Distributed Algorithmic Mechanism Design: Recent Results and Future Directions

Joan Feigenbaum
Yale

Scott Shenker
ICSI

Slides: http://www.cs.yale.edu/~jf/DIALM02.{ppt,pdf}
Paper: http://www.cs.yale.edu/~jf/FS.{ps,pdf}

Two Views of Multi-agent Systems

CS	ECON
Focus is on Computational & Communication Efficiency	Focus is on Incentives
Agents are Obedient, Faulty, or Adversarial	Agents are Strategic

Internet Computation

- Both incentives and computational and communication efficiency matter.

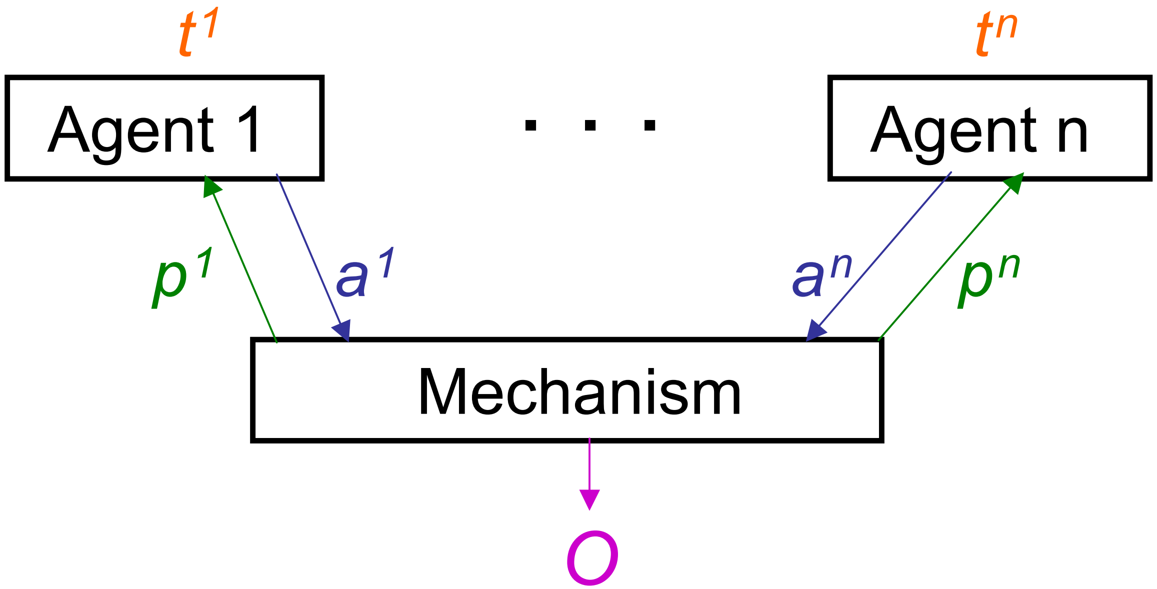
- “Ownership, operation, and use by numerous independent self-interested parties give the Internet the characteristics of an economy as well as those of a computer.”

$\Rightarrow$ DAMD: “Distributed Algorithmic Mechanism Design”

Outline

- DAMD definitions and notation
- Example: Multicast cost sharing
- Example: Interdomain routing
- General open questions

Definitions and Notation



(Private) types: $t^1, \ldots, t^n$

Strategies: $a^1, \ldots, a^n$

Payments: $p^i = p^i(a^1, \ldots, a^n)$

Output: $O = O(a^1, \ldots a^n)$

Valuations: $v^i = v^i(t^i, O)$

Utilities: $u^i = v^i + p^i$

Agent i chooses a^i to maximize u^i.

