Salus

Seny Kamara - Microsoft Research
Payman Mohassel – U. of Calgary
Ben Riva – Tel Aviv U.
Cooperation without Trust

Alice

Eve

Bob

\( f(x,y,z) \)
Cooperation without Trust

- Examples
  - Data mining
  - Negotiations
  - Electronic Voting
  - Auctions
  - Exchanges
  - Distributed constraint satisfaction & optimization
  - Location privacy
  - Bioinformatics
  - Electronic commerce
  - Healthcare
  - ...

Cooperation without Trust

- **Q**: how do we achieve this?

Diagram:
- Trusted Party
- NDAs
- Arrows indicating interaction between parties
Secure Function Evaluation

\[ f(x, y, z) \approx x y z \]
SFE is Great!

- Really powerful
  - Solves large number of problems that occur in practice
  - Can be combined with other techniques to solve even more problems

- We can do it for *any* function!
  - negotiations, data mining, search, ...

- We have many protocols with different properties
  - 30 years’ worth of MPC research

- **Q:** So why aren’t we using this on a daily basis?
SFE is Too Slow!

- Early work on SFE was theoretical
- Researchers recognized its importance
  - But didn’t know how to make it practical yet
- It was dismissed as pie-in-the-sky
  - Similar to how FHE is perceived today
Why is SFE so Expensive?

• Bottlenecks (in 2SFE):
  o Malicious behavior: ZK proofs to make sure Garbler does not cheat
    • Cut & Choose [Malka-Nisan-Pinkas-Sella04, Mohassel-Franklin06, Kiraz-Schoehnmakers06, Lindell-Pinkas07, Woodruff07]
  o Circuit size: $O(\text{size of circuit})$ work to garble and evaluate circuit
    • Free XOR [Kolesnikov-SchneiderS08]
  o Oblivious transfer: $O(|y|)$ number of 1-out-of-2 oblivious transfers
    • OT Extension [Ishai-Kilian-Nissim-Petrank03]
  o Memory: need to load and process $O(\text{size of circuit})$ gates
    • Pipelined Execution [Huang-Evans-Katz-Malka11, Malka11]
SFE Frameworks

- Fairplay
  - Implementations of 2PC & MPC
- FairplayPF
  - Implementation of private function evaluation using UCs
- VIFF
  - Sharing-based MPC & real-life use-case
- Sharemind
  - Sharing-based MPC for data analytics
- TASTY
  - Mixed MPC framework (sharing + garbled circuits)
- Fast Garbled Circuits
  - Highly-optimized garbled circuit framework
- VMCrypt
  - Highly-optimized garbled circuit framework with pipelined execution
Inherent Limitations of SFE

• Linear work
  - All protocols require $O(|C|)$ work from each party
  - Circuits can be very large
    - AES $\approx 30,000$ gates
    - Edit distance (50 char strings) $\approx 250,000$ gates
    - Dot product (255 dims over 64-bit field) $\approx 30$ million gates

• Fairness
  - Either all parties get output or none do
  - Fairness is impossible in general [Cleve86]

• Symmetric work
  - All parties do same amount of work
  - MPC-based systems will not scale if parties are heterogeneous
Server-Aided SFE
Server-Aided SFE

SFE

Server-aided SFE
Server-Aided SFE

- [Asharov-Jain-Lopez-Alt-Tromer-Vaikuntanathan-Wichs12]
  - Protocol based on FHE
  - From $O(\text{size of circuit + size of input}) \Rightarrow O(\text{size of input})$
  - Mostly of theoretical interest

- [K.-Mohassel-Raykova12]
  - Protocol based on garbled circuits
  - $O(\text{size of circuit + size of input}) \Rightarrow O(\text{size of input})$
  - Of practical interest but...
  - Limitations!
    - Assumes parties do not collude with server
      - Removing this implies general-purpose sub-linear 2PC
    - One party does $O(\text{size of circuit})$ work
      - Reducing this implies non-interactive secure delegation
Is Server-Aided SFE Practical?
Salus

