Ad Hoc PSM Protocols: Secure Computation without Coordination

Amos Beimel (BGU)

Yuval Ishai (Technion, UCLA)

Eyal Kushilevitz (Technion)

Eurocrypt 2017

Ad Hoc MPC [BGIK16]

The (basic) problem:

- Universe of n (honest but curious) parties.
- Set of k parties S, not known in advance, participate in the actual computation of some f (say, symmetric).

Examples:

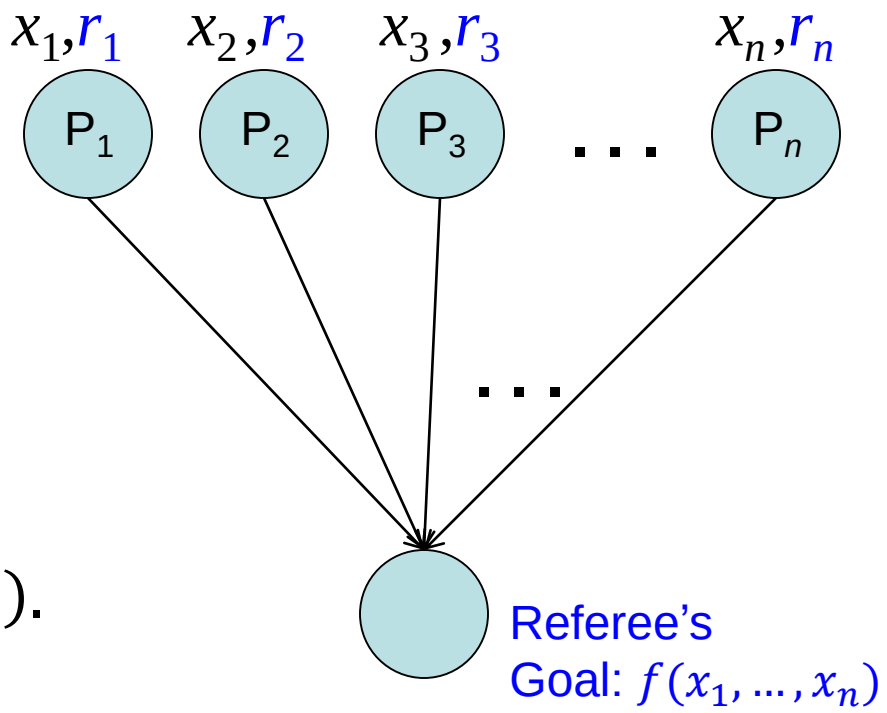
- Voting_k: output majority vote of k participants.
- Dating: 2 out of n players want to know if they match.

Easy in "standard" MPC model where parties can talk to each other.

Can this be done without adding communications rounds?

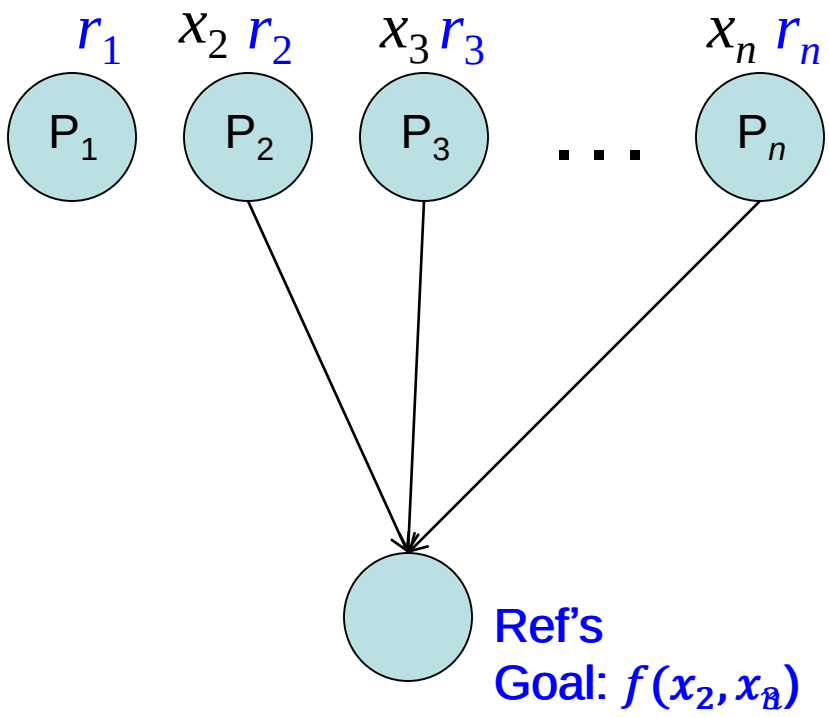
Private Simultaneous Messages (PSM) model [FKN94,IK97]

