

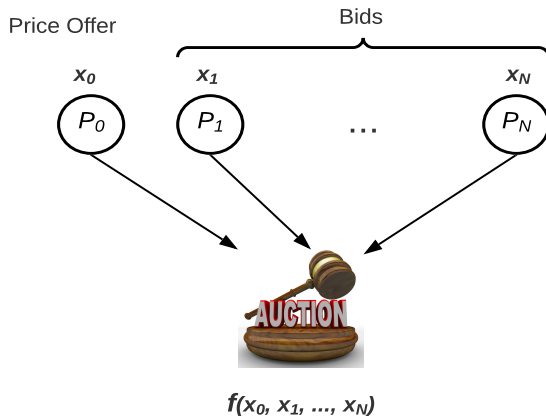
# Gage MPC: Bypassing Residual Function Leakage for Non-Interactive MPC

**Ghada Almashaqbeh**<sup>1</sup> Fabrice Benhamouda<sup>2</sup> Seungwook Han<sup>3</sup>  
Daniel Jaroslawicz<sup>3</sup> Tal Malkin<sup>3</sup> Alex Nicita<sup>3</sup> Tal Rabin<sup>4,2</sup>  
Abhishek Shah<sup>3</sup> Eran Tromer<sup>3,5</sup>

<sup>1</sup>UConn, <sup>2</sup>Algorand, <sup>3</sup>Columbia, <sup>4</sup>UPenn, <sup>5</sup>Tel-Aviv University

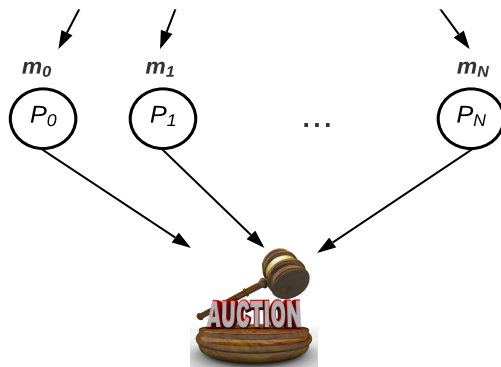
PETS 2021

# NIMPC — Auction



The winner is ...

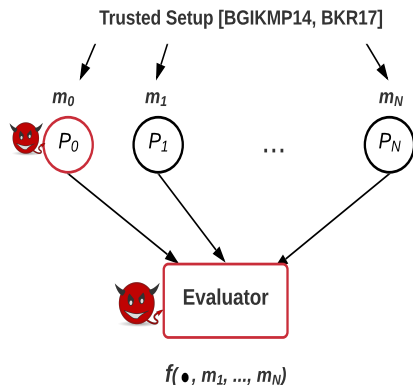
Trusted Setup [BGIKMP14, BKR17]



$f(m_0, m_1, \dots, m_N)$

The winner is ...

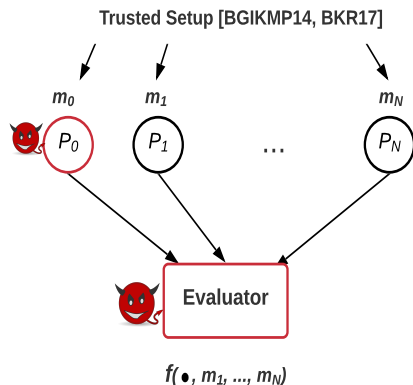
- Leakage of the *Residual Function* is inherent.  
 Evaluator and say  $P_0$  can compute  $f(\bullet, m_1, \dots, m_N)$ .



- Leakage of the *Residual Function* is inherent.

Evaluator and say  $P_0$  can compute  $f(\bullet, m_1, \dots, m_N)$ .

- Robustness to collusion.  
Only the residual function is leaked!

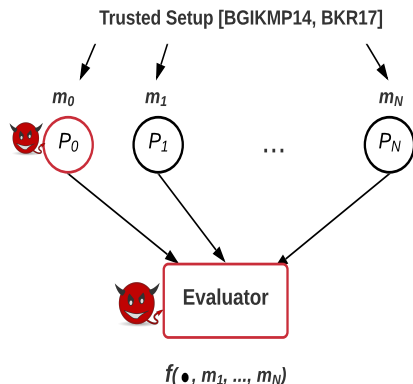


- Leakage of the *Residual Function* is inherent.

Evaluator and say  $P_0$  can compute  $f(\bullet, m_1, \dots, m_N)$ .

- Robustness to collusion.  
Only the residual function is leaked!

- Setup assumptions.

