Multi-Prover Zero-Knowledge From MPC-in-Multi-Heads

Presented by Hongrui Cui

Jul. 2020 — Nov. 2020

Synopsis

Introduction

Related Works
- MPC/ZK
- PV-MPC
- ZK on Shared Instances

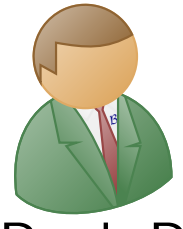
Current Status
- A Black-Box Construction
- A More Advanced Construction

Inception

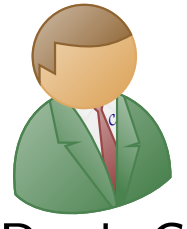
The **Double Financing** Problem

Borrower

Bank A



Bank B

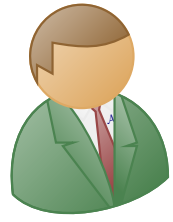


Bank C

Inception

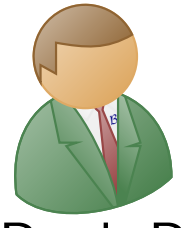
The **Double Financing** Problem

Borrower



Bank A

z-value 

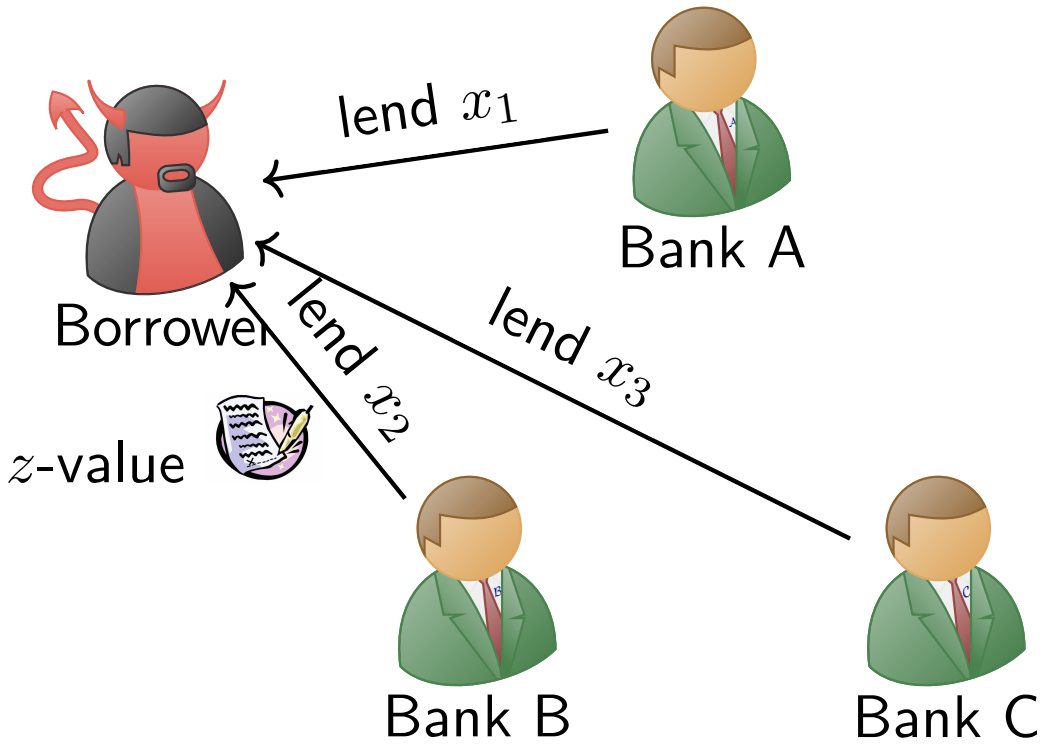


Bank B

Bank C

