ENCRYPTO
CRYPTOGRAPHY AND
PRIVACY ENGINEERING

Don't Eject the Impostor

Fast Three-Party Computation With a Known Cheater

paper @ IEEE S&P'24

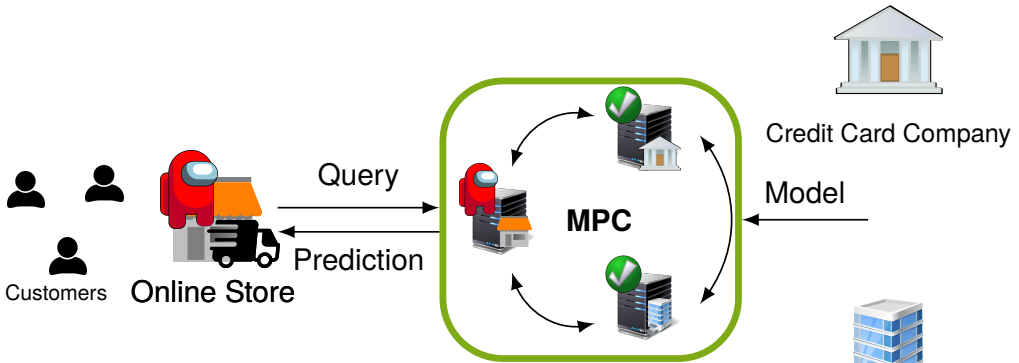
Andreas Brüggemann, Oliver Schick, Thomas Schneider, Ajith Suresh, Hossein Yalame

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Contact

https://encrypto.de

Motivating Example: Credit Scoring & Fraud Detection



- Semi-honest security model ⇒ Not sufficient
- Malicious security model ⇒ Too inefficient
- Exclude cheater ⇒ Also too inefficient

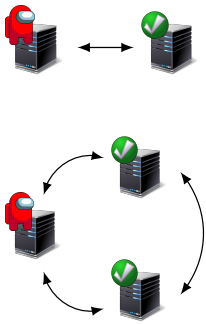
This setting is not restricted to secure inference

Trust in different people or institutions is commonly asymmetric

Prior Works ⇒ Honest Majority

ENCRYPTO
CRYPTOGRAPHY AND
PRIVACY ENGINEERING

- Asymmetric 2PC ⇒ dishonest majority
  - Yao's garbled circuits* [Yao86]
  - Secure inference with a malicious client [LMS+21; CGO+22]
  - Secure inference with a malicious server [DWL+23]
- **This work:**
  - **asymmetric 3PC**
  - **one known malicious party**
  - ⇒ **honest majority**



[Yao86] A. C.-C. Yao. "How to Generate and Exchange Secrets (Extended Abstract)". *FOCS*. 1986

[LMS+21] R. Lehmkuhl, P. Mishra, A. Srinivasan, and R. A. Popa. "Muse: Secure Inference Resilient to Malicious Clients". *USENIX Security*. 2021

[CGO+22] N. Chandran, D. Gupta, S. L. B. Obbattu, and A. Shah. "SIMC: ML Inference Secure Against Malicious Clients at Semi-Honest Cost". *USENIX Security*. 2022

[DWL+23] C. Dong, J. Weng, J.-N. Liu, Y. Zhang, Y. Tong, A. Yang, Y. Cheng, and S. Hu. "Fusion: Efficient and Secure Inference Resilient to Malicious Servers". *NDSS*. 2023

Key Takeaways

ENCRYPTO
CRYPTOGRAPHY AND
PRIVACY ENGINEERING

- **Think Big!** Compute on matrices, not scalars.
- **Don't Eject the Impostor!** Cooperate with untrustworthy parties if this yields an honest majority.
- **Embrace Diversity Between Parties!** There exists an interesting world between the fully semi-honest and malicious settings.

