The Economics of Cybersecurity — Lecture 11 Notes

Adam Hastings

April 2, 2024

1 Introduction

We are going to touch on Mechanism Design, cover some of the well-known and celebrated results from this field, and then discuss how we can apply mechanism design to problems facing security.

2 Game Theory Recap

2.1 Backwards Induction

3 Mechanism Design

I'm going to lecture a bit on mechanism design and build up the paradigm and the motivation. Later we can get back into the actual paper (which is on control systems).

3.1 The Divide-and-Choose Mechanism

- The paper begins with a description of the divide-and-choose mechanism. Can someone tell me what this is? (A mechanism for creating a fair split)
- We learned game theory last week. Let's write this out as a game:
  - Players: $\mathcal{I} = \{1, 2\}$
  - Actions: $A = \{\text{cut} \mapsto p \in \{l, r\},\ \text{choose} \mapsto \text{argmax}_p(S(l), S(r)),\ \text{receive} \mapsto \text{argmin}_p(S(l), S(r))$ (where p is a piece is $S(p)$ is the size of the piece)
  - Utility: $u_i = S(p)$
- Does this work for more than two players? What if we wanted to split a cake three ways?
  - I could cut a cake into three pieces. What does this look like as a game? (TODO outcomes?)

- $*$ Players: $\mathcal{I} = \{1, 2, 3\}$
- $*$ Actions: $\mathcal{A} = \{a_1 = \text{Choose piece } 1, a_2 = Choosepiece2, a_3 = Choosepiece3$
- $*$ Utility: $u_i = \text{size}(a)$

3.2 Auctions

Note: Much of this section on auctions is inspired by and borrowed from Tim Roughgarden's class notes on Algorithmic Mechanism Design—a class here at Columbia! Go take it if this topic interests you.

Auctions are a classic area of study in mechanism design. Auctions are one of those things that may seem so simple you don't need to study them at all, but it turns out that there are many different types of auctions and many reasons why you might choose form over another.

We're going to go over auctions first to help us develop a taste for what mechanism design and what its uses are before we try to tackle the mathematical definition of it.

3.3 Simple Auctions (English auctions or open outcry auctions)

Let's start with an open-air auction at, say, and art gallery:

- Bidders will place bids that are monotonically increasing. There is usually minimum "bid increment" of about 10%, so that if the previous bid is \$100, you have to bid at least \$110, not \$100.01.
- The bidding repeats until a highest bid is placed. The person who places the highest bid pays the value of the bid and receives the item.

From a computability perspective: What is the complexity of this algorithm? It is $O(n)$, where n is the number of bids[1]. If you're in an auction house, maybe that is tolerable, but what if you're in a situation where time is critical?

Ask: Does anyone know of any real-world situations where the runtime of an auction is of critical importance? Online advertisements. Ad personalization is a huge part of the online economy, and pays for a large portion of the tech economy. But did anyone know that it's all based on auctions? Google or Facebook provide your data (age, sex, search history, etc.) and then advertisers place bids based on your data to determine which advertisement you see. This all happens on the order of milliseconds. This can't rely on a $O(n)$ process if you consider the relative slowness of network connections. It relies on a $O(1)$ auction[2].

[1] Technically $O(n)$ transactions, but then computing the maximum bid is $O(1)$

[2] Technically you still have to find the highest bid, which is $O(n)$. But the number of sequential transactions between auction players and auction house is now $O(1)$

What are some of the other drawbacks of this auction design? It is very public. Even if the auction is done anonymously, you are revealing to the market what your valuation of goods is; if the same good or type of good (like ads) is auctioned repeatedly, this public information can be used against you.

Ask: If you're an auction designer, how can you design an auction that doesn't require a back-and-forth $O(n)$ algorithm? And perhaps one that isn't quite so public? (Answer: Students will probably suggest a first-price sealed auction, a.k.a. a "silent auction")

3.4 First-price sealed bids

This is another type of auction commonly seen in fundraising events: People interested in the good write their bids on a piece of paper, seal it in an envelope, and drop it in a basket; the person who writes the highest bid pays the amount on the bid and receives the good.

What are the available strategies?

1. You can bid the amount of how much you value the auctioned good. E.g. if you value the good at $100, you could place your bid at $100. *Is this a rational strategy?*
    - No. If your valuation is $100, then you are indifferent between having $100 or having the good. If you were to win the auction with this bid, your net utility would remained unchanged.
2. You can (and should) bid less than your valuation of the good. *Ask: what's the challenge here?*
    - It's difficult to compute how much less than your valuation you should bid, because this relies on knowing information about the valuations of all the bidders you're bidding against. You're forced to do a Bayesian analysis of the game and—based on your estimates of the other players—you ideal strategy is to bid the smallest amount possible that you think will still win you the auction.
    - An extension of this is that it's likely that, in the end, the auctioned good goes to a bidder whose valuation was *not* the highest out of the bidders, if they happened to do their calculations differently. This is an economic inefficiency! In an ideal auction, the good would go to the person who has the highest valuation of the good.