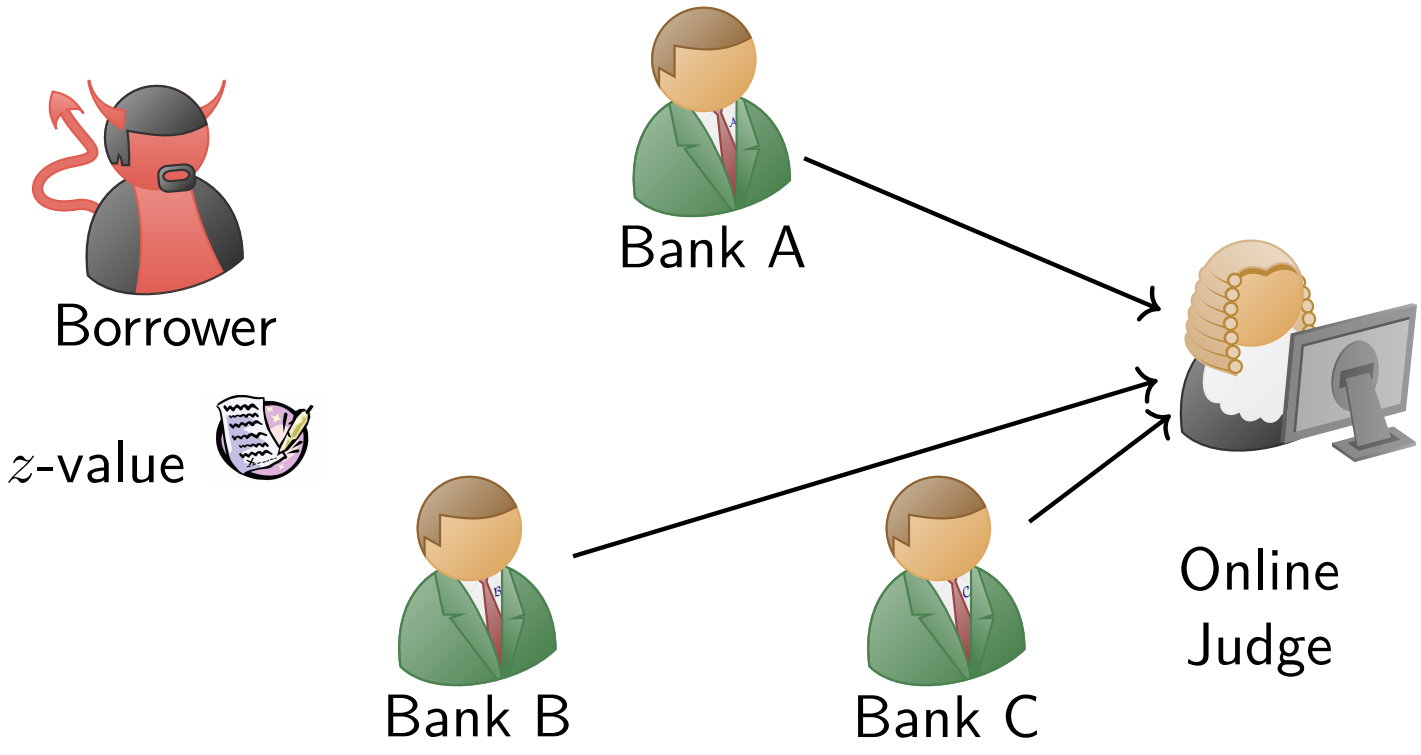
Inception

The **Double Financing** Problem



Inception

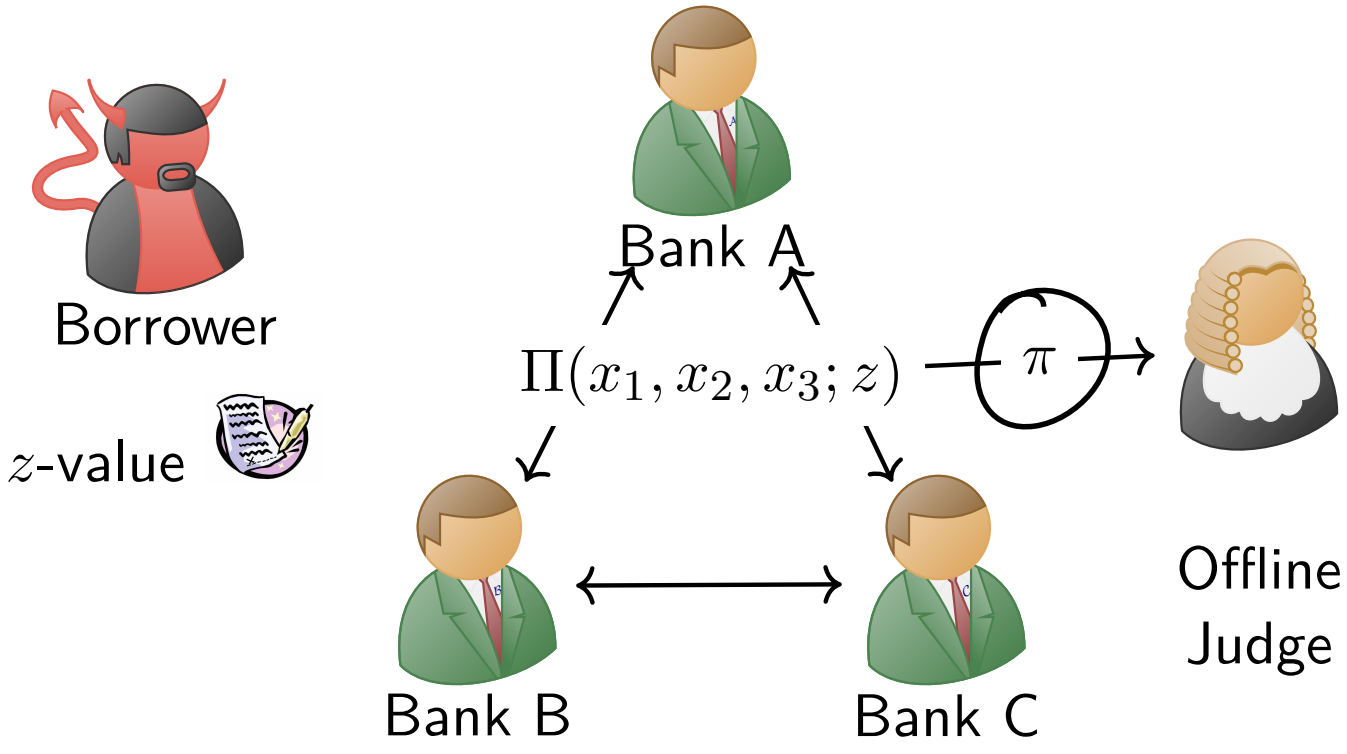
The **Double Financing** Problem



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

Inception

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

Introduction

Related Works
- MPC/ZK
- PV-MPC
- ZK on Shared Instances

Current Status
- A Black-Box Construction
- A More Advanced Construction

MPC+ZK

Solutions Implied by Feasibility Results

- One can easily design a protocol by computing $\mathcal{F}^{\text{mpzk}}$ via general MPC framework
- Constructions like [JKO13] roughly follows this approach

