

An End-to-End System for Large Scale P2P MPC-as-a-Service and Low- Bandwidth MPC for Weak Participants

Yehuda Lindell

Bar-Ilan University, Israel

Based on joint works with: A. Barak, K. China, J. Furukawa D. Genkin, K. Hamada, M. Hirt, D. Ikarashi, R. Kikuchi, L. Koskas and A. Nof at CRYPTO'18, ACM CCS'18 and under preparation

Secure Multiparty Computation (MPC)

- A set of parties with private inputs wish to compute a joint function of their inputs
 - Ensuring that **nothing but the output is learned** (privacy)
 - Ensuring that the **output is correctly computed** (correctness)
- These properties should be guaranteed even in the face of adversarial behavior
- Additional properties
 - Independence of inputs
 - Fairness
 - Guaranteed output delivery

Security Requirements

- Consider **comparing DNA to know if two people are close family**
 - Wish to do this without revealing actual DNA
- Adversarial threats
 - An adversary may try to learn the other person's DNA or some property of it like tendency to some illness (breach of **privacy**)
 - An adversary may wish to have the result be that s/he's close family to get the inheritance (breach of **correctness**)

Modeling Adversaries

- **Adversarial behavior**

- **Semi-honest**: follows the protocol specification
 - Tries to learn more than allowed by inspecting transcript
- **Malicious**: follows any arbitrary strategy
 - Much stronger security guarantees; much more expensive

- **Corruption threshold**

- **Honest majority** (or 2/3 majority):
 - Can get information-theoretic security
- **Dishonest majority**:
 - Better security guarantee; much more expensive

Feasibility – Fundamental Theorems from the 80s

- **Any polynomial-time functionality** can be securely computed with computational security (assuming oblivious transfer), with and without an honest majority [Yao,GMW]
- **Any polynomial-time functionality** can be securely computed with information theoretic security (assuming ideal channels), with a 2/3 honest majority [BGW,CCD], and with an honest majority (assuming broadcast) [RB]
- **These are theoretical feasibility results; can they be realized in practice?**
 - A lot of work has been done in the past decade and we can carry out significant computations today
 - But cannot compute on massive databases!

Secure Computation – Potential and Reality

- Secure computation is now being used in practice and there is increasing interest from industry
 - Processing of encrypted data
 - Secure statistics
 - Key and biometric protection

Privacy-Preserving Analytics

Privacy and Security User-Centric Distributed Solutions for Privacy- Preserving Analytics

How can cryptography empower users with sensitive data to access large-scale computing platforms in a privacy-preserving manner?

FOR OVER A YEAR, a high-profile initiative spearheaded by the City of Boston and the Boston Women's Workforce Council (BWWC) strived to identify salary inequities across various employee gender and ethnic demographics at different levels of employment, from executive to entry-level positions.¹¹ While the effort was supported by a diverse set of more than 100 employer organizations in the city—including major corporations, small businesses, and public/non-profit organizations—it was stalled by concerns about the confidentiality of the data to be collected in order to calculate aggregate metrics.²

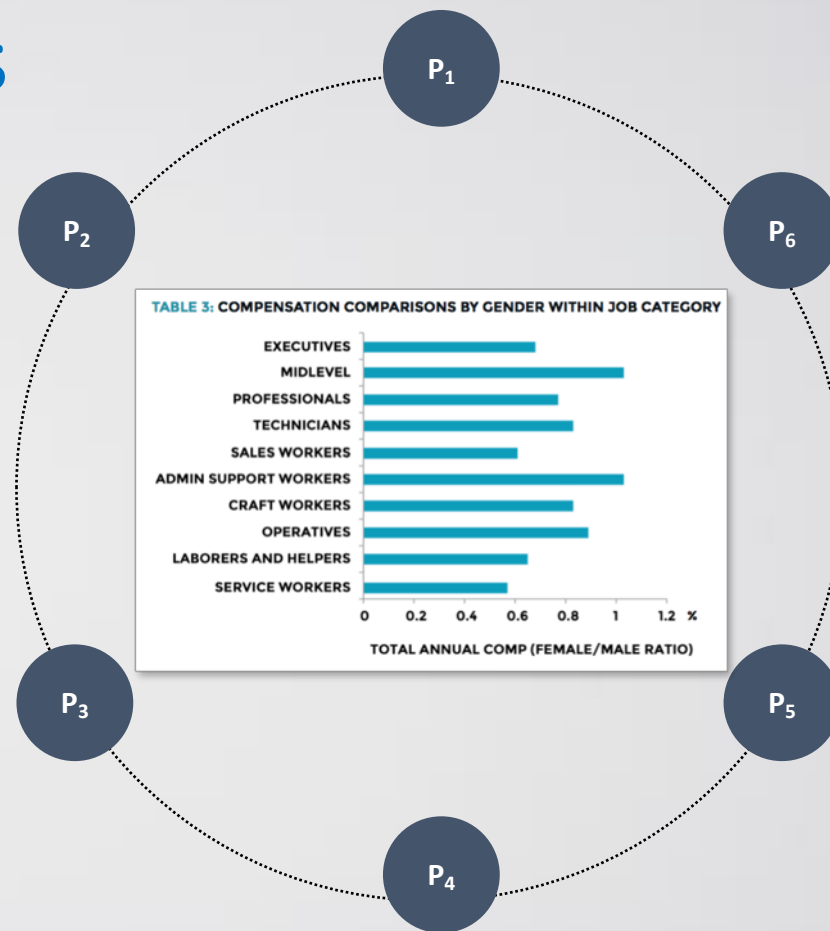
MPC privately shards users' sensitive data across multiple servers in such a way that analytics may be jointly computed and released while ensuring that (small collections of) servers cannot learn any user's data. Theoretical constructs for MPC have been known for 35 years, with several existing software frameworks designed over the past 10 years.^{7,9}

MPC techniques can possess substantial social value: they enable society to benefit from collective data aggregation and analysis in contexts where the raw data is encumbered by legal and corporate policy restrictions on data sharing. Other examples of deploying MPC for social good include tax fraud

