

Distributed Games

From Mechanisms to Protocols

Dov Monderer and Moshe Tennenholtz
Technion



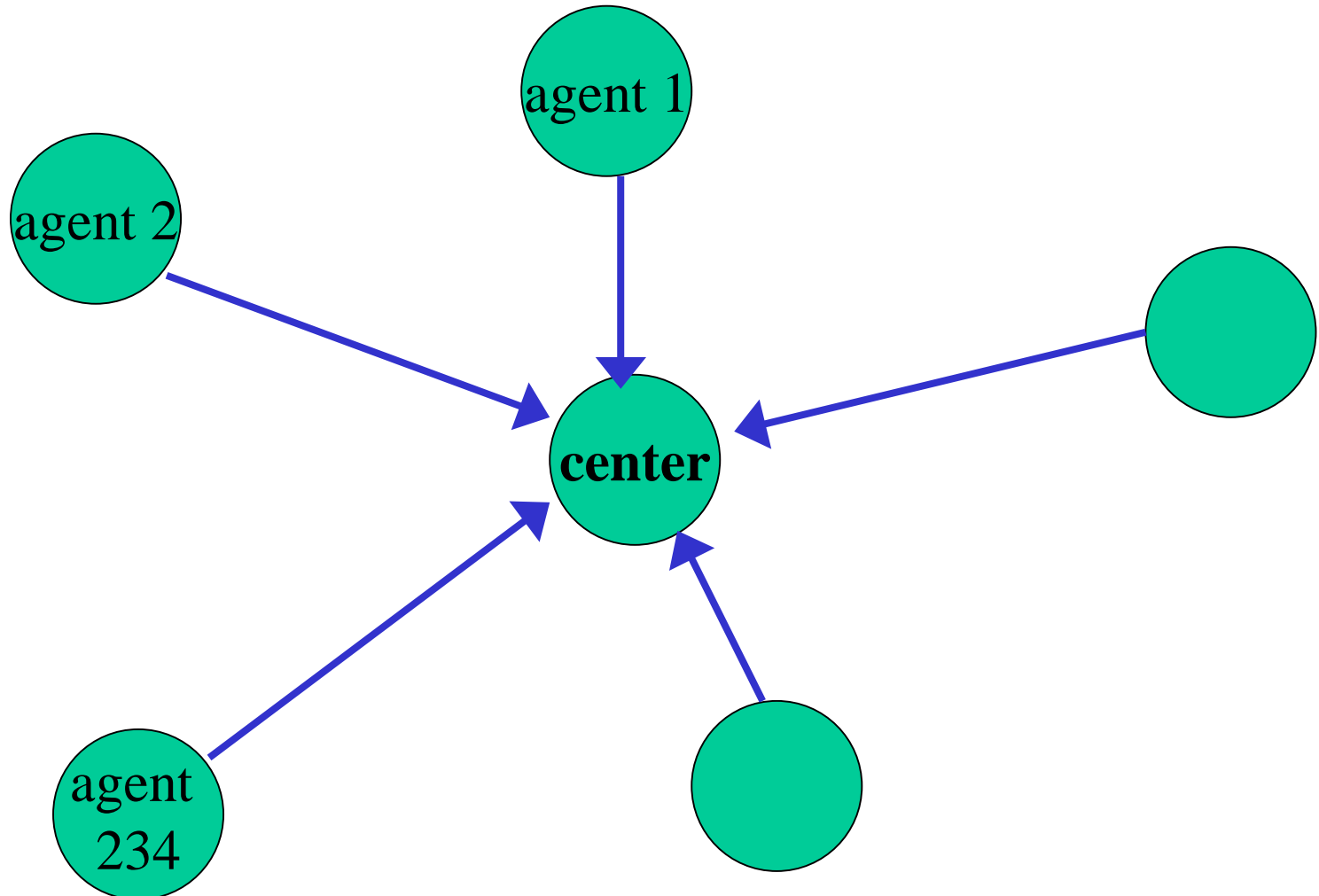
A preliminary and unfriendly version can be found at:

<http://ie.technion.ac.il/dov.phtml>

Example

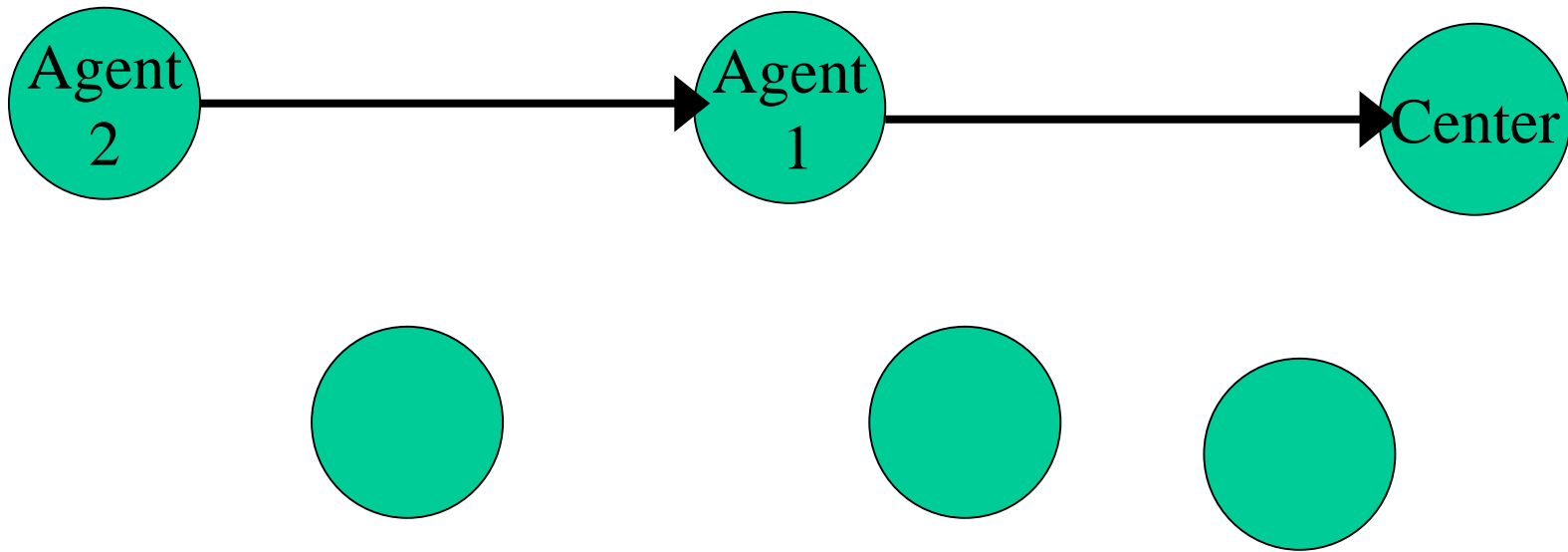
First-Price Auction (in economics):

(The highest bid wins, and the winner pays her bid).



Example

First-Price Auction in a communication network (e.g., the internet)



1. Agent 1 can read 2's bid.
2. Agent 1 can modify 2's bid.
3. Agent 1 can destroy 2's bid.

Cryptography Versus Game Theory

Problem 1: Agent 1 can read 2's bid.

Solutions:

Cryptography: Agent 1 cannot deduce the true bid from the encrypted bid, that is:

$$\text{Prob}(2\text{'s bid}|\text{encrypted bid})=\text{Prob}(2\text{'s bid}).$$

Game Theory: Change the mechanism (first-price auction) to another equivalent mechanism, in which knowing 2's bid does not help Agent 1.

A Game Theoretic Solution

Game theory solution in this particular case:
use a **second price auction**: The highest bid wins,
and the winner pays the highest non-winning bid.
That is,

if 1's bid is 70 2's bid is 80 and 3's bid is 100,
3 wins the object and pays 80.

if 1's bid is 70 2's bid is 100 and 3's bid is 100,
The winner is 2 or 3. The winner determined by a lottery
with equal probabilities, and she pays 100.