AND → 2 K bit

MPC+ZK

Solutions Implied by Feasibility Results

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

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
- If MPC1$(\vec{w}, x) \mapsto (a, \vec{s})$, $c = H(a)$, MPC2$(\vec{w}, x, c, \vec{s}) \mapsto z$, intuitively, we have to enforce consistency between MPC1 and MPC2

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

Introduction

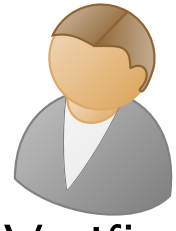
Related Works
- MPC/ZK
- PV-MPC
- ZK on Shared Instances

Current Status
- A Black-Box Construction
- A More Advanced Construction

Extending [IKOS07]

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

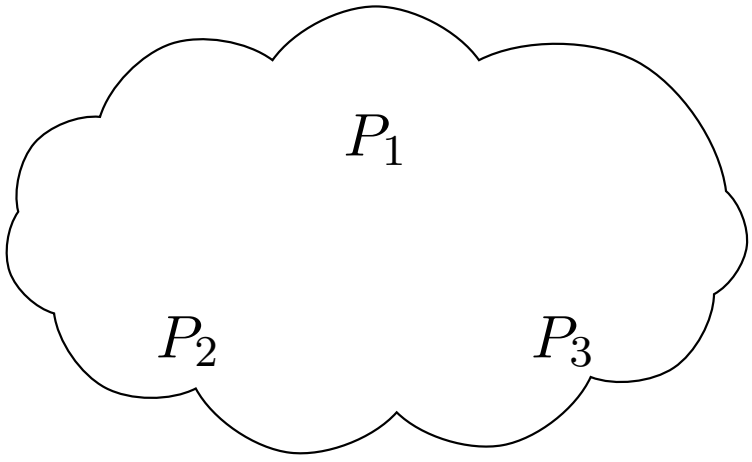
Prover



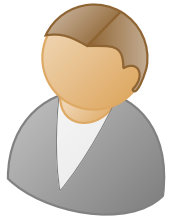
Verifier

Extending [IKOS07]

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



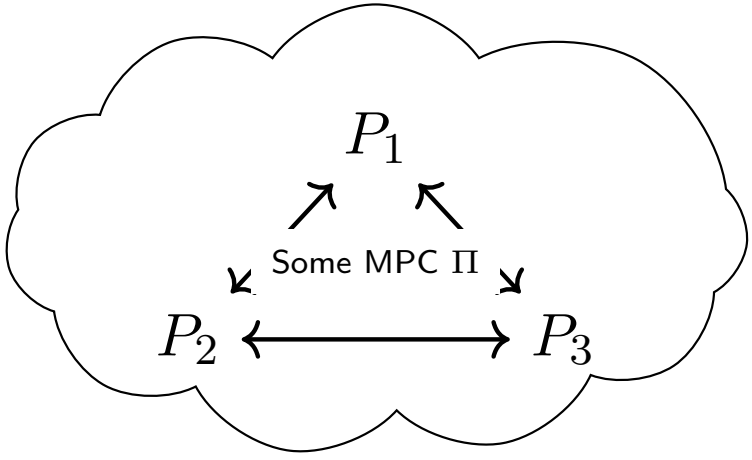
Prover



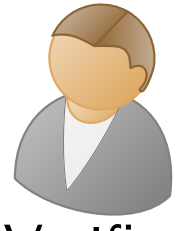
Verifier

Extending [IKOS07]

