Systematizing Secure Computation for Research and Decision Support

Jason Perry, Debayan Gupta, Joan Feigenbaum
and Rebecca N. Wright

Rutgers University, Yale University

SCN 2014, Amalfi

Slides available at
`http://paul.rutgers.edu/~jasperry/scn-slides-jp.pdf`

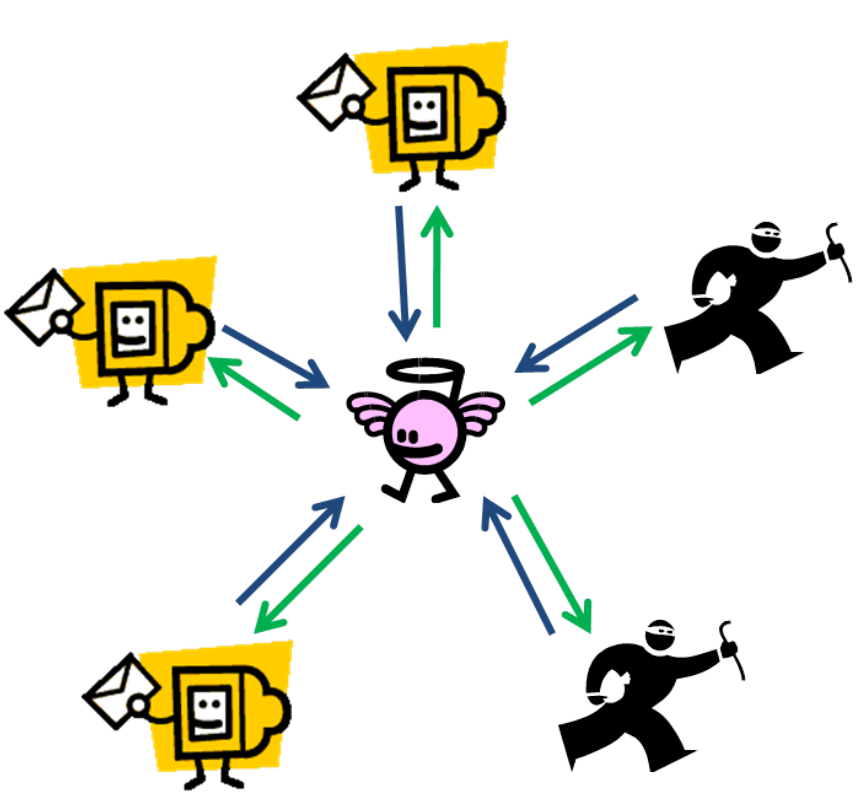
Secure Multi-party Computation = MPC

There are n parties who wish to jointly compute a functionality based on their individual inputs $(y_1, \ldots y_n) = f(x_1, \ldots x_n)$, while preserving

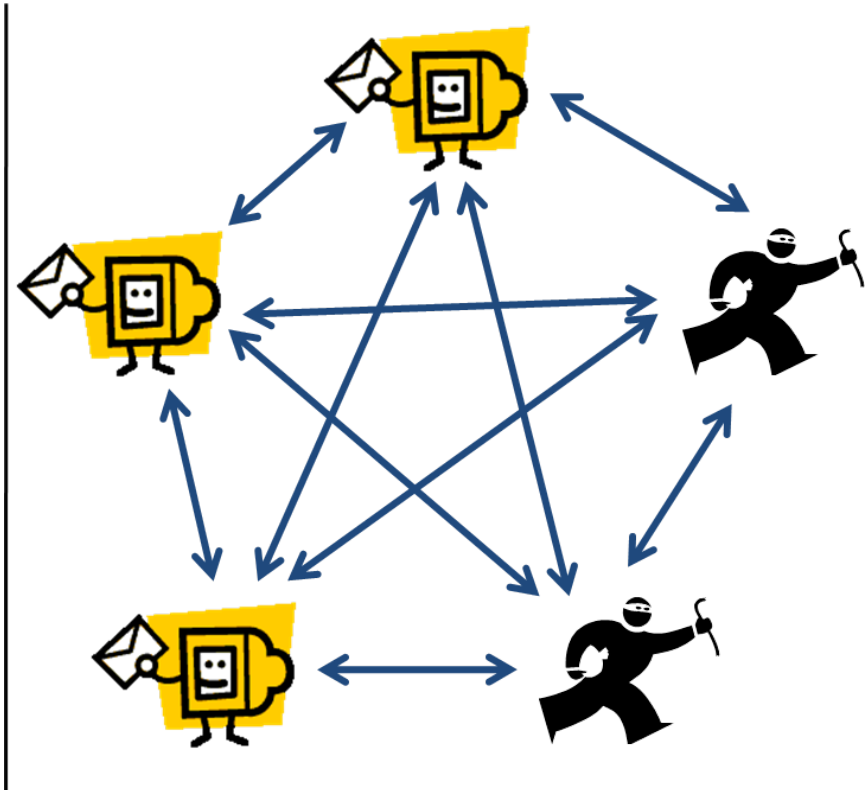
- **Privacy**: Not revealing anything about their own inputs
- **Correctness**: An adversary cannot prevent honest parties from obtaining the answer