- Simplest communication pattern.
- Each party sends one message.
- Shared (correlated) randomness.
- Correctness: Ref learns $f(x_1, \ldots, x_n)$.
- Security: Ref learns nothing else.



Ad Hoc PSM model

- *n* parties.
- Correlated randomness.
- Exactly *k* parties show up.
- Participants not known in advance.



Ad-Hoc PSM: assumptions + variants

- Exactly k parties show up.
  - If allow $|S| > k$ "best possible security" definition gives Ref f's value on all size-k subsets and nothing else.
- f symmetric; else can sort by id's or specific f_S, for any S.
- S not known to the parties but will be known to Ref.
  - If require anonymity, we need anonymous channels.
- Information-Theoretic or computational security.

Our Results

- Constructions of ad hoc PSM protocols:
  - Every function has an IT ad hoc PSM.
  - All functions known to have an efficient IT PSM have an efficient IT ad hoc PSM.
  - All poly-time functions have an efficient computational ad hoc PSM.

- Connections with other primitives:
  - Order revealing encryption from IT ad hoc PSM.
  - NIMPC (*t*-robust PSM) iff best possible ad hoc PSM.
  - Best possible computational ad hoc PSM iff iO exists.
  - (fuzzy) point function obfuscation.

Example 1: difference (k=2)

For S={P_i,P_j}, i<j, output $x_i - x_j \bmod p$.

Common randomness: $r \in_R \mathbb{Z}_p$.

Protocol:

1. P_i: $m_i = x_i + r \bmod p$.
2. Ref: given m_i,m_j, where i<j, outputs $m_i - m_j = x_i - x_j \bmod p$.

Correctness: √

Security: √

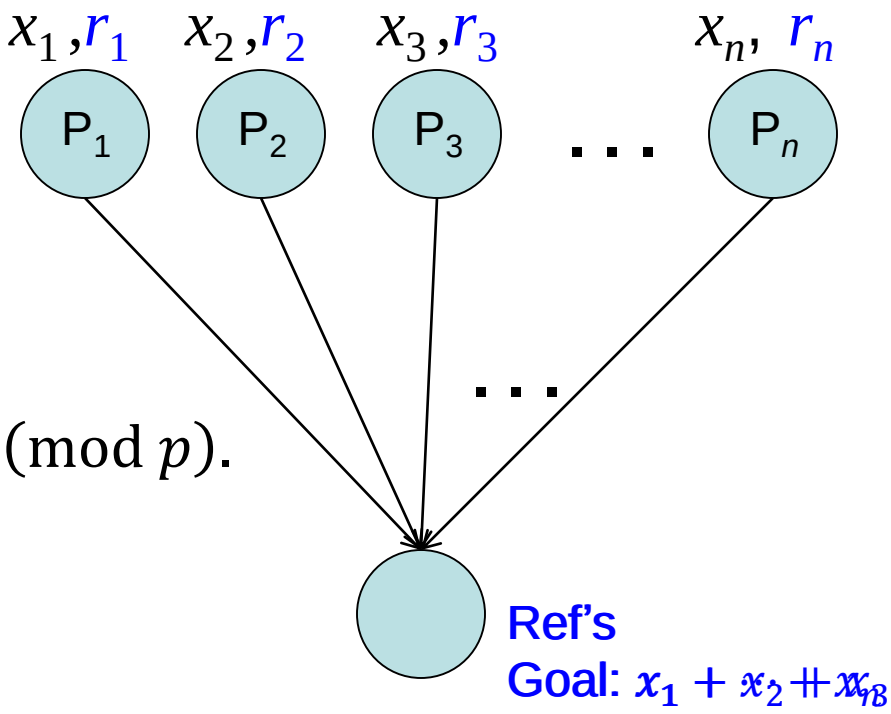
Example 2: ~~Ad Hoc~~ PSM for Sum_n

Input: Each P_i is given $x_i \in \mathbb{Z}_p$.