"Strategyproof" Mechanism

For all i, t^i, a^i, and $a^{-i} = (a^1, \ldots, a^{i-1}, a^{i+i}, \ldots a^n)$

$$v^i(t^i, O(a^{-i}, t^i)) + p^i(a^{-i}, t^i)$$
$$\geq v^i(t^i, O(a^{-i}, a^i)) + p^i(a^{-i}, a^i)$$

- "Dominant-Strategy Solution Concept"
  Said to be appropriate for Internet-based games; see, *e.g.,* Nisan-Ronen '99, Friedman-Shenker '97
- "Truthfulness"

Algorithmic Mechanism Design

N. Nisan and A. Ronen
Games and Economic Behavior **35** (2001), pp. 166--196

- Introduced computational efficiency into mechanism-design framework.
- Polynomial-time computable functions
  $O(\,)$ and $p^i(\,)$
- Centralized model of computation

Example: Task Allocation

Input: Tasks $z_1, \ldots, z_k$

Agent i's type: $\vec{t^i} = (t^i_1, \ldots, t^i_k)$
(t^i_j is the minimum time in which i can complete z_j.)

Feasible outputs: $Z = Z^1 \sqcup Z^2 \sqcup \ldots \sqcup Z^n$
(Z^i is the set of tasks assigned to agent i.)

Valuations: $v^i(\vec{t^i}, Z) = -\sum_{z_j \in Z^i} t^i_j$

Goal: $\text{Minimize}_{Z} \; \max_{i} \sum_{z_j \in Z^i} t^i_j$

Min-Work Mechanism [NR '99]

$O(\vec{a^1}, \ldots, \vec{a^n})$: Assign z_j to agent with smallest a^i_j

$$p^i(\vec{a^1}, \ldots, \vec{a^n}) = \sum_{z_j \in Z^i} \min_{i \neq i'} a^{i'}_j$$

Theorem: Strategyproof, n-Approximation

Open Questions:

- Average case(s)?
- Distributed algorithm for Min-Work?

Distributed AMD [FPS '00]

Agents *1, …, n*

Interconnection network *T*

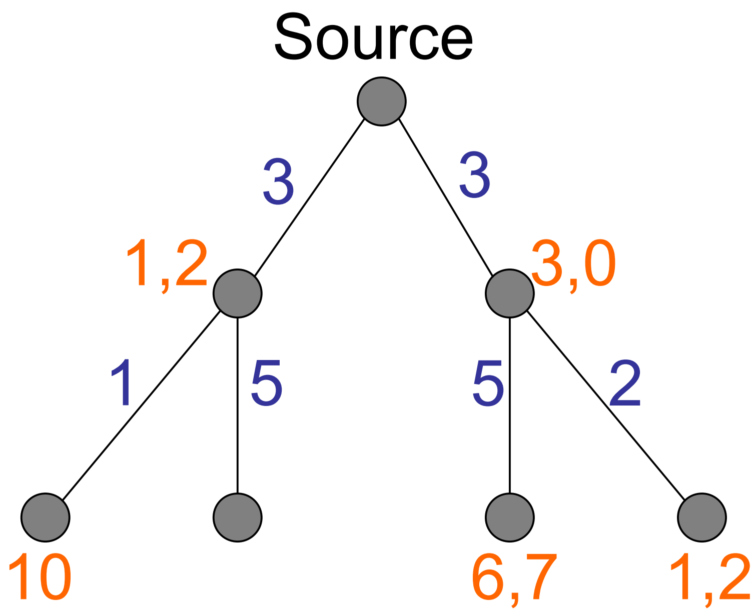
Numerical input $\{c_1, \dots c_m\}$

- $O(|T|)$ messages total
- $O(1)$ messages per link
- Polynomial-time local computation
- Maximum message size is

  $polylog(n, |T|)$ and $poly(\sum_{j=1}^{m} ||c_j||)$.

"Good network complexity"

Multicast Cost Sharing Mechanism-Design Problem



Users' types

Link costs

Receiver Set

Which users receive the multicast?

Cost Shares

How much does each receiver pay?