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



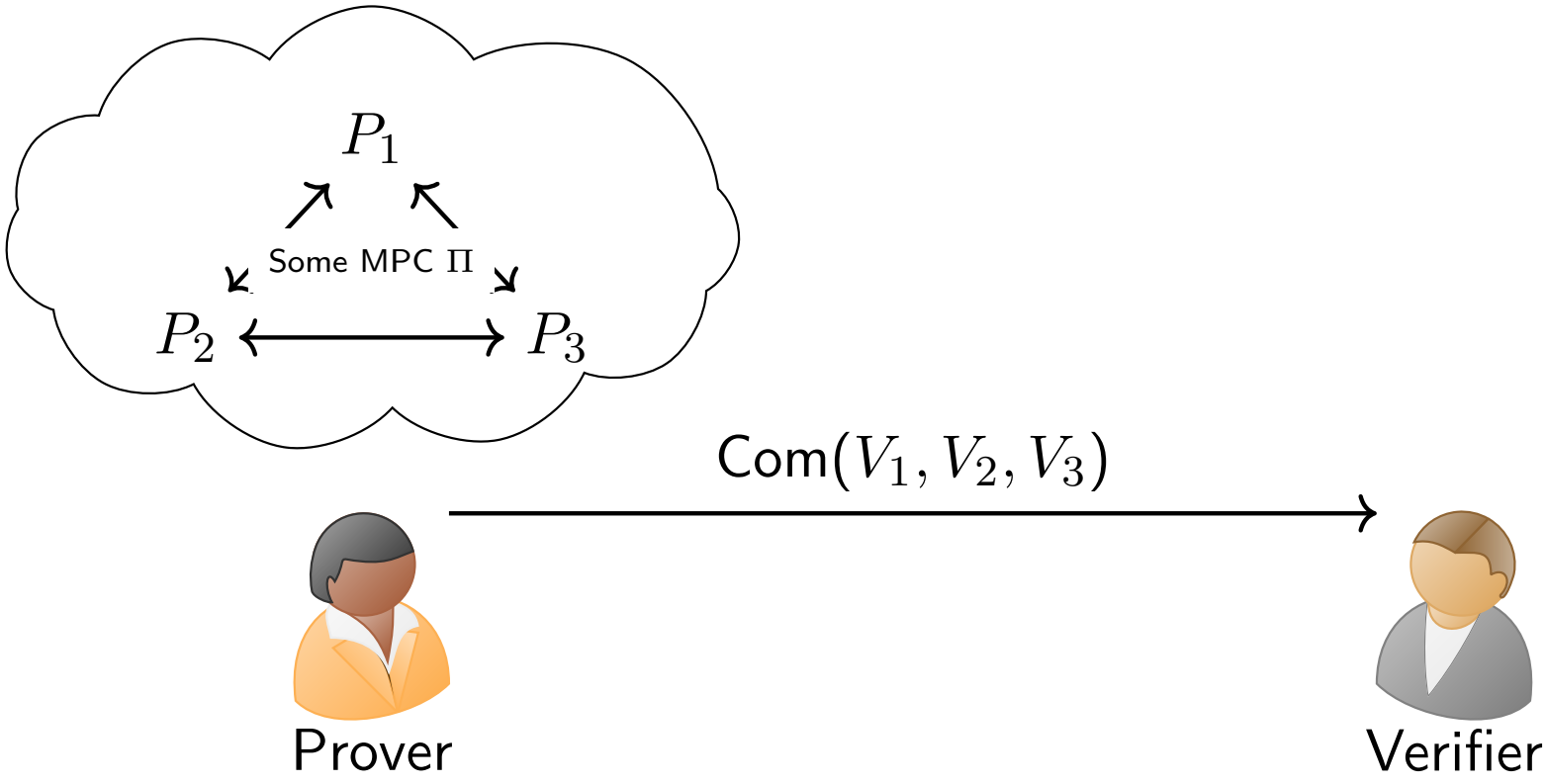
Prover



Verifier

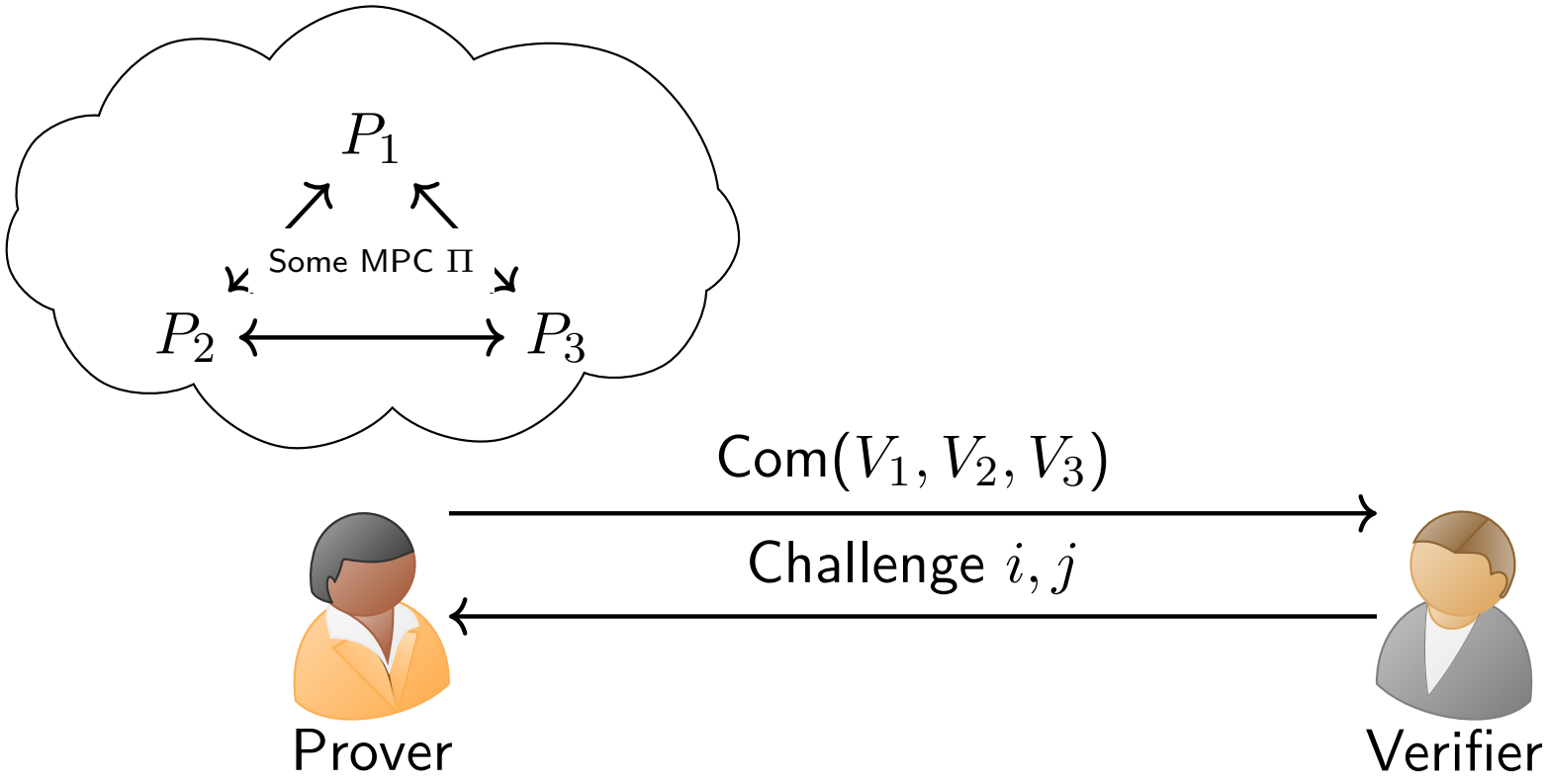
Extending [IKOS07]

