MPC Complexity

Manoj Prabhakaran :: IIT Bombay

The World of Functionalities

The World of Functionalities

- Distributed functions display interesting features the are not apparent when they are not distributed

The World of Functionalities

- Distributed functions display interesting features the are not apparent when they are not distributed
  - Classical example: Communication Complexity [Yao]

The World of Functionalities

- Distributed functions display interesting features the are not apparent when they are not distributed
  - Classical example: Communication Complexity [Yao]
  - MPC provides another lens to look at the complexity of functions

Complexity w.r.t. MPC

Complexity w.r.t. MPC

- We saw OT is complete for MPC
  - Any other functionality can be reduced to OT
  - Under all notions of reduction (passive-secure, or UC secure)

Complexity w.r.t. MPC

- We saw OT is complete for MPC
  - Any other functionality can be reduced to OT
  - Under all notions of reduction (passive-secure, or UC secure)
- The Cryptographic Complexity question:
  - Can F be reduced to G (for different reductions)?

Complexity w.r.t. MPC

- We saw OT is complete for MPC
  - Any other functionality can be reduced to OT
  - Under all notions of reduction (passive-secure, or UC secure)
- The Cryptographic Complexity question:
  - Can F be reduced to G (for different reductions)?
  - G **complete** if everything reduces to G

Complexity w.r.t. MPC

- We saw OT is complete for MPC
  - Any other functionality can be reduced to OT
  - Under all notions of reduction (passive-secure, or UC secure)
- The Cryptographic Complexity question:
  - Can F be reduced to G (for different reductions)?
  - G **complete** if everything reduces to G
  - F **trivial** if F reduces to everything (in particular, to NULL)

Quiz

Quiz

- What's the complexity of the following 3 functions, w.r.t, IT passive secure MPC?

Quiz

- What's the complexity of the following 3 functions, w.r.t, IT passive secure MPC?
  - max(x,y)

Quiz

- What's the complexity of the following 3 functions, w.r.t, IT passive secure MPC?
  - max(x,y)
  - [x < y]

Quiz

- What’s the complexity of the following 3 functions, w.r.t, IT passive secure MPC?
  - max(x,y)
  - [x < y]
  - (max(x,y), [x < y])

Complexity w.r.t. MPC

- Several notions of reductions
  - Passive, Active/Standalone or Active/UC
  - Information-theoretic (IT) or PPT
    - If PPT, also specify any computational assumptions used
- Will restrict to 2-party functionalities (mostly SFE)
  - In particular, omitting honest majority security

RECALL

Is MPC Possible?

- Can we securely realize <u>every</u> functionality?
- No & Yes!

Univ. Composable Angel-UC Standalone Passive	All subsets corruptible	Honest Majority
Computationally Unbounded (IT)	No	Yes
Computationally Bounded (PPT)	No Yes Yes Yes	

RECALL

Is MPC Possible?

- Can we securely realize every functionality?
- No & Yes!

> Yes means all are trivial.
> No is more interesting!

Univ. Composable / Angel-UC / Standalone / Passive	All subsets corruptible	Honest Majority
Computationally Unbounded (IT)	No	Yes
Computationally Bounded (PPT)	No / Yes / Yes / Yes	Yes

RECALL

Is MPC Possible?

every functionality?

In fact interesting: What computational hardness assumption makes it switch from No to Yes?

Yes means all are trivial.
No is more interesting!

Univ. Composable Angel-UC Standalone Passive	All subsets corruptible	Honest Majority
Computationally Unbounded (IT)	No	Yes
Computationally Bounded (PPT)	No Yes Yes Yes	

RECALL

Is MPC Possible?

every functionality?

Yes ⇔ sh-OT assumption

Yes means all are trivial.
No is more interesting!

Univ. Composable / Angel-UC / Standalone / Passive	All subsets corruptible	Honest Majority
Computationally Unbounded (IT)	No	Yes
Computationally Bounded (PPT)	Univ. Composable: No Angel-UC: Yes Standalone: Yes Passive: Yes	Yes

RECALL

Is MPC Possible?

Yes ⇔ sh-OT assumption

... every functionality?

Yes means all are trivial.
No is more interesting!