Part I: Building Blocks

How to Multiply (Optimistically)

Recap of Replicated Secret-Sharing [AFL+16]

- $[\vec{X}]$: Party P_i holds $\vec{X}^i, \vec{X}^{i-1}$ s.t. $\vec{X}^0 + \vec{X}^1 + \vec{X}^2 = \vec{X}$
- Multiplication of $[\vec{A}]$ and $[\vec{B}]$:
  1. Precompute $[\vec{0}]$
  2. P_i computes $\vec{C}^i = \vec{A}^i \cdot \vec{B}^i + \vec{A}^i \cdot \vec{B}^{i-1} + \vec{A}^{i-1} \cdot \vec{B}^i + \vec{0}^i$
  3. P_i sends $\vec{C}^i$ to P_{i+1}
- **Easily generalizes to matrix multiplication!**
- Also works with additive sharings + Beaver triples and *masked secret-sharing* (ABY2.0 [PSS+21], Astra [CCP+19], Swift [KPP+21], ...)

[PSS+21] A. Patra, T. Schneider, A. Suresh, and H. Yalame. "ABY2.0: Improved Mixed-Protocol Secure Two-Party Computation". *USENIX Security*. 2021

[CCP+19] H. Chaudhari, A. Choudhury, A. Patra, and A. Suresh. "ASTRA: High Throughput 3PC over Rings with Application to Secure Prediction". *CCSW@CCS*. 2019

[KPP+21] N. Koti, M. Pancholi, A. Patra, and A. Suresh. "SWIFT: Super-fast and Robust Privacy-Preserving Machine Learning". *USENIX Security*. 2021

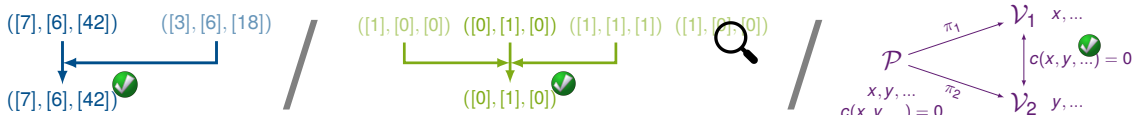
[AFL+16] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara. "High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority". *CCS*. 2016

How to Detect and Prevent Cheating

A Selection of Efficient Verification Methods

ENCRYPTO
CRYPTOGRAPHY AND PRIVACY ENGINEERING

- Triple Sacrificing (ring version) [CDE+18] — **low comp.**, linear comm.
- Cut-and-Choose (binary) [ABF+17] — **low comp.**, linear comm.
- Distributed ZKPs [BBC+19; BGI+19] — high comp., **sublinear comm.**



[CDE+18] R. Cramer, I. Damgård, D. Escudero, P. Scholl, and C. Xing. "SPD$\mathbb{Z}_{2^k}$: Efficient MPC mod 2^k for Dishonest Majority". *CRYPTO*. 2018

[ABF+17] T. Araki, A. Barak, J. Furukawa, T. Lichter, Y. Lindell, A. Nof, K. Ohara, A. Watzman, and O. Weinstein. "Optimized Honest-Majority MPC for Malicious Adversaries - Breaking the 1 Billion-Gate Per Second Barrier". *IEEE S&P*. 2017

[BBC+19] D. Boneh, E. Boyle, H. Corrigan-Gibbs, N. Gilboa, and Y. Ishai. "Zero-Knowledge Proofs on Secret-Shared Data via Fully Linear PCPs". *CRYPTO*. 2019

[BGI+19] E. Boyle, N. Gilboa, Y. Ishai, and A. Nof. "Practical Fully Secure Three-Party Computation via Sublinear Distributed Zero-Knowledge Proofs". *CCS*. 2019

Close-Up: Asymmetric Triple Sacrificing

ENCRYPTO
CRYPTOGRAPHY AND
PRIVACY ENGINEERING

- Generate multiplication triples optimistically
- Check correctness of a triple $([\vec{A}], [\vec{B}], [\vec{C}])$ (i.e., $\vec{C} = \vec{A}\vec{B}$),
  using a correlated triple $([\hat{\vec{A}}], [\vec{B}], [\hat{\vec{C}}])$
  1. $r \leftarrow_{\$} \mathbb{Z}_{2^\sigma}$ (batched [EKO+20])
  2. $[\vec{V}] = r[\vec{A}] - [\hat{\vec{A}}]$
  3. Recover $\vec{V}$
  4. $[\vec{W}] = \vec{V}[\vec{B}] - r[\vec{C}] + [\hat{\vec{C}}]$
  5. Verify $\vec{W} \equiv_{2^{\ell+\sigma}} \vec{0}$ (batched [EKO+20])

