Non-Interactive Secure Computation from One-Way Functions

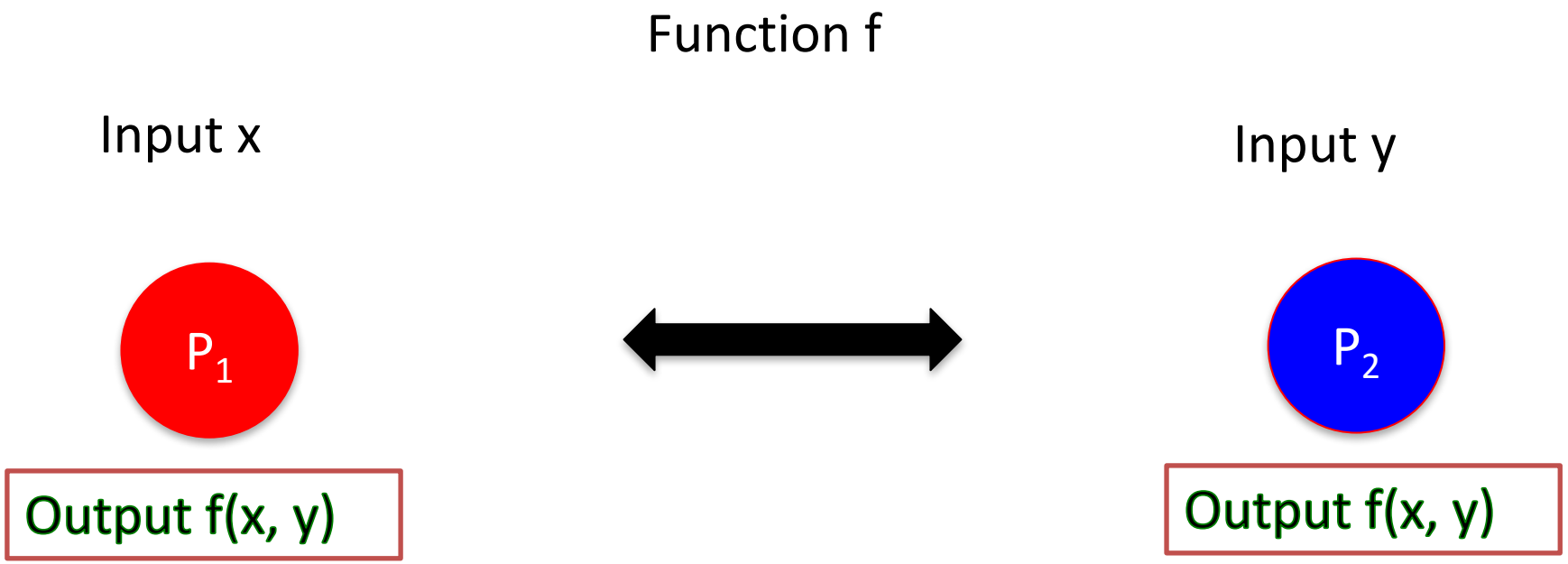
Saikrishna Badrinarayanan
(UCLA)

Abhishek Jain
(JHU)

Rafail Ostrovsky
(UCLA)

Ivan Visconti
(University of Salerno)

Secure Two Party Computation



Goal: Both parties wish to run a protocol at the end of which they both learn the output of the function on their joint inputs.