Two Natural Mechanisms [MS '97]

- Marginal cost
  - Strategyproof
  - Efficient
  - Good network complexity [FPS '00]
- Shapley value
  - Group-strategyproof
  - Budget-balanced
  - Minimum worst-case efficiency loss
  - Bad network complexity [FKSS '02]

Marginal Cost

Receiver set: $R^* = \underset{R}{\arg\max}\ \mathrm{NW}(R)$

$$\mathrm{NW}(R) \equiv \sum_{i \in R} t^i \quad - \quad C(T(R))$$

Cost shares:

$$p^i = \begin{cases} 0 & \text{if } \; i \notin R^* \\ t^i - \left[\mathrm{NW}(R^*(\vec{t}\,)) - \mathrm{NW}(R^*(\vec{t}\,|^i 0))\right] & \text{o.w.} \end{cases}$$

Computable with two (short) messages per link and two (simple) calculations per node. [FPS ’00]

Shapley Value

Cost shares: $c(l)$ is shared equally by all receivers downstream of l. (Non-receivers pay 0.)

Receiver set: Biggest R^* such that $t^i \geq p^i$, for all $i \in R^*$

Any distributed algorithm that computes it must send $\Omega(n)$ bits over $\Omega(|T|)$ links in the worst case. [FKSS ’02]

Profit Maximization [FGHK '02]

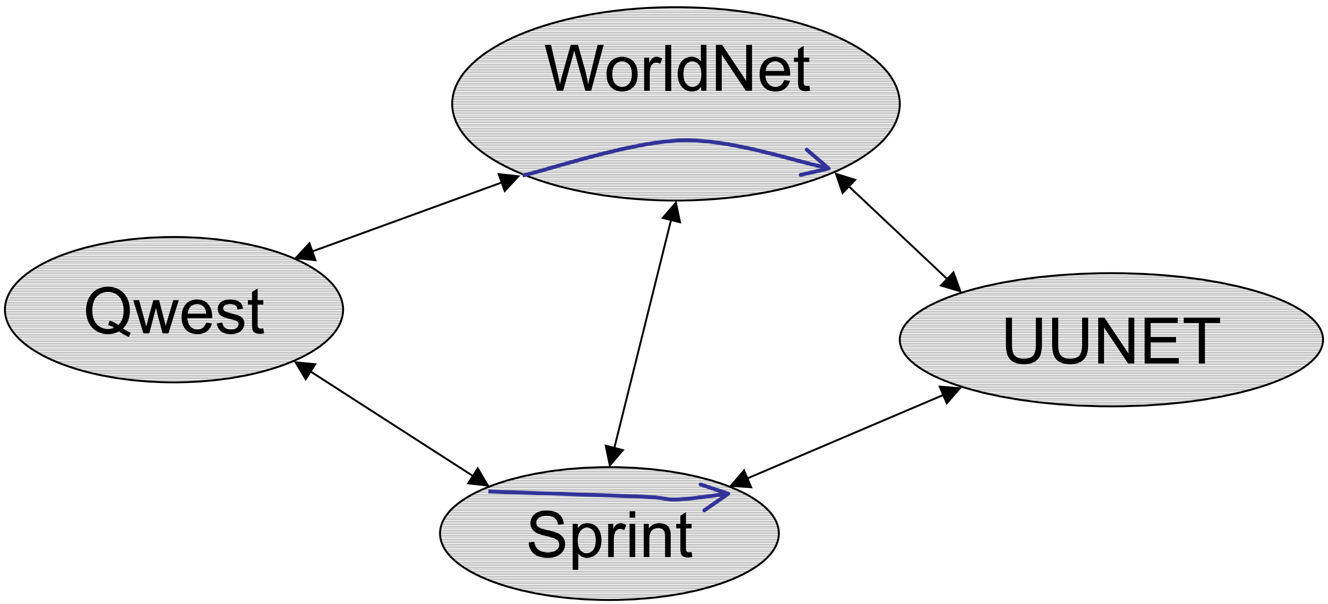
Mechanism:

- Treat each node as a separate "market."
- Choose clearing price for each market to maximize market revenue (approximately).
- Find profit-maximizing subtree of markets.