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



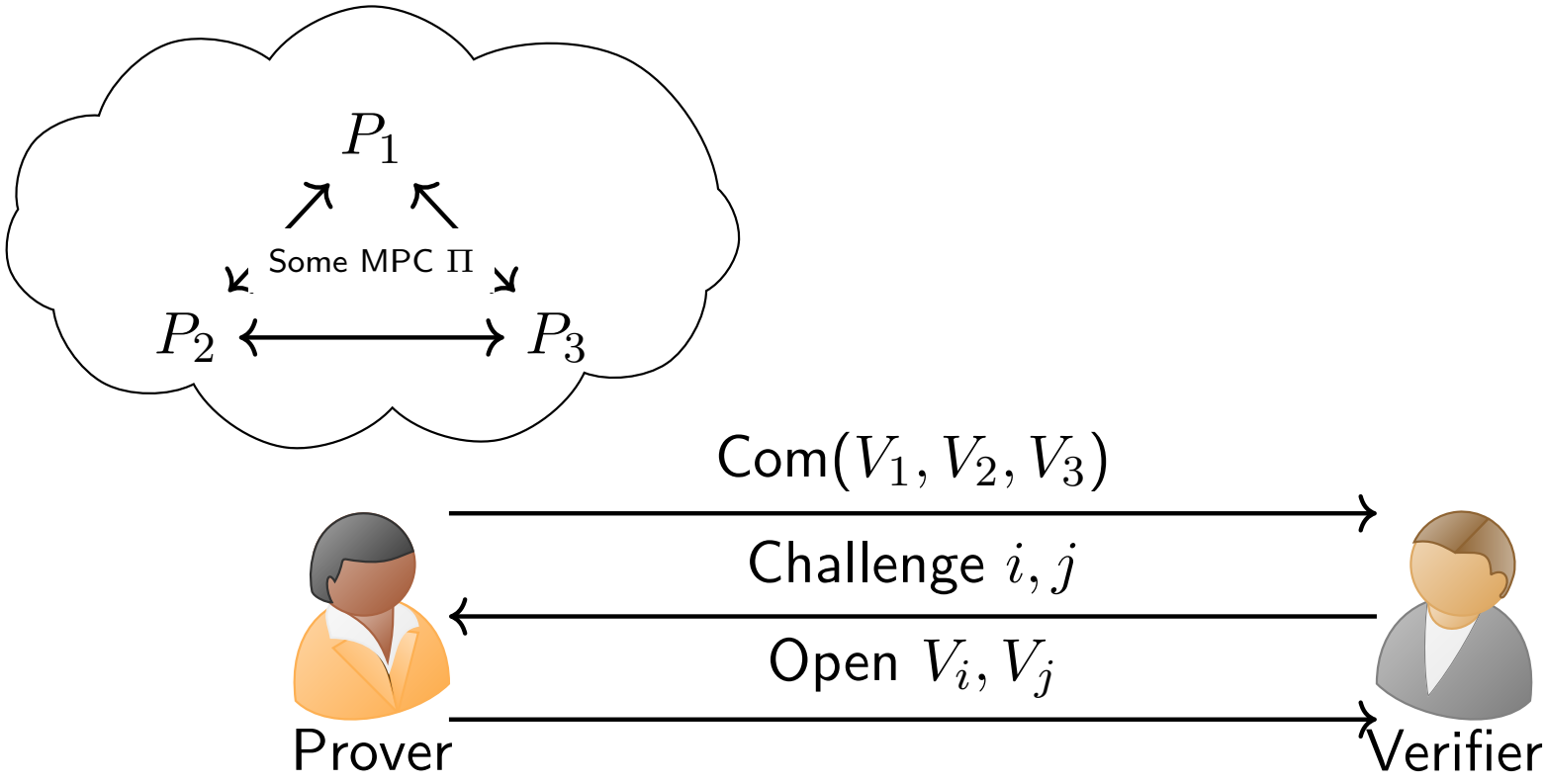
Extending [IKOS07]