3.5 Vickrey auctions and Generalized Second Price auctions

In terms of game theory an mechanism design, you are forced to engage in strategic bidding. *Ask: Imagine you are auction holder, like Google or Ebay or Sotheby's. Does anyone know of an auction "game" you can design that eliminates the problem of strategic bidding?*

A commonly-used auction design—one that has been used for centuries—is the *sealed-bid second* *price auction*, also known as the *Vickrey auction*[3] In a Vickrey auction, everyone submits one bid ($O(1)$ transactions) the price goes to the person who submits the highest bid, but the amount they pay is the price of the *second-highest bid.*

How does this change the stragies of the players? I'm not going to go through the proof—but there is a proof—that this auction design has a dominant strategy (*Someone remind me—what's a dominant strategy?* One that is optimal regardless of all the other players' actions). And the dominant strategy, believe it or not, is to simply bid your true valuation!

The benefits over the sealed-bid first-price auction:

- If you are a bidder, your strategy is very simple! Just bid your true valuation. You don't need to estimate other player's strategies or valuations
- It is incentive compatible. In fact, because there is a dominant strategy, this "game" has a special name in game theory: It is a **dominant strategy incentive compatible game**, or **DSIC** game.
- The auctioned good always goes to the bidder with the highest valuation.
- If you're the highest bidder, you certainly walk away happy—you end up paying less than your valuation! You end up with positive surplus utility.
- Computationally efficient (1 transaction from each bidder, finding two highest bids is $O(n)$).
- Privacy-preserving (same with first-price sealed bid auction).

This is not just academic hot air—these are used in the real world:

- Facebook used a Vickrey-Clarke-Groves auction, which is a generalization of the Vickrey auction that extends an auction of one good into an auction of multiple goods.
- Google used a Generalized Second Price auction (similar to a Vickrey auction with a few modifications)
- I read somewhere that due to "trust issues" among advertisers there has been a switch to different types of auctions (source: https://www.cmu.edu/tepper/news/stories/2020/april/display-advertisting-research-ravi.html) but I haven't really been able to corroborate this (I found conflicting information when I looked).

Pretty cool, right?

[3] named after William Vickrey, professor of economics here at Columbia who was the first to describe the mathematical properties of the sealed-bid second-price auction. For this and other work on incentives under asymmetric information, Vickrey was the co-recipient of the 1996 Nobel Memorial Prize in Economics. Fun fact—William Vickrey is also regarded by many to be "the father of congestion pricing", which, if you've been following the news, is a politically hot topic as Governer Hochul has recently approved congestion pricing for New York City south of 60th st.

Do you understand why mechanism design is called "reverse game theory?" The "game" of mechanism design is *designing a game* (like an auction) whose rules lead strategic players to choose actions that improve overall welfare and utility.

Any questions?

3.6 Mathematical Definition

The definition of a mechanism is similar to that of a game:

- A set of players $\mathcal{I}, |\mathcal{I}| = n \in \mathbb{N}$
- A set of outcomes $\mathcal{O} = \{o_1, o_2, o_3 ... o_k\}$
- Each agent $i \in \mathcal{I}$ has private information $\theta_i \in \Theta_i$
  - Since private information may affect the player's strategies i.e. decisionmaking, we call θ_i the player's *type.*
- "We write $(\theta_i)_{i \in \mathcal{I}} = \theta, \theta \in \Theta$, where $\Theta = \Pi_{i \in \mathcal{I}} \Theta_i$ to represent the type profile of all agents"
  - WTF?
  - Let's break this down—recall from our game theory discussion that "action profile" is the set of possible combinations of actions that players can take.
  - So intuitively, we should expect that the "type profile" is the set of possible combinations of possible types that players can be
  - Maybe the Π is throwing you off—you've seen Σ in math before, maybe even Π in algebra—it's the product of terms rather than the sum.
  - We're dealing with sets here, so Π is Caretsian product of i types θ.
- Each agent i's preferences represented by a utility function $u_i : \mathcal{O} \times \Theta_i \to \mathbb{R}$
  - Intepretation: player's utility is dependent on two things—the outcome of the game, and the player's type. The Cartesian product means there is a utility score for each possible combination of outcome and type.
- Utility is assumed to be of the quasi-linear form

  $$u_i(o, \theta_i) = v_i(o, \theta_i) - p_i(\theta_i)$$

  where $v_i : \mathcal{O} \times \Theta_i \to \mathbb{R}_{\geq 0}$ represents an arbitrary valuation function and $p_i \mapsto \mathbb{R}$
  - See? Utility $u_i(o, \theta_i)$ is a combination of outcome and type (just like the it says above).
  - Quasi-linear because while the functions p and v may be differential equations, at this level it looks like a linear equation.
  - Any time you see a p and a v in economics, your first thought should be "price" and "value"

- If $o \in \mathcal{O}$ is an outcome, then p_i can be thought of as the cost (or price) imposed upon player i for that particular outcome o.
- *Do we understand why the function v depends on the type θ? What about the price p?*
- *Ask: the value v is dependent on the outcome o but not the price p. Why?*

- The social welfare W can be defined as the collective summation of all agents valuations, that is
$$SW(o, \theta) = \sum_{i \in \mathcal{I}} v_i(o, \theta) \tag{1}$$

- The goal of the players is to maximize u_i. But the goal of the social planner or mechanism designer is to maximize $SW(o, \theta)$!