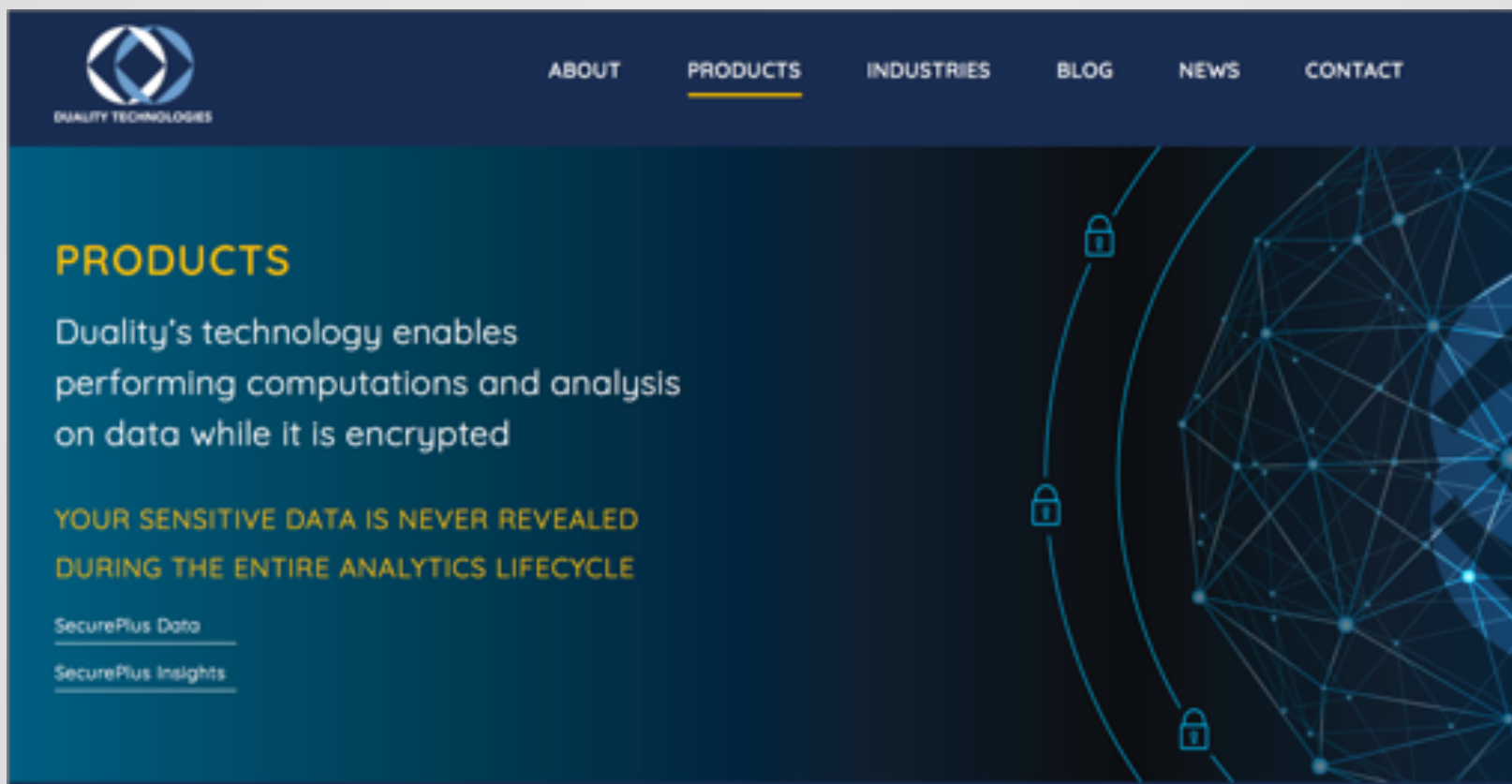
non-CIOs
yers
tions
of th
the s
take
ence:
board
both
conci
vidin
and h
guar
ly, th
demy
and,
stand

**BOSTON
WOMEN'S
WORKFORCE
COUNCIL
REPORT
2016**

“Moreover, the BWWC collected actual wage data from 69 companies earlier this year. This represented a first ever totally confidential reporting of such data aimed at providing an average wage gap as a baseline, against which to measure progress for the city.”

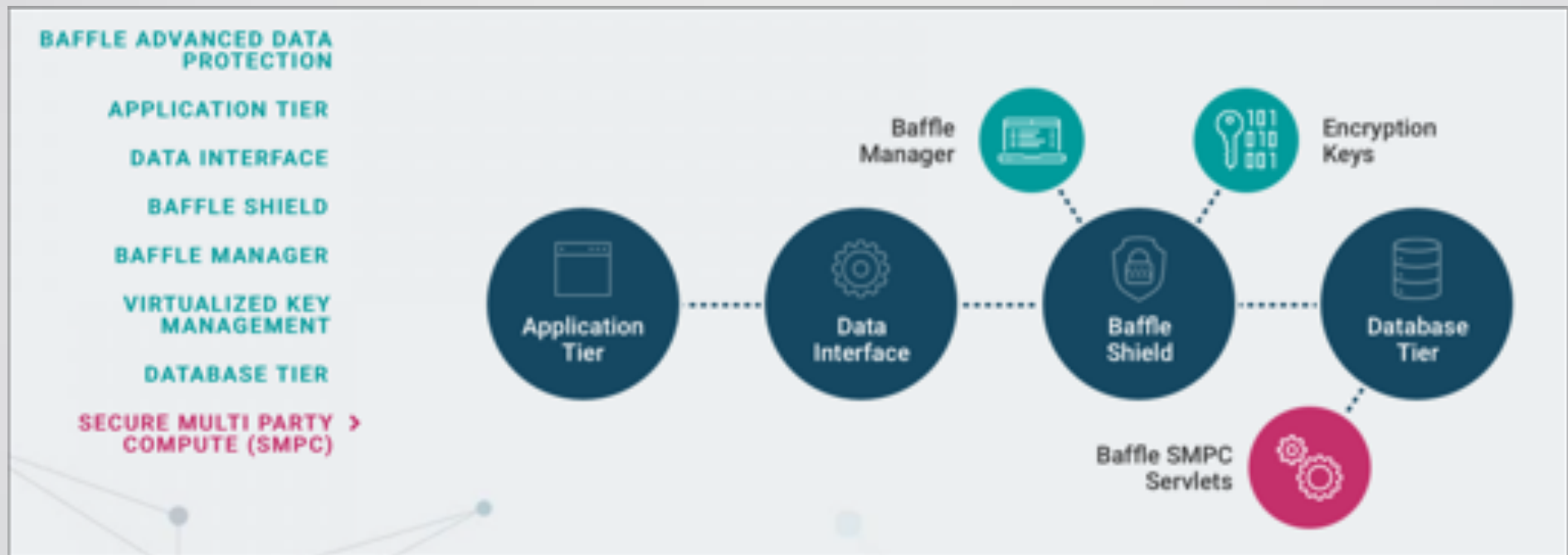


Duality: Collaborate by Computing on Encrypted Data



The screenshot shows the Duality Technologies website. The top navigation bar includes the company logo and links for ABOUT, PRODUCTS (which is underlined), INDUSTRIES, BLOG, NEWS, and CONTACT. The main content area features the heading "PRODUCTS" in yellow, followed by the text "Duality's technology enables performing computations and analysis on data while it is encrypted". Below this is a yellow quote: "YOUR SENSITIVE DATA IS NEVER REVEALED DURING THE ENTIRE ANALYTICS LIFECYCLE". At the bottom of the main content, there are two links: "SecurePlus Data" and "SecurePlus Insights". The background of the main content area is a dark blue network diagram with several padlock icons.

Baffle: Compute on Encrypted Data – Protect Your Data While in Use



Unbound: Protection of Cryptographic Keys

UNBOUND
(MATH OVER MATTER)

PRODUCTS USE CASES SOLUTIONS TECHNOLOGY RESOURCES COMPANY BLOG

LET'S TALK
FREE TRIAL

Security Based on Mathematical Proof

Unbound's Distributed Trust Platform applies revolutionary breakthroughs in Multi-Party Computation mathematics to split secrets into parts that are never again united, completely eliminating the single point of failure for your most sensitive assets.

- 1** Each secret exists as separate random shares stored in separate locations
- 2** The shares are never combined at any point in time – not even when used
- 3** Secrets never exist in complete form throughout their lifecycle

Private P2P – The Basic Promise of MPC

- All current use-case examples are B2B (or maybe B2C)
- The basic MPC promise
 - An arbitrary set of parties (decentralized P2P setting)
 - Compute on their private data (their own private data)
 - Obtain output (**they gain utility from their own data**)
- Why don't we have peer-to-peer (P2P) MPC?