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



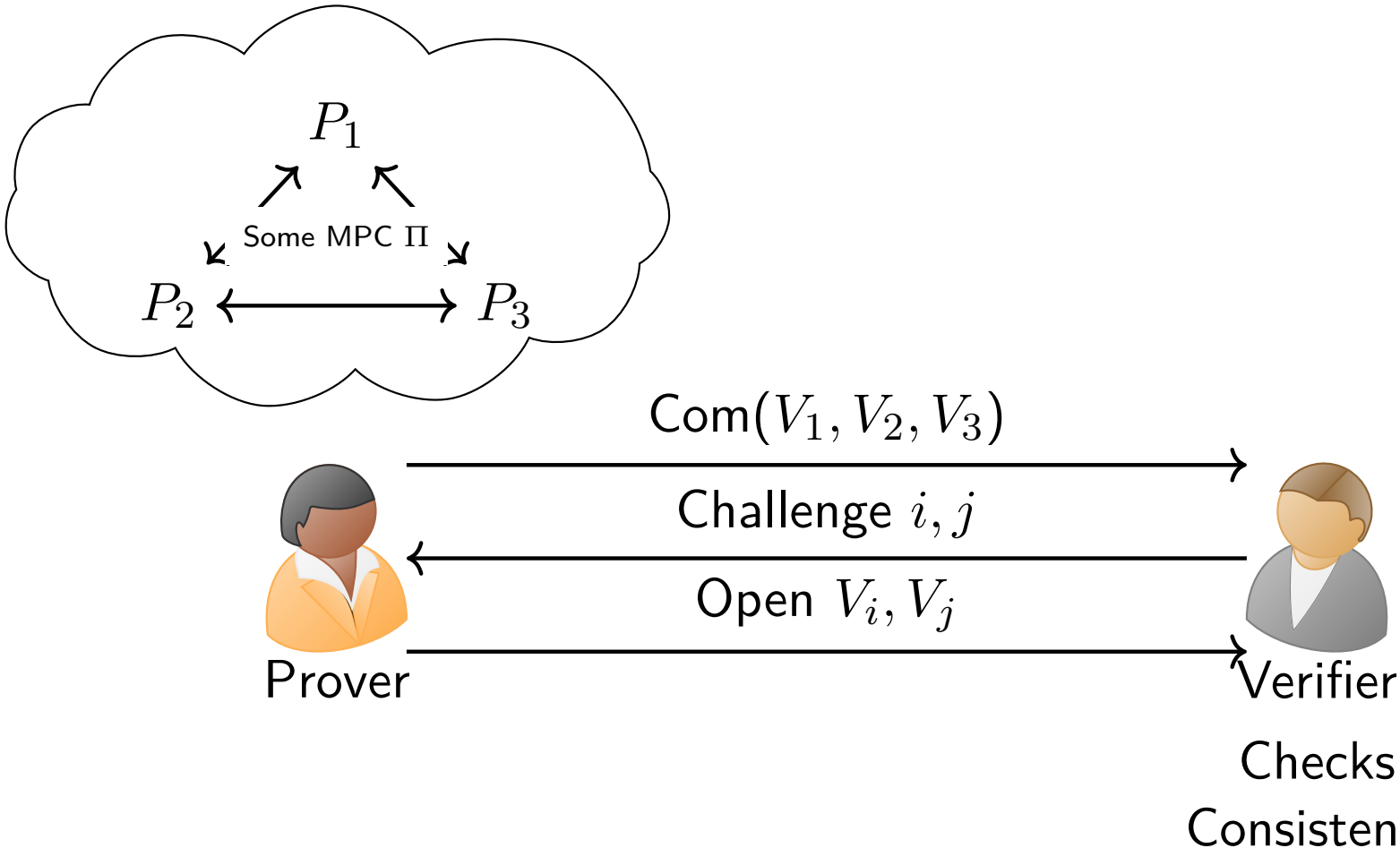
Extending [IKOS07]

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



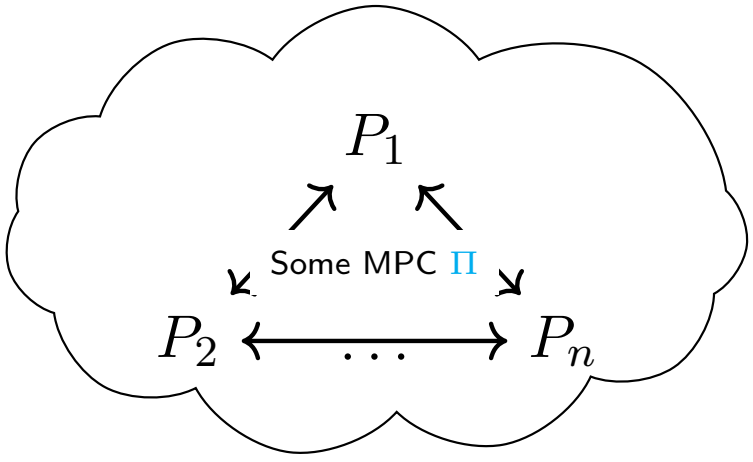
Extending [IKOS07]

