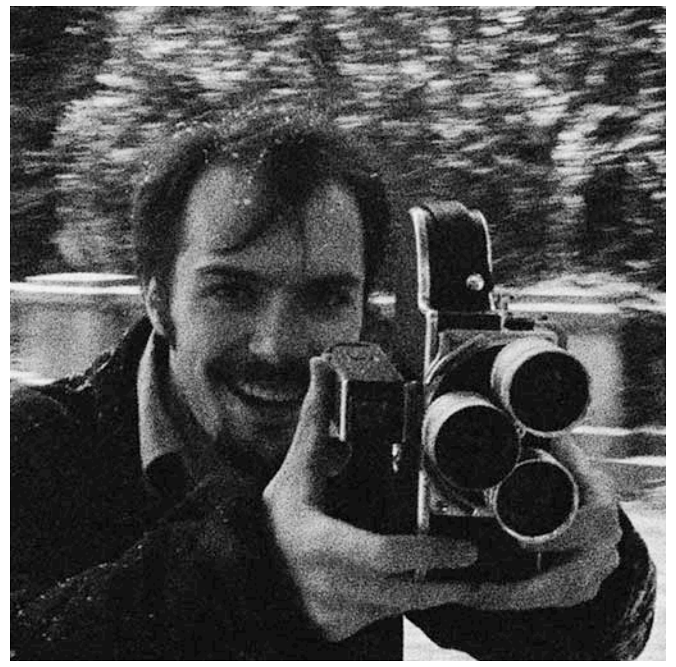
CS4501: Cryptographic Protocols

Instructor
Jack Doerner
jhd3pa@virginia.edu
Rice 106



TA
Jinye He (Clara)
qfn5bh@virginia.edu

https://jackdoerner.net/teaching/#2026/Spring/CS4501

☝ All Course Details Here ☝

What is Cryptography?

Greek: *kryptós gráfein*
English: *hidden writing*

Concise Oxford English Dictionary:
the art of writing or solving codes

This was true until ~1980

Concise Oxford English Dictionary: *the art of writing or solving codes*

A heuristic process: artists use their *intuition* to come up with very clever codes that seem to be secure.

Later, people who are even more clever come along and solve (i.e. *break*) them.

Concise Oxford English Dictionary:
the art of writing or solving codes

A heuristic process: artists use their *intuition* to come up with very clever codes that seem to be secure.

Later, people who are even more clever come along and solve (i.e. *break*) them.

Q: What constitutes a good code?

A: The enemy general doesn't find out when your army will attack.

Concise Oxford English Dictionary:
the art of writing or solving codes

A heuristic process: artists use their *intuition* to come up with very clever codes that seem to be secure.

Later, people who are even more clever come along and solve (i.e. *break*) them.

Q: What constitutes a good code?

A: The enemy general doesn't find out when your army will attack.

Q: What does it mean when a code is broken?

A: The artist wasn't clever enough...

Concise Oxford English Dictionary:
the art of writing or solving codes

A heuristic process: artists use their *intuition* to come up with very clever codes that seem to be secure.

Later, people who are even more clever come along and solve (i.e. *break*) them.

Q: What constitutes a good code?

A: The enemy general doesn't find out when your army will attack.

Q: What does it mean when a code is broken?

A: The artist wasn't clever enough…
…and now you need another code.

Modern Cryptography:

A scientific* discipline:
Formal definitions, rigorous proofs,
precise mathematical assumptions.

*there is still some art. We'll talk about it later.

Modern Cryptography:

A scientific* discipline:
Formal definitions, rigorous proofs,
precise mathematical assumptions.

In order to know if we have achieved security,
we must first know what *security* means!

*there is still some art. We'll talk about it later.

Modern Cryptography:

A scientific* discipline:
Formal definitions, rigorous proofs,
precise mathematical assumptions.

Guarantees absolutely that properties hold without requiring us to enumerate specific attacks or measure system behavior.

*there is still some art. We'll talk about it later.

Modern Cryptography:

A scientific* discipline:
Formal definitions, rigorous proofs,
precise mathematical assumptions.

Often related to important open problems
in math and computer science

*there is still some art. We'll talk about it later.