Many Ifs

If we assume that valuations are private, independent, and their distribution is commonly known, and **if** we accept the equilibrium assumption of economic theory:

1. The **revenue equivalence principle** states that first- and second-price auctions yield the same expected revenue.
2. In a second-price auction, bidding your true valuation is a (weakly) dominating strategy. In particular, if you know your opponents' bids, you cannot do better by deviating from
6 your true valuation.

Telling the truth is a selfish strategy

Assume Agent's 1 willingness to pay (valuation) is \$100, and Agent 1 knows the bids of agents 2 and 3. What will she do? (w.l.o.g 2's bid < 3's bid)

Agent 2	Agent 3
80	90
80	120
120	130
80	100
100	120

Intermediate Summary

Cryptography

Game Theory

First-Price
auctions.

General
Mechanism

1. Agent 1 can
read 2's bid.

Encryption

Second-Price
auctions.

?

2. Agent 1 can
modify 2's bid.

Digital
Signature

?

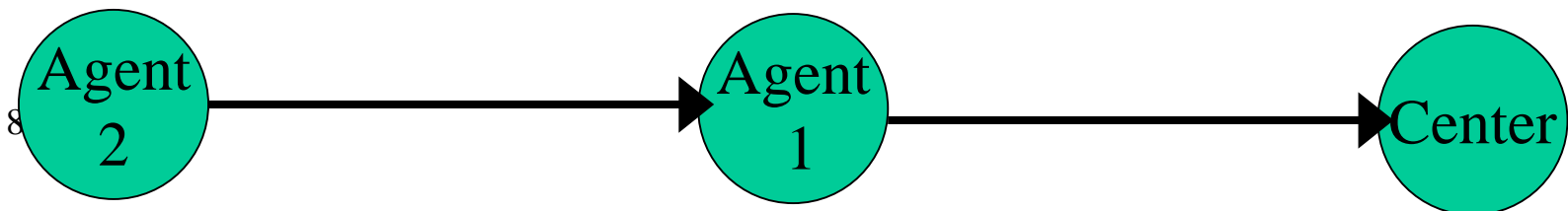
?

3. Agent 1 can
destroy 2's bid.

?

?

?



Our Goal

Our goal in this paper is to replace the question marks with new mechanisms that would function efficiently in networks, and would be less costly than the cryptographic solutions, when such solution exist.