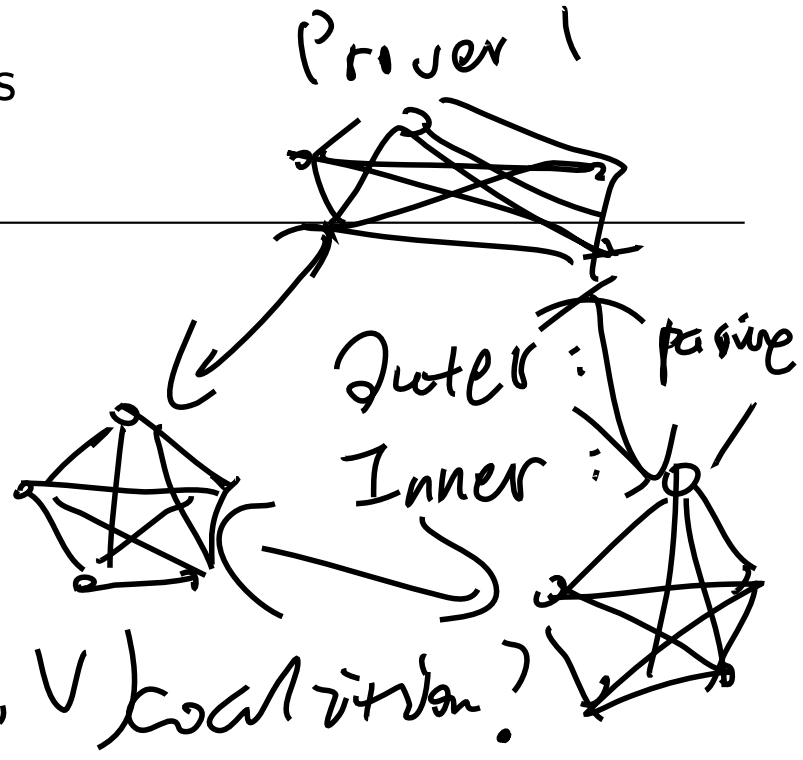
Consider the original MPC-in-the-Head construction of Ishai et al.



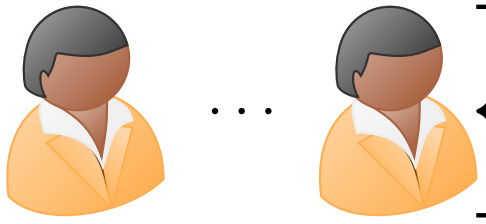
Extending [IKOS07]

Now we extend the number of provers





(P, V) Local virtual?

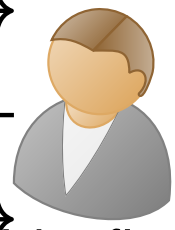


Prover 1 $\cdots$ Prover m

$\text{Com}(V_1, \ldots, V_n)$

Challenge S

Open $V_i : i \in S$



Verifier

Checks Consistency

$t \geq n/m$

More Details

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

A | B | C |

$R(\Sigma w; x)$

More Details

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

- Comm. complexity is $\Omega(|C|)$
- Π needs strong security to protect honest prover's privacy

Advanced Constructions (Attempt)

Attempting to Lower Comm. Complexity

- The LevioSA paper [HIMV19] proposed an implementation of the IPS compiler [IPS08, IPS09] in the 2PC setting
- if we can make it publicly-verifiable, then our goal is achieved

Advanced Constructions (Attempt)

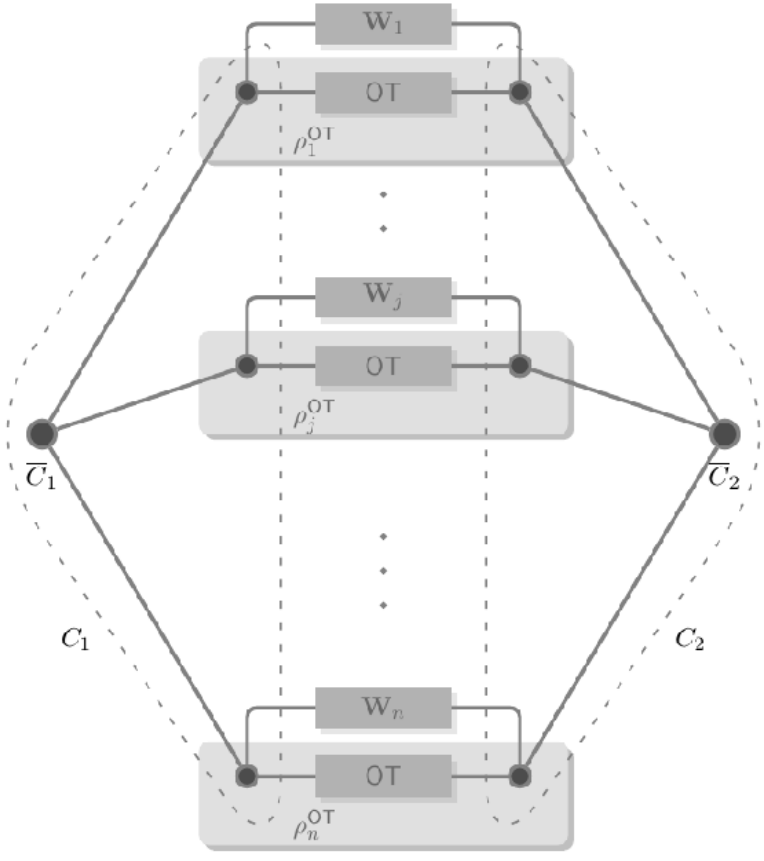
Attempting to Lower Comm. Complexity

- The LevioSA paper [HIMV19] proposed an implementation of the IPS compiler [IPS08, IPS09] in the 2PC setting
- if we can make it publicly-verifiable, then our goal is achieved

---

Caveats: their protocol only handles "layered" circuits. So efficiency gain is only obvious on "shallow-and-wide" circuits.