Obstacles to P2P MPC

- How can decentralized parties agree what to run and when, and set up an appropriate environment?
 - How do they deploy software?
 - How do they agree upon who joins, and how do they know their IDs?
- End users use browsers and mobile apps, and don't install software
- Almost all MPC protocols require all parties to be online simultaneously
- The high bandwidth of many MPC protocols is an obstacle to mobile deployment
- A much better gender gap study would be P2P and involve individuals
 - Less legal problems, larger sample, diverse geographics

MPC With Inputs From Many Parties

- Currently, in order to run MPC with inputs from many parties
 - A small set of servers are defined to run the actual MPC
 - All parties send shares of their inputs to the servers
 - The servers run the MPC and provide output
- Disadvantages
 - Who runs the servers?
 - Do we trust them?
 - Do we all agree that we can trust them?

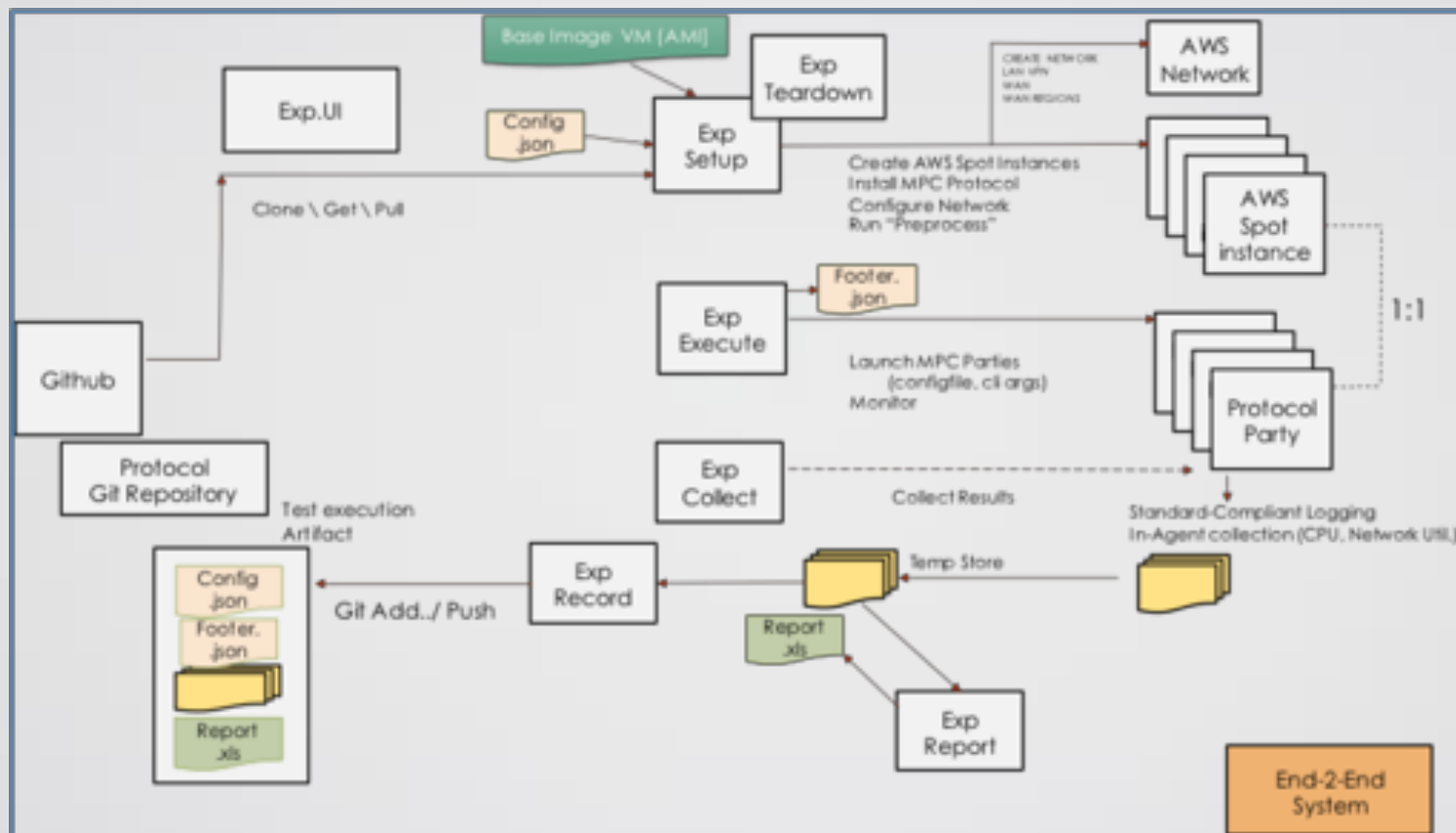
An End-to-End System for MPC

- Works the way modern software works
 - End users use browsers or mobile apps
 - Service model: cloud service provider offers the MPC service
 - Subscribers purchase/use the service to initiate MPC executions
 - End users actually run the MPC and ***trust no one but themselves***
 - If honest majority protocols are used, then they must trust this

Automation Backend Component

- Automation backend – fully automated MPC execution deployment
- Capabilities
 - Automatic setup of parties in cloud (AWS, Azure, etc.)
 - Multiple execution coordination (bid for instances, setup parties, tear down)
 - Monitoring and results collection
- Admin defines parties, types, protocols executions, etc.
- Works for arbitrary protocols (have ≈ 10 incorporated)