**A short course in
private key cryptography**

**A short course in
public key cryptography**

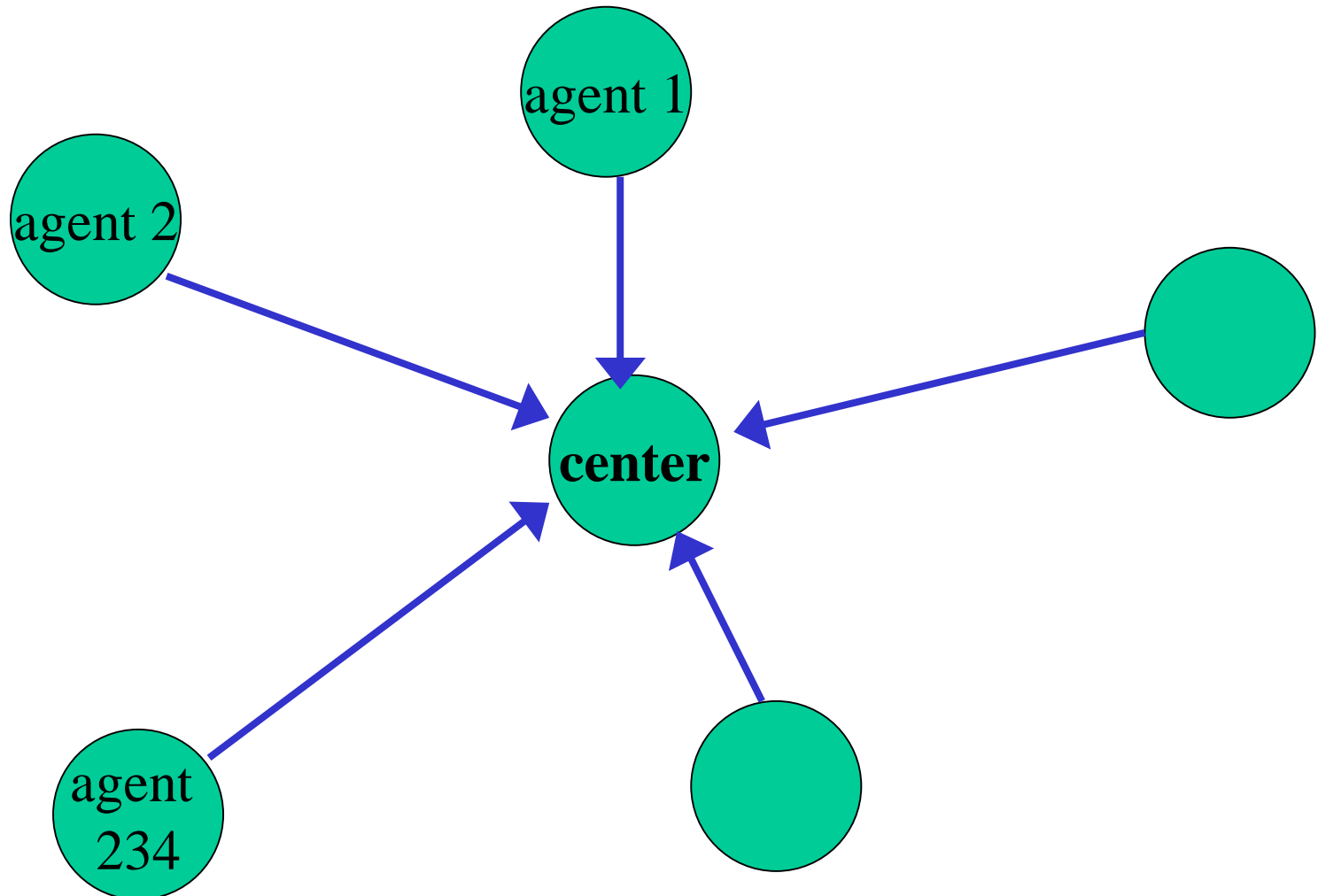
Mechanisms in Economics



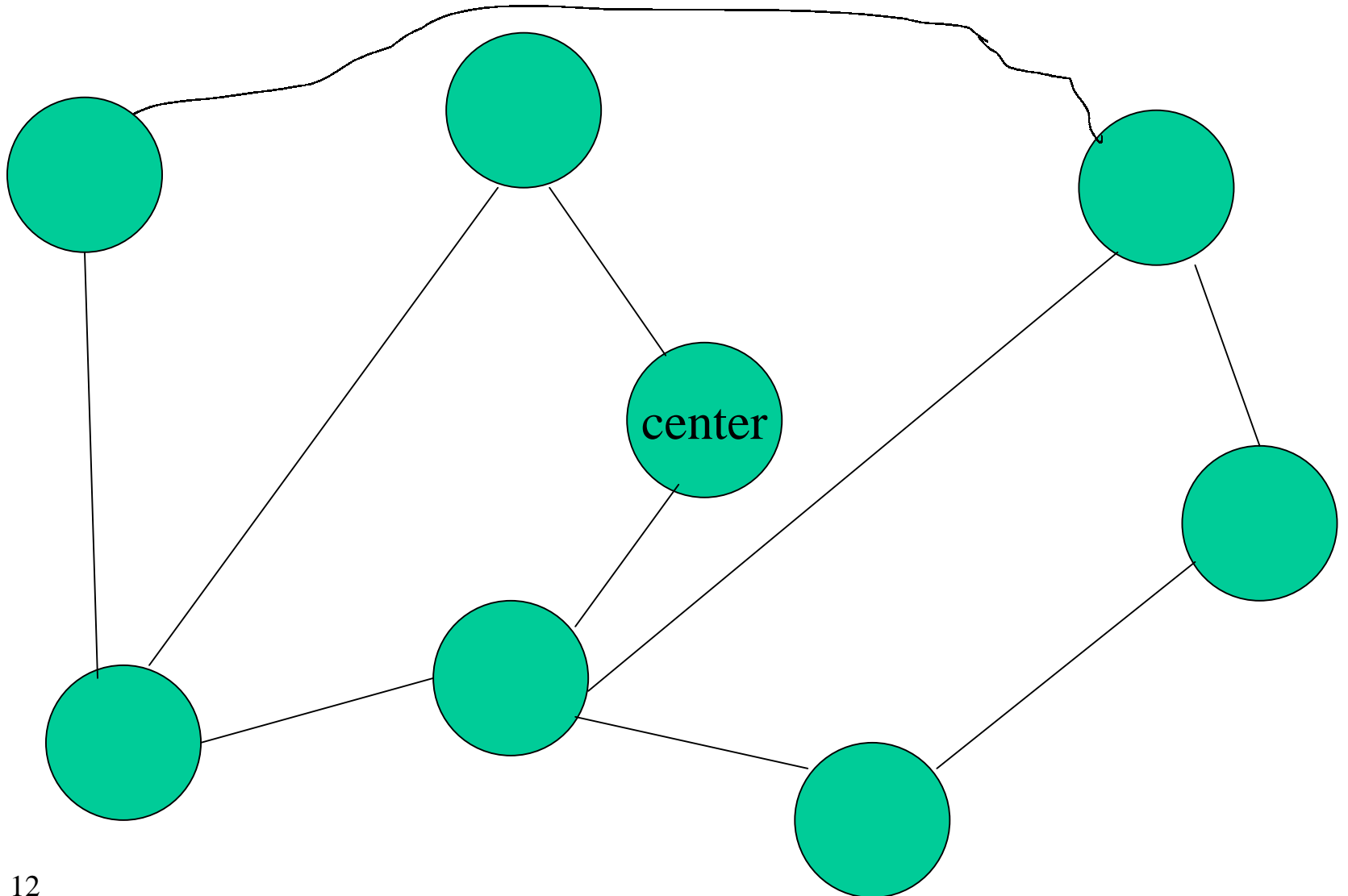
Mechanism Design
in Economics

Networks

Mechanisms in economics work well in such network:



Communication Networks



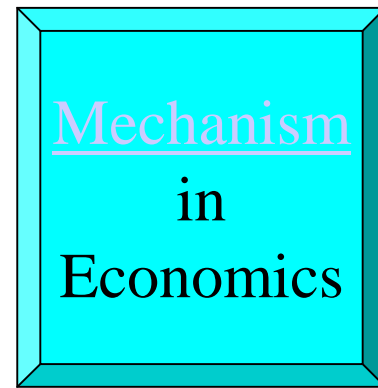
Communication Networks II

A communication network is described by a simple graph $L=(W,D)$, where W is the set of $n+1$ vertices, and D is the set of edges. Every vertex is labeled by a distinct $i=0,1,\dots,n$, where 0 denotes the center (i may be an IP number).

It is assumed that every i is either directly connected to the center, or there exist at least two disjoint paths that connect i to the center.

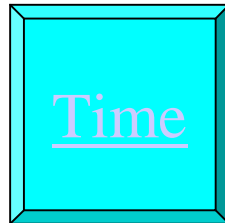
An environment and a communication network generate
A Distributed Environment

However, we assume that (normalized) types are dyadic numbers, of the form $k/2^r$, $0 \leq k \leq 2^r - 1$. Hence, the set of types is $V = \{0,1\}^r$ – the set of all vectors of length r of zeros and ones.