- If you're a social planner, in order to compute (1), you need to know each $v_i(o, \theta_i)$. I think you can assume that, because the rules of the game are known, then the social planner knows how to compute v_i if they know the outcome o and player type θ_i.
  - What's the problem here? The outcome can't be misreported but the player can lie about their player type θ_i.
  - In fact, in many games there is an incentive to lie about your type. E.g. in auctions like a sealed-bid first price auction, you can gain an advantage if you lie to the other bidders and make them think that you hate the art being auctioned, making them lower their bids, and then you can more easily outbid them.

- The question then becomes, *how can be incentivize agents to report their type truthfully?*. As the authors state, the answer is through the appropriate design of the price function p_i.
  - *Why does the social planner modify p_i instead of v_i?* v_i is players' personal valuation. A social planner can't change the amount of value I get out of some good or some outcome (like an apple) but they can make the desirability of buying apples either higher or lower depending on the costs of acquiring apples.

- Now we get to the definition of a mechanism: A mechanism is a tuple $\langle f, p \rangle$.

- p is (like I mentioned) a vector of payment functions $p = (p_i)_{i \in \mathcal{I}}$, with $p_i : \Theta \to \mathbb{R}$.

- f is a *Social Choice Function (SCF)* $f : \Theta \to \mathcal{O}$
  - Recall—Θ is the space of all possible combinations of player types. Don't worry if you can't fully wrap your head around what's going on. Just know that there's a function f that maps from the space of possible player types Θ to an outcome $\mathcal{O}$, and a vector of price functions that for each player maps from Θ to $\mathbb{R}$.

- "In other words, a mechanism $\langle f, p \rangle$ defines the rules by which we [the social planner] can implement a system objective by mapping the agents' types to an outcomes (by means of the SCF) while using the payments to ensure the optimality or efficiency of that outcome"

- The social planner's problem is
$$\max_{o \in \mathcal{O}} SW(o, \theta) \tag{2}$$
(i.e. find the outcome that maximizes social welfare)

subject to:

$$\hat{\theta}_i = \theta_i, \forall i \in \mathcal{I} \tag{3}$$

where $\hat{\theta}_i$ is the reported type. (Interpretation: the outcome that we're trying to find that maximizes social welfare also has to incentivize players to not lie about their type).

Subject to:

$$\sum_{i \in \mathcal{I}} v_i(o, \theta_i) \geq \sum_{i \in \mathcal{I}} v_i(o', \theta_i), \quad \forall o' \in \mathcal{O} \tag{4}$$

Interpretation: the outcome is efficient, i.e. produces higher utility than all other outcomes that fit the constraints

Subject to:

$$\sum_{i \in \mathcal{I}} p_i(s(\theta)) \geq 0, \ \forall \theta \in \Theta \tag{5}$$

where $s(\cdot)$ is the equilibrium strategy profile (for example, a Nash Equilibrium).

(Interpretation: The price p of behaving optimally is greater than 0 for all possible player types)

Subject to:

$$v_i(f(s(\theta))) - p_i(s(\theta)) \geq 0, \ \forall i \in \mathcal{I}, \ \forall \theta \in \Theta \tag{6}$$

Interpretation: The value of participating in this game minus the price of participating in the game is ≥ 0 for all players regardless of what player type they are.

I am not an optimization theorist but the authors claim that if the social planner knows the true types of all players, they can solve this problem using standard optimization techniques. But as was mentioned earlier, the social planner may not know all the player types, who we can expect to be strategic and selfish, so the social planner needs to *elicit* θ from each player by designing the right price function p.

- The social planner has to answer two questions: The *preference aggregation*, which asks what is the best outcome $o \in \mathcal{O}$ for any give type profile $\theta \in \Theta$, and the *information elicitation*, which asks how one can extract truthfully the type $\theta_i \in \Theta$ of any agent $i \in \mathcal{I}$.
- "The theory of mechanism design essentially helps us answer both questions by providing the mathematical framework to construct mechanisms $\langle f, p \rangle$ that can achieve our desireable outcome"
- Example: In the three-person cake-cutting, the social planner (the parent) designed a protocol that incentivizes each child to be truthful in how much they value each piece of cake. Recall that in the naive chocolate bar implementation, the rules of the game made it so that the the first child cutting could be dishonest about their type, i.e. dishonest about how much they actually valued each of the initial three chocolate pieces.

3.7 The Revelation Principle

The challenge, obviously, is to design the functions f and p that achieve what we want. Unfortunately, in terms of theory, this is where our train stops. There are many more stops along this route if you're like to learn more; there is a large body of literature on how to algorithmically find mechanisms.