Modern Cryptography:

A scientific* discipline:
Formal definitions, rigorous proofs,
precise mathematical assumptions.

Q: What constitutes a good cryptosystem?

A: It was proven to satisfy the definition under well-understood assumptions.

*there is still some art. We'll talk about it later.

Modern Cryptography:

A scientific* discipline:
Formal definitions, rigorous proofs,
precise mathematical assumptions.

Q: What constitutes a good cryptosystem?

A: It was proven to satisfy the definition under well-understood assumptions.

Q: What does it mean when a cryptosystem is broken?

A: The assumption was false! A breakthrough in Computer Science!

*there is still some art. We'll talk about it later.

Modern Cryptography:

A scientific* discipline:
Formal definitions, rigorous proofs, precise mathematical assumptions.

A **win-win** proposition. If the assumption is true, the scheme cannot be broken. If the scheme is broken, we solve an important open problem!

A: The assumption was false! A breakthrough in Computer Science!

*there is still some art. We'll talk about it later.

Where is the art now?

Consider the limits of our rigorous methodology:

Choosing the *right* definition is a matter of human judgment.

Proposing mathematical assumptions and proof techniques requires creativity and insight.

The proof doesn't guarantee anything if the implementation differs from what was proven.

These limits also tell us where we can still hope for attacks.

Who uses Cryptography and for What?

Historically:

A: The enemy general doesn't find out when your army will attack.

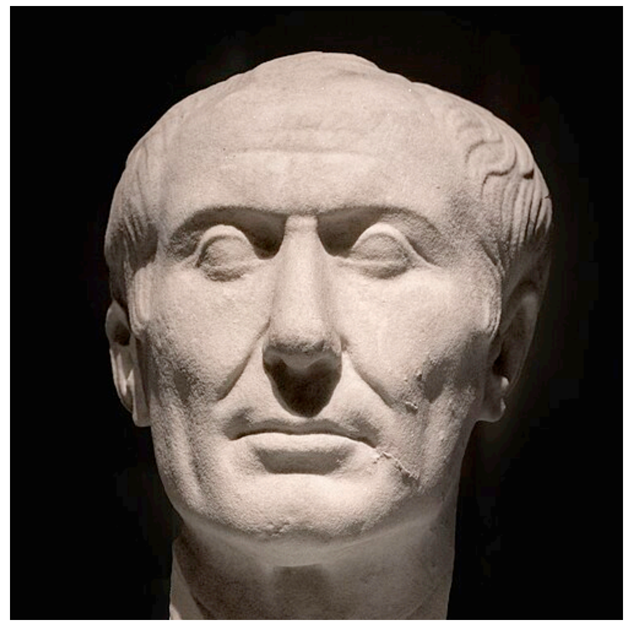
the art of writing or solving codes

Who uses Cryptography and for What?

Historically:

A: The enemy general doesn't find out when your army will attack.

the art of writing or solving codes

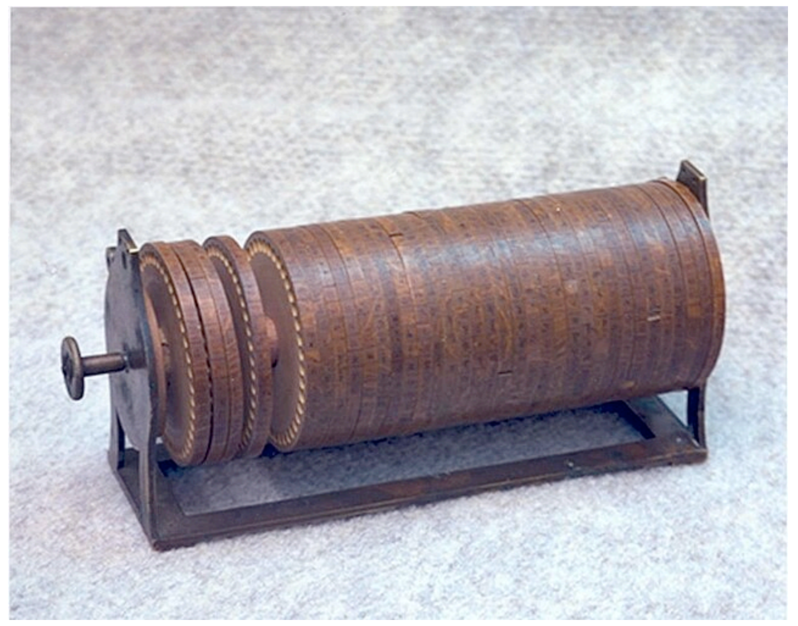


Governments and Militaries. Solves the problem of Private Two-Party Communication with pre-agreed participants.