  } **Executed between semi-honest servers only**

- $\vec{W} \equiv_{2^{\ell+\sigma}} \vec{0} \Leftrightarrow r(\vec{C} - \vec{A}\vec{B}) \equiv_{2^{\ell+\sigma}} \hat{\vec{C}} - \hat{\vec{A}}\vec{B} \Rightarrow_{\text{exc. negl. prob.}} \vec{C} \equiv_{2^\ell} \vec{A}\vec{B}$
- Our Observation: Seamlessly generalizes to matrix multiplication

[EKO+20] H. Eerikson, M. Keller, C. Orlandi, P. Pullonen, J. Puura, and M. Simkin. “Use Your Brain! Arithmetic 3PC for Any Modulus with Active Security”. *ITC*. 2020

Think Big!

Compute on matrices, not scalars.

This Work

ENCRYPTO
CRYPTOGRAPHY AND
PRIVACY ENGINEERING

- Do expensive work & verification in setup phase
- Use masked secret-sharing (as in ABY2.0 [PSS+21], Astra [CCP+19], Swift [KPP+21], ...)
- **Two new 3PC protocols: Auxiliator & Socium**

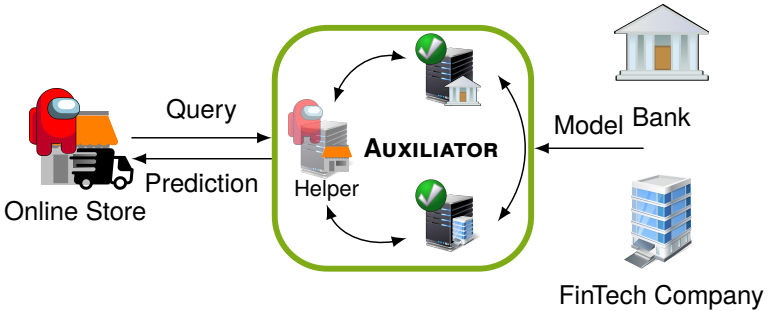
[PSS+21] A. Patra, T. Schneider, A. Suresh, and H. Yalame. "ABY2.0: Improved Mixed-Protocol Secure Two-Party Computation". *USENIX Security*. 2021

[CCP+19] H. Chaudhari, A. Choudhury, A. Patra, and A. Suresh. "ASTRA: High Throughput 3PC over Rings with Application to Secure Prediction". *CCSW@CCS*. 2019

[KPP+21] N. Koti, M. Pancholi, A. Patra, and A. Suresh. "SWIFT: Super-fast and Robust Privacy-Preserving Machine Learning". *USENIX Security*. 2021

Part II: Letting Cheaters Assist

Auxiliator
3PC With an Untrusted Helper



- Like Astra [CCP+19] or ABY2.0 [PSS+21], but with an untrusted helper

[CCP+19] H. Chaudhari, A. Choudhury, A. Patra, and A. Suresh. “ASTRA: High Throughput 3PC over Rings with Application to Secure Prediction”. *CCSW@CCS*. 2019

[PSS+21] A. Patra, T. Schneider, A. Suresh, and H. Yalame. “ABY2.0: Improved Mixed-Protocol Secure Two-Party Computation”. *USENIX Security*. 2021

Communication per Operation

ENCRYPTO
CRYPTOGRAPHY AND PRIVACY ENGINEERING

$\ell = 32/64$: ring size $\kappa = 128$: computational security parameter $\sigma = 40$: statistical security parameter