Just a teaser:

Technically the above is the definition of a **direct mechanism**, meaning that players' types are observable. Some mechanisms are indirect, meaning that types can only be inferred indirectly. The **revelation principle** says that all indirect mechanisms can be implemented in a direct mechanism, meaning that any indirect mechanism can be modified in such a way to achieve the same result but while being *incentive compatible*.

This is one of the major results of mechanism design and makes it computationally much easier to actually find mechanisms since you only need to consider at the incentive compatible outcomes.

If you want to learn more about algorithmic mechanism design, take Tim Roughgarden's class.

4 Mechanism Design in Behavioral Economics

Recall my paper "How much is performance worth to users?", where I measured users' willingness to accept performance losses on their personal devices.

The challenge was that I wanted to measure how much people valued performance, but I could not trust "player" (i.e. research participants) to tell me their "type" (how much they valued performance) without adjusting the price function. What was the price function that I imposed on the game players? The price was that I actually slowed down their computers!

4.1 Single Discrete Binary Choice Mechanism

Specifically, the mechanism that I used was called the Single Discrete Binary Choice Mechanism. Essentially a take-it-or-leave-it option. Can we confirm that it satisfies the above constraints for mechanisms?

Was I trying to optimize social welfare? Technically no. So was this a true mechanism design? No, not really, I wasn't trying to design a mechanism, I was trying to find out their types! But most the other conditions held:

- Players revealed to me their true type
- I wasn't really concerned that the outcome I chose (assignment of offer prices to participants) maximized their values
- The prices that were imposed on them were greater than zero
- And there was an incentive to play this game because the value from playing their optimal

strategy minus the price of playing their optimal strategy was greater than zero if they accepted the offer, and zero if they declined the offer.

I was an indifferent mechanism designer, I suppose.

4.2 Becker-DeGroot-Marschak lotteries

An incentive-compatible way to measure willingness to pay or (willingness to accept).

Suppose you want to know how much some good is worth to people. You can't ask people to name their price, since they're going to lowball you. And you can't name the price, because the point is not that you're trying to sell something but figure out what is the most they'd be willing to pay for the good (i.e. in economics research).

The BDM lottery protocol is as follows:

1. The buyer submits a bid b for the maximum they'd be willing to pay for the good
2. You draw a random number r from a uniform distribution (where the high end of the distribution is assumed to be well above the max WTP, I think)
3. If $b < r$, the buyer pays nothing and receives nothing
4. If $b > r$, the buyer pays b and receives the item.

What are the strategies?

- The buyer should never bid a b higher than their true WTP because this means they will pay more for the good than it's worth to them if $b > r$.
- The buyer should never bid a b *lower* that their true WTP because this proportionally decreases the chances of receiving the good that they truly would be happy paying their WTP to receive.
  - Example: if $0 < r < 100$, and I value some good at \$50, I'm not going to give a lowball bid of \$1 because this essentially eliminates my odds of receiving the item.

What's the result? The optimal strategy as a buyer is to submit a bid b that equals your WTP! It's incentive compatible. The rules of the game incentivize you to act in a way that reveals your true type.

5 Mechanism Design for Hardware Security

(Coming back down to earth—and security—now)

This is not a peer-reviewed paper. This is a report from a workshop event a couple of summers ago, where me, Prof. Sethumadhavan, and about 50 other people from academia, industry, and government got together and discussed what mechanism design means for hardware security.

Aside: For anyone who doesn't know, *Hardware Security* is exactly what it sounds like: It's a subfield of security that focuses on securing systems from the ground up, starting from the circuit level to the physical device level and typically including up to computer architecture. If this is a topic that interests you, and you are not graduating this semester, there is a hardware security class in this department, taught by Prof. Sethumadhavan. (Today's class is really just an advertisement for other graduate-level classes at Columbia, isn't it?)

The workshop was from people who work in the hardware security area. I.e. topics like side channel detection and exploitation, hardware trojans, speculation safety, Rowhammer, hardware-based defenses like CFI, secure embedded systems, tamper-proof hardware, hardware enclaves and trusted execution environments (*to Simha: any others I'm missing?*) To prevent an echo chamber, there were also some people from economics and a few people who worked in government policy (next week's topic).

Ask: Why did I have you read a workshop report rather than a peer-reviewed paper on mechanism design for security? Mostly because there is no work on this topic! This is an emerging research area. The purpose of this workshop is to spur research in this direction. But mechanism design is such an interesting tool and one that security practitioners ought to have in their toolbox. It's economics applied directly to improving security. Extremely relevant to this class, (hopefully) extremely important to the field of security moving forward.

5.1 Recommendations

The format of the event was a mix of presentations and discussion, with the end goal of coming up with a set of recommendations for what the hardware and computing communities can do to improve hardware security through mechanism design.