• Server-aided SFE framework
  o Fairplay circuit format
  o New (fair) protocols
    • vs. malicious servers
    • vs. covert servers
  o Pipelined execution (new approach for malicious setting)
  o Free XOR
  o Batched Peikert-Vaikuntanathan-Waters OT
Garbled Circuits [Yao82]

1. $GC(C) \Rightarrow (\tilde{C}, sk, dk)$
2. $GI(sk, x) \Rightarrow \tilde{x}$

1. $Eval(\tilde{C}, \tilde{x}) \Rightarrow z$
2. $Decode(dk, z) \Rightarrow C(x)$
Garbled Circuits [Yao82]

- What happens if evaluator cheats?
  - Garbled circuits have a verifiability property
Garbled Circuits [Yao82]

- What if Garbler cheats?
  - Zero-knowledge proofs [GMW87]
  - Cut-and-choose [MNSP04,MF06,LP07,W07,...]
    - Send many garbled circuits
    - Evaluator asks Garbler to open some and verifies them
    - Evaluates the rest and outputs majority
Cut-and-Choose [MNPS04,MF06,LP07]

Garbler

\((\tilde{C}, ..., \tilde{C})\)  
Open 1/2

\((sk, ..., sk)\)  

\((\tilde{x}, ..., \tilde{x}) \& EQ(\tilde{x}, ..., \tilde{x})\)

Evaluator

1. Verify all \(\tilde{x}\) are equal
2. Evaluate remaining \(\tilde{C}\)
3. Output majority bits

\(C(x)\)
Server-Aided C-&-C [K.-Mohassell-Raykova12]

1. Eval(\(\tilde{C}, \tilde{x}, \tilde{y}\)), ... , Eval(\(\tilde{C}, \tilde{x}, \tilde{y}\))
2. How does the Server take majority?
   1. **Oblivious-MAJ**(\(\tilde{z}, ..., \tilde{z}\))
Protocol 1

- **Input equality checking**
  - [Mohassel-Franklin06, Lindell-Pinkas07]: $O(s^2 \cdot n)$ based on hash functions
  - [Woodruff07]: $O(s \cdot n)$ but based on expander graphs
  - [Lindell-Pinkas11-shelat-Shen11]: $O(s \cdot n)$ based on ZK and WI proofs (exps)
  - **Our work**: $O(s \cdot n)$ based only on hash functions

- **Oblivious majority**
  - [K.-Mohassel-Raykova12]: based on polynomial evaluation & interpolation
  - **Our work**: based only on symmetric encryption

- **Pipelined execution**
  - [HEKM11,Malka11]: does not work vs malicious adversaries
  - **Our work**: new pipelined exec for cut-and-choose [Kreuter-shelat-Shen12]
Protocol 2

- Server garbles circuits & P1 verifies and evaluates
- Problem #1: fairness
  - Hash-based mechanism
- Problem #2: garbled input delivery
  - Distributed OT
  - XOR secret sharing & hash functions
Experiments
Functionalities

• AES
  o with $|K| = 128$ and $|m| = 128$
  o 31 512 gates
  o 13 904 non-XOR gates

• Edit Distance
  o $|x| = |y| = 50$ and 8-bit characters
  o 254 930 gates
  o 94 472 non-XOR gates
## Protocol 1

<table>
<thead>
<tr>
<th></th>
<th>2P-AES</th>
<th>4P-AES</th>
<th>Edit Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>[PKSS09]</td>
<td>1114s</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>[selat-Shen11]</td>
<td>192s w/o comm.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Protocol 1</td>
<td>45s (4x-24x)</td>
<td>46s</td>
<td>240s</td>
</tr>
</tbody>
</table>

- **Note**: time is independent of number of parties!
### Protocol 2 (Covert)

<table>
<thead>
<tr>
<th></th>
<th>2P-AES</th>
<th>4P-AES</th>
<th>Edit Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>[PKSS09]</td>
<td>60s</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Protocol 2</td>
<td>9.12s (6x)</td>
<td>14.8s</td>
<td>33.5s</td>
</tr>
</tbody>
</table>
Thanks