MATRIX – The Automation Backend

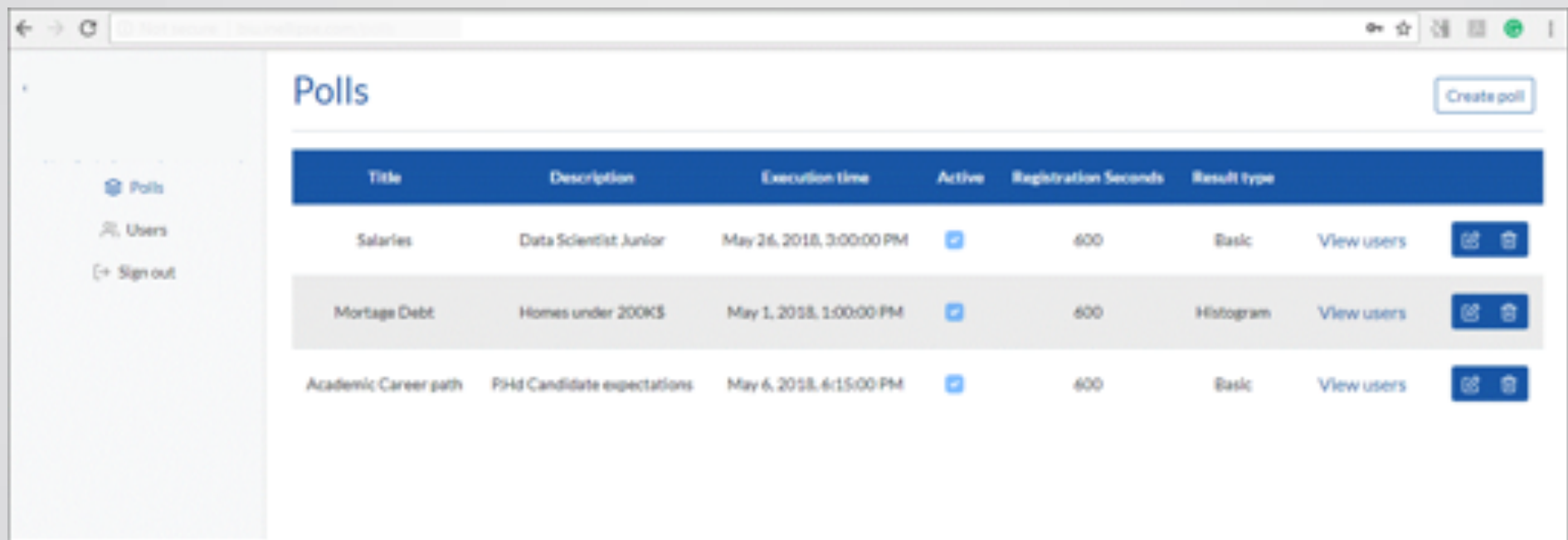


Administrator Component







- Provider (or anyone running open source) manages execution
- Capabilities
 - Publishes “invite” to participate
 - Track how many users (and potentially which users) have registered
 - Not aimed for anonymity of participants
 - Obtain results (as well as all participants)
 - Linked to backend to actually deploy
- We will demonstrate on “**PrivatePoll**”: a system for generic end-to-end private polls/surveys via MPC

Administrator Component for PrivatePoll

Main Admin Page



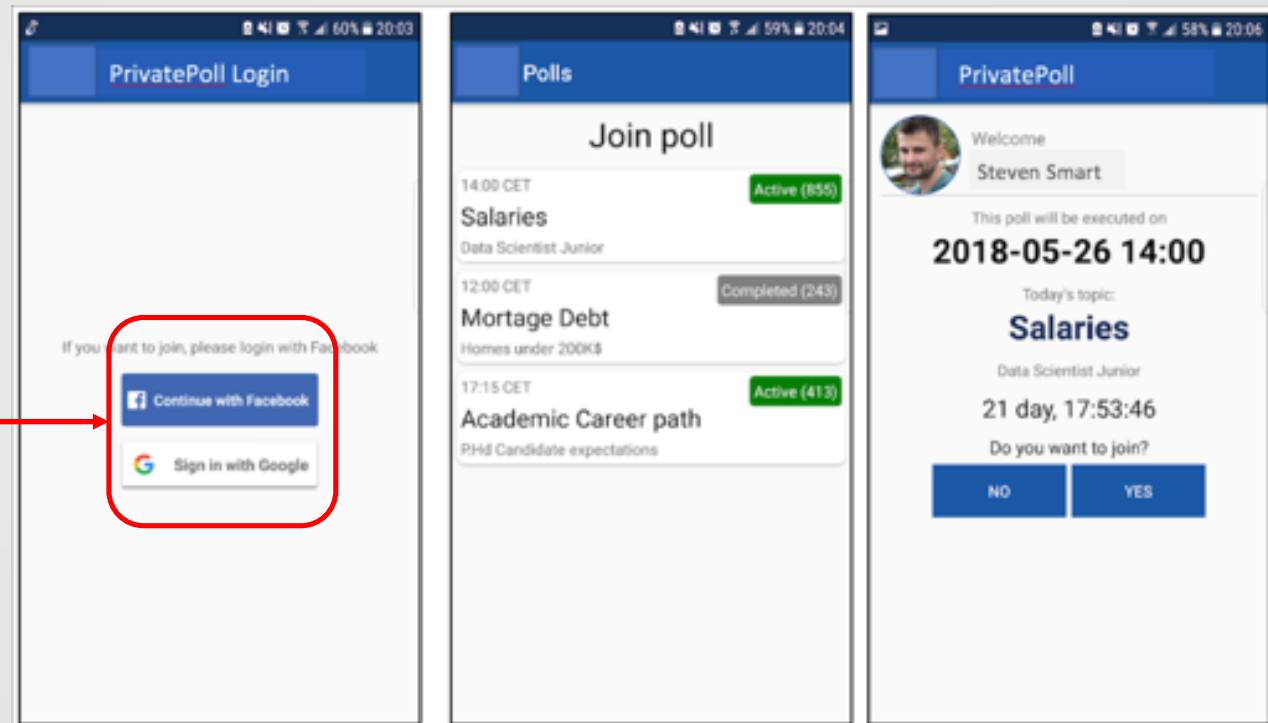
The screenshot displays the 'Polls' management page in a web browser. The browser's address bar shows 'https://polls.com/polls'. On the left, a navigation sidebar includes 'Polls', 'Users', and 'Sign out'. The main content area features a 'Create poll' button and a table of existing polls. The table has columns for Title, Description, Execution time, Active status, Registration Seconds, Result type, and actions (View users, Edit, Delete). Three polls are listed: 'Salaries' (Data Scientist Junior), 'Mortgage Debt' (Homes under 200K\$), and 'Academic Career path' (Phd Candidate expectations).

Title	Description	Execution time	Active	Registration Seconds	Result type	
Salaries	Data Scientist Junior	May 26, 2018, 3:00:00 PM	<input checked="" type="checkbox"/>	600	Basic	View users  
Mortgage Debt	Homes under 200K\$	May 1, 2018, 1:00:00 PM	<input checked="" type="checkbox"/>	600	Histogram	View users  
Academic Career path	Phd Candidate expectations	May 6, 2018, 6:15:00 PM	<input checked="" type="checkbox"/>	600	Basic	View users  

End User Component

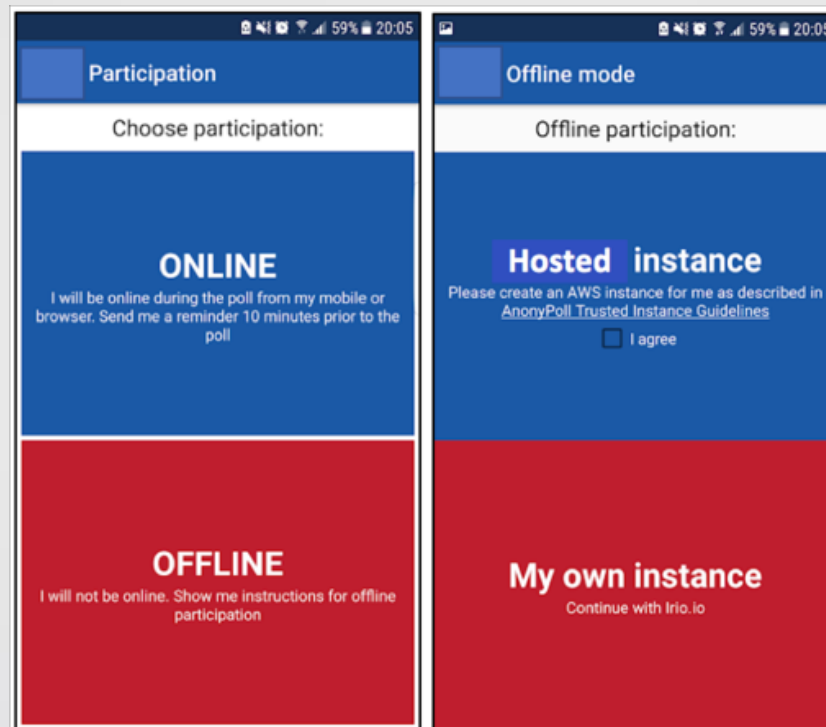
Login, poll join and poll status pages (in mobile app)