Canonical example—"Millionaires' problem": find out which of us is the richest without revealing how much money I actually have

MPC Simulates a Trusted Third Party



Ideal world



Real world

State of MPC Research

- 2-party garbled circuits paradigm suggested by Yao [Y82, Y86], first general protocol for any n parties by Goldreich et al. [GMW87]
- Hundreds of research papers since, many giving new general protocols with varying sets of assumptions, more rigorous formulations of security, and efficiency improvements
- Since Fairplay [MNPS04], a growing number of implementations
- Several practical applications proposed:
  - Satellite collision avoidance
  - Auctions
  - Personal appointment scheduling

...but still only a handful of documented real-world deployment experiments

Obstacles to MPC Research and Adoption

Why the low adoption rate?

- Field is genuinely complicated: MPC protocols are complex objects with many axes of variation
- Difficult to compare protocols or evaluate their suitability to any given problem
- Understanding and organizing a large number of results might be a thankless job...

Our Contribution

A *Systematization of Secure Computation* can improve this situation by:

- Helping security consultants and implementers understand the relative merits of protocols, so they can recommend and deploy solutions.
- Helping new researchers come up to speed on the area more quickly
- Helping researchers explore the problem space and discover new openings for improved protocols

Roadmap of the Work

1. Survey many research papers in the area and create an annotated bibliography
2. Develop a system for classifying MPC protocols by their distinguishing features (security, efficiency etc.) *and* modeling their interdependencies
3. Classify published protocols using our system
4. Implement a GUI for interacting with the systematization database

The Secure Computation Annotated Bibliography

Currently over 190 papers and growing, annotated with description of result and cross-references

- Includes some key background papers on oblivious transfer, secret sharing, commitment
- Entries in source are tagged, allowing creation of sub-bibliographies for smaller problem areas

Available online at
`http://paul.rutgers.edu/~jasperry/ssc-annbib.pdf`.

Axes of Systematization

Goal: a means of classifying protocols that captures all significant distinctions (at least asymptotically) and makes it easy to compare & contrast protocols

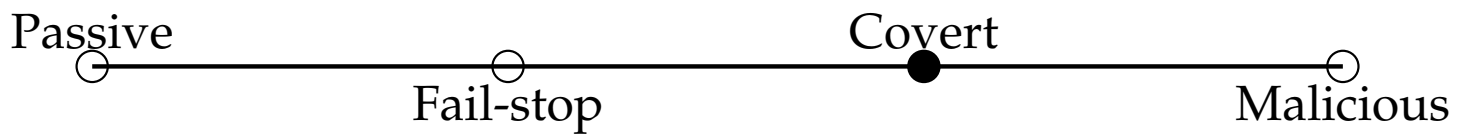
- especially in terms of *tradeoffs*: strength of assumptions vs. security/efficiency, security vs. efficiency

Axes of Systematization

We factored the features of MPC protocols into a set of 22 linear axes, ordered from weaker to stronger result.

- Each axis populated with a discrete set of known values; new results may define new intermediate values, though some are inherently binary
- Axes fall into four categories, highlighting the tradeoffs at a high level