These were the recommendations we came up with *(write on board)*:

5.1.1 Foster Diverse Educational, Professional, and Industrial Communities in Hardware Security

5.1.2 Lay the Scientific Foundations for Work that Combines Incentives and Technology

5.1.3 Make Security Accountable and Explainable

5.1.4 Co-Develop Emerging Technologies with the Understanding of their Hardware Security Ramifications

5.1.5 Prioritize the Human Impact of Hardware Security

5.2 Discussion

What do we think? Did we get it right? What did we miss? What did we get wrong? *Write recommendations on board.*

One thing I'd like to point out is that you should not assume that just because the audience of people who contributed to this list was a bunch of established professors that their level of wisdom or insight in the topic is much better than yours. Don't think of this list as infallible, and don't think of your intuitions as invalid. Like I said, this is an emerging area, so someone who has spent all their life working in security may barely have a leg up on you (if at all!) when it comes to taking an economic perspective at security problems.

Ask Simha: In retrospect, what changes would you make to the set of recommendations?

5.2.1 What does this have to do with auctions, study design, etc.?

Very little! I think there is massive potential in applying mechanism design to problems of security and security policy.

Open ended discussion: What are some areas in security where we can apply what we've learned here today to solve problems in security?

- Are there mechanisms we can use to improve patching practices?
- Are there mechanisms we can use to get the right people to invest in security in the right places?
- Can we use *liability* as a mechanism to balance the costs of security?
  - (Remember my Doctrine of Shared Burdens?)

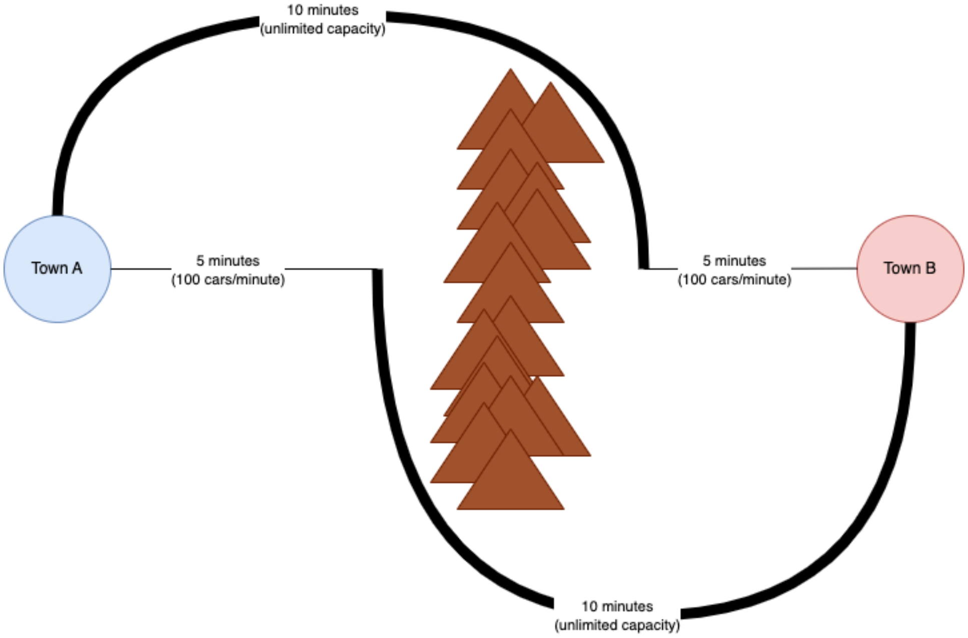

Figure 1: Before the tunnel, it takes 1200 cars 14 minutes to travel from Town A to Town B

- (Seems like this is happening behind the scenes in Washington—based on statements and reports, there's clear interest in getting software vendors to take on greater liability for releasing software that is full of vulnerabilities.
  - *What are some of the challenges here?*
  - Most software is a tangled web of dependencies. Who heard about the xz vulnerability over the weekend? Who should be responsible if your open source dependency leads to a vulnerability in your software? (You? The open source programmer? Someone else? Who?)

6 Braess's Paradox

Ask: What do we think—do we need a social planner (i.e. government) to come in and solve all our problems?

I want to point out that if you don't fully understand the system you're modifying, you can very easily make things worse.

Imagine there are two town, separated by a mountain range. There is one highway to the north, and one highway to the south. Connecting the highways to the towns are smaller country roads. The highways take 10 minutes to drive, and the country roads take 4 minutes, so no matter on which route you take, it takes 14 minutes to drive from Town A to Town B.

Suppose 1200 cars drive from Town A to Town B every hour.