Time

**The agents are about to send messages to the center.
However, we need to treat the issue of **time**.**



Mechanisms in Distributed Environments

Recall that a mechanism in economics is defined by a set of messages and by a (probabilistic) function h that determines the center's action as a function of the vector of bids.

In a distributed environment, a mechanism is called a **center's protocol**, and its description is more complex...

Center's Protocol

A protocol must define what is the format of a **standard message**:

In our model, a standard message has 4 parts:

Part 1 : Sender's Name (not encrypted)

Part 2: Receiver Name (not encrypted).

Part 3: A bid-- a vector in $\{0,1\}^k$, for some $k \geq r$.

Part 3: Additional Data.

A protocol must specify what is a **feasible message**, by defining the structure and content of the data , and the length k of a **feasible bid**.

A protocol may specify the interpretation of a bid; It provides an **interpretation function**:

$$I: \{0,1\}^k \longrightarrow \{0,1\}^r$$

A protocol also provides a maximal number of rounds T^* , after which it stops.

Center's Protocol II

The center expects the agents to send it standard messages via their neighbors. The main part of the protocol specify the **behavior** of the center at every round, as a function of the messages (some of them may be non-standard) received by it at previous rounds. For simplicity, let us assume that the center performs one of the following actions:

- **Do nothing.**
- **Terminate the process.**
- **Choose a (random) outcome and stop.**

The center's protocol is published (these are the rules of commerce).

Distributed games

The communication environment and a center protocol define a (Bayesian) (multistage) **distributed game**.

Agents' Behavior

At each round an agent can send messages via its neighbors (or it can do nothing).

An **agent's protocol** (strategy) is a function that assigns to every type and every history of received and sent messages, the description of messages (and addresses) to be sent on the next round.

We assume that every non-active agent forwards as is every message that goes through it according to the instructions given in the message.

Distributed protocols

A **distributed protocol** is an $(n+1)^{\text{th}}$ tuple consisting of a center protocol and an equilibrium profile of agents' protocols.

Like a mechanism with a give equilibrium profile, every distributed protocol defines an outcome function

$$g: (\{0,1\} \times V)^n \longrightarrow \Delta(A).$$

Theorem 1:

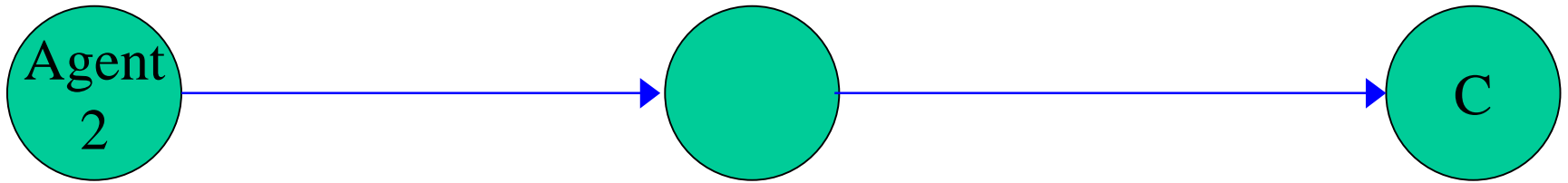
Let (E,L) be a distributed environment, where E is the environment and L is the communication network. Every outcome function that is implementable by a mechanism in E is implementable by a distributed protocol executed in (E,L) .