Secure Two Party Computation

Function f

Input x

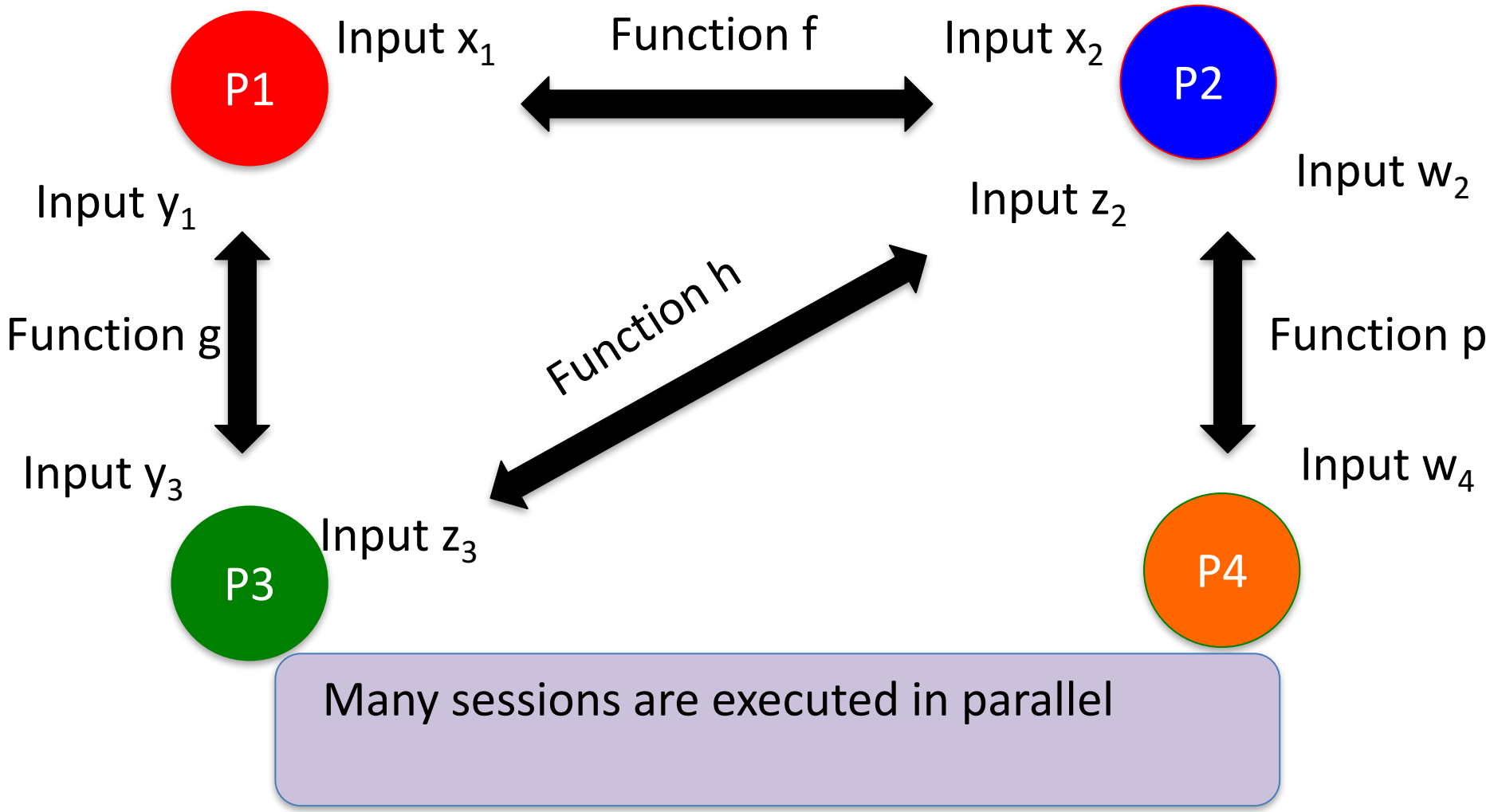
Input y

Output f(x, y)