- Necessary if we want to assume an honest majority
- Even if not, unclear what ramifications on result is vast majority corrupted



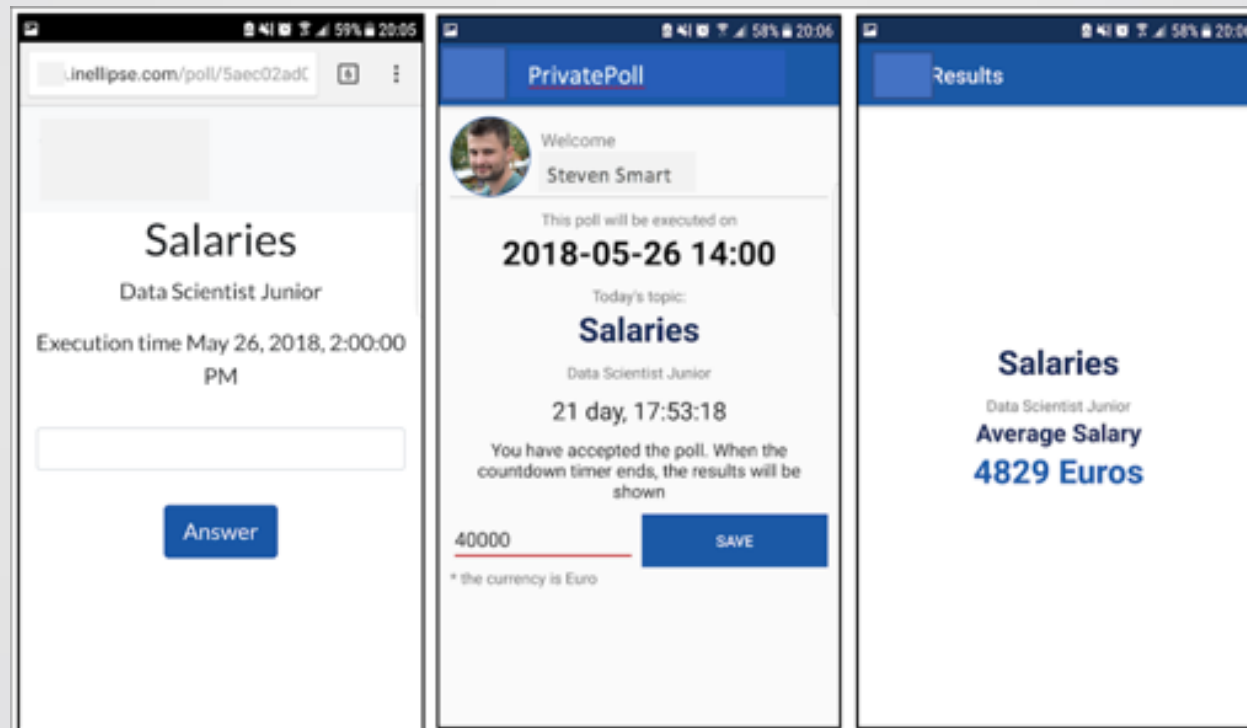
End User Component

User instance generation pages (online vs offline modes)



End User Component

Input/output pages



The Cryptographic Challenge

- The end-to-end system provides the capabilities for true decentralized MPC
- But, in such real scenarios, BANDWIDTH constraints are a huge concern
 - Relates to actual cost (with bandwidth limitations on cellular, etc.)
 - High bandwidth means much higher chance of failure
- We assume honest majority (or 2/3 majority)
 - Appropriate for true end-to-end MPC, **assuming authentication**

Low-Bandwidth MPC

- A warmup – consider three parties, at most one corrupted

Basic Additive Secret-Sharing

$$x = x_1 + x_2 + x_3$$

$$y = y_1 + y_2 + y_3$$



x_1
 y_1



x_2
 y_2



x_3
 y_3

- $z = x + y$: each computes $z_i = x_i + y_i$ (no interaction)
- $z = x \cdot y = (x_1 + x_2 + x_3) \cdot (y_1 + y_2 + y_3) =$

Basic Additive Secret-Sharing

$$x = x_1 + x_2 + x_3$$

$$y = y_1 + y_2 + y_3$$



x_1
 y_1



x_2
 y_2



x_3
 y_3

- $z = x + y$: each computes $z_i = x_i + y_i$ (no interaction)
- $z = x \cdot y = (x_1 + x_2 + x_3) \cdot (y_1 + y_2 + y_3) =$

$$x_1 \cdot y_1 + x_1 \cdot y_3 + x_3 \cdot y_1$$

+

$$x_2 \cdot y_1 + x_2 \cdot y_2 + x_1 \cdot y_2$$

+

$$x_2 \cdot y_3 + x_3 \cdot y_2 + x_3 \cdot y_3$$

Replicated Secret Sharing

$$x = x_1 + x_2 + x_3$$

$$y = y_1 + y_2 + y_3$$



(x_1, x_3)
 (y_1, y_3)



(x_2, x_1)
 (y_2, y_1)



(x_3, x_2)
 (y_3, y_2)

- $z = x + y$: each computes $z_i = x_i + y_i$, $z_{i-1} = x_{i-1} + y_{i-1}$ (no interaction)
- $z = x \cdot y = (x_1 + x_2 + x_3) \cdot (y_1 + y_2 + y_3) =$

$$x_1 \cdot y_1 + x_1 \cdot y_3 + x_3 \cdot y_1 \quad \mathbf{z_1}$$

+

$$x_2 \cdot y_1 + x_2 \cdot y_2 + x_1 \cdot y_2 \quad \mathbf{z_2}$$

+

$$x_2 \cdot y_3 + x_3 \cdot y_2 + x_3 \cdot y_3 \quad \mathbf{z_3}$$

Replicated Secret Sharing

$$x = x_1 + x_2 + x_3$$

$$y = y_1 + y_2 + y_3$$



(x_1, x_3)
 (y_1, y_3)

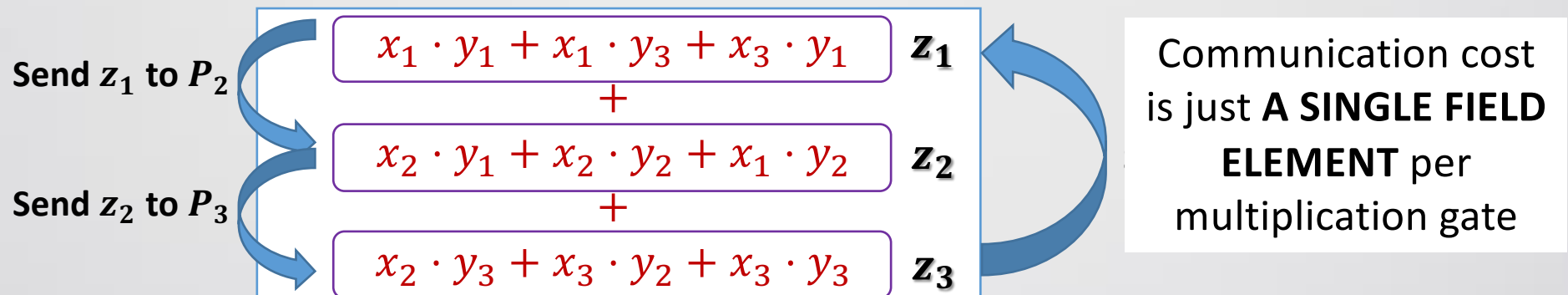


(x_2, x_1)
 (y_2, y_1)



