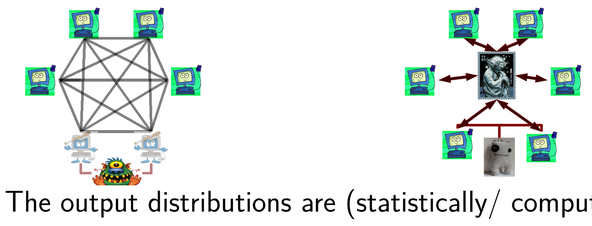
Secure Multi-Party Computation With Minimal Interaction

Anat Paskin-Cherniavsky, Joint work with Yuval Ishai and Eyal Kusilevitz
Technion, Israel Institute of Technology

Secure Multi Party Computation (MPC)

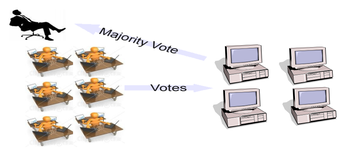
- A set of parties with private inputs wish to compute some joint function of their inputs.
- The protocol proceeds in rounds.
- Parties wish to preserve some security properties. E.g., privacy and correctness:
  - Semi-honest setting: parties follow the protocol, but try to learn extra information.
  - Malicious setting (default): corrupted parties may behave arbitrarily.
- Capturing it all: simulation-based security.
  The real model. The ideal model.



- The output distributions are (statistically/ computationally) indistinguishable.

A motivating example

- A manager wants to learn the (majority) vote among employees on issue X.
- A minimally interactive solution: use a trusted external server.



The general client-server setting:
- Two or more clients hold inputs to a function f.
- They wish to learn the output of f with the help of $n \geq 1$ servers.
- No communication among clients or among servers.

Towards practical MPC

- Client-server setting: 2-round MPC is particularly useful.
  - Quantitatively: Round complexity is a major bottle-neck as to concrete efficiency.
  - Qualitatively: The clients can send a message, and get an answer in their spare time.
- How about 2-round MPC in the standard setting ?
- Implementing broadcast is costly (think of MPC over the Internet).
- The focus of this work: Feasibility of 2-round MPC, over point to point channels.

MPC round complexity - state of the art

threshold	rounds	setting	broadcast	source
$t < n/3$	circ. depth	malicious	no	[bgw88]
$t < n/2$	cric. depth	malicious	yes	[rb89]
$\mathbf{t = 1, n \geq 5}$	**2**	**malicious**	**no**	**this work**
$t = 1, n \geq 4$	2	semi-honest	no	[bgw88,ik00]
t=2	2	fairness	yes	[gikr02]
$\mathbf{t < n/3}$	**2**	**sel. abort**	**no**	**this work**
$t = \Omega(n)$	3	malicious	yes	[gikr01]
$t < n/3$	2	semi-honest	no	[bgw88,ik00]
$\mathbf{(1 \vert t < n/3)}$	**2**	**client-server**	**no**	**this work**
$(1, 1)$	2	client-server	yes	[gikr02]

To conclude:
- 3 rounds are sufficient for general MPC with $t = \Omega(n)$ (and broadcast).
- 2 rounds are not sufficient even for $t = 2$, and "fairness" [ik00].
- Our results: In some settings, 2-round MPC is possible.

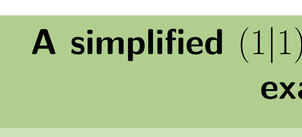
Some details on our results

- In all cases, we achieve computational security for general $f \in Poly$.
  - Security of 2 of the three relies on existence of arbitrary PRG (a relatively week assumption).
- Security is statistical for $f \in NC_1$.

Toolbox 1

PSM

- A minimalistic setting for secure computation.



- Correctness: Referee outputs $f(x)$.
- Privacy: Referee's view depends only on $f(x)$.

Set systems

A set system over $[n]$ satisfying:
- Large pairwise intersection: at least $t + 1$.
- High resilience: each t-tuple in $[n]$ intersects a strict minority of the sets.

Secret sharing

- (t, n) threshold secret sharing schemes.
- Every t parties learn nothing a bout the secret s.
- Every $t + 1$ parties recover the secret.
- Examples: (t, n)-Shamir. (t, n)-bivariate Shamir.

A simplified $(1|1)$-secure client-server example.

- Two clients wish to compute f with the help of $n = 20$ servers. Servers hold common randomness (can be waived).
- Round 1: Client i secret-shares x_i among all servers.
- Round 2: Server set T executes PSM with Client i as referee computing:

$$f(s_T^1, \ldots, s_T^n) = \begin{cases} f(s^1, \ldots, s^n) & \text{all sharings are valid} \\ \perp & s^i \text{ is invalid} \\ (\text{"accuse } j\text{"}, v) & s_T^j \text{ is invalid for } j \neq i \end{cases}$$

  where $v = f(s^1, \ldots, s^{j-1}, 0, s^{j+1}, \ldots, s^n)$
- Reconstruction:
  - If all sets contain accusations, output v (same for all sets).
  - Otherwise, output the majority reply over non-accusing sets.

