MPC-in-Multi-Heads: a Multi-Prover Zero-Knowledge Proof System

(or: How to Jointly Prove Any NP Statements in ZK)

Hongrui Cui[1] Kaiyi Zhang[1] Yu Chen [2,3,4] Zhen Liu [1]
Yu Yu [1,5]

[1]Department of Computer Science, Shanghai Jiao Tong University

[2]School of Cyber Science and Technology, Shandong University

[3]State Key Laboratory of Cryptology

[4]Key Laboratory of Cryptologic Technology and Information Security

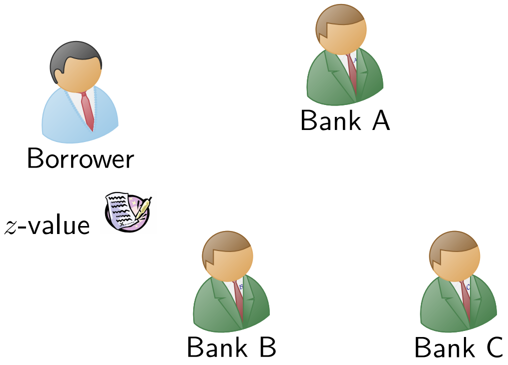
[5]Shanghai Qizhi Institute

ESORICS 2021

Synopsis

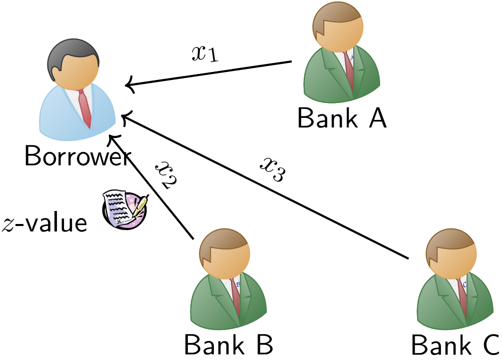
Motivation

The **Double Financing** Problem



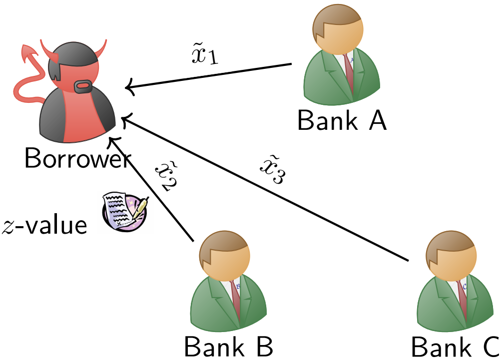
Motivation

The **Double Financing** Problem



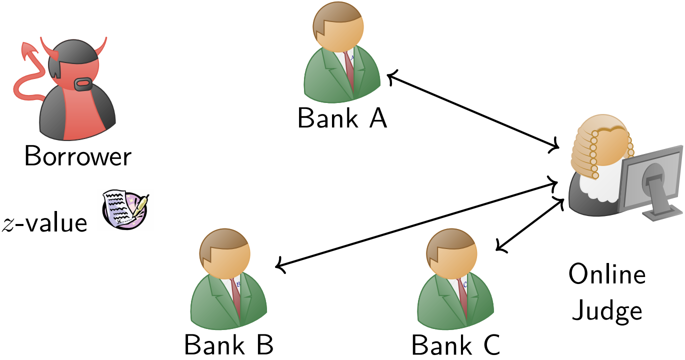
Motivation

The **Double Financing** Problem



Motivation

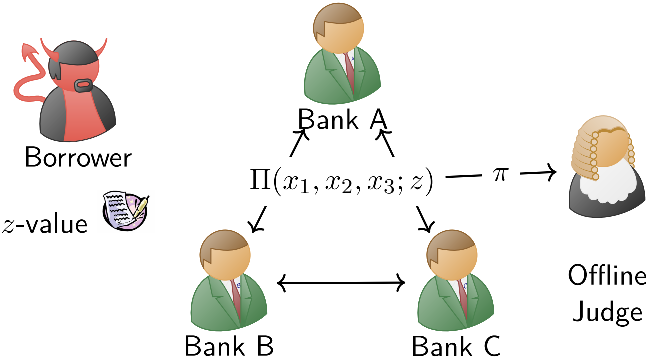
The **Double Financing** Problem



Prove $(x_1 + x_2 + x_3) < 0.9 \cdot z$!