Proof (just kidding)

Before I present the idea of the proof we nevertheless need a minimal knowledge in private key cryptography

Private Key Encryption

Recall this picture:



Private
Key

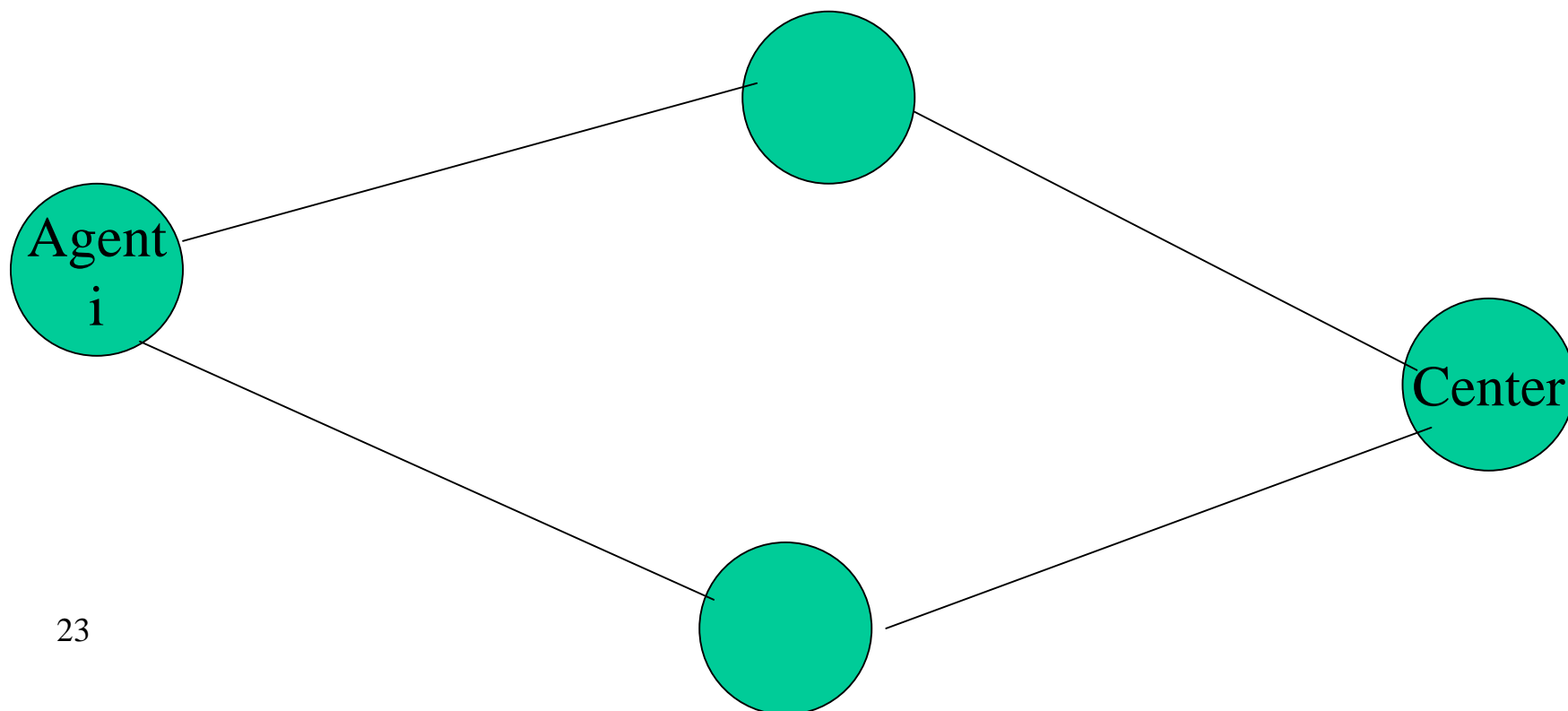
Slide
9

The Proof by an Example

Case 1: Types are uniformly distributed

That is, the probability of v in $\{0,1\}^r$ is $1/2^r$.

The original mechanism is truth-telling .