(above: some historical cryptographers)

Who uses Cryptography and for What?

Historically:



The "Jefferson Disk"

A derivative was used until WWII...

... and then it was broken by the Germans.

Who uses Cryptography and for What?

Now:

Everyone (including you)!

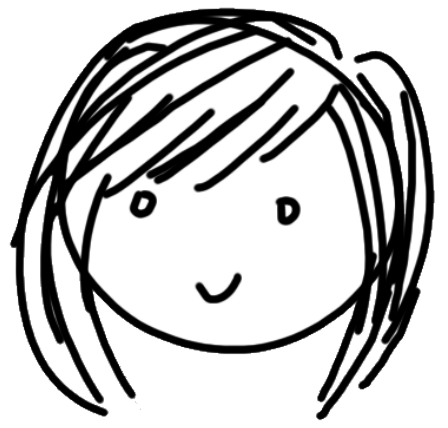
A methodology for rigorously *reasoning about*
and *limiting* the power of an adversary
whenever we interact with others using a computer

This goes far beyond sending secret messages,
but communication is at the heart of it

What is a multi-party protocol?

Simply a set of instructions that allows
a group to achieve a task together in a distributed way

e.g. a two-party cake-baking protocol:



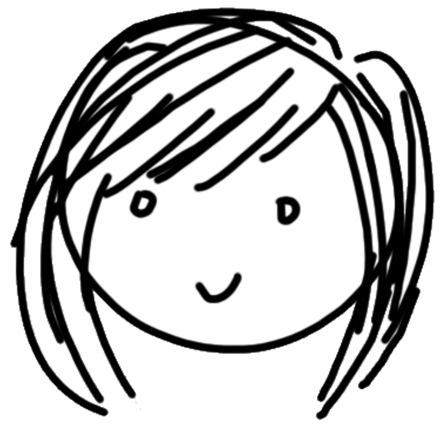
Alice's Instructions:
1. Mix butter and sugar
2. Add eggs one at a time
3. Stir in vanilla
4. Send mixture to Bob
5. Wait 30 minutes
6. Remove cake from oven

Bob's Instructions:
1. Preheat oven to 350°
2. Butter cake pan liberally
3. Combine flour, baking powder, salt
4. Receive mixture from Alice, add to dry ingredients
5. Pour batter into pan
6. Place pan in oven

What is a multi-party protocol?

Simply a set of instructions that allows
a group to achieve a task together in a distributed way

e.g. a two-party cake-baking protocol:



Each party has
inputs and outputs

Alice's Instructions:
1. Mix butter and sugar
2. Add eggs one at a time
3. Stir in vanilla
4. Send mixture to Bob
5. Wait 30 minutes
6. Remove cake from oven

Bob's Instructions:
1. Preheat oven to 350°
2. Butter cake pan liberally
3. Combine flour, baking powder, salt
4. Receive mixture from Alice, add to dry ingredients
5. Pour batter into pan
6. Place pan in oven

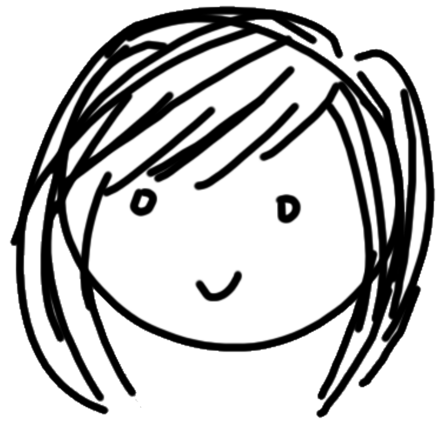
What is a multi-party protocol?

