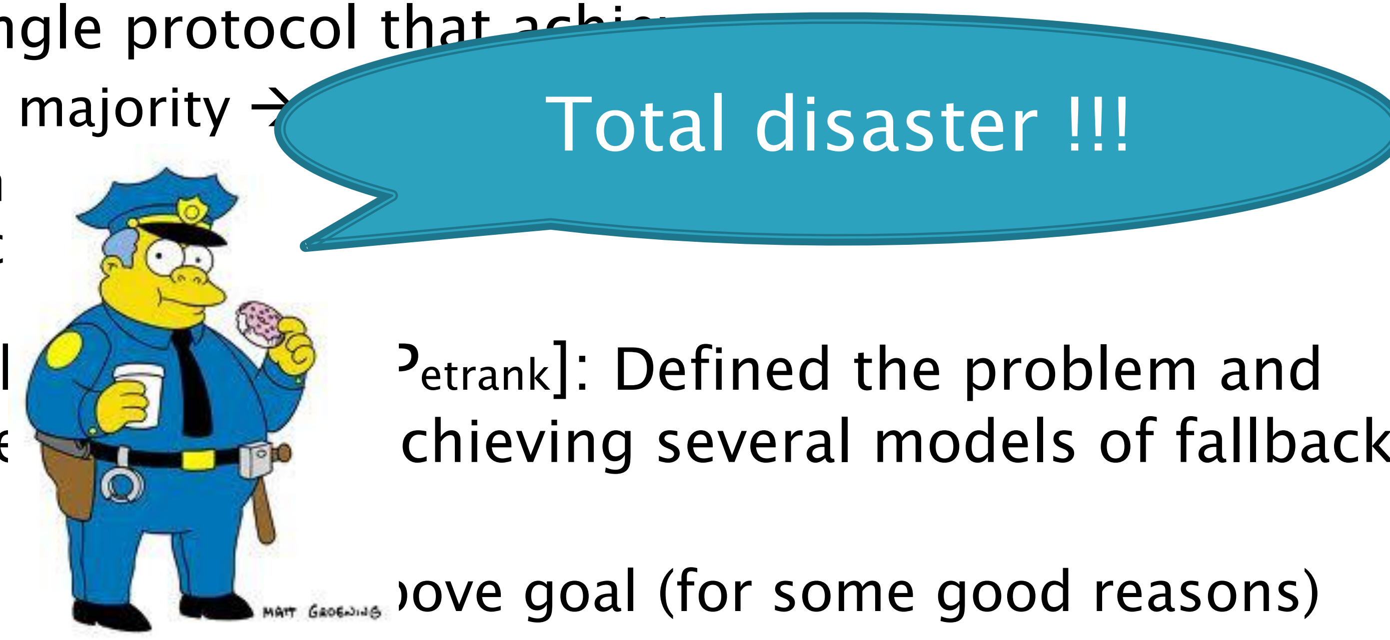
An Impossibility Result

- Special case of possibility result: There exists a $1/p$-secure protocol when
  - $\mathbf{m}$ is constant
  - $\mathbf{F}$ is deterministic
  - |Domain| of each party is polynomial

- Impossibility: Such protocol is not possible when $\mathbf{m}$ is non-constant
  - Explains why $\mathbf{m}=O(1)$ in our result

Best of Both Worlds

- [GMW 87]: Any polynomial-time **F** can be computed by a protocol with full security with an honest majority
- If there is no honest majority, the above protocol does not guarantee any security
- Goal: Single protocol that achie
  - Honest majority →
  - No hon (fallbac
- [IshaiKatzl Petrank]: Defined the problem and suggeste chieving several models of fallback security
  - Do not ove goal (for some good reasons)



Our Results: 1/p-Security is Possible as a Fallback

- For every function F for **m** parties, if
  1. Both the domain and the range are polynomial
  2. **m** is constant

  then, there exists a (single) protocol
  - Honest majority → Full security
  - No honest majority → 1/**p**-security
- This is best of both worlds!
- Secure-with-abort **is not possible** as a fallback [IKKLP]
  - Strong motivation for 1/**p**-security

Informally:

1/p-security is possible as a fallback security for constant number of parties

Talk Outline

- Background
- Our Results
- The Ideas of Our Protocol
- Summary and Open Problems

The Structure of Our Protocol

- The protocol has 2 steps:
  - Preprocessing step
  - **r** rounds of interaction

- Prepressing: The parties execute a secure-with-abort protocol:
  - The parties input their inputs
  - Receive a set of shares and signed messages for executing an **r**-round protocol

- Rounds of Interaction: There are **r** rounds, in each round:
  - Each party broadcasts its message
  - Each subset of parties learns a value
  - The value is used if other parties abort

The Structure of Our Protocol (2)

- There is a special round, called **i***
  - After round **i***, each subset of parties receives the actual output of **F**
  - Before round **i***, each subset of parties receives a value that depends only on its inputs
- To cause "computational distance", the adversary must guess **i***
- The value of **i*** is concealed
- This structure was used in previous constructions: [IKLP06, Katz06, GK06, GHKL06, MNS09, GK10, BOO10, …]

New Challenges and New Ideas

- How to conceal the value of **i*** in a multiparty setting?
- How to deal with any possible abort of any subset?
- Some of the solutions:
  - The information is shared in a few layers of secret sharing
  - After an abort, the remaining parties execute a protocol
    - This protocol has to conceal **i***

Talk Outline

- Background
- Our Results
- The Ideas of Our Protocol
- Summary and Open Problems

Summary

- We explore **1/p**-secure multiparty protocols **without** an honest majority
- Positive result:
  - **1/p**-secure protocols for **constant** number of parties*
- Impossibility result:
  - There is no general **1/p**-secure protocol for **non-constant** number of parties*
- Best of both worlds
  - Single protocol that
    - Honest majority → Full security
    - No honest majority → **1/p**-security

* Some restriction might apply


The Future
NEXT EXIT

Open Problems

- Is there a $1/p$-secure protocol for **F** with non-constant number of parties and polynomial-sized range and domain?
- Are there more efficient $\mathbf{1/p}$-secure protocols?
- Can we guarantee full-privacy and partial fairness in secure multiparty computation without an honest majority?
  - $\mathbf{1/p}$ security: With prob. $\mathbf{1/p}$ privacy can be totally lost
  - Maybe suggest new definitions?

Thank you !