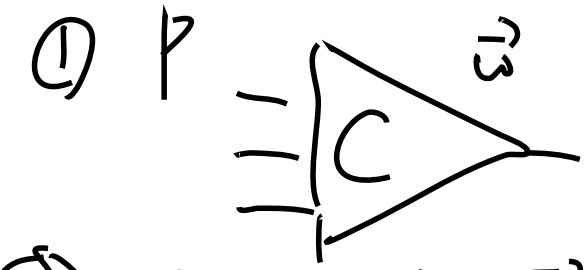
Overview of IPS Compiler

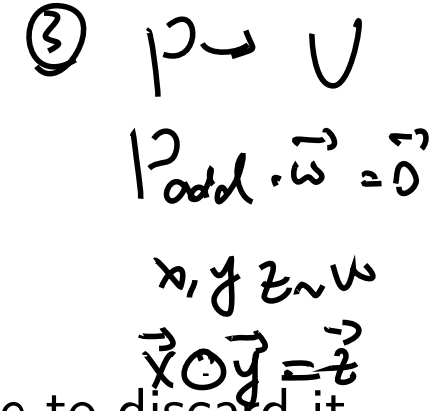


Basic Ideas

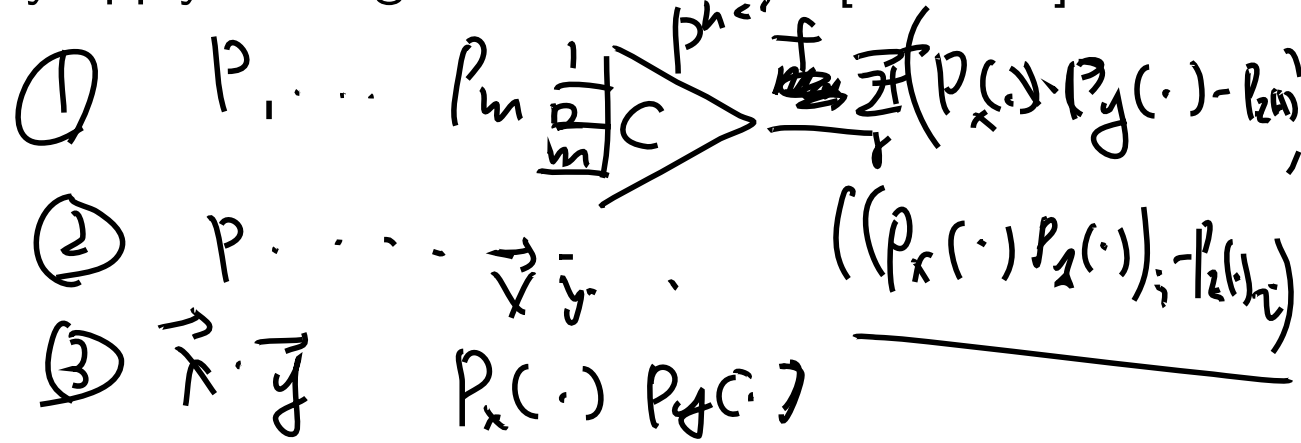
- Outer MPC uses n additional parties, actively-secure
- Inner MPC simulates those additional parties, passively-secure
- Deviation is to be caught by OT-based "watchlist"

Utilizing LevioSA





- Claim: watchlist is inherently interactive, we have to discard it
- We can make the parties commit to their views of inner protocol once they are ready, and check t of them before output phase
- This effectively equals to postponing all checks before output
- Claim: to ensure privacy, we need more security out of inner protocol (e.g. passive-OLE to active-OLE)
- After that, we may apply the argument similar to [AHIV17] and [DPSZ12]



Reference I

Benchmark: Constant Round (BMR)?

Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam.
Ligero: Lightweight sublinear arguments without a trusted setup.
In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 2087–2104. ACM Press, October / November 2017.

Gilad Asharov and Claudio Orlandi.
Calling out cheaters: Covert security with public verifiability.
In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 681–698. Springer, Heidelberg, December 2012.

Reference II

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 III


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 IV


Robert K. Cunningham, Benjamin Fuller, and Sophia Yakoubov.
Catching MPC cheaters: Identification and openability.
In Junji Shikata, editor, *ICITS 17*, volume 10681 of *LNCS*, pages 110–134. Springer, Heidelberg, November / December 2017.


Ivan Damgård and Yuval Ishai.
Scalable secure multiparty computation.
In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 501–520. Springer, Heidelberg, August 2006.


Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias.
Multiparty computation from somewhat homomorphic encryption.

Reference V

In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 643–662. Springer, Heidelberg, August 2012.

Irene Giacomelli, Jesper Madsen, and Claudio Orlandi.
ZKBoo: Faster zero-knowledge for Boolean circuits.
In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016*, pages 1069–1083. USENIX Association, August 2016.

Carmit Hazay, Yuval Ishai, Antonio Marcedone, and Muthuramakrishnan Venkitasubramaniam.
LevioSA: Lightweight secure arithmetic computation.
In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 327–344. ACM Press, November 2019.

Reference VI

Cheng Hong, Jonathan Katz, Vladimir Kolesnikov, Wen-jie Lu, and Xiao Wang.
Covert security with public verifiability: Faster, leaner, and simpler.
In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 97–121. Springer, Heidelberg, May 2019.

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.

Yuval Ishai, Manoj Prabhakaran, and Amit Sahai.
Founding cryptography on oblivious transfer - efficiently.
In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 572–591. Springer, Heidelberg, August 2008.

Reference VII


Yuval Ishai, Manoj Prabhakaran, and Amit Sahai.
Secure arithmetic computation with no honest majority.
In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 294–314. Springer, Heidelberg, March 2009.


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.

Reference VIII