Motivation

The **Double Financing** Problem



Non-Interactive Proof is Better !

Multi-Prover Zero-Knowledge

One possible solution for $\mathcal{NP}$ relations:

Ideal MPZK Functionality

$$\mathcal{F}^{\mathsf{mpzk}}(\underbrace{x_1, \ldots, x_m}_{m \text{ Provers}};\ \underbrace{\ y\ }_{1 \text{ Verifier}}) \mapsto \mathcal{R}(x_1, \ldots, x_m; y)$$

where $\mathcal{R}$ defines an $\mathcal{NP}$ relation.

Multi-Prover Zero-Knowledge

One possible solution for $\mathcal{NP}$ relations:

Ideal MPZK Functionality

$$\mathcal{F}^{\mathsf{mpzk}}(\underbrace{x_1, \ldots, x_m}_{m \text{ Provers}}; \underbrace{y}_{1 \text{ Verifier}}) \mapsto \mathcal{R}(x_1, \ldots, x_m; y)$$

where $\mathcal{R}$ defines an $\mathcal{NP}$ relation.

Discussions:

- Implies traditional ZK when $m = 1$
- If $\mathcal{V}$ only broadcasts random coins, we can apply FS/BCS transformation

Synopsis

MPC+ZK

Solutions Implied by Feasibility Results

- One can easily design a protocol by computing $\mathcal{F}^{\mathsf{mpzk}}$ via general MPC framework
- GCZK follows this approach [JKO13, FNO15]

MPC+ZK

Solutions Implied by Feasibility Results

- One can easily design a protocol by computing $\mathcal{F}^{\mathsf{mpzk}}$ via general MPC framework
- GCZK follows this approach [JKO13, FNO15]

Discussions

- Claim: the above construction is not public-coin

MPC+zk-SNARK

More Advanced Solutions

One can also distribute the proving program of zk-SNARK among multi-provers.

MPC+zk-SNARK

More Advanced Solutions

One can also distribute the proving program of zk-SNARK among multi-provers.

Discussions

- Assuming a 3-round protocol w/ messages (a, c, z).
- If MPC outputs (a, c, z), then some hash function has to be evaluated inside MPC
- The prover's computational complexity tends to be high

Publicly Verifiable MPC

This is the closest to our goal

- PV-MPC allows any external party to verify that the computation is correct
- Claim: this property suffices for our goal

Publicly Verifiable MPC

This is the closest to our goal

- PV-MPC allows any external party to verify that the computation is correct
- Claim: this property suffices for our goal

Caveats

Existing works have some significant drawbacks

- Works of Baum et al. [BDO14, BOSS20] relies on **bulletin board**—an unalterable broadcast
- Works of Schoenmakers and Veeningen [SV15] relies on honest majority setting to preserve privacy

ZK with Shared Instances

Secret-Shared Proof Instance

- Boneh et al. proposed "ZKP on Secret-Shared Data" in [BBC+19]
- In their formulation, the **single** prover holds x entirely while **multiple** verifiers only hold shares
- This primitive is already being used in MPC (cf. [BGIN20])

ZK with Shared Instances

Secret-Shared Proof Instance

- Boneh et al. proposed "ZKP on Secret-Shared Data" in [BBC$^+$19]
- In their formulation, the **single** prover holds x entirely while **multiple** verifiers only hold shares
- This primitive is already being used in MPC (cf. [BGIN20])

Conclusion

- Quite orthogonal

Synopsis

Extending [IKOS07]

Consider the original MPC-in-the-Head construction of Ishai et al.