Univ. Composable / Angel-UC / Standalone / Passive	All subsets corruptible
Computationally Unbounded (IT)	No
Computationally Bounded (PPT)	No (Univ. Composable) Yes (Angel-UC) Yes (Standalone) Yes (Passive)

Trivial ones are *really* trivial
(called *Splittable*)

RECALL

An example

- Protocol:
  - Count down from 100
  - At each even round Alice announces whether her bid equals the current count; at each odd round Bob does the same
  - Stop if a party says yes
- Dutch flower auction

RECALL

An example

- Protocol:
  - Count down from 100
  - At each even round Alice announces whether her bid equals the current count; at each odd round Bob does the same
  - Stop if a party says yes
- Dutch flower auction

Perfect Standalone Security
But doesn't compose!

Attack on
Dutch Flower Auction

Attack on Dutch Flower Auction

- Alice and Bob are taking part in two auctions

Attack on Dutch Flower Auction

- Alice and Bob are taking part in two auctions
- Alice's goal: ensure that Bob wins at least one auction and the winning bids in the two auctions are within ±1 of each other

Attack on Dutch Flower Auction

- Alice and Bob are taking part in two auctions
- Alice's goal: ensure that Bob wins at least one auction and the winning bids in the two auctions are within ±1 of each other
- Easy in the protocol: run the two protocols lockstep. Wait till Bob says yes in one. Done if Bob says yes in the other simultaneously. Else Alice will say yes in the next round.

Attack on Dutch Flower Auction

- Alice and Bob are taking part in two auctions
- Alice's goal: ensure that Bob wins at least one auction and the winning bids in the two auctions are within ±1 of each other
- Easy in the protocol: run the two protocols lockstep. Wait till Bob says yes in one. Done if Bob says yes in the other simultaneously. Else Alice will say yes in the next round.
- Why is this an attack?

Attack on Dutch Flower Auction

- Alice and Bob are taking part in two auctions
- Alice's goal: ensure that Bob wins at least one auction and the winning bids in the two auctions are within ±1 of each other
- Easy in the protocol: run the two protocols lockstep. Wait till Bob says yes in one. Done if Bob says yes in the other simultaneously. Else Alice will say yes in the next round.
- Why is this an attack?
    - Impossible to ensure this in IDEAL!

Attack on Dutch Flower Auction

- Alice's goal: ensure that the outcome in the two auctions are within ±1 of each other, and Bob wins at least one auction
- Impossible to ensure this in IDEAL!

Attack on Dutch Flower Auction

- Alice's goal: ensure that the outcome in the two auctions are within ±1 of each other, and Bob wins at least one auction
- Impossible to ensure this in IDEAL!
- Alice could get a result in one session, before running the other. But what should she submit as her input in the first one?

Attack on Dutch Flower Auction

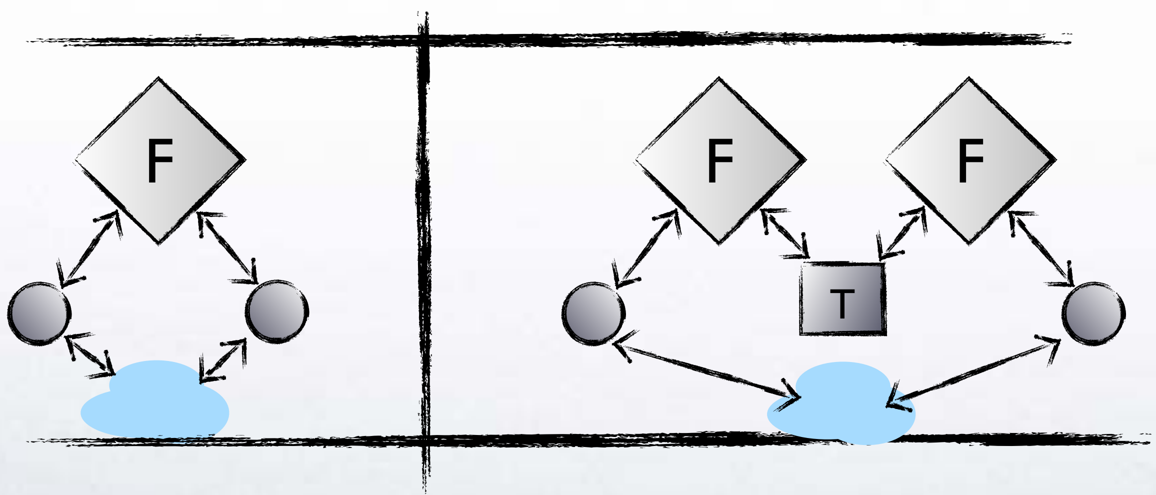
- Alice's goal: ensure that the outcome in the two auctions are within ±1 of each other, and Bob wins at least one auction
- Impossible to ensure this in IDEAL!
- Alice could get a result in one session, before running the other. But what should she submit as her input in the first one?
  - If a high bid, in trouble if she wins now, but Bob has a very low bid in the other session (which he must win).