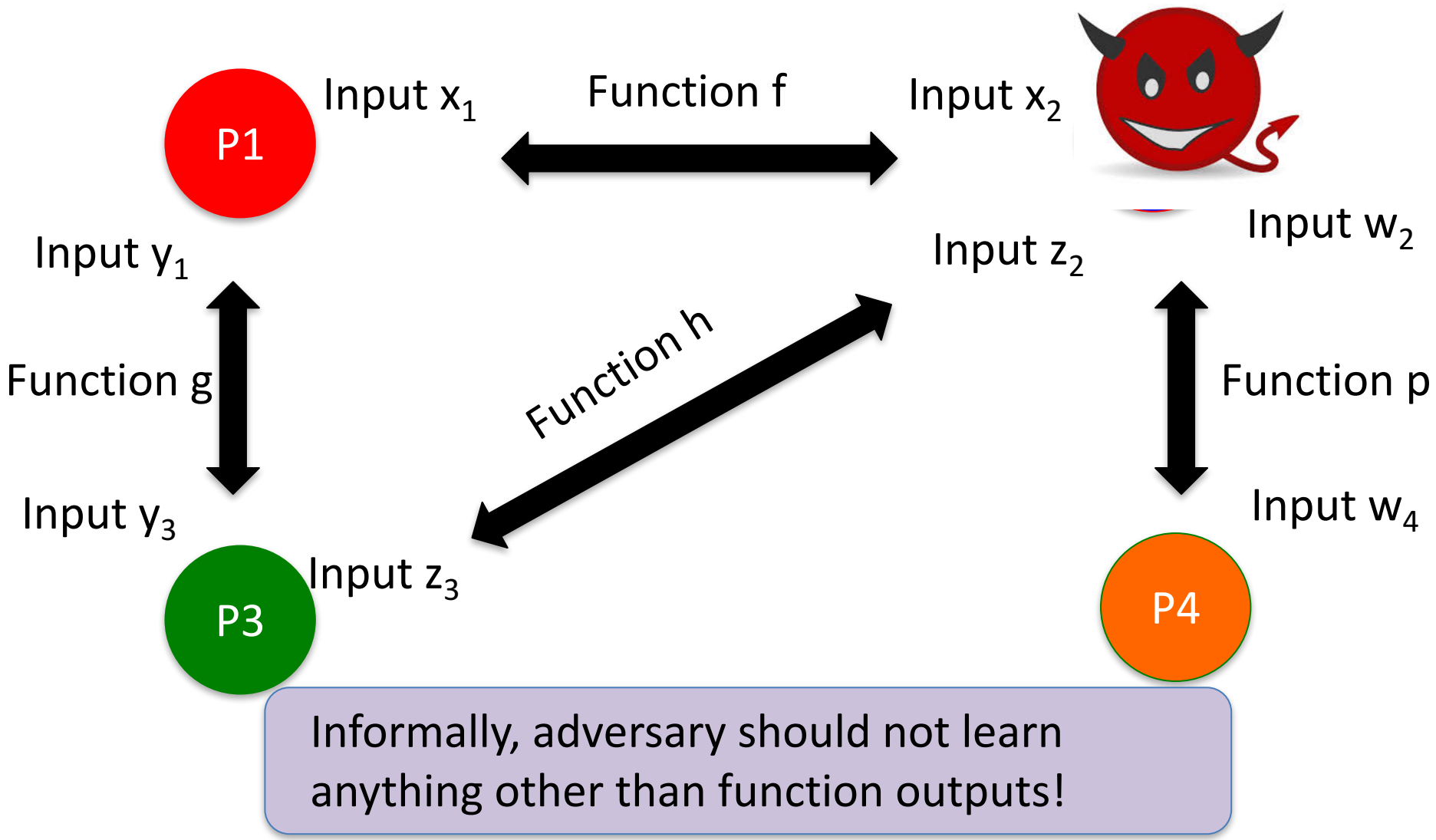
Output f(x, y)

Security: Informally, adversary should not learn anything about y other than f(x, y) !

UC-Secure Computation [Canetti 01]



UC-Secure Computation [Canetti 01]



Informally, adversary should not learn anything other than function outputs!

Continued..

- Numerous applications
- Unfortunately, impossible to construct without a setup assumption!

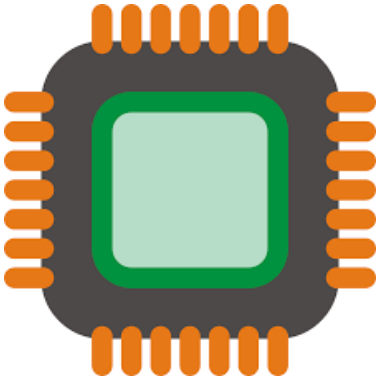
Setup Assumptions

- Common Reference String [Canetti-Lindell-Ostrovsky-Sahai 02]