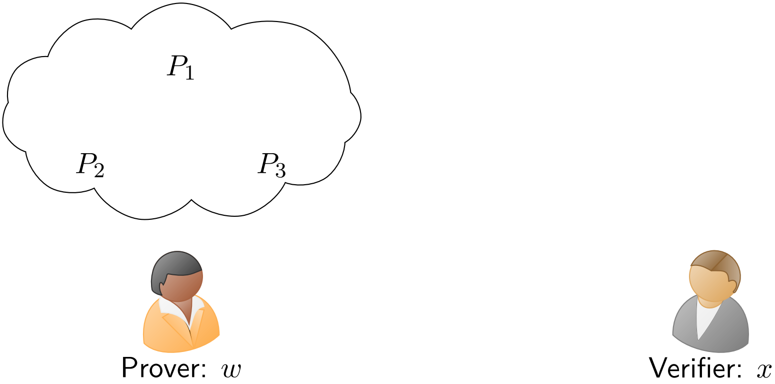
Prover: w

$$\mathcal{R}(x, w) = C(x, w) = 1$$

Verifier: x

Extending [IKOS07]

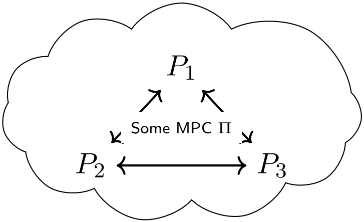
Consider the original MPC-in-the-Head construction of Ishai et al.



$$\mathcal{R}(x, w) = C(x, w) = 1$$

Extending [IKOS07]

Consider the original MPC-in-the-Head construction of Ishai et al.



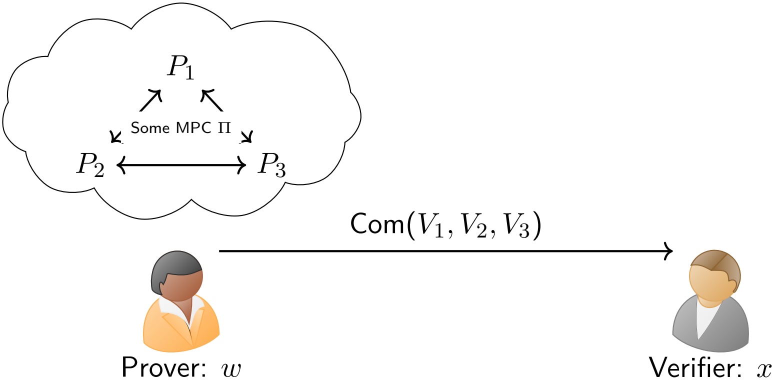
Prover: w

$$\mathcal{R}(x, w) = C(x, w) = 1$$

Verifier: x

Extending [IKOS07]

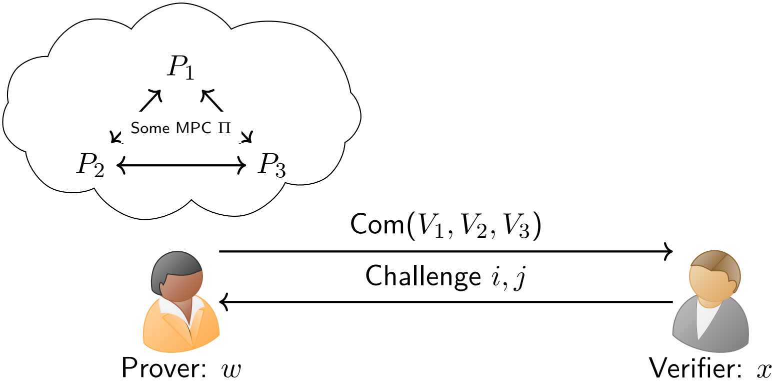
Consider the original MPC-in-the-Head construction of Ishai et al.