Analysis sketch

- Upto t corrupted servers. By resilience, a strict majority of sets sends a correct, non-accusing reply $f(x_1, x_2)$.
- Client 1 is corrupted.
  - Case 1: all partial sharings s_T^i are inconsistent. Then the adversary gets $\perp$, and learns nothing. All honest parties output $f(0, x_2)$.
  - Case 2: For some T, the s_T^i's are consistent. The induced secret v s the same for all such sets by pairwise intersection + properties of secret sharing. All non-blaming sets report $f(v, x_2)$.

Some hints on Result 3

1. Start with a t-private $dt + 1$-party protocol $\Pi_{\text{priv-poly}}$ for degree-d multivariate polynomials.
2. Transform $\Pi_{\text{priv-poly}}$ into a protocol $\Pi_{\text{sec-poly}}$ secure with selective abort for degree-3 polynomials.
   - Use statistical MACs.
3. Transform $\Pi_{\text{sec-poly}}$ into a protocol $\Pi_{\text{sec-gen}}$ secure with selective abort for general $f \in$ Poly.
   - Use representations of general functions by (vectors of) degree-3 randomizing polynomials [ik00,aik05].

Toolbox 2

- d-multiplicative scheme (over $\mathcal{F}$): Given shares of d secrets $s_i^1, \ldots, s_i^d$ to party i, it can locally compute an additive share m_i of $s^1 \cdot \ldots \cdot s^d$.
- Sharing a monomial $\alpha x_{i_1} \cdot \ldots \cdot x_{i_d}$: αm_i.
- Sharing a degree-d polynomial: add up the shares of the monomials.
- $(dt + 1, t)$-Shamir is d-multiplicative: simply multiply the shares.
  - Bivariate version of $(dt + 1, t)$-Shamir is also d-multiplicative.

Starting point - a standard 2-round private protocol for semi-honest parties [bgw88,ik00]

Evaluate a degree-d polynomial $P(x_1, \ldots, x_n)$.
- Round 1: Party i
  - Shares x_i via $(dt + 1, t)$-Shamir. Denote sharing by $p_i(y)$.
  - Additively shares 0 via $(dt + 1, dt)$-Shamir. Denote sharing by $z_i(y)$.
- Round 2: Party i:
- Lets $Z(y) = \Sigma_{i \in [n]} z_i(y)$.
- Sends $P'(i) = P(p_1(i), \ldots, p_n(i)) + Z(i)$ to all parties.
- Reconstruction: Output $P'(0)$.

using d-multiplicativity of Shamir

How about malicious adversaries?

- Remains private for degree-2 polynomials ($d = 2$).
- Privacy breaks at $d = 3$:
  - For a case as simple as $P(x_1, x_2, x_3) = x_1 \cdot x_2 \cdot x_3,\ n = 4\ t = 1$.
- The main problem: distributing inconsistent shares can allow the adversary to learn extra information.

Solution idea

Punish the adversary for distributing inconsistent shares.

More specifically

- Each party discloses its Round 2 message conditioned on the shares of all other parties being consistent.
- Should be done without knowing anything about the other parties' shares are!

Toolbox 3 - pairwise verifiability

- First idea: Make the consistency condition as simple as possible.
- Tool: pairwise verifiable secret sharing scheme. There exist local tests $V_{i,j}$, such that if all pairwise tests $V_{i,j}(s_i) = V_{j,i}(s_j)$ pass, then the sharing is globally consistent.
  - Bivariate $(dt + 1, t)$-Shamir has this property!

Toolbox 3 - CDS

- Second idea: Implement disclosing secrets conditioned on a single test's $V_{i,j}(s_i) = V_{j,i}(s_j)$ success.
- Tool: A from CDS (Conditional disclosure of secrets):
  - It has two rounds.
  - Party S holds a secret s, parties A, B hold values a, b respectively. There are $n - 3$ other parties holding no inputs.
  - If A, B, S are honest, and $a = b$, all honest parties output s.
  - If $a \neq b$, and A, B, S are honest, then the adversary's view is independent of s (even conditioned on a, b.)
  - Some additional mild technical requirements...
- Such a CDS can be easily implemented.

Amending the basic protocol

- Round 1:
  - Party i shares x_i via bivariate $(3t + 1, t)$-Shamir. Denote sharing by $p_i(v, w)$.
  - Party i additively shares 0 via bivariate $(3t + 1, 3t)$-Shamir. Denote sharing by $z_i(v, w)$.
- Round 2: Party i:
  - Lets $Z(v, w) = \Sigma_{i \in [n]} z_i(v, w)$.
  - Computes an additive share of $P'(0, 0)$ for $P'(v, w) = P(p_1(v, w), \ldots, p_n(v, w)) + Z(v, w)$.
  - Discloses the share conditioned that all pairwise conditioned $V_{i,j} = V_{j,i}$ are satisfies by all sharings (run a separate CDS)
- Reconstruction: Recover $P'(v, w)$, and output $P'(0, 0)$.

bibliography

[aik05] Computationally private randomizing polynomials and their applications.
[bgw88] Completeness Theorems for Noncryptographic Fault-Tolerant Distributed Computations.
[ik00] Randomizing polynomials: A new representation with applications to round-efficient secure computation.
[gikr01] The Round Complexity of Verifiable Secret Sharing and Secure Multicast.
[gikr02] On 2-Round Secure Multiparty Computation.
[rb89] Verifiable Secret Sharing and Multiparty Protocols with Honest Majority.