Focus of this paper

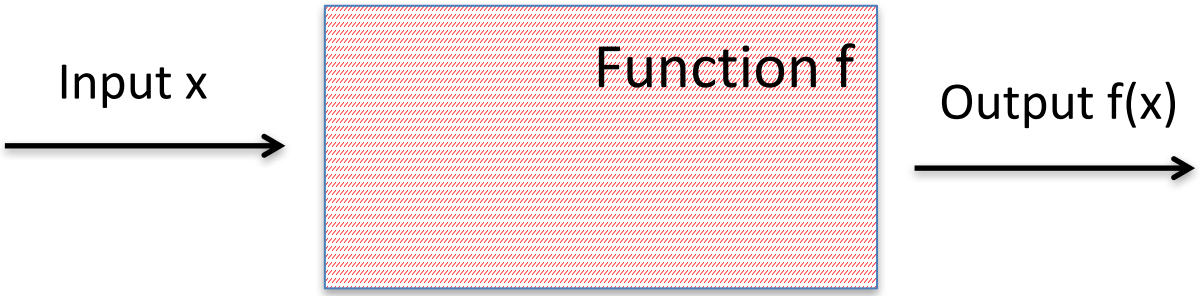
- Physical Assumptions
  - Hardware Tokens [Katz 07]

  - Physically Uncloneable Functions (PUFs)



Hardware Tokens [Katz07]

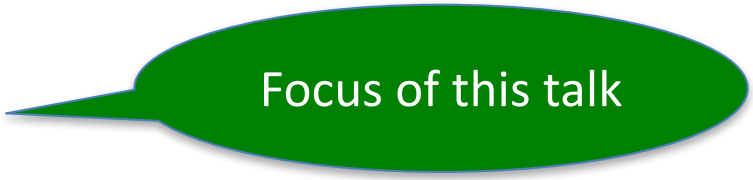
- A piece of hardware that can evaluate any function (embedded inside it) on input queries.



- Physical manifestation of ideal obfuscation?
- Difference: Need the hardware object in hand to be able to query and recover output.

Types of Hardware Tokens

- Stateless:



  - Honest token does not have any memory across invocations.

Challenge: Adversarial tokens can be stateful!

- Stateful:
  - Token can maintain memory.
  - Harder to design such tokens.
  - Easier to design protocols using them.

Motivating Scenario

Reusable message

Input: DNA matching algorithm

Input: DNA Data x

Encrypt (x)

Encrypt f(x)

Decrypt and learn f(x) = list of relatives

Goal: Alice wants to publish an encryption of her private DNA data to an organization that can compute the set of relatives of a given DNA sample.

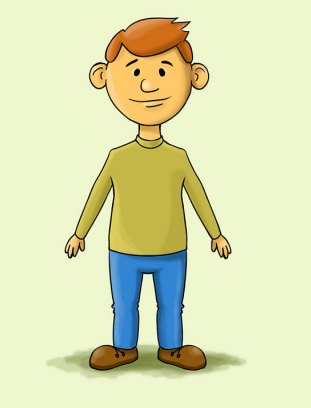
Security requirement: Alice's private data and company's data should be hidden.

Non-interactive secure computation (NISC)
[IKOPS'11]

- Formalizes the scenario in the previous s

Reusable message for input x

Input: y



Input: x

Encrypt (x)

Encrypt f(x, y)

Decrypt and learn f(x, y)

Security requirement: as in standard two party computation

Prior work

- [IKOPS11] : NISC in OT Hybrid model.
- [AMPR14,MR17] : NISC in CRS model from OT + one way functions.
- [CJS14]: UC-secure NISC in Global Random Oracle model from OT + one way functions.
- [BGISW17]: NISC in plain model from sub-exponentially secure OT + one way functions.

Question

- Can we achieve NISC from the minimal assumption of One-Way Functions?
- Further, can we achieve UC security?

Our Result

- UC-secure non-interactive secure computation assuming one-way functions using a single stateless hardware token.

  - Optimal in terms of assumptions and number of tokens.
  - Achieves UC security unlike all prior work except CJS14.

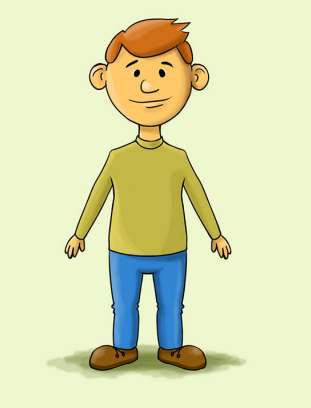
Our Results: Corollary

- Two message UC-secure two party computation where both parties receive output, assuming one way functions using a single stateless hardware token.