(x_3, x_2)
 (y_3, y_2)

- $z = x + y$: each computes $z_i = x_i + y_i$, $z_{i-1} = x_{i-1} + y_{i-1}$ (no interaction)
- $z = x \cdot y = (x_1 + x_2 + x_3) \cdot (y_1 + y_2 + y_3) =$



Replicated Secret Sharing

$$x = x_1 + x_2 + x_3$$

$$y = y_1 + y_2 + y_3$$



(x_1, x_3)
 (y_1, y_3)



(x_2)
 (y_2)

- $z = x + y$
- $z = x \cdot y$

Send z_1 to P_1

Send z_2 to P_3

The z_1, z_2, z_3 values also need to be masked; this can be achieved utilizing correlated randomness which can be generated using pseudorandom functions, without interaction (after sending keys once)

(no interaction)

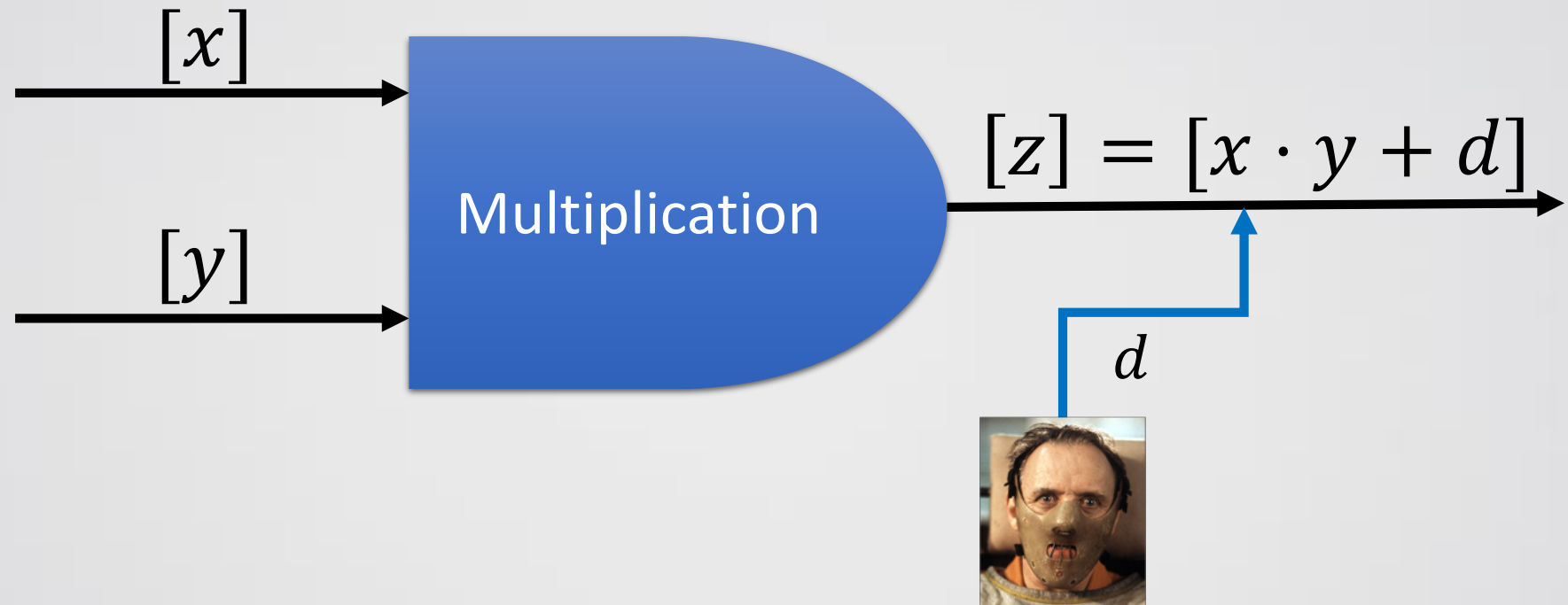
Communication cost is just **A SINGLE FIELD ELEMENT** per multiplication gate

$$x_2 \cdot y_3 + x_3 \cdot y_2 \quad z_3$$

Achieving Security for Malicious Adversaries

- Cheating party can send incorrect z_i value
- Can prove that this is all it can do
 - Formalize as **security up to additive attack** [GIPST14]
 - Multiplication is secure, but adversary can send d and result computed by trusted party is $x \cdot y + d$ (honest hold shares of x, y)
- Notation: sharing of x amongst parties by $[x]$

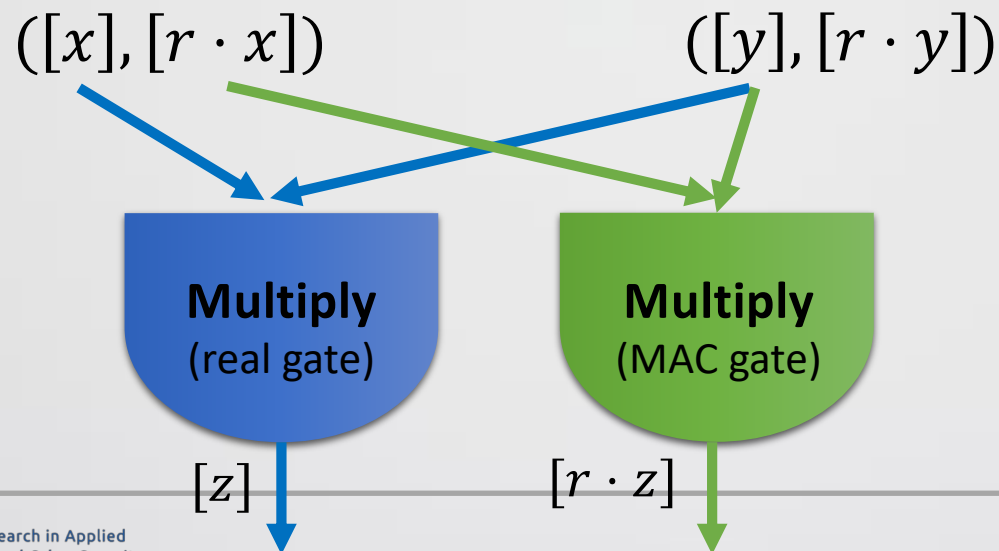
Achieving Malicious Security



How can the honest parties detect (and abort) when $d \neq 0$?