$\mathcal{R}(x, w) = C(x, w) = 1$

Extending [IKOS07]

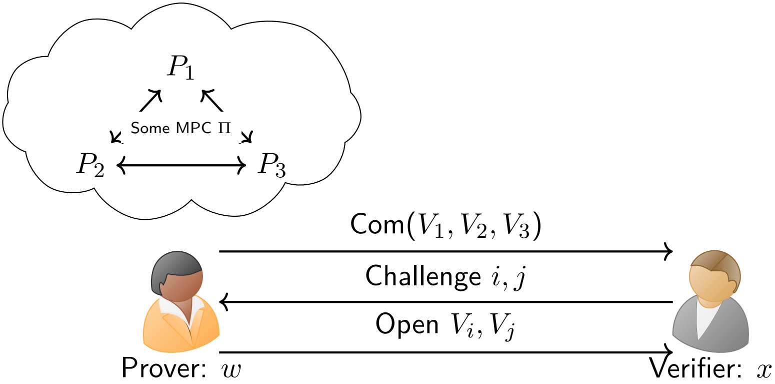
Consider the original MPC-in-the-Head construction of Ishai et al.



$$\mathcal{R}(x, w) = C(x, w) = 1$$

Extending [IKOS07]

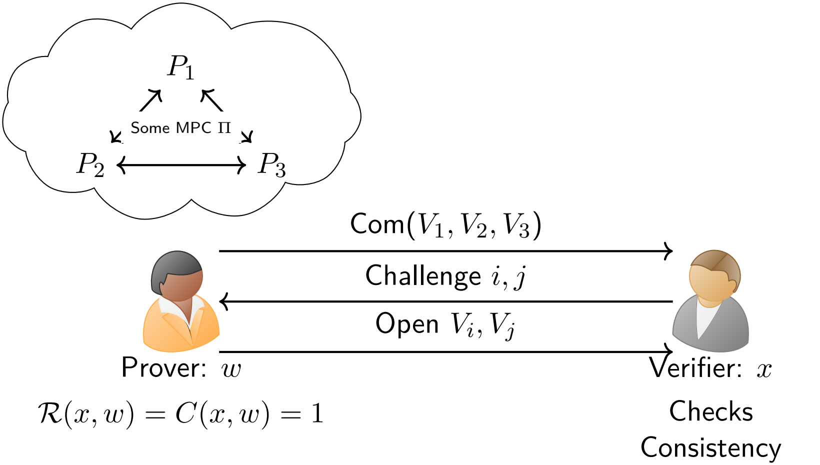
Consider the original MPC-in-the-Head construction of Ishai et al.



$$\mathcal{R}(x, w) = C(x, w) = 1$$

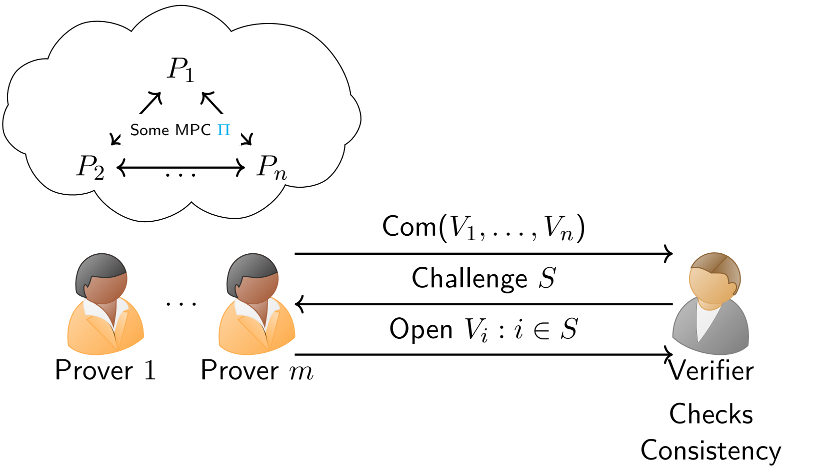
Extending [IKOS07]

Consider the original MPC-in-the-Head construction of Ishai et al.