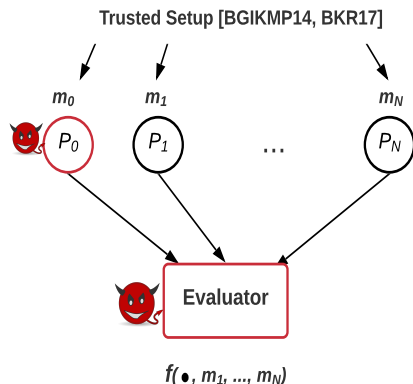


- Leakage of the *Residual Function* is inherent.

Evaluator and say  $P_0$  can compute  $f(\bullet, m_1, \dots, m_N)$ .

- Robustness to collusion.  
Only the residual function is leaked!

- Setup assumptions.



*Avoid such limitations??!*

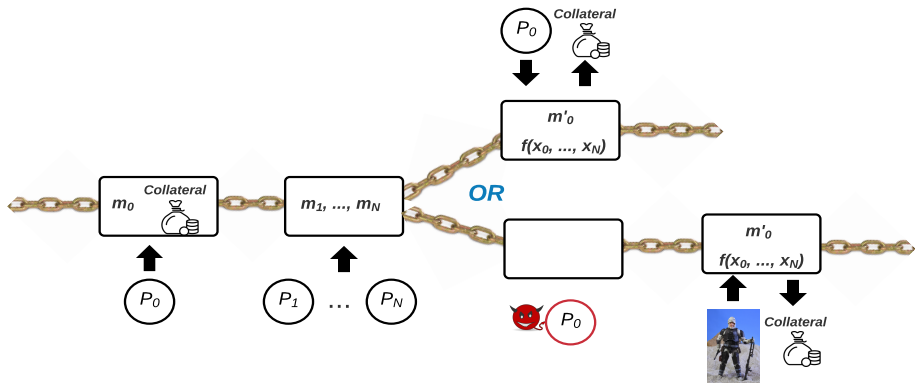
- **Gen I.** A blockchain implements a broadcast channel.

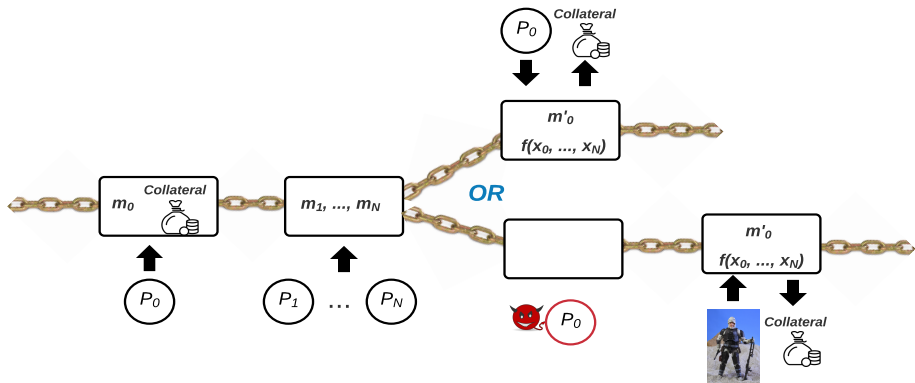


- **Gen I.** A blockchain implements a broadcast channel.
- **Gen II.** Payments are incorporated into MPC.

- **Gen I.** A blockchain implements a broadcast channel.
- **Gen II.** Payments are incorporated into MPC.
- **Gen III.** *This work; Gage MPC!* Smart contracts and miners are active participants in MPC.

# Gage MPC





**A monetary assumption.** An honest party can put a collateral of value much higher than what an adversary can expend on computation.

# On Circumventing the Lower Bounds

- Eliminate the leakage of the residual function.

# On Circumventing the Lower Bounds

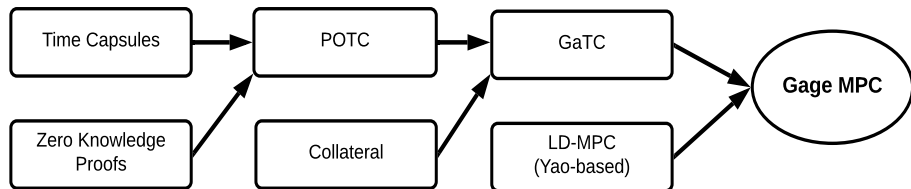
- Eliminate the leakage of the residual function.
- Eliminate setup assumptions.
  - A PKI or pre-shared correlated randomness.
  - The need for a dedicated online evaluator.

# On Circumventing the Lower Bounds

- Eliminate the leakage of the residual function.
- Eliminate setup assumptions.
  - A PKI or pre-shared correlated randomness.
  - The need for a dedicated online evaluator.

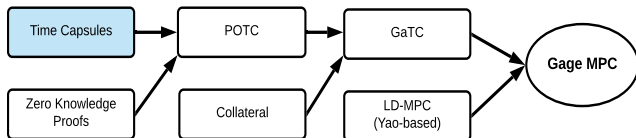
Gage MPC guarantees *short term* security!