	Operation	$\mathbf{x} \wedge \mathbf{y} = \mathbf{z}$	$\mathbf{a} \cdot \mathbf{b} = \mathbf{c}$	$\underset{u \times v}{\square} \cdot \underset{v \times w}{\square} = \underset{u \times w}{\square}$
	Setup			
ABY2.0 [PSS+21]	IKNP-style OT [ALS+13]	134 bits	$2(\kappa + \ell) \times \mathbb{Z}_{2^\ell}$	$2(\kappa + \ell)uvw \times \mathbb{Z}_{2^\ell}$
	Silent OT [BCG+19]	4 bits	$2(1 + \ell) \times \mathbb{Z}_{2^\ell}$	$2(1 + \ell)uvw \times \mathbb{Z}_{2^\ell}$
Auxiliator (**this work**)	C&C/Sacrifice	11 bits	$4 \times \mathbb{Z}_{2^{\ell+\sigma}}$	$2 \underset{u \times w}{\square} + 2 \underset{u \times v}{\square}$
	DZKP	1 bit	$1 \times \mathbb{Z}_{2^\ell}$	$\underset{u \times w}{\square}$
	Online			
ABY2.0/Auxiliator		2 bits	$2 \times \mathbb{Z}_{2^\ell}$	$2 \underset{u \times w}{\square}$

[PSS+21] A. Patra, T. Schneider, A. Suresh, and H. Yalame. “ABY2.0: Improved Mixed-Protocol Secure Two-Party Computation”. *USENIX Security*. 2021

[ALS+13] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner. “More Efficient Oblivious Transfer and Extensions for Faster Secure Computation”. *CCS*. 2013

[BCG+19] E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, P. Rindal, and P. Scholl. “Efficient Two-Round OT Extension and Silent Non-Interactive Secure Computation”. *CCS*. 2019

Communication for Secure Inference

Setup Phase

Layers		Non-linear	Linear	Total
ABY2.0 [PSS+21]	IKNP-style OT [ALS+13]	524.25 MB	174.78 GB	175.29 GB
	Silent OT [BCG+19]	23.04 MB	59.17 GB	59.19 GB
Auxiliator (**this work**)	C&C/Sacrifice	62.43 MB	10.46 MB	72.89 MB
	DZKP	6.42 MB	1.48 MB	7.90 MB

using the 7-layer CNN from MiniONN [LJL+17] trained on the CIFAR-10 dataset

[PSS+21] A. Patra, T. Schneider, A. Suresh, and H. Yalame. “ABY2.0: Improved Mixed-Protocol Secure Two-Party Computation”. *USENIX Security*. 2021

[ALS+13] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner. “More Efficient Oblivious Transfer and Extensions for Faster Secure Computation”. *CCS*. 2013

[BCG+19] E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, P. Rindal, and P. Scholl. “Efficient Two-Round OT Extension and Silent Non-Interactive Secure Computation”. *CCS*. 2019

[LJL+17] J. Liu, M. Juuti, Y. Lu, and N. Asokan. “Oblivious Neural Network Predictions via MiniONN Transformations”. *CCS*. 2017

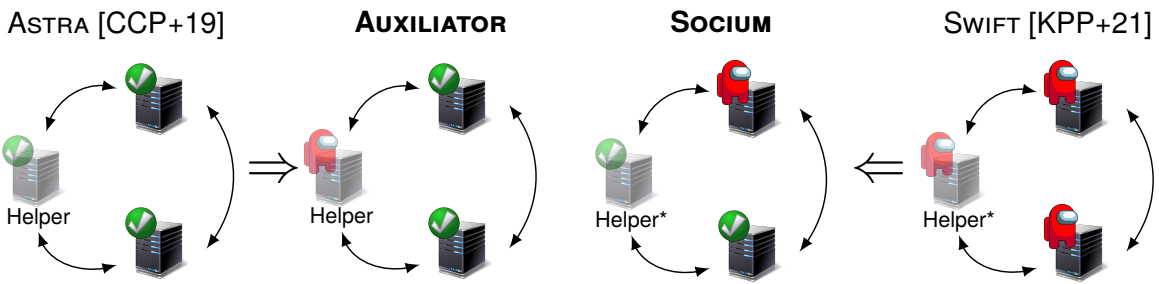
Don’t Eject the Impostor!

Cooperate with untrustworthy parties if this yields an honest majority.