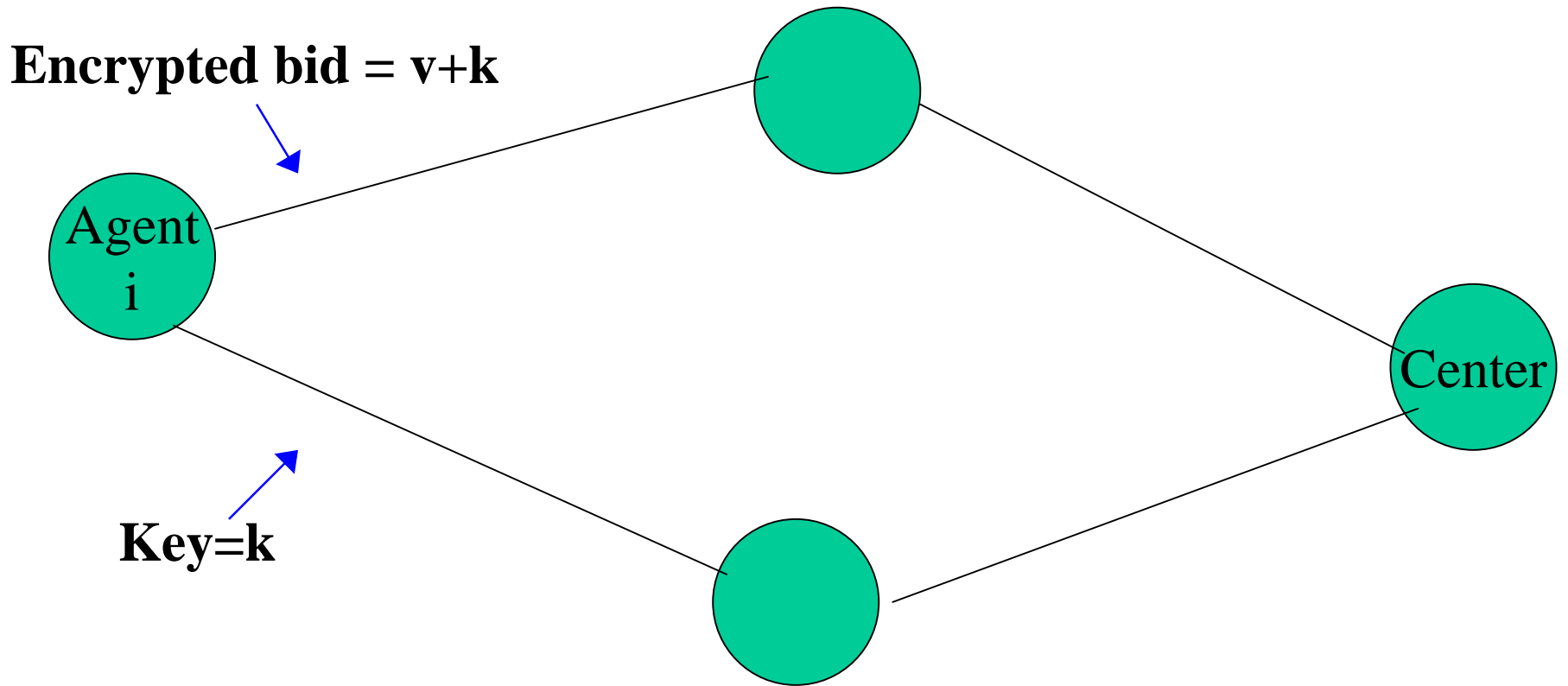
Agent i's protocol

Agent i has a binary type $v=(a_1, a_2, \dots, a_r)$. She chooses a binary key $k=(k_1, \dots, k_r)$ that is the outcome of a lottery with equal probabilities over $\{0,1\}^r$.

Agent i sends the key and the encrypted message $y=k+v$ in two disjoint paths, recording these paths on both messages.

Agent i forward every message that seems to be consistent with the above 2 rules, and she destroys any other message.

Another Graph



Center's Protocol

Decrypt any bid in a legal message-- a message that has a consistent counter part; $v=(v+k)+k$ (recall that $k+k=0$).

If the center gets two non consistent messages, it punishes **every agent** in all paths appearing on Part 4 of these messages, **including the sender**, by removing them from the mechanism (i.e., assuming that they do not participate).

Note that an agent does not have an incentive to deviate from its protocol if it believes that all other agents obey their protocol.

When types are not uniformly distributed