Results:

- Strategyproofness
- $O(1)$ messages per link
- Expected constant fraction of maximum profit if
  - Maximum profit margin is large (> 300%), and
  - There is real competition in each market.

Lowest-Cost Routing Mechanism-design Problem



Agents: Transit ASs

Inputs: Transit costs

Outputs: Routes, Payments

Problem Statement

Agents' types: Per-packet costs $\{c_k\}$

(Unknown) global parameter: Traffic matrix $[T_{ij}]$

Outputs: $\{route(i, j)\}$

Payments: $\{p^k\}$

Objectives:

- Lowest-cost paths (LCPs)
- Strategyproofness
- "BGP-based" distributed algorithm

(Some) AMD Results on Lowest-Cost Routing

- **Nisan-Ronen** [STOC ’99]
  – Polynomial-time centralized mechanism
  – Strategic agents are the edges
  – Single source-destination pair
- **Hershberger-Suri** [FOCS ’01]
  – Same formulation as in NR’99
  – Compute payments for *all* edges on the path in the same time it takes to compute payment for *one* edge
- **Feigenbaum-Papadimitriou-Sami-Shenker** [PODC ’02]
  – BGP-based, distributed algorithm
  – Strategic agents are the nodes
  – All source-destination pairs

Notation

- LCPs described by an indicator function:

$$I_k(c;\, i,j) \equiv \begin{cases} 1 & \text{if } k \text{ is on the LCP from } i \text{ to } j, \\ & \quad \text{when cost vector is } c \\ 0 & \text{otherwise} \end{cases}$$

- $c|_{\infty}^{k} \equiv (c_1, c_2, \ldots, c_{k-1}, \infty, c_{k+1}, \ldots, c_n)$

A Unique VCG Mechanism

Theorem 1:

For a biconnected network, if LCP routes are always chosen, there is a unique strategyproof mechanism that gives no payment to nodes that carry no transit traffic. The payments are of the form

$$p^k = \sum_{i,j} T_{ij}\, p^k_{ij}, \qquad \text{where}$$

$$p^k_{ij} = c_k\, I_k(c;\, i, j) + \left[\sum_r I_r(c|^k\infty;\, i, j)\, c_r - \sum_r I_r(c;\, i, j)\, c_r \right]$$

Features of this Mechanism

- Payments have a very simple dependence on traffic T_{ij} : payment p^k is the sum of per-packet prices p_{ij}^k.
- Price p_{ij}^k is *0* if k is not on LCP between i, j.
- Cost c_k is independent of i and j, but price p_{ij}^k depends on i and j.
- Price p_{ij}^k is determined by cost of min-cost path from i to j not passing through k (min-cost "k-avoiding" path).

Performance of Algorithm

$$d = max_{i,j} \, || \, P(c; i, j) \, ||$$

$$d' = max_{i,j,k} \, || \, P^{-k}(c; i, j) \, ||$$

Theorem 2:

Our algorithm computes the VCG prices correctly, uses routing tables of size $O(nd)$ (a constant factor increase over BGP), and converges in at most $(d + d')$ stages (worst-case additive penalty of d' stages over the BGP convergence time).

Policy-Routing MD Problem [FGSS '02]

- Per-packet c_k is an unrealistic cost model.
- AS route preferences are influenced by reliability, customer-provider relationships, peering agreements, *etc*.