# Gage MPC: Our Construction





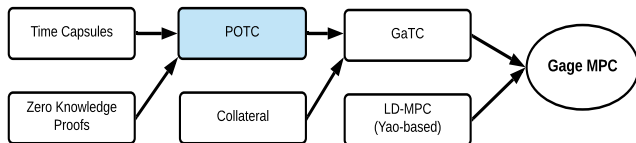
# Our Construction — Time Capsules



Simply commit to a value that can be opened after expending a pre-specified amount of computation.

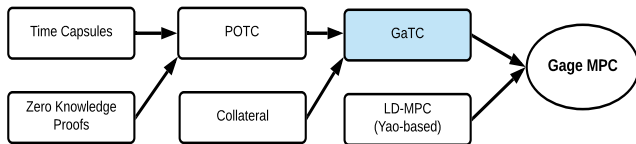
E.g.,  $h(s)$  where  $s \leftarrow \{0, 1\}^{\lambda^*}$

# Our Construction — Proof of Time Capsules (POTC)

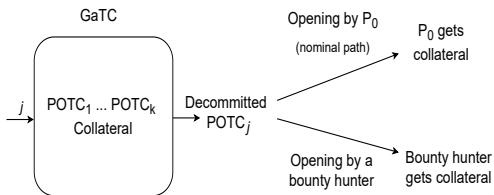


Instead of announcing the decommitment itself (i.e.,  $s$ ), prove in zero knowledge that the decommitment has been found.

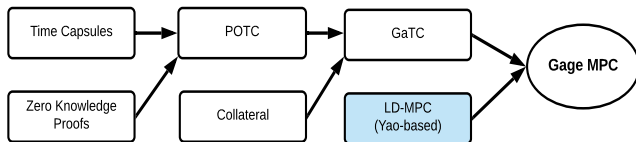
# Our Construction — Gage Time Capsules (GaTC)



Bundle several POTCs together, and utilize a smart contract to provide a monetary incentive to open the intended POTC.



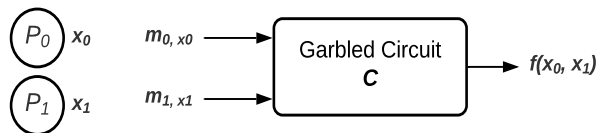
# Our Construction — Label Driven MPC (LD-MPC)



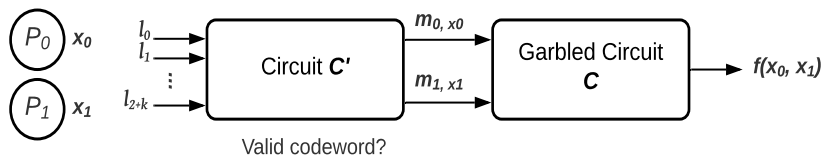
A generalization of Garbled Circuits that is robust to the exposure of additional labels.

# Our Construction — Label Driven MPC (LD-MPC)

Conventional Yao; Exposure of any additional label compromise input privacy.



LD-MPC = Error Correcting Codes + Yao

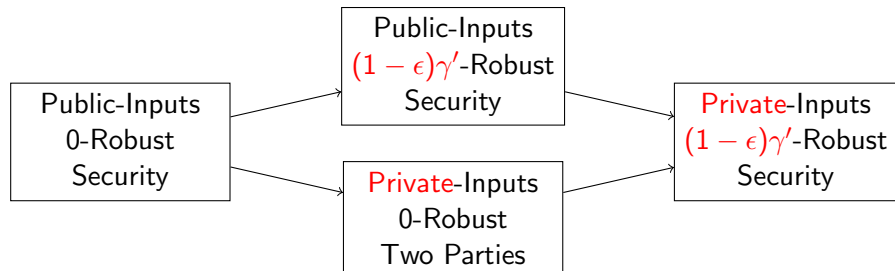


# Main Result — Gage MPC

Combines LD-MPC with GaTC.

Simplest case; Only the input of  $P_0$  is private.

- $P_0$  prepares a garbled circuit, GaTCs for input labels for  $P_1, \dots, P_N$ , and a controller smart contract.
- $P_1, \dots, P_N$  submit their inputs.
- Either  $P_0$  will come back and open the corresponding labels, or bounty hunters will do.
- Smart contract evaluates the circuit over the labels and record the output.



The private input versions support only two party computation.

# Conclusion

## Main Result — Gage MPC

NIMPC for  $f$  leaking  $R$  and requiring  $TS \rightarrow$  NIMPC with no  $R$  and  $TS$

Gen III of MPC + blockchain

## Side Result

Several new primitives (POTC, GaTC, and LD-MPC) that could be of independent interest.

A proof-of-concept implementation in Ethereum-like blockchain.



Thank you!

Questions?