Adversary Maliciousness



Axis Categories

Environmental Assumptions

Private Channels
Broadcast Channel
Trusted Setup
Synchronous Network

Cryptographic Assumptions

Computational Assumption Level
Assumption Specificity

Security Features

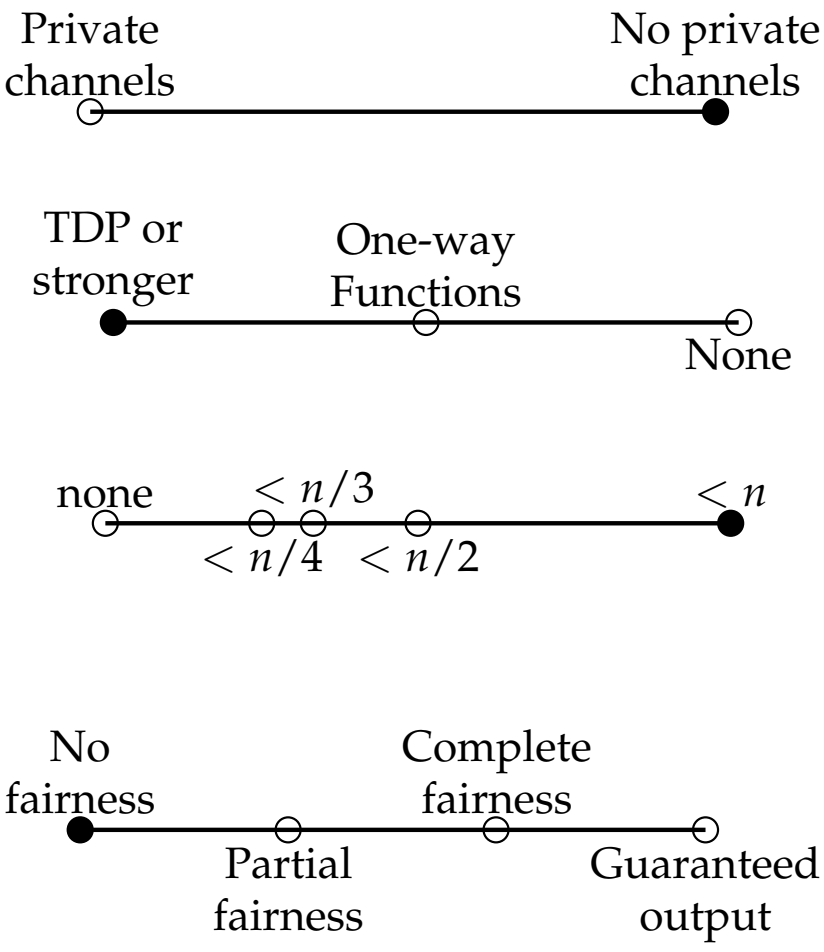
Security type
Adversary Maliciousness
Adversary Mobility
Threshold of Corrupted Parties
Add'l passively corrupted parties
Add'l corrupted with weaker security
Fairness
Composability
Leakage Security
Auditability

Efficiency Achieved

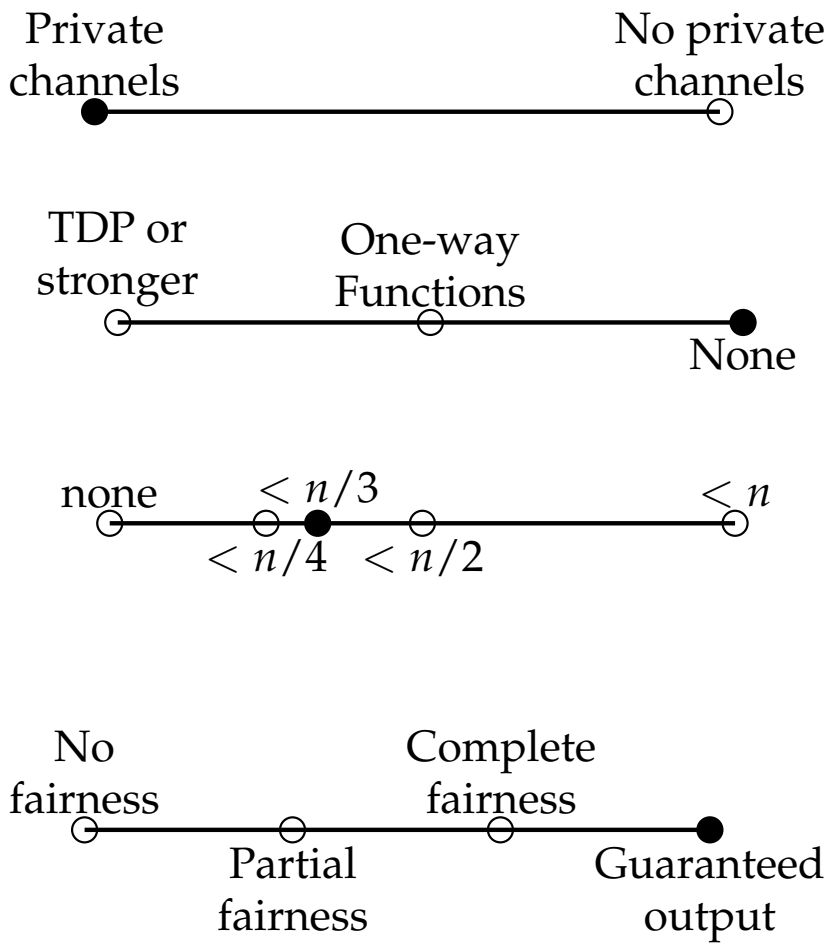
Online computation complexity
Online round complexity
Online per-gate comm complexity
Preprocessing comm complexity
Preprocessing dependency
Preprocessing reuse