Output: Ref gets $\Sigma x_i \bmod p$.

Randomness: $r_1,\ldots,r_n \in_R \mathbb{Z}_p$ s.t. $\Sigma r_i \equiv 0 \pmod p$.

Protocol:



1. Each P_i computes $m_i = x_i + r_i \bmod p$ and sends to Ref.
2. Ref computes $\Sigma m_i \equiv \Sigma x_i + \Sigma r_i \equiv \Sigma x_i \bmod p$.

Examples 2: Ad Hoc PSM for SUM_k

Output: Ref gets $\Sigma_{i \in S} x_i \bmod p$.

Randomness: $r_1, \ldots, r_n \in_R \mathbb{Z}_p$ s.t. $\Sigma r_i \equiv 0 \bmod p$.

k-of-n secret sharing of each r_j into $\{r_{j,i}\}_{i \in [n]}$.

P_i receives r_i and $\{r_{j,i}\}_{j \neq i}$.

Messages: P_i sends $m_i = x_i + r_i \bmod p$ and all the shares it got.

Output of Ref (on S of size k):

- For $i \in S$ knows $x_i + r_i \bmod p$.
- For $j \notin S$ can reconstruct r_j (knows k shares).
- Output $\Sigma_{i \in S} (x_i + r_i) + \Sigma_{j \notin S} r_j \equiv \Sigma_{i \in S} x_i \pmod p$.

Security: for $i \in S$, value of r_i hidden; view of Ref can be generated from its view in SUM_n where each $P_j \notin S$ has $x_j = 0$.

Constructions of Ad Hoc PSM

- Trivial: An ad hoc PSM with overhead of $\binom{n}{k}$ compared to standard PSM for f.
  - Best possible security.
  - All functions have an (inefficient) ad hoc PSM.

- For symmetric functions there is an ad hoc PSM with overhead of $2^{O(k)} \cdot \log n$ compared to standard PSM for f.

- Construction of an ad hoc PSM protocol for f from a PSM for a related function g.
  - All functions known to have efficient IT PSM have efficient IT ad hoc PSM.
  - All poly-time functions have an efficient computational ad hoc PSM.

Application: Order Revealing Encryption (ORE) [AKSX04,BCLO09,BCO11]

A private-key encryption equipped with a comparison.

- A public procedure Comp:
  - $c_1 = \mathrm{Enc}(x_1, k), c_2 = \mathrm{Enc}(x_2, k)$.
  - $\mathrm{Comp}(c_1, c_2) = 1$ iff $x_1 \leq x_2$.
- Encryption does not leak additional information.

IT Ad Hoc PSM $\Rightarrow$ ORE

- Use ad hoc PSM for the Greater-Than function with $n = 2^{\lambda}$ parties and $k = 2$.
  - λ – security parameter.
  - Greater-Than has a IT PSM with complexity $\text{poly}(\ell)$.
  - Has an IT ad hoc PSM with complexity $\log n \cdot \text{poly}(\ell) = \lambda \cdot \text{poly}(\ell)$.

- Statistical IT-security for two messages.
- Complexity: $\lambda \cdot \text{poly}(\ell)$.

- For more than two messages: leakage 1/poly.

Best possible Ad Hoc PSM

- [BGIK16]: Multi-Input Functional Encryption (MIFE) $\Rightarrow$ Distribution Design $\Rightarrow$ Computational best possible ad hoc PSM (w/indistinguishability def.)

- Best possible ad hoc PSM $\Rightarrow$ NIMPC $\Rightarrow$ iO.

- Best possible comp. ad hoc PSM for AND

  $\Rightarrow$ point function obfuscation.

- Best possible comp. ad hoc PSM for Threshold func.

  $\Rightarrow$ fuzzy point function obfuscation.

Conclusion: Best possible ad hoc PSM requires strong assumptions.

Summary

- We present constructions of Ad Hoc PSM protocols.
  - Every function has an ad hoc PSM.
  - All functions known to have efficient IT PSM have efficient IT ad-hoc PSM.
  - All poly. time functions have an efficient comp. ad hoc PSM.
- Connections to ORE, NIMPC, iO, point function obfuscation.

Obvious open problems: more protocols, improved complexity and parameters, more connections with other primitives.

- Best possible security.

Thank you!