General Policy Routing:

- For all i,j, AS *i* assigns a value $v^i(P_{ij})$ to each potential route P_{ij}.
- Mechanism Design Goals:
  - Maximize $V = \sum_i v^i(P_{ij})$.
  - For each destination *j*, $\{P_{ij}\}$ forms a tree.
  - Strategyproofness, good network complexity

General Policy Routing is Hard

NP-hard even to approximate V closely

Approximation-preserving reduction from Maximum Independent Set

Possible approaches:

Restricted class of networks

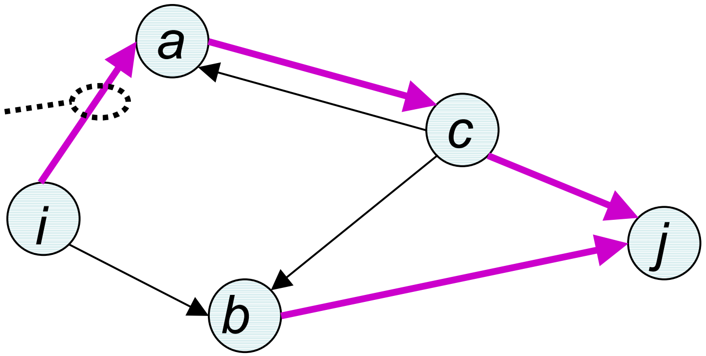
Restricted class of valuation functions $v^i(\ )$

Next-Hop Preferences

- $v^i(P_{ij})$ depends only on next-hop AS *a*.
- Captures preferences due to customer/provider/peer agreements.

For each destination *j* , we need to find a Maximum-weight Directed Spanning Tree (MDST).

Edge weight = $v^i([i\ a\ \ldots\ j])$



A Strategyproof Mechanism

Notation:

$T^*(S, j)$ = MDST on vertex set S, with destination j

Payments: $p^i = \sum_j p_j^i$, where

$$p_j^i = \text{Weight}\left[T^*(N, j)\right] - v^i(T^*(N, j)) - \text{Weight}\left[T^*(N - \{i\}, j)\right]$$

- Belongs to the family of "Vickrey-Clarke-Groves" (VCG) utilitarian mechanisms.

Towards a DAM for Next-Hop Preferences

- Centralized and distributed algorithms for MDST are known (*e.g.,* [Humblet ’83]).
- Need to compute VCG payments efficiently.
- Need to solve for *all* destinations simultaneously.
- Can we find a “BGP-based” algorithm?

Open Question: More "Canonically Hard Problems"

Hard "to solve on the Internet" if

- No solution simultaneously achieves:
  - Good network complexity
  - Incentive compatibility
- Can achieve either requirement separately.

GSP, BB multicast cost sharing is canonically hard.

Open Question: Find other canonically hard problems.

Open Question: More (Realistic) Distributed Algorithmic Mechanisms

- Caching
- P2P file sharing
- Distributed Task Allocation
- Overlay Networks

* Ad-hoc and/or Mobile Networks

Ad-Hoc and/or Mobile Networks

- Nodes make same incentive-sensitive decisions as in traditional networks, *e.g.*:
  - Should I connect to the network?
  - Should I transit traffic?
  - Should I obey the protocol?

- These decisions are made more often and under faster-changing conditions than they are in traditional networks.

- Resources (*e.g.*, bandwidth and power) are scarcer than in traditional networks. Hence:
  - Global optimization is more important.
  - Selfish behavior by individual nodes is potentially more rewarding.

Open Question: Strategic Modeling

- Agent behavior:
  In each DAMD problem, which agents are
  Obedient, Strategic, [Adversarial], [Faulty] ?

- Reconciling strategic and computational models:
  - "Strategyproofness" $\Rightarrow$ Agents have no incentive to *lie* about their private inputs.
  - But, output and payments may be *computed* by the strategic agents themselves (*e.g.,* in interdomain routing).
  - "Quick fix" : Digital Signatures
    [Mitchell, Sami, Talwar, Teague]