Attack on Dutch Flower Auction

- Alice's goal: ensure that the outcome in the two auctions are within ±1 of each other, and Bob wins at least one auction
- Impossible to ensure this in IDEAL!
- Alice could get a result in one session, before running the other. But what should she submit as her input in the first one?
  - If a high bid, in trouble if she wins now, but Bob has a very low bid in the other session (which he must win).
  - If a low bid (so Bob may win with a low bid), in trouble if Bob has a high bid in the other session.

UC Triviality:
Splittability

- UC-trivial: "Splittable" [CKL'03,PR'08]
    - Literally trivial ones!



    - Extends to reactive, randomized functionalities, both PPT and IT

RECALL

Is MPC Possible?

every functionality?

Yes ⇔ sh-OT assumption

Yes means all are trivial.
No is more interesting!

Univ. Composable / Angel-UC / Standalone / Passive	All subsets corruptible
Computationally Unbounded (IT)	No
Computationally Bounded (PPT)	No / Yes / Yes / Yes

Trivial ones are *really* trivial (called *Splittable*)

Under sh-OT, everything else complete!

(Zero-One-Law)

IT Setting: Trivial Functionality

- Information-Theoretic Passive security
  - Deterministic SFE: Trivial ⇔ Decomposable

Decomposable Function

Decomposable

	1	3
0	1	3
2	2	3

	0	1
0	0	1
1	1	0

1	1	2
3	4	4

1	1	2	2
3	4	4	3

Undecomposable

	0	1
0	0	0
1	0	1

1	1	2
4	5	2
4	3	3

1	1	4	2
4	3	3	2
4	2	1	1

Decomposable Function

Decomposable

	1	3
0	1	3
2	2	3

	0	1
0	0	1
1	1	0

1	1	2
3	4	4

1	1	2	2
3	4	4	3

Undecomposable

	0	1
0	0	0
1	0	1

1	1	2
4	5	2
4	3	3

1	1	4	2
4	3	3	2
4	2	1	1

Decomposable Function

Decomposable

	1	3
0	1	3
2	2	3

	0	1
0	0	1
1	1	0

1	1	2
3	4	4

1	1	2	2
3	4	4	3

Undecomposable

	0	1
0	0	0
1	0	1

1	1	2
4	5	2
4	3	3

1	1	4	2
4	3	3	2
4	2	1	1

Decomposable Function

Decomposable

	1	3
0	1	3
2	2	3

	0	1
0	0	1
1	1	0

1	1	2
3	4	4

1	1	2	2
3	4	4	3

Undecomposable

	0	1
0	0	0
1	0	1

1	1	2
4	5	2
4	3	3

1	1	4	2
4	3	3	2
4	2	1	1

Decomposable Function

Decomposable

	1	3
0	1	3
2	2	3

	0	1
0	0	1
1	1	0

1	1	2
3	4	4

1	1	2	2
3	4	4	3

Undecomposable

	0	1
0	0	0
1	0	1

1	1	2
4	5	2
4	3	3

1	1	4	2
4	3	3	2
4	2	1	1

Decomposable Function

Decomposable

	1	3
0	1	3
2	2	3

	0	1
0	0	1
1	1	0

1	1	2
3	4	4

1	1	2	2
3	4	4	3

Undecomposable

	0	1
0	0	0
1	0	1

1	1	2
4	5	2
4	3	3

1	1	4	2
4	3	3	2
4	2	1	1

Decomposable Function

Decomposable

	1	3
0	1	3
2	2	3

	0	1
0	0	1
1	1	0

1	1	2
3	4	4

1	1	2	2
3	4	4	3

Undecomposable

	0	1
0	0	0
1	0	1

1	1	2
4	5	2
4	3	3

1	1	4	2
4	3	3	2
4	2	1	1

IT Setting: Trivial Functionality

- Information-Theoretic Passive security
  - Deterministic SFE: Trivial $\Leftrightarrow$ Decomposable

IT Setting: Trivial Functionality

- Information-Theoretic Passive security
  - Deterministic SFE: Trivial ⇔ Decomposable
  - Open for randomized SFE!