Extending [IKOS07]

Now we extend the number of provers



An Example

Consider the 3-prover example:

3 Real Provers Simulating 9 Virtual Parties

- Alice (resp. Bob, Charlie) shares a into a_1, a_2, a_3 (resp. b, c)
- They compute the function $\mathcal{R}(\sum a_i, \sum b_i, \sum c_i; x)$ using some 9-party MPC Π
- Each prover simulates 3 parties, "group-wise" communication is sent via "prover-wise" channels

An Example

Consider the 3-prover example:

3 Real Provers Simulating 9 Virtual Parties

- Alice (resp. Bob, Charlie) shares a into a_1, a_2, a_3 (resp. b, c)
- They compute the function $\mathcal{R}(\sum a_i, \sum b_i, \sum c_i; x)$ using some 9-party MPC Π
- Each prover simulates 3 parties, "group-wise" communication is sent via "prover-wise" channels

---

Discussion

- Communication complexity is $\Omega(|C|)$
- Π needs to protect honest prover's privacy

Experiment Setup

We tested on three relations:

- $\mathcal{R}^{\text{hash}}(y; (x_1, x_2))$: $y = \texttt{SHA256}(x_1 \oplus x_2)$
- $\mathcal{R}^{\text{comp}}((y, h_1, h_2); ((x_1, r_1), (x_2, r_2)))$:

$$\underbrace{y < (x_1 + x_2)}_{\text{32-bit integer}} \wedge h_1 = \texttt{SHA256}(x_1 || r_1) \wedge h_2 = \texttt{SHA256}(x_2 || r_2)$$

- $\mathcal{R}^{\text{sum}}(y; (x_1, ..., x_8))$: $y = \underbrace{x_1 + ... + x_8}_{\text{32-bit integer}}$

Experiment Results

- We instantiate the inner protocol Π with semi-honest GMW
- Each round the verifier checks 2 views per prover

Relation	$\mathcal{R}^{\mathsf{hash}}$	$\mathcal{R}^{\mathsf{comp}}$	$\mathcal{R}^{\mathsf{sum}}$
Circuit Size	94,302/22,528	189,450/45,312	1,821/288
Simulated Party	2×3	2×3	8×3
Soundness Error	2^{-40}	2^{-40}	2^{-40}
Repetition Count	70	70	70
Proving Time	109min	223min	26min31s
Verification Time	23.7s	50.0s	1.44s
Proof Size	4.0MB	8.0MB	1.3MB

Synopsis

Conclusion

Our contributions:

- A new primitive from practical applications
- A simple construction of the primitive
- Implementation and experiments

Further Improvement

The current protocol only utilizes the original (simplest) MPC-in-the-head construction, adaptation of new techniques (e.g., Ligero, ZKB++) is left as a future work.

Reference I

Dan Boneh, Elette Boyle, Henry Corrigan-Gibbs, Niv Gilboa, and Yuval Ishai.
Zero-knowledge proofs on secret-shared data via fully linear PCPs.
In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 67–97. Springer, Heidelberg, August 2019.

Carsten Baum, Ivan Damgård, and Claudio Orlandi.
Publicly auditable secure multi-party computation.
In Michel Abdalla and Roberto De Prisco, editors, *SCN 14*, volume 8642 of *LNCS*, pages 175–196. Springer, Heidelberg, September 2014.

Reference II

Elette Boyle, Niv Gilboa, Yuval Ishai, and Ariel Nof.
Efficient fully secure computation via distributed zero-knowledge proofs.
Cryptology ePrint Archive, Report 2020/1451, 2020. https://eprint.iacr.org/2020/1451.

Carsten Baum, Emmanuela Orsini, Peter Scholl, and Eduardo Soria-Vazquez.
Efficient constant-round MPC with identifiable abort and public verifiability.
In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 562–592. Springer, Heidelberg, August 2020.

Reference III

Tore Kasper Frederiksen, Jesper Buus Nielsen, and Claudio Orlandi.
Privacy-free garbled circuits with applications to efficient zero-knowledge.
In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 191–219. Springer, Heidelberg, April 2015.

Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai.
Zero-knowledge from secure multiparty computation.
In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007.

Reference IV

- Marek Jawurek, Florian Kerschbaum, and Claudio Orlandi.
  Zero-knowledge using garbled circuits: how to prove non-algebraic statements efficiently.
  In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013*, pages 955–966. ACM Press, November 2013.
- Berry Schoenmakers and Meilof Veeningen.
  Universally verifiable multiparty computation from threshold homomorphic cryptosystems.
  In Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis, editors, *ACNS 15*, volume 9092 of *LNCS*, pages 3–22. Springer, Heidelberg, June 2015.