Cheating Detection – Randomized Computation

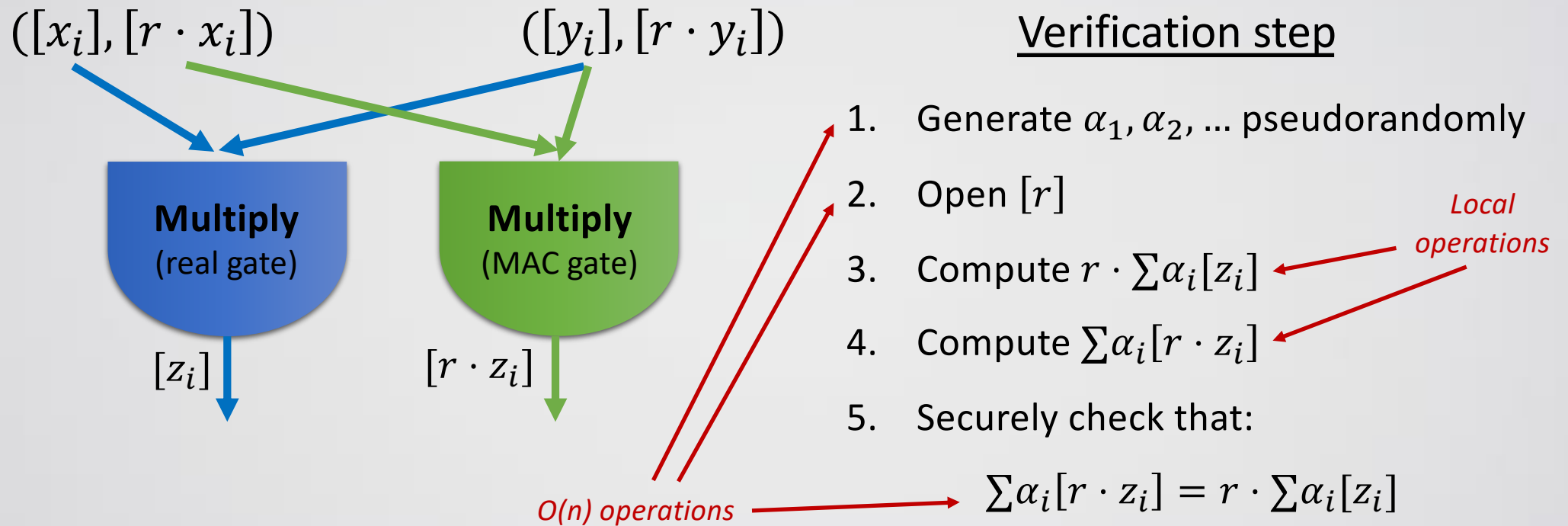
- Generate a random sharing $[r]$; serves as a type of MAC
- Invariant: for each wire of the circuit, compute the pair $([x], [r \cdot x])$:
 - Use multiplication to randomize the input wires of the circuit
- For each multiplication gate:



Cheating Detection – Verification

- Recall: in every multiplication, adversary can add some d
 - In first multiplication, can cheat with $x \cdot y + d_1$
 - In second multiplication, can cheat with $r \cdot x \cdot y + d_2$
- Observation: these “match” only if $d_2 = r \cdot d_1$
 - In that case, $r \cdot x \cdot y + d_2 = r \cdot x \cdot y + r \cdot d_1 = r \cdot (x \cdot y + d_1)$
 - It’s hard for adversary to make it match, since doesn’t know r (**up to $1/|\mathbb{F}|$**)
- Aim: detect if there are wires that do not “match”

Cheating Detection Procedure



Multiparty Computation (> 3)

- The same method works for multiparty computation as well
- Semi-honest multiplication protocols with Shamir sharings are secure up to additive attacks
 - Damgård-Nielsen 2007 protocol has very low complexity
 - Exactly 6 field elements per party per multiplication
- Resulting complexity for malicious = twice semi-honest (for large fields)
 - 2 field elements per multiplication for 3 party
 - 12 field elements per multiplication for multiparty

Malicious Security at the Cost of Semi-Honest

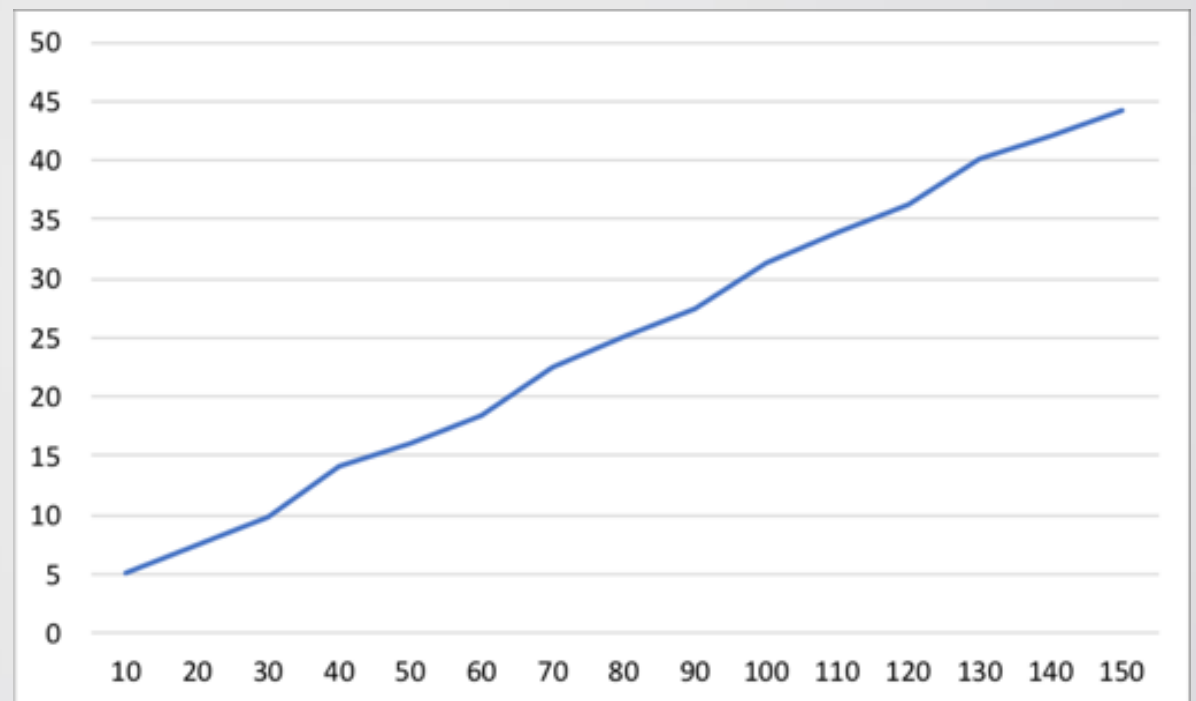
- We assume less than $1/3$ parties corrupted (out of n)
- Consider a single execution using the semi-honest protocol
 - Assume additive attack security (but actually need less)
 - The **best known semi-honest protocols** have this property
- For every multiplication gate with input x, y and output z , it should hold that $z = x \cdot y$; **we need to verify this equality**

Complexity

- A single semi-honest multiplication per multiplication gate plus verification
 - The communication of the verification is $O(n)$, **independent of circuit size**
 - Local computation is over entire circuit, but insignificant in practice
 - For small fields, repeat verification until small enough
 - Very useful for $GF[2^8]$ which enables computation of Boolean circuits
- **Overall:** with known optimizations to Damgård-Nielsen, only $\frac{8}{3} < 3$ elements per multiplication gate + some small overhead