Sample Protocol Comparison Using Axes – 1

[GMW87]-mal



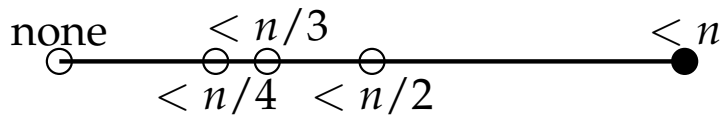
[BGW88]-mal



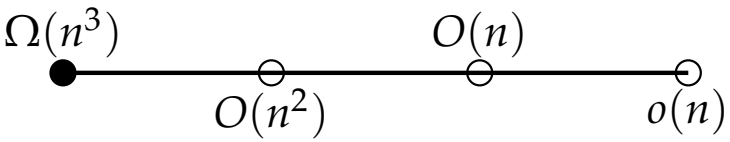
Sample Protocol Comparison Using Axes – 2

[GMW87]-mal

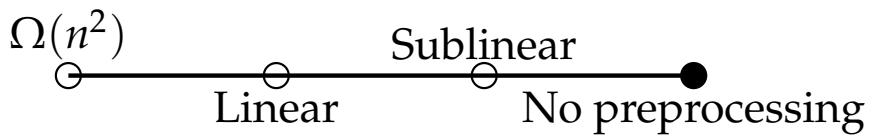
Threshold of corrupted parties



Online communication complexity per gate

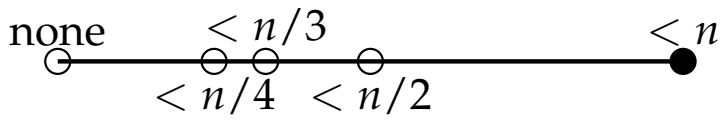


Preprocessing communication complexity per gate

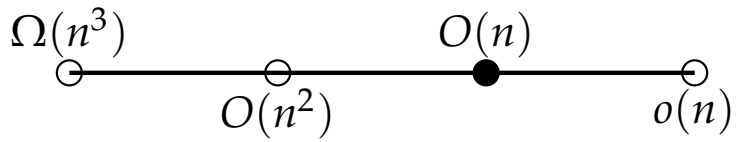


[DPSZ12]

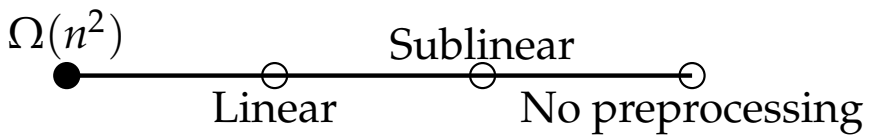
Threshold of corrupted parties



Online communication complexity per gate



Preprocessing communication complexity per gate



MPC Protocol Database

- Currently over 30 protocols scored on axes
- Freely available; currently distributed as part of GUI tool

Dependencies

Impossibility & lower-bound theorems of the MPC literature can be stated as a set of dependencies between axis values

Example:

Theorem [BGW88]

For unconditional security against t maliciously corrupted players, $n/3 \leq t < n/2$, a broadcast channel is required.