IT Setting: Trivial Functionality

- Information-Theoretic Passive security
  - Deterministic SFE: Trivial ⇔ Decomposable
  - Open for randomized SFE!
- Information-Theoretic Standalone security

IT Setting: Trivial Functionality

- Information-Theoretic Passive security
  - Deterministic SFE: Trivial ⇔ Decomposable
  - Open for randomized SFE!
- Information-Theoretic Standalone security
  - Deterministic SFE:
    Trivial ⇔ Uniquely Decomposable and Saturated

Decomposable Function

Decomposable

	1	3
0	1	3
2	2	3

	0	1
0	0	1
1	1	0

1	1	2
3	4	4

1	1	2	2
3	4	4	3

Decomposable Function

Decomposable

	1	3
0	1	3
2	2	3

	0	1
0	0	1
1	1	0

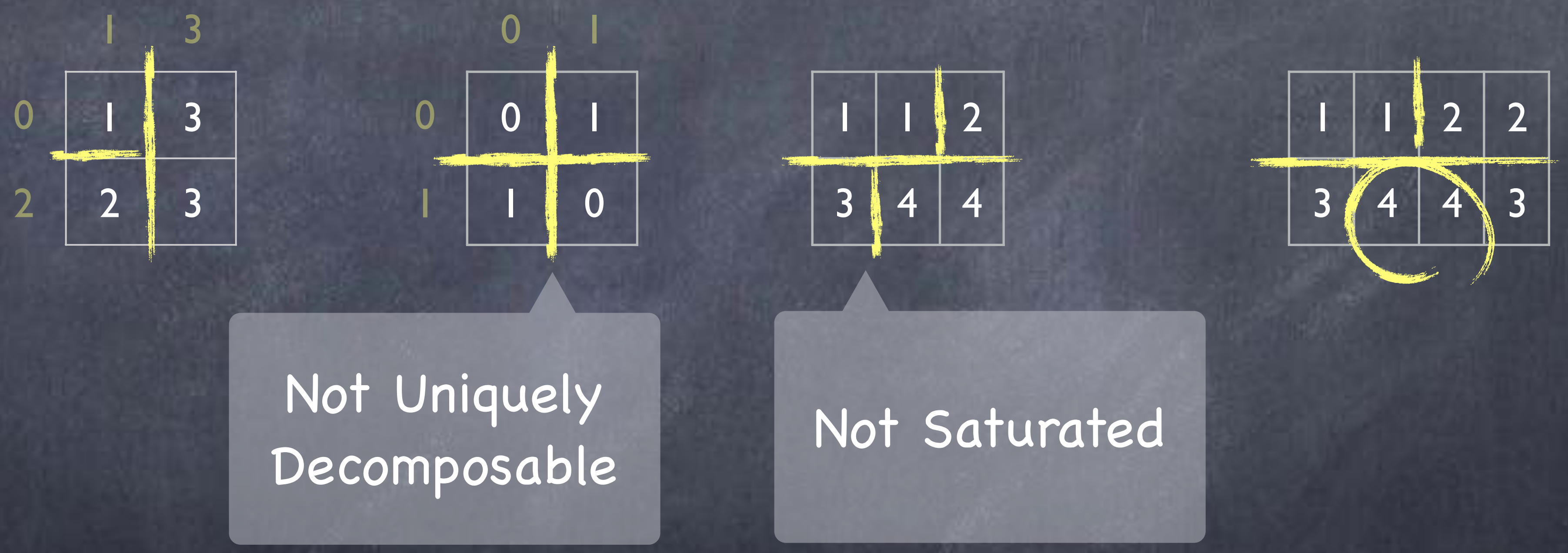
Not Uniquely Decomposable

1	1	2
3	4	4

1	1	2	2
3	4	4	3

Decomposable Function

Decomposable



Decomposable Function

Decomposable

	1	3
0	1	3
2	2	3

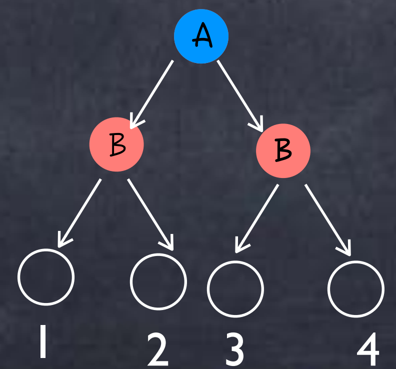
	0	1
0	0	1
1	1	0

Not Uniquely Decomposable

1	1	2
3	4	4

Not Saturated

1	1	2	2
3	4	4	3



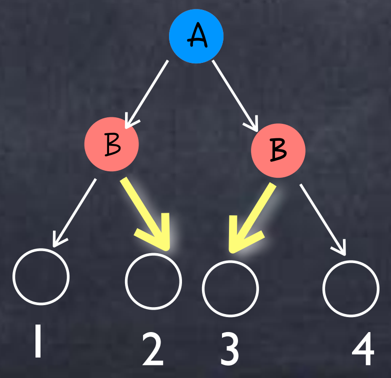
Decomposable Function

Decomposable