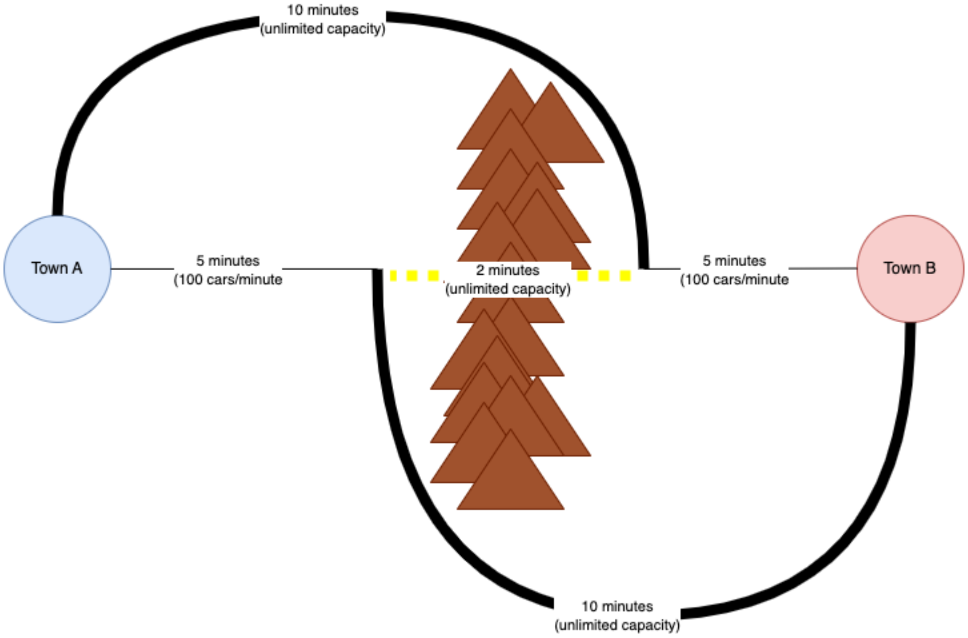

Figure 2: A naive urban planner might expect the tunnel to reduce the drive time to 10 minutes

Let's look at this as a game from the persepctive of the players.

- *What is the number of players?* $\mathcal{I} = \{1, 2, \ldots 1200\}$.
- *What is the set of actions available to the players?* Take the north route or take the south route: $a = \{\text{North}, \text{South}\}$.
- *How much utility do players get from either strategy?* In both cases the players are trying to minimize negative utility (lost travel time). We can write this as $u(R_N) = u(R_S) = -14{:}00$.
  - *Why is the utility negative?* Negative utility represents a bad thing. In this case, the bad thing is travel time. So the longer the drive time is, the greater the negative utility is, and the worse it is.

So what is the strategy? The strategy is to maximize utility, i.e. choose $\text{argmax}_a(u(a))$.

Since the utility of both is equal, we can assume that 50% of players will choose the north route and that 50% of players take the south route. In terms of game theory, we call this a *mixed strategy* since strategies are probabilistic in nature. Just in case you end up reading or working more on game theory later on and you see the term *mixed strategy*, that's all it means.

Let's say you're an urban planner, and you just been hired by the Inter-Town Transit Authority to reduce the travel time between the two towns. What might you do to reduce travel times? You could dig a tunnel through the mountains.

Does anyone see a problem here? (If a student notices: We're going to incentivize all the traffic to go on the small country roads)

Let's suppose that you, the urban planner, did a little bit more research on the roads before digging the tunnel, and you found that the country roads have a limited capacity. Suppose they can only allow 12 cars/minute.

Recall that there are 1200 cars per hour. Since there are two routes, and they're identical, we can assume that every hour 600 cars take the north route and 600 cars take the south route per hour, or 10 cars per minute.

$$\frac{600 \text{ cars}}{\text{hour}} = \frac{10 \text{ cars}}{\text{minute}}$$

Can our country roads handle this capacity? Yes. In other words, the travel time is unconstrained by the number of cars. We can write the travel time of the country roads as

$$t = \max(4, 4n/12)$$

where n is the number of cars per minute.

TODO draw graph and explain!!

Any questions?

What happens if *every* car takes the tunnel? I.e. instead of 10 cars per minute, the country roads now have 20 cars per minute? The travel time for each stretch of the country road becomes

$$t = \max(4, 4(20)/12) = 6\text{:}40 \text{ minutes}$$

What happens if we dig the tunnel? In the best case the travel time will be 10 minutes, if there were no other cars on the road. But if everybody takes the tunnel, the travel time will be 6:40 + 6:40 + 2:00 = 15:20 minutes. This isn't any faster than the old route, which was 14 minutes.

But what is the time of the old route now? 6:40 + 10:00 = 16:40. Taking the highway is still the slowest option, but it too is slower than before!

Let's look at the game.

- $\mathcal{I} = \{1, 2, \ldots 1200\}$
- $a = \{\text{North}, \text{South}, \text{Tunnel}\}$
- $u(R_N) = -16\text{:}40, \quad u(R_S) = -16\text{:}40, \quad u(R_T) = -15\text{:}20$

Recall that the strategy is to maximize utility. The highest utility is achieved by taking the tunnel. But note that this is no longer a mixed strategy. This is a dominant strategy! Every player is going to take the tunnel.