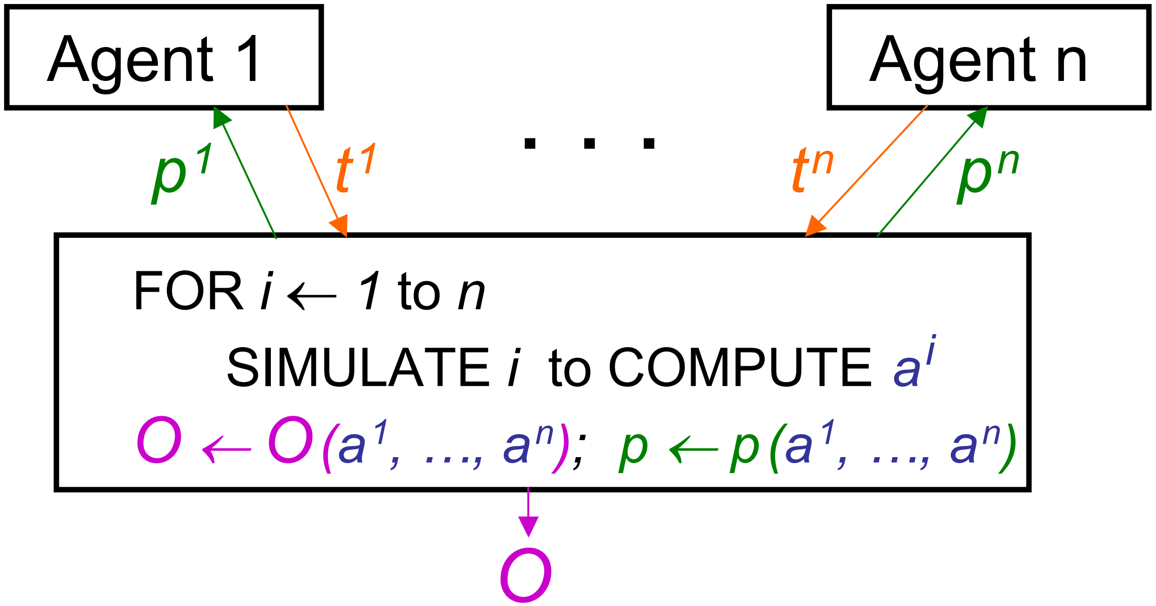
Is there a way to do this without a PKI?

Open Question: What about "provably hard" DAMD problems?

- AMD approximation is subtle. One can easily destroy strategyproofness.
- "Feasibly dominant strategies" [NR '00]
- "Strategically faithful" approximation [AFKSS '02]
- "Tolerable manipulability" [AFKSS '02]

Revelation Principle

If there is a mechanism (O, p) that implements a design goal, then there is one that does so truthfully.



Note: Loss of privacy
Shift of computational load

Is truthtelling really "dominant"?

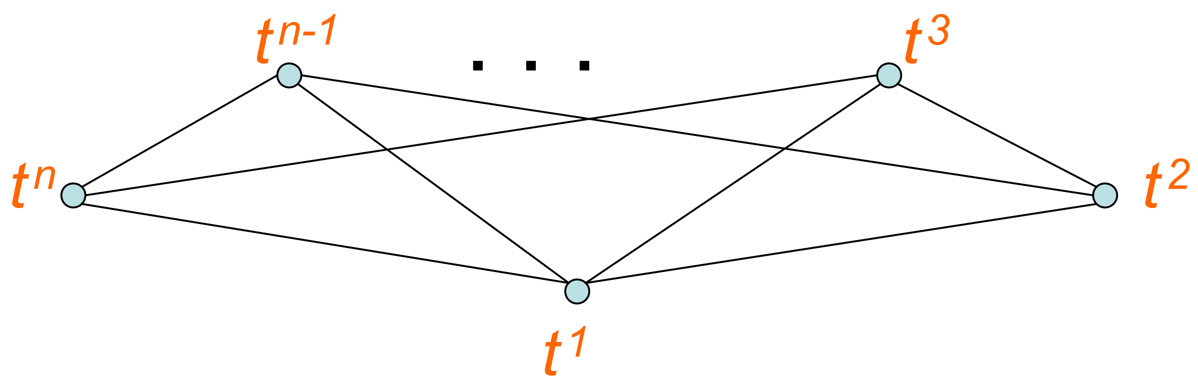
Consider Lowest-Cost Routing:

- Mechanism is strategyproof, in the technical sense: Lying about its cost cannot improve an AS's welfare *in this particular game*.