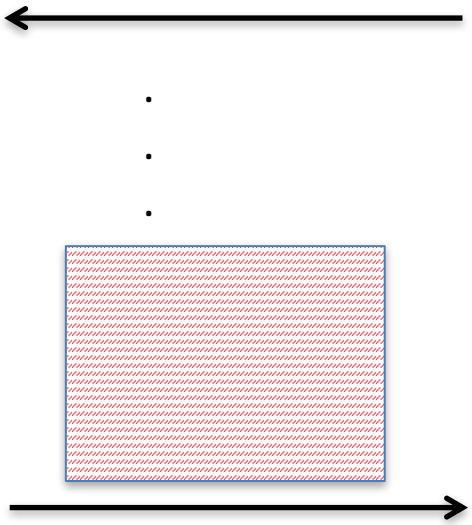
Techniques

Token direction: Prior works

Sender: input y



Receiver: input x



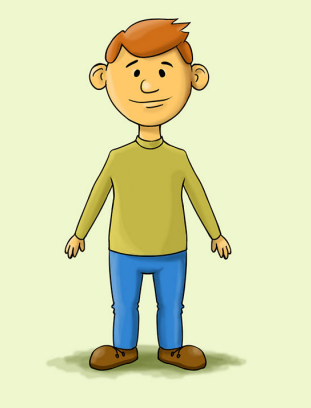
In all prior works, token sent by the sender.

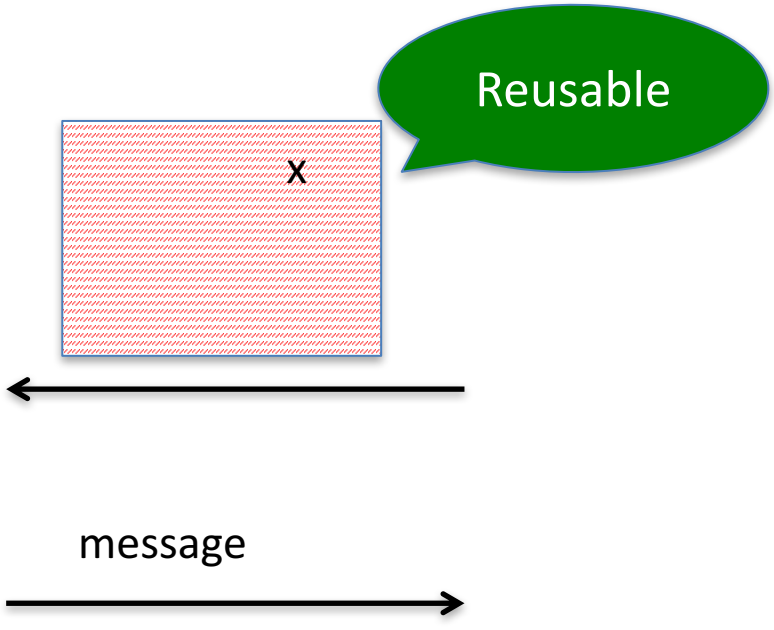
Issues in our setting:

1. Need a fresh token for each new interaction with a fixed reusable receiver message.
2. All prior works require atleast two rounds of interaction after sender token.

Solution: Token from Receiver

Sender: input y





Receiver: input x

Main Challenge: Resetting Attacks

1. How to prevent sender from resetting the token and trying different inputs y?
2. Need the receiver to authenticate the sender's input to the token before it processed by the token.
3. But that will take at least 2 rounds!

Solution:

1. We allow the sender to reset the token!
2. However, token is carefully designed to perform only "encrypted" computation that is later decrypted by the receiver.
3. Hence, even on trying different inputs, sender doesn't learn anything meaningful from the token.

Other Challenges

- Selective abort attacks.
- Subliminal channel information through token.
- Achieving straight line simulation to get UC security.

- Please refer to the paper for more details!
  https://eprint.iacr.org/2018/1020

Thank you!