Simply a set of instructions that allows
a group to achieve a task together in a distributed way

e.g. a two-party cake-baking protocol:

Each party has
inputs and outputs

Involves local work



Alice's Instructions:

1. Mix butter and sugar
2. Add eggs one at a time
3. Stir in vanilla
4. Send mixture to Bob
5. Wait 30 minutes
6. Remove cake from oven

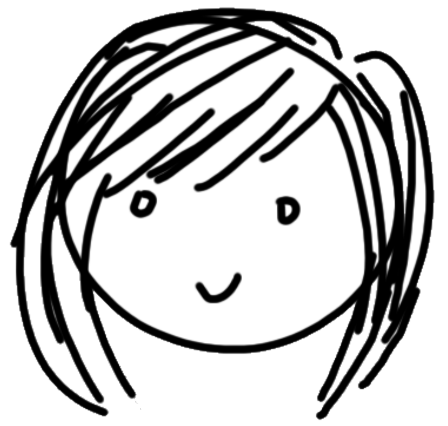
Bob's Instructions:

1. Preheat oven to 350°
2. Butter cake pan liberally
3. Combine flour, baking powder, salt
4. Receive mixture from Alice, add to dry ingredients
5. Pour batter into pan
6. Place pan in oven

What is a multi-party protocol?

Simply a set of instructions that allows
a group to achieve a task together in a distributed way

e.g. a two-party cake-baking protocol:



Each party has
inputs and outputs

Involves local work
and communication

Alice's Instructions:

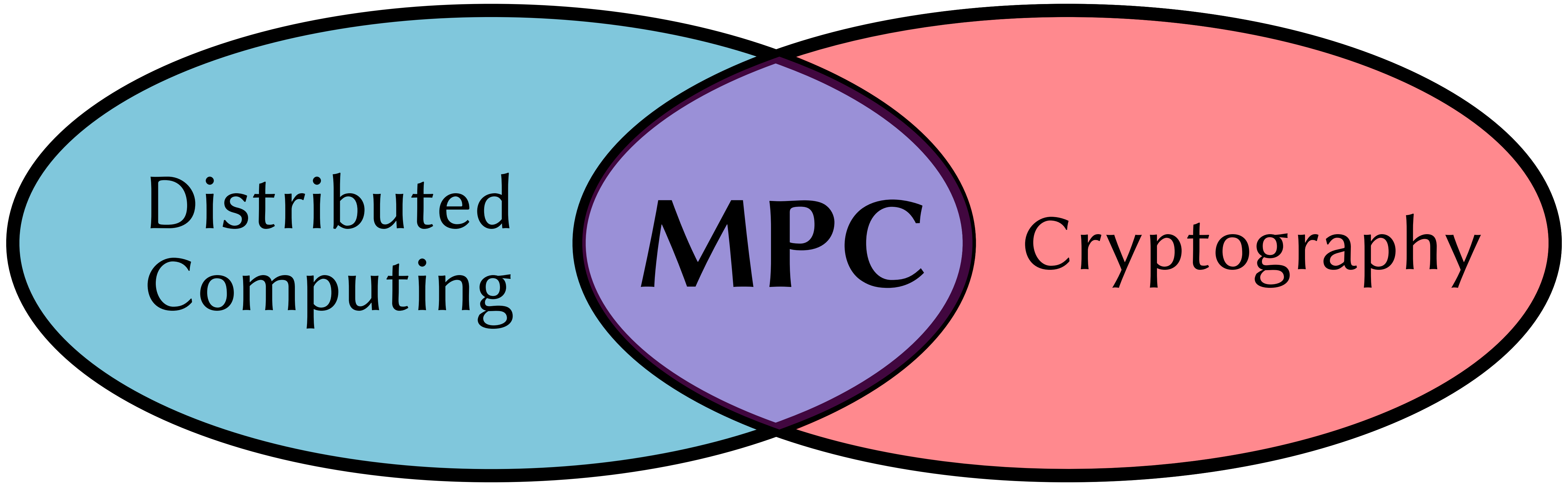
1. Mix butter and sugar
2. Add eggs one at a time
3. Stir in vanilla
4. Send mixture to Bob
5. Wait 30 minutes
6. Remove cake from oven

