E0 337: Topics in Advanced Cryptography 28th Aug, 2023

Lecture 3

Instructor: Dr. Bhavana Kanukurthi, Dr. Chaya Ganesh *Scribe: Sourabh*

1 Welcome to E0 337

1.1 Course Information

Contact information and office hours:

- **Lecturer:** Bhavana Kanukurthi (bhavana@iisc.ac.in), Chaya Ganesh (chaya@iisc.ac.in).
- **Presenter:** Girisha (girishabs@iisc.ac.in).

1.2 Course Topics

The course will cover the following topics:

- Mechanism Design (Setting and examples)

2 Mechanism Design

2.1 Introduction

We previously looked at game theory, where the setting involves a set of players playing a "game" which has well-defined outcomes. The players all have their evaluation of what outcome suits them the best (which is captured in their utility functions), and they are assumed to be rational (i.e. they operate in a way so as to maximize their utility functions).

Mechanism Design essentially asks the opposite question: If there is a result that you want to achieve, can you (and how, if yes) design a game that can get you that result? For a relevant example, in cryptography, the result of the mechanism is generally privacy.

Example 1. *Suppose there is a cake, and the parent wants to divide it between the two children. The parent wants to achieve a "fair" division, in the sense that neither child has a reason to complain or be envious of the piece of cake that the other child gets. Obviously, if the parent knows what the children like, it is trivial. So assume not. How do you think you can ensure a fair division?*

The answer is surprisingly neat: The parent asks one of the children to divide the cake into two pieces, and asks the other child to pick first. Intuitively, you can see that no child would have a reason to complain.

2.2 Setting

As in Game Theoretic settings, there is a set of players $N = 1, 2, ..., n$, but there is also an entity, who we call a Mechanism Designer, who designs and executes the game. The Designer themself does not participate in the game. Each player has their own preferences, which are captured by their *types*. Types are modelled as follows: Each party i has type $\theta_i \in \Theta_i$. Θ_i's are known to the Mechanism Designer, but θ_i's remain unknown. For example, the players' valuations of a good in an auction. They are not known to the Mechanism Designer, whose job here is to design an auction.

Like in Game Theory again, there are outcomes $o \in \mathcal{O}$, and there are utility functions, which now not only depend on the outcome, but also the type of each of the players: $u_i(\theta_i, o) : \Theta_i \times \mathcal{O} \to \mathbb{R}$

Notice there is something missing here; there are no strategies available to the players to play the game. This is where Mechanism Designer comes in. Their goal is to furnish the actions for each of the players so as to meet a goal – some social choice function like privacy, fairness etc.

2.3 Examples

Now we discuss some examples to illustrate the setting in practice.

Example 2. *(Reference) Say A and B are stake holders for a power plant. There are two possible "states of the world", State I and State II. A and B can both either be in State I or in State II. Call the Mechanism Designer to be D. Now, D does not know what the state of the world is, but knows what A and B's preference lists are in each state. Note that A and B know each others' preferences. This is given in Table 1:*

	State I		State II	
Utilities	A	B	A	B
4	Gas	Nuclear	Nuclear	Oil
3	Oil	Oil	Gas	Gas
2	Coal	Coal	Coal	Coal
1	Nuclear	Gas	Oil	Nuclear

Table 1: Preferences of A and B in each state

Now, to design the mechanism, D gives A the action set $S_A = \{Up, Down\}$ *and B the action set* $S_B = \{Left, Right\}$ *and the game:*

A/B	Left	Right
Up	Oil	Coal
Down	Nuclear	Gas

Mapping each others preferences, we can analyze that $(Up, Left)$ *is a Nash Equilibrium in State I, and* $(Down, Right)$ *is a Nash Equilibrium in State II. So A and B when they play the game end up choosing Oil in State I and Gas in State II.*

2.4 Types of Mechanisms

Two types of Mechanisms can be studied based on whether Players' types are revealed or not to the Mechanism Designer:

1. Direct Mechanism (Players' types are revealed to the designer)
2. Indirect Mechanism (Players' types are unknown to the designer)

Intuitively, it seems that indirect mechanisms are more powerful than direct mechanisms. But Revelation Principle roughly says that any indirect mechanism can be achieved by a direct mechanism, implying that it is enough to study direct mechanisms.

In the next lecture, we will turn our attention to various common mechanisms like auctions, fair division, matching, elections, etc.