ENCRYPTO
CRYPTOGRAPHY AND
PRIVACY ENGINEERING

Part III: The World Between Symmetrical Semi-Honest and Malicious Models

The World Between Astra & Swift



[CCP+19] H. Chaudhari, A. Choudhury, A. Patra, and A. Suresh. “ASTRA: High Throughput 3PC over Rings with Application to Secure Prediction”. *CCSW@CCS*. 2019

[KPP+21] N. Koti, M. Pancholi, A. Patra, and A. Suresh. “SWIFT: Super-fast and Robust Privacy-Preserving Machine Learning”. *USENIX Security*. 2021

Communication per Operation

ENCRYPTO
CRYPTOGRAPHY AND
PRIVACY ENGINEERING

$\ell = 32/64$: ring size $\quad \sigma = 40$: statistical security parameter

Operation	$\mathbf{a} \cdot \mathbf{b} = \mathbf{c}$		$(\blacksquare\blacksquare\blacksquare) \cdot (\blacksquare\blacksquare) = (\blacksquare\blacksquare)$		$\mathbf{x} \wedge \mathbf{y} = \mathbf{z}$	
Setup						
Verification	Sacrifice	DZKP	Sacrifice	DZKP	C&C	DZKP
Astra [CCP+19]	$1 \times \mathbb{Z}_{2^\ell}$		$(\blacksquare\blacksquare)$		1 bit	
Auxiliator (**this work**)	$4 \times \mathbb{Z}_{2^{\ell+\sigma}}$	$1 \times \mathbb{Z}_{2^\ell}$	$2(\blacksquare\blacksquare) + 2(\blacksquare\blacksquare\blacksquare)$	$(\blacksquare\blacksquare)$	11 bits	1 bit
Socium (**this work**)	$5 \times \mathbb{Z}_{2^{\ell+\sigma}}$	$2 \times \mathbb{Z}_{2^\ell}$	$3(\blacksquare\blacksquare) + 2(\blacksquare\blacksquare\blacksquare)$	$2(\blacksquare\blacksquare)$	12 bits	2 bits
Swift [KPP+21]	$9 \times \mathbb{Z}_{2^{\ell+\sigma}}$	$3 \times \mathbb{Z}_{2^\ell}$	$6(\blacksquare\blacksquare) + 3(\blacksquare\blacksquare\blacksquare)$	$3(\blacksquare\blacksquare)$	21 bits	3 bits
Online						
Astra/Auxiliator	$2 \times \mathbb{Z}_{2^\ell}$		$2(\blacksquare\blacksquare)$		2 bits	
Swift/Socium	$3 \times \mathbb{Z}_{2^\ell}$		$3(\blacksquare\blacksquare)$		3 bits	

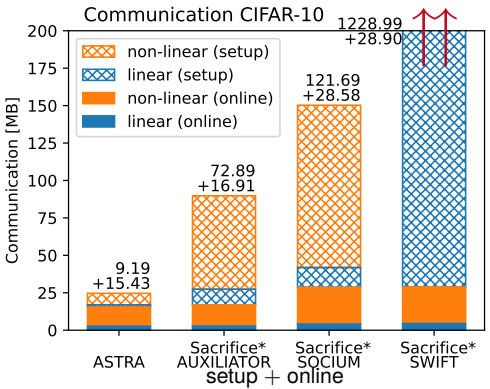
[CCP+19] H. Chaudhari, A. Choudhury, A. Patra, and A. Suresh. "ASTRA: High Throughput 3PC over Rings with Application to Secure Prediction". *CCSW@CCS*. 2019

[KPP+21] N. Koti, M. Pancholi, A. Patra, and A. Suresh. "SWIFT: Super-fast and Robust Privacy-Preserving Machine Learning". *USENIX Security*. 2021

Communication for Secure Inference

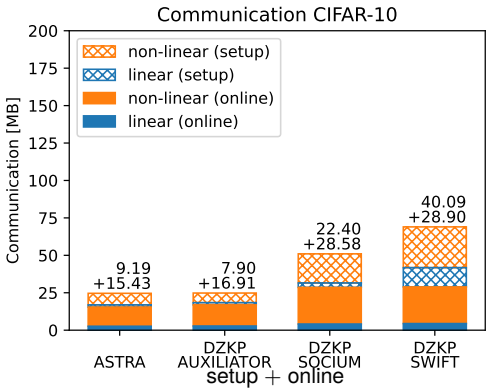
ENCRYPTO
CRYPTOGRAPHY AND
PRIVACY ENGINEERING