Not Uniquely Decomposable

Not Saturated

This strategy doesn't correspond to an input



IT Setting: Trivial Functionality

- Information-Theoretic Passive security
  - Deterministic SFE: Trivial ⇔ Decomposable
  - Open for randomized SFE!
- Information-Theoretic Standalone security
  - Deterministic SFE:
    Trivial ⇔ Uniquely Decomposable and Saturated

IT Setting: Trivial Functionality

- Information-Theoretic Passive security
  - Deterministic SFE: Trivial ⇔ Decomposable
  - Open for randomized SFE!
- Information-Theoretic Standalone security
  - Deterministic SFE:
    Trivial ⇔ Uniquely Decomposable and Saturated
- Information-Theoretic UC security
  - Trivial ⇔ Splittable

IT Setting: Completeness

IT Setting: Completeness

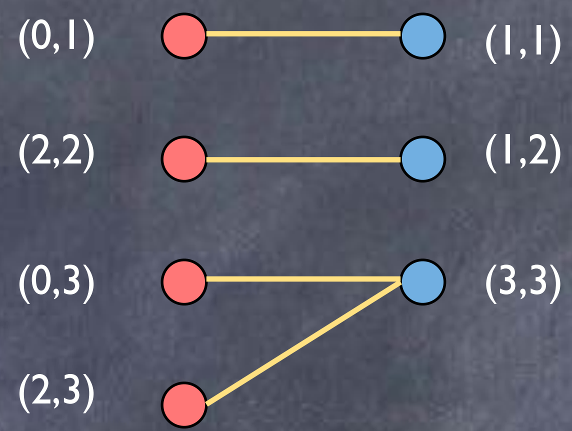
- Information-Theoretic Passive security
  - (Randomized) SFE: Complete $\Leftrightarrow$ Not Simple

IT Setting: Completeness

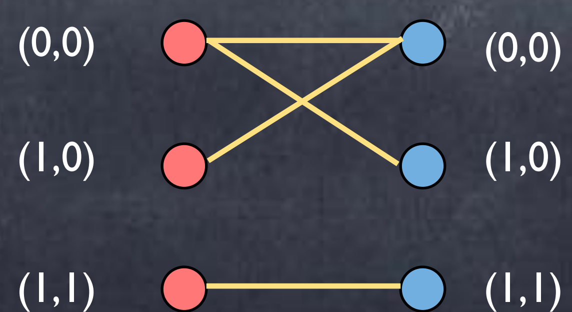
- Information-Theoretic Passive security
  - (Randomized) SFE: Complete ⇔ Not Simple
  - What is Simple?