- But truthtelling reveals to competitors information about an AS's internal network. This may be a disadvantage *in the long run.*

- Note that the goal of the mechanism is not acquisition of private inputs *per se*, but rather evaluation of a function of those inputs.

Secure, Multiparty Function Evaluation



$$O = O\,(t^1, \ldots, t^n)$$

- Each i learns O.
- No i can learn anything about t^j (except what he can infer from t^i and O).

Extensive SMFE Theory Developed by Cryptographic Researchers

- Agents are either "good" or "bad."
  - "Good" is what's called "obedient" in DAMD.
  - "Bad" could mean, *e.g.*,

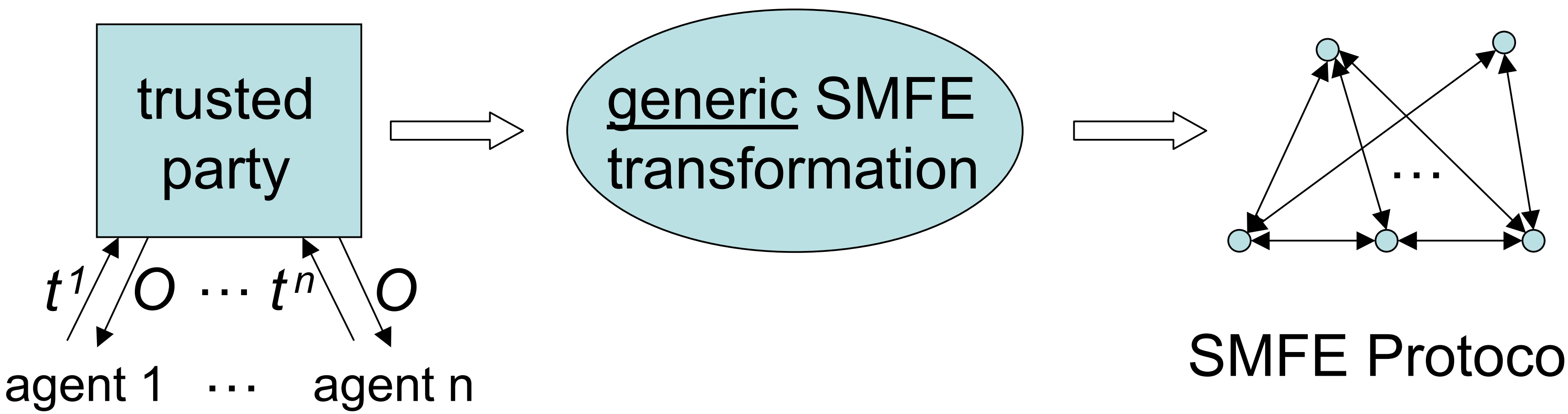
    Honest but curious

    Byzantine adversary

- Typical Results
  - If at most $r < n/2$ agents are honest but curious, every function has an *r-private* protocol.
  - If at most $r < n/3$ agents are byzantine, every function has an *r-resilient* protocol.

([BGW '88] uses threshold-*r* secret sharing and error-correcting codes.)

Constructive, "Compiler"-style Results



Tempting to think:

centralized mechanism ≈ trusted party

DAM ≈ SMFE protocol

Can SMFE Techniques Be Used for Agent Privacy in DAMs?

- In general, cannot simply "compose" a DAM with a standard SMFE protocol.
  - Strategic models may differ (*e.g.,* there may be Ω(n) obedient agents in an SMFE protocol but zero in a DAM).
  - Unacceptable network complexity
  - Agents don't "know" each other.
- Are SMFE results usable "off-the-shelf" at all?

Other General Questions

- New solution concepts

- Use of *indirect* mechanisms for goals other than privacy. Tradeoffs among

  - agent computation
  - mechanism computation
  - communication
  - privacy
  - approximation factors