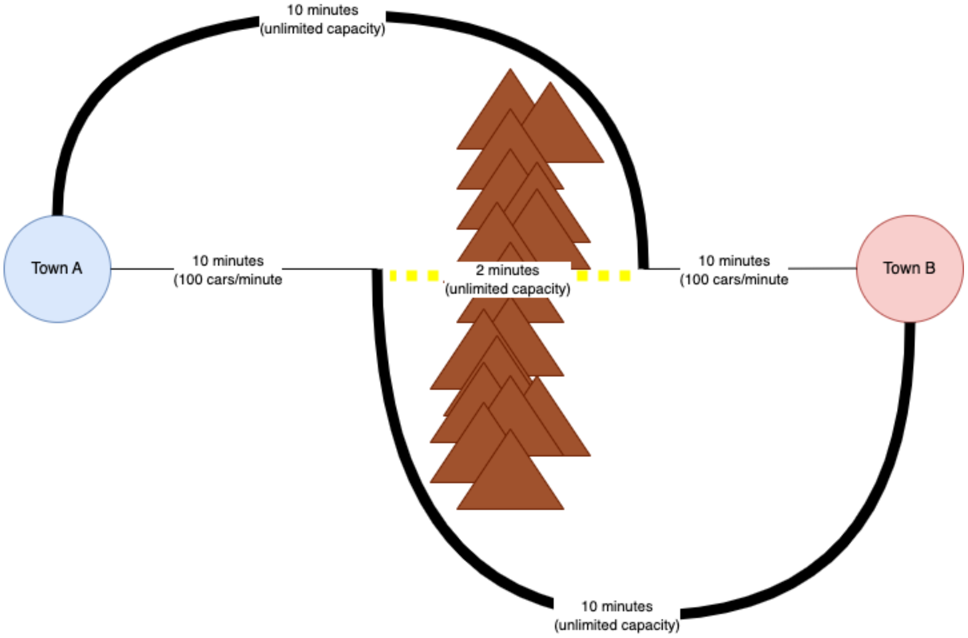

Figure 3: A naive urban planner might expect the tunnel to reduce the drive time to 12 minutess

Can someone tell me the plain-English interpretation of what I just said? We built a tunnel. The tunnel incentivized everybody to take the tunnel. But because of the road capacity issues, this route ended up being slower than the old routes around the mountains. But the tragedy is that the old routes now also became slower as well! You can't get from Town A to Town B in 14 minutes anymore.

Note that the old roads still exist! Everyone could cooperate and have 50% take the north road and 50% take the south road. But that's really hard to enforce! Because then what happens? That tunnel is still there, and is just too tempting—If you're the only one taking the tunnel, your travel time is 10 minutes!

6.0.1 Price of Anarchy

In economics, game theory, and mechanism design, there is a concept called the *POA*, or *Price of Anarchy*. This is a quantitative measure of the overhead imposed by agents choosing selfish behavior, and is defined as the ratio between the Nash equilibrium outcomem and the socially optimal outcome. In this game, the socially optimal outcome was -14 minutes while the equilibrium outcome was -15 minutes 20 seconds. *(Note that the socially optimal outcome was -14 minutes even after the tunnel is built).* So we have

$$PoA = \frac{\text{15:20}}{\text{14:00}} \approx 1.095 = 9.5\%$$

In other words, the system is 9.5% worse off because the rules of the game allow for selfish behavior which comes at the expense of the other drivers.

In the prisoner's dilemma, we had a (-1,-1) payoff for (cooperate, cooperate) and a (-2,-2) payoff for (defect, defect). The social cost of (cooperate, cooperate) is 2 and the social cost of (defect, defect) is 4. This is a 4/2 = 2X = 100% price of anarchy. *Any questions?*

Ask: In the prisoner's dilemma, who is the social planner? The party who has a degree of influence or control over the players, so in this case, the mob :)

As a fun fact, the term *Price of Anarchy* comes from a 1999 paper "Worst-case equilibria" by Elias Koutsoupias and our very own Christos Papadimitriou (although the concept of measuring the inefficiency of selfish behaviour predates this paper by quite a bit. But within the algorithmic game theory and algorithmic mechanism design, this is a very commonly-used term.)

6.1 Extending Braess' Paradox to Security

The general concept of what is going on here is called Braess' Paradox. You may have also seen this with two springs and a weight. *(Has anyone seen this before?)* It's a physical analog to this situtation with roads and tunnels. It's counterintuitive but cutting the middle string causes the weight to *rise*, not fall. The reason is because with the middle string, the two springs are acting in series, and both springs are bearing the full weight, whereas once the middle string is cut, the two springs are acting in parallel, meaning that they each are only holding half the weight.

In traffic engineering and urban planning, it's often explained in terms of roads between two cities.

6.1.1 Discussion

What are the takeaways?

- If we're going to meddle with systems, we have to be very confident that we fully understand the system we are meddling with. Otherwise we can make things much worse. I suppose this is a mathematical example of a perverse incentive.
- *Any others*?
- The takeaway that I've led you to thus far is that a misguided planner can make things worse. Equivalently, a smart planner can make things better by doing counterintuitive roads. Imagine that you're the city planner hired to improve travel times between Town A and Town B, *and the tunnel already existed*. How do you think the public would react if your solution is to *close the tunnel*?
  - Sometimes you need the political will and determination to do things that are beneficial but unpopular.
  - *In general though, how do you know if you're the smart planner who is going to make things better, or the naive planner who is going to make things worse??* I don't know. Would love some answers to this question!