Simple vs. Non-Simple

	1	3
0	1	3
2	2	3

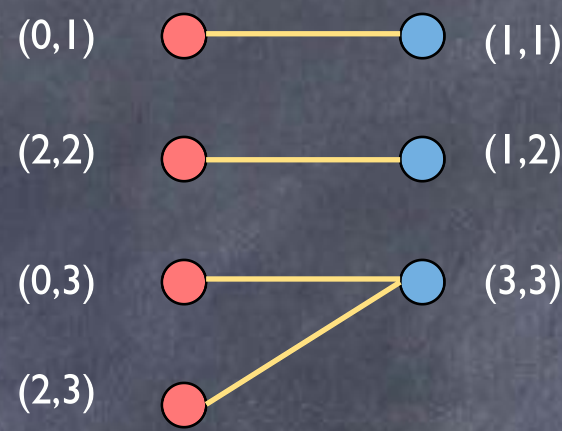


	0	1
0	0	0
1	0	1



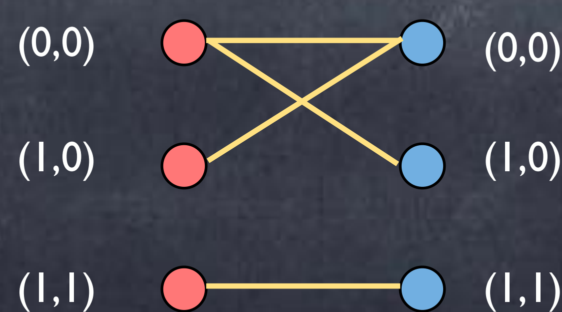
Simple vs. Non-Simple

	1	3
0	1	3
2	2	3



Simple:
Each connected component is a biclique

	0	1
0	0	0
1	0	1



IT Setting: Completeness

- Information-Theoretic Passive security
  - (Randomized) SFE: Complete ⇔ Not Simple
  - What is Simple?
    - Deterministic SFE: In the characteristic bipartite graph, each connected component is a biclique
    - More generally, using a weighted characteristic graph, with $w(u,v) = \Pr[\text{outputs} \mid \text{inputs}]$
      - Simple: $w(u,v) = w_A(u) \times w_B(v)$
    - "Isomorphic" to the "common information"

IT Setting: Completeness

- Information-Theoretic Passive security
  - (Randomized) SFE: Complete ⇔ Not Simple

IT Setting: Completeness

- Information-Theoretic Passive security
  - (Randomized) SFE: Complete $\Leftrightarrow$ Not Simple
- Information-Theoretic Standalone & UC security

IT Setting: Completeness

- Information-Theoretic Passive security
  - (Randomized) SFE: Complete ⇔ Not Simple
- Information-Theoretic Standalone & UC security
  - (Randomized) SFE: Complete ⇔ Core is not Simple

IT Setting: Completeness

- Information-Theoretic Passive security
  - (Randomized) SFE: Complete ⇔ Not Simple
- Information-Theoretic Standalone & UC security
  - (Randomized) SFE: Complete ⇔ Core is not Simple
  - What is the core of an SFE?

IT Setting: Completeness

- Information-Theoretic Passive security
  - (Randomized) SFE: Complete ⇔ Not Simple
- Information-Theoretic Standalone & UC security
  - (Randomized) SFE: Complete ⇔ Core is not Simple
  - What is the core of an SFE?
    - SFE obtained by removing "redundancies" in the input and output space

Quiz

- What's the complexity of the following 3 functions, w.r.t, IT passive secure MPC?
  - max(x,y)
  - [x < y]
  - (max(x,y), [x < y])

Quiz

- What's the complexity of the following 3 functions, w.r.t, IT passive secure MPC?
  - max(x,y)
  - [x < y]
  - (max(x,y), [x < y])

	0	1	2	3
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0

	0	1	2	3
0	0	1	2	3
1	1′	1	2	3
2	2′	2′	2	3
3	3′	3′	3′	3

Quiz

- What's the complexity of the following 3 functions, w.r.t, IT passive secure MPC?
  - max(x,y) — Complete
  - [x < y]
  - (max(x,y), [x < y])

	0	1	2	3
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0

	0	1	2	3
0	0	1	2	3
1	1′	1	2	3
2	2′	2′	2	3
3	3′	3′	3′	3

Quiz

- What's the complexity of the following 3 functions, w.r.t, IT passive secure MPC?
  - max(x,y) — Complete
  - [x < y] — Complete
  - (max(x,y), [x < y])

	0	1	2	3
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0

	0	1	2	3
0	0	1	2	3
1	1′	1	2	3
2	2′	2′	2	3
3	3′	3′	3′	3

Quiz

- What's the complexity of the following 3 functions, w.r.t, IT passive secure MPC?
  - max(x,y) — Complete
  - [x < y] — Complete
  - (max(x,y), [x < y]) — Trivial (Passive and Standalone/Active)

	0	1	2	3
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0

	0	1	2	3
0	0	1	2	3
1	1′	1	2	3
2	2′	2′	2	3
3	3′	3′	3′	3

Quiz

- What's the complexity of the following 3 functions, w.r.t, IT passive secure MPC?
  - max(x,y) — Complete
  - [x < y] — Complete
  - (max(x,y), [x < y]) — Trivial (Passive and Standalone/Active)

	0	1	2	3
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0

	0	1	2	3
0	0	1	2	3
1	1'	1	2	3
2	2'	2'	2	3
3	3'	3'	3'	3

Between Trivial & Complete?