Sacrifice



*: Cut-and-Choose for binary domain

DZKP

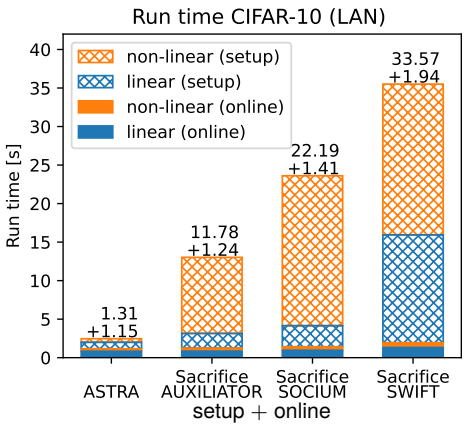


using the 7-layer CNN from MiniONN [LJL+17] trained on the CIFAR-10 dataset

[LJL+17] J. Liu, M. Juuti, Y. Lu, and N. Asokan. “Oblivious Neural Network Predictions via MiniONN Transformations”. *CCS*. 2017

Run Time for Secure Inference

Sacrifice



Full Open-Source Implementation of all protocols and sacrifice verification available at:

https://encrypto.de/code/MOTION-FD

Embrace Diversity Between Parties!

There exists an interesting world between the fully semi-honest and malicious settings.

ENCRYPTO
CRYPTOGRAPHY AND
PRIVACY ENGINEERING

Think Big!
Don't Eject the Impostor!
Embrace Diversity Between Parties!

Questions?

Paper

https://ia.cr/2023/1744

Code

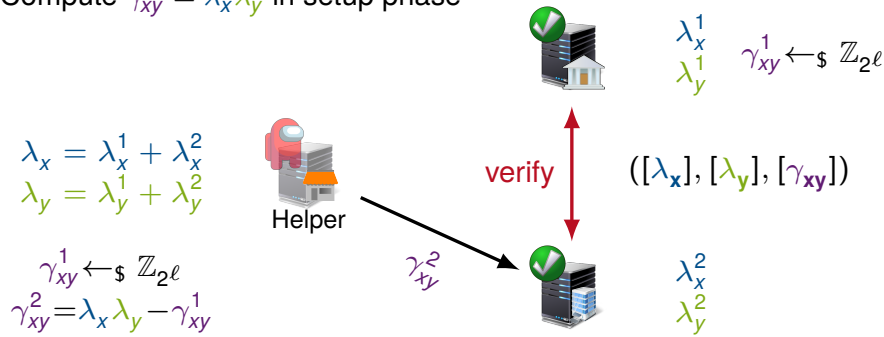
https://encrypto.de/code/MOTION-FD

Website

https://encrypto.de

Close-Up: Astra ⇒ Auxiliator

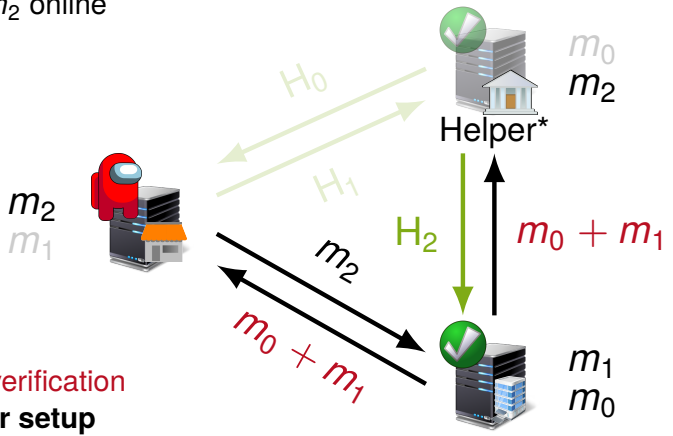
Goal: Compute $\gamma_{xy} = \lambda_x \lambda_y$ in setup phase



- Semi-honest parties verify correctness
- Helper provides additional triples or proof

Close-Up: Swift ⇒ Socium

Goal: Recover $m = m_0 + m_1 + m_2$ online



- Setup based on replicated secret-sharing [AFL+16] + verification
- Less m_i required ⇒ **cheaper setup**
- Reduction to 2 shares ⇒ cheaper truncation

[AFL+16] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara. "High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority". *CCS*. 2016