Better Privacy and Security via Secure Computation

Jonathan Katz
Security/privacy would be much easier...
...if there were someone we could all TRUST with our data
Better data mining -- using MORE data, while respecting users’ PRIVACY
Office of Financial Research
Working Paper #0011
September 4, 2013

Cryptography and the Economics of Supervisory Information: Balancing Transparency and Confidentiality

Mark Flood,\(^1\) Jonathan Katz,\(^2\) Stephen Ong,\(^3\)
and Adam Smith\(^4\)

\(^1\)Office of Financial Research, mark.flood@treasury.gov
\(^2\)University of Maryland, jkatz@cs.umd.edu
\(^3\)Federal Reserve Bank of Cleveland, stephen.j.ong@clev.frb.org
\(^4\)Pennsylvania State University and Office of Financial Research, asmith@cse.psu.edu
CONTROLLED information sharing
Achieving Higher-Fidelity Conjunction Analyses Using Cryptography to Improve Information Sharing

Brett Hemenway, William Welser IV, Dave Balocchi

RAND Project AIR FORCE

Prepared for the United States Air Force
Approved for public release; distribution unlimited
Better privacy/security for EVERYONE
Would be nice if there were someone we could all TRUST with our data...
• Legal/regulatory restrictions
• Not economically viable (cost + liability vs. value)
• Central point of failure/attack
• Incompatible trust frameworks
Would be even better if we could AVOID the need for trust with a first place.
Secure computation ensures:

- **Confidentiality**
  - No party’s input is revealed

- **Integrity**
  - Correct output is computed

- **Availability**
  - All parties obtain the output

- **Input independence**
  - Each party’s input is *independent* of the others’
Assumptions/caveats

- **Number** of malicious parties (sometimes)
- **Actions** of malicious parties (sometimes)
- **Cryptographic** hardness (sometimes)
- **Weaker** guarantees (sometimes)
Secure computation of any function, with security against arbitrary behavior of any number of parties, is possible
Two-party setting

• Start with a boolean circuit for $f$
• $P_1$ sends a “garbled circuit” for $f$ to $P_2$ along with keys for its own input
• $P_2$ obtains the keys for its input using oblivious transfer
• $P_2$ evaluates the garbled circuit

This gives semi-honest security only!
General feeling (~2000):
Hopelessly impractical
Efficiency (semi-honest)

AES

- Fairplay
- PSSW09
- TaSTY
- HEKM11
- LR15

Time (log scale):

0.5 ms
Efficiency (malicious)

AES, 40-bit statistical security

- PSSW09
- SS11
- AMPR14
- LR15
- WMK16

Time (log scale)

65 ms
Real-world interest

• **Partisia (3-party)**
  – Danish sugar-beet auction (2008-present(?))
  – Wireless-spectrum auctions

• **Sharemind (3-party)**
  – Statistical analysis of financial data

• **Sepior, Dyadic (2-party)**
  – AES

• **IARPA SPAR, DARPA PROCEED/Brandeis**
Research questions

• “Cryptographic”
  – Multi-party setting
    • Protocols, “real-world” issues
  – Post-quantum security
  – Alternate models of computation
  – Composability
  – What functions are “safe” to compute?
Research questions

• “Non-cryptographic”
  – Usability
  – PL/compiler support
  – Formal verification of protocols, implementations
Real-world questions

• Will secure computation be of **niche** interest, or will it be more **widespread**?
• What is the **business model**?
• What **security requirements** suffice?
• What are the right **cost metrics**?
• What is the **barrier** to more widespread use of secure computation?
Real-world questions

• Will there be multiple applications of secure computation, or just a few?
  – Should we focus on generic systems, or optimize for specific “killer applications”?
  – What are the “killer applications”?

• Who will be writing code?
  – Where should we focus our attention when writing compilers?
Conclusions

• **Tremendous advances** in past few years

• Greater **deployment** in the near future(?)
Acknowledgments

Research supported by

– NSF ("TC: Large: Collaborative Research: Practical Secure Computation: Techniques, Tools, and Applications")

– US ARL/UK MoD ("Secure Information Flows in Hybrid Coalition Networks")

– DARPA ("Toward Practical Cryptographic Protocols for Secure Information Sharing")
Thank you!

Papers and code available from
http://www.cs.umd.edu/~jkatz/papers.html