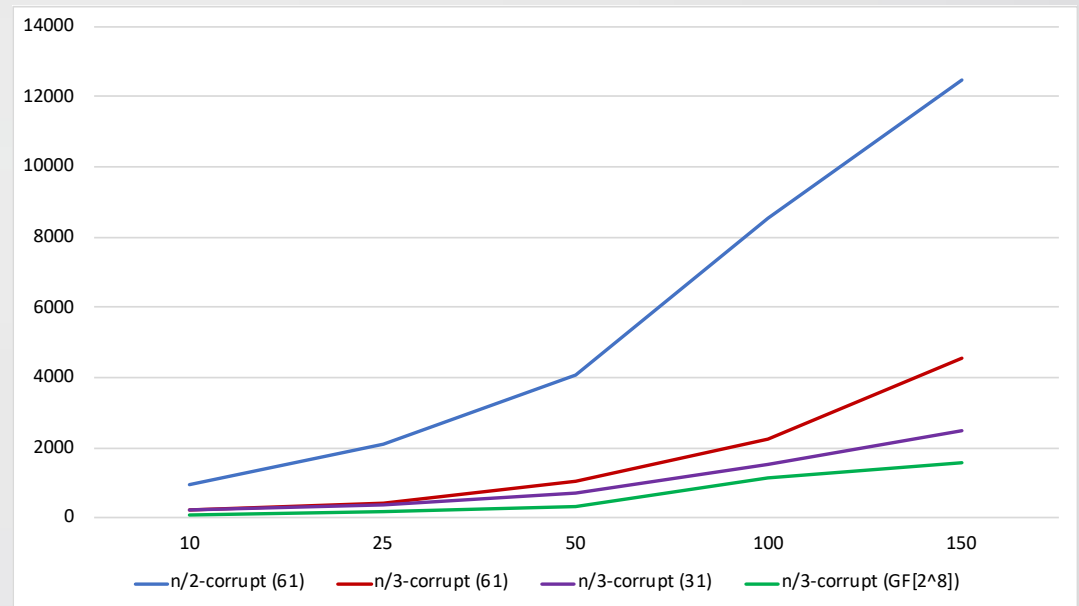
Experiments – a Real Statistics Computation for Honest Majority Protocol

- Statistics computation (mean, variance and linear regression)
- Circuit parameters
 - 4,000,000 inputs
 - 6,000,000 mult gates
 - Depth = 1
 - 31-bit field
- Execution environment
 - AWS single region
 - m5.12xlarge instances
- Results
 - **5 seconds for 10 parties**
 - **45 seconds for 150 parties**



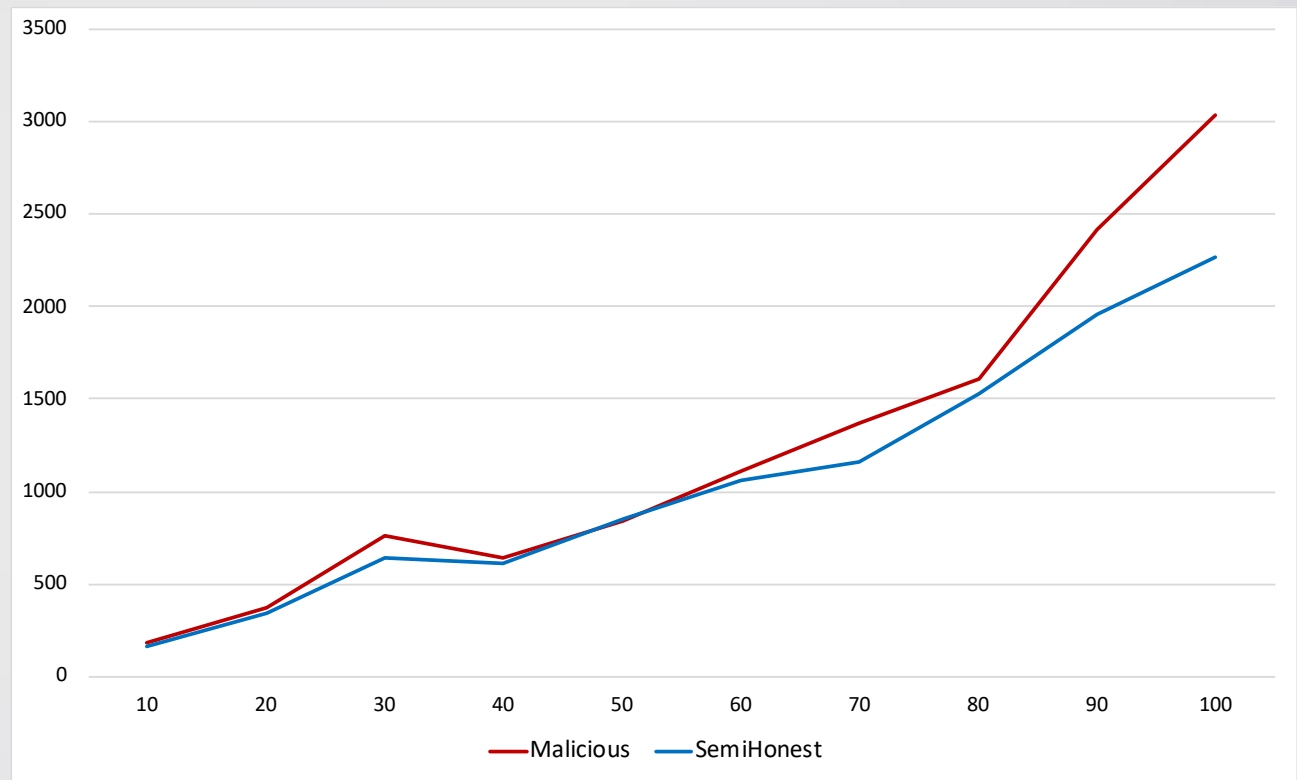
Experiments – Protocol Comparison

- Circuit
 - 1,000,000 multiplication gates
 - Depth 20
 - 61, 31, 8 bit fields
- Execution environment
 - AWS single region
 - c5.xlarge instances
- Results (for n/3-corrupt)
 - **$GF[2^8]$ = 1.5 seconds for 150 parties**
 - **31-bit = 2.5 seconds for 150 parties**
 - **61-bit = 4.5 seconds for 150 parties**



Protocol for 1/3 Corrupt Setting

- Circuit of 1,000,000 multiplication gates and depth 20 over 61-bit field
- Malicious and semi-honest almost same cost (difference is basically noise)



Experiments – Mobile Executions

Parties Configuration	Network Latency	Running Time
10 ARM a1.large	90ms	9.9
50 ARM a1.large	90ms	46.4
50 ARM a1.large and 50 servers c5.xlarge	90ms	95.9
10 ARM a1.large	300ms	22.1
50 ARM a1.large	300ms	101.7
50 ARM a1.large and 50 servers c5.xlarge	300ms	303.2

Table 3. Running times in *seconds* for a circuit of 1,000,000 multiplication gates and depth-20 with a 31-bit Mersenne prime.

Additional Challenges

- **Cryptographic challenges**

- Deal with honest failures without penalty of fully robust protocol with guaranteed output delivery
- Achieve guaranteed output delivery at low cost in cases of no attack
- Achieve low-cost dishonest majority protocols
 - Seems very difficult but would enable better trust model
- Incorporate differential privacy

- **Systems and other challenges**

- Scale up to thousands of parties
- Enable better performance from browsers
- Collaborate with social scientists (or others) to see what they need

Thank You