7 Control Systems

The first paper I had you read was a tutorial for mechanism design, but was published in an academic magazine for control engineers. Why did I have you read this one?

1. Believe it or not, I think this is one of the more readable and gentle introductions to the topic.
2. The focus was on control engineering but they were kind enough to include some examples of mechanism design applied to security games.
3. Control engineering is not something that most computer scientists are exposed to and is not something that most people in security are even aware of but in my opinion it is a useful toolbox for approaching systems problems (of which security is one).

7.1 What is control engineering?

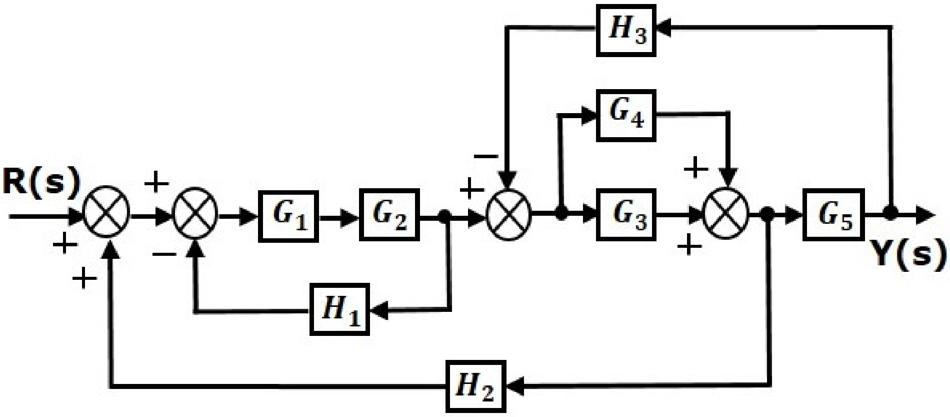

Figure 4: Example of a control systems problem (source: https://www.tutorialspoint.com/control_systems/images/reduction_diagram.jpg)

This is, in many ways, very similar to mechanism design, and so I wanted to expose you to it so that it can sit in the back of your mind when you think about security problems. Control systems engineering is typically going to be found in electrical and mechanical engineering, while mechanism design is going to be found in economics and computer science departments, but they have many of the same goals and are worth knowing about:

Similarities:

- Shared goal of mathematically describing a complex system with feedback
- Shared goal of designing ways to interact with and modify the system in a way that achieves some desireable outcome.

Differences:

- Control systems engineering lacks a notion of agents and in particular, strategies.
- Continuing this theme: Control systems engineering is primarily used when dealing with the physical world, where problems like "what if the game players lie to each other or to the rule maker" don't come up.

7.2 Control systems engineering and systems thinking

What does this look like? (A system diagram! Remember HW1?)

Ask: Why don't security practitioners or policymakers apply control systems theory to security problems? Something to think about. Some possible explanations:

- Is it because the "signals" in security are too noisy? (Recall: there is a general lack of information in security, because there are incentives for companies to not be open with others about the details of breaches and attacks).
  - Possibly. But control systems engineering is speficially all about how to maintain stability in a system that is subject to random and unpredictable noise and perturbations.
  - Counterpoint: There are mathematical notions of stability in control systems engineering, i.e. your system can remain in this stabilized position if the size of the perturbations are bounded by some amount; maybe given the strength of the signals we are working with and the scale of attacks and the *adversarial nature*, it is not feasible to apply control systems engineering principles to security.
- Perhaps it's not just the signals that are the problem. Perhaps due to the lack of information it's the functions between the signals that are unknown.
  - Example: If you're designing the shock system on an automobile, you know that perturbations to your system are due to variations in the road (things like potholes) and are going to be in one direction only (up and down). Your modeling is going to account for the functions you know of because you're very confident that these are the functions that are going to impact your real-world system, andmain assumption in this approach is that the interactions among strategic agents in a system in which securit you're probably going to be right about it. Your shock system is not meant to dampen forces from a lateral blow. But in security, you don't know which direction the blows are going to come from (omnidirectional and multidimensional). Thoughts?

7.3 Mechanism Design for Security Games

The Design of Security Games — writes out how security games we talked about last week can be mathematically expressed first as a game (which we did last week), and then as a mechanism design problem, i.e. what a social planner is trying to optimize. The challenge (unspecified how to do it) is to find the right price functions for all agents to incentivize participation.

Security in Communication Systems — examines how mechanisms like auctions have been used in allocating resources

Ask: What types of security goods are auctioned off? What types of security goods should be auctioned off?

8 Homework

I'm going to release the homework shortly.

I want you to revisit your systems diagram from Homework 1 and, with the information that you have acquired from this class, try to take a mechanism design and policy approach towards improving the outcomes of your system.

If you were not present for the first week of class and didn't submit Homework 1 as a result, this is your chance to go do it (it's pretty short) and submit it penalty-free.

Still working on details but I'm going to have you analyze the effects of various interventions onto your system.