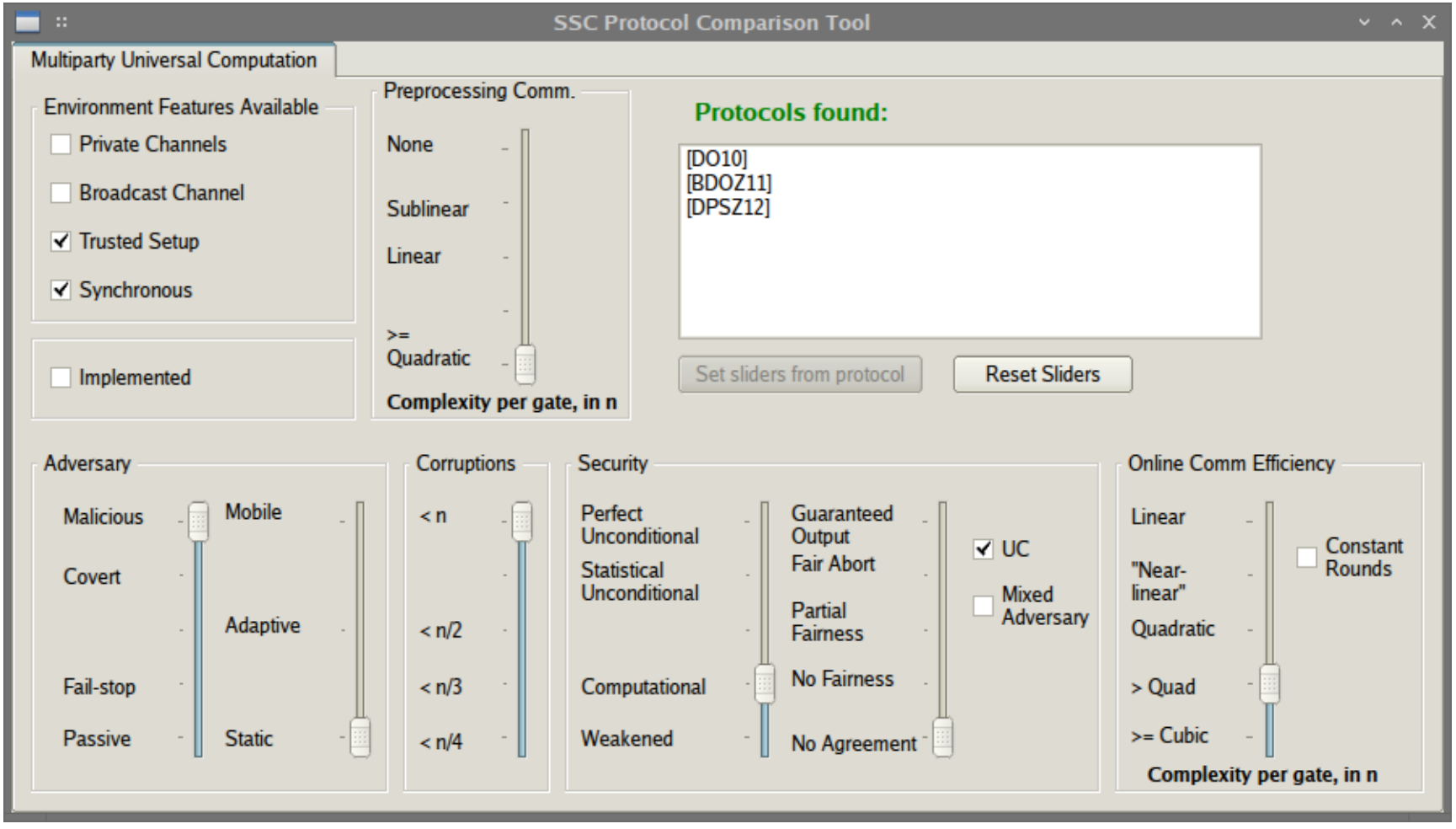
$=$ `If the Security type axis value is to the right of "Computational" and the Maliciousness axis is at "Malicious" and the Corrupted parties axis is to the right of "n/3", then the Broadcast axis must be at "Broadcast channel"`

Putting the Systematization to Work

Developed a graphical tool, *SysSC-UI*, for exploring the MPC protocol database

- Reads axis values of protocols directly from database
- Has encoding of the dependencies in its internal logic
- User sets sliders and checkboxes to the desired parameters, and sees references to all papers with protocols *at least* as good.

SysSC-UI


SSC Protocol Comparison Tool
Multiparty Universal Computation
Environment Features Available
Private Channels
Broadcast Channel
Trusted Setup
Synchronous
Implemented
Preprocessing Comm.
None
Sublinear
Linear
>= Quadratic
Complexity per gate, in n
Protocols found:
[DO10]
[BDOZ11]
[DPSZ12]
Set sliders from protocol
Reset Sliders
Adversary
Malicious
Covert
Fail-stop
Passive
Mobile
Adaptive
Static
Corruptions
< n
< n/2
< n/3
< n/4
Security
Perfect Unconditional
Statistical Unconditional
Computational
Weakened
Guaranteed Output
Fair Abort
Partial Fairness
No Fairness
No Agreement
UC
Mixed Adversary
Online Comm Efficiency
Linear
"Near-linear"
Quadratic
> Quad
>= Cubic
Constant Rounds
Complexity per gate, in n

SysSC-UI

Nice things:

- Immediately see the history of papers for a given sub-problem
- Reveals protocols most suited to given requirements, and potential gaps for research.

Open source; python code and database available at
`https://code.google.com/p/syssc-ui/`

Web version also in progress:
`http://work.debayangupta.com/ssc/`

Future Work

- Moving toward a community-based model
  - To keep our database up-to-date, we have developed an online survey in which researchers can enter their protocols and their properties:
    `http://goo.gl/T4ORzr`
  - Feedback welcome
- Many potential ways to visualize/interact with the protocol database
- Applying this systematization approach to other messy bodies of theoretical knowledge

Thank you

Questions?