Between Trivial & Complete?

- In the PPT setting, assuming sh-OT, there can be only one or two classes (two for UC security)

Between Trivial & Complete?

- In the PPT setting, assuming sh-OT, there can be only one or two classes (two for UC security)
- In the IT setting, infinitely many levels!

Between Trivial & Complete?

- In the PPT setting, assuming sh-OT, there can be only one or two classes (two for UC security)
- In the IT setting, infinitely many levels!
- Question: Do these levels yield infinitely many "distinct" complexity assumptions corresponding to which levels collapse in the PPT setting?

Between Trivial & Complete?

- In the PPT setting, assuming sh-OT, there can be only one or two classes (two for UC security)
- In the IT setting, infinitely many levels!
- Question: Do these levels yield infinitely many "distinct" complexity assumptions corresponding to which levels collapse in the PPT setting?
  - Maybe not for UC security reductions

Between Trivial & Complete?

- In the PPT setting, assuming sh-OT, there can be only one or two classes (two for UC security)
- In the IT setting, infinitely many levels!
- Question: Do these levels yield infinitely many "distinct" complexity assumptions corresponding to which levels collapse in the PPT setting?
  - Maybe not for UC security reductions
    - Only two such assumptions known so far: shOT & OWF

Between Trivial & Complete?

- In the PPT setting, assuming sh-OT, there can be only one or two classes (two for UC security)
- In the IT setting, infinitely many levels!
- Question: Do these levels yield infinitely many "distinct" complexity assumptions corresponding to which levels collapse in the PPT setting?
  - Maybe not for UC security reductions
    - Only two such assumptions known so far: shOT & OWF
  - Conjecture: Yes, for passive security reductions

Between Trivial & Complete?

- In the PPT setting, assuming sh-OT, there can be only one or two classes (two for UC security)
- In the IT setting, infinitely many levels!
- Question: Do these levels yield infinitely many “distinct” complexity assumptions corresponding to which levels collapse in the PPT setting?

Few Worlds Conjecture

  - Maybe not for UC security reductions
    - Only two such assumptions known so far: shOT & OWF
  - Conjecture: Yes, for passive security reductions

Between Trivial & Complete?

- In the PPT setting, assuming sh-OT, there can be only one or two classes (two for UC security)
- In the IT setting, infinitely many levels!
- Question: Do these levels yield infinitely many "distinct" complexity assumptions corresponding to which levels collapse in the PPT setting?
  - Maybe not for UC security reductions
    - Only two such assumptions known so far: shOT & OWF
  - Conjecture: Yes, for passive security reductions

Few Worlds Conjecture

Many Worlds Conjecture

Summary

- 2-Party:
  - PPT, assuming sh-OT: 3 complexity classes. UC-trivial, UC-complete, All (= Passive/Standalone trivial/complete)
  - IT: Infinitely many complexity classes. Several open problems.
    - Computational assumptions related to collapse of classes in the PPT setting (so far OWF, shOT)
- m-Party (m>2):
  - Non-Honest-Majority: largely open

Quantitative Complexity

- Qualitative question: Does F reduce to G?
- Quantitative question: How many instances of G are needed to implement one instance of F (amortized)?
  - G-complexity of F
  - Upto constants, G-complexity remains the same for all complete G
  - "Cryptographic Complexity" of F
- Cryptographic Complexity is a lower bound on Circuit Complexity

Conclusion

- A detailed picture of deterministic 2-party SFE, under various MPC reductions
  - Completeness characterised for randomised SFE too
  - But complexity questions largely open for randomised SFE, m-party SFE for m > 2
- Computational hardness related to MPC reductions
  - We know that OWF is one of the "F reduces to G" assumptions, and sh-OT is the "maximal" assumption
  - Few Worlds Conjecture & Many Worlds Conjecture
- Quantitative Complexity
  - Crypto complexity is a lower bound on circuit complexity