Bob's Instructions:

1. Preheat oven to 350°
2. Butter cake pan liberally
3. Combine flour, baking powder, salt
4. Receive mixture from Alice, add to dry ingredients
5. Pour batter into pan
6. Place pan in oven

Cryptographic Protocols are a tool to achieve

Secure Multiparty Computation (MPC)



i.e. to compute together without trusting one another

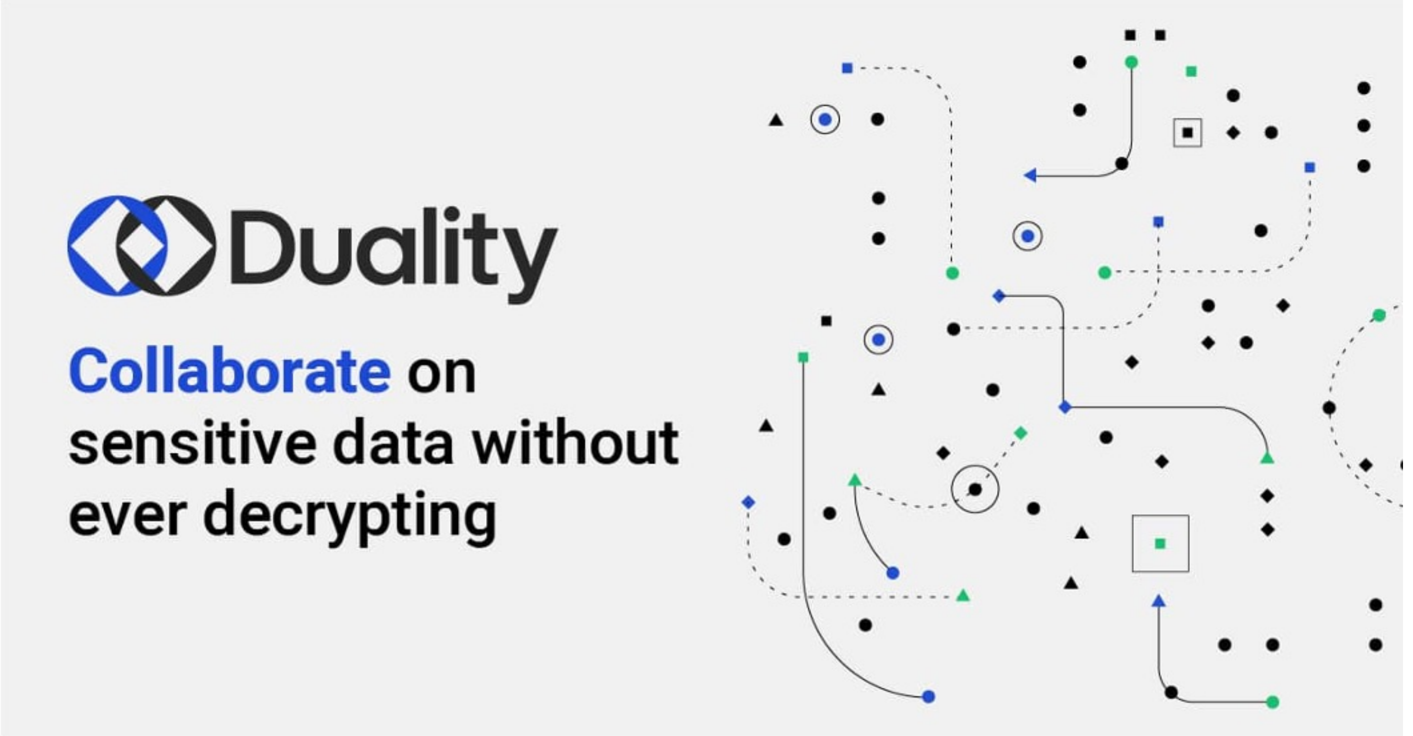
Secure Multiparty Computation (MPC)

20 years ago this was still a purely theoretical idea, but now…

Secure Multiparty Computation (MPC)

20 years ago this was still a purely theoretical idea, but now…

…we use it to achieve privacy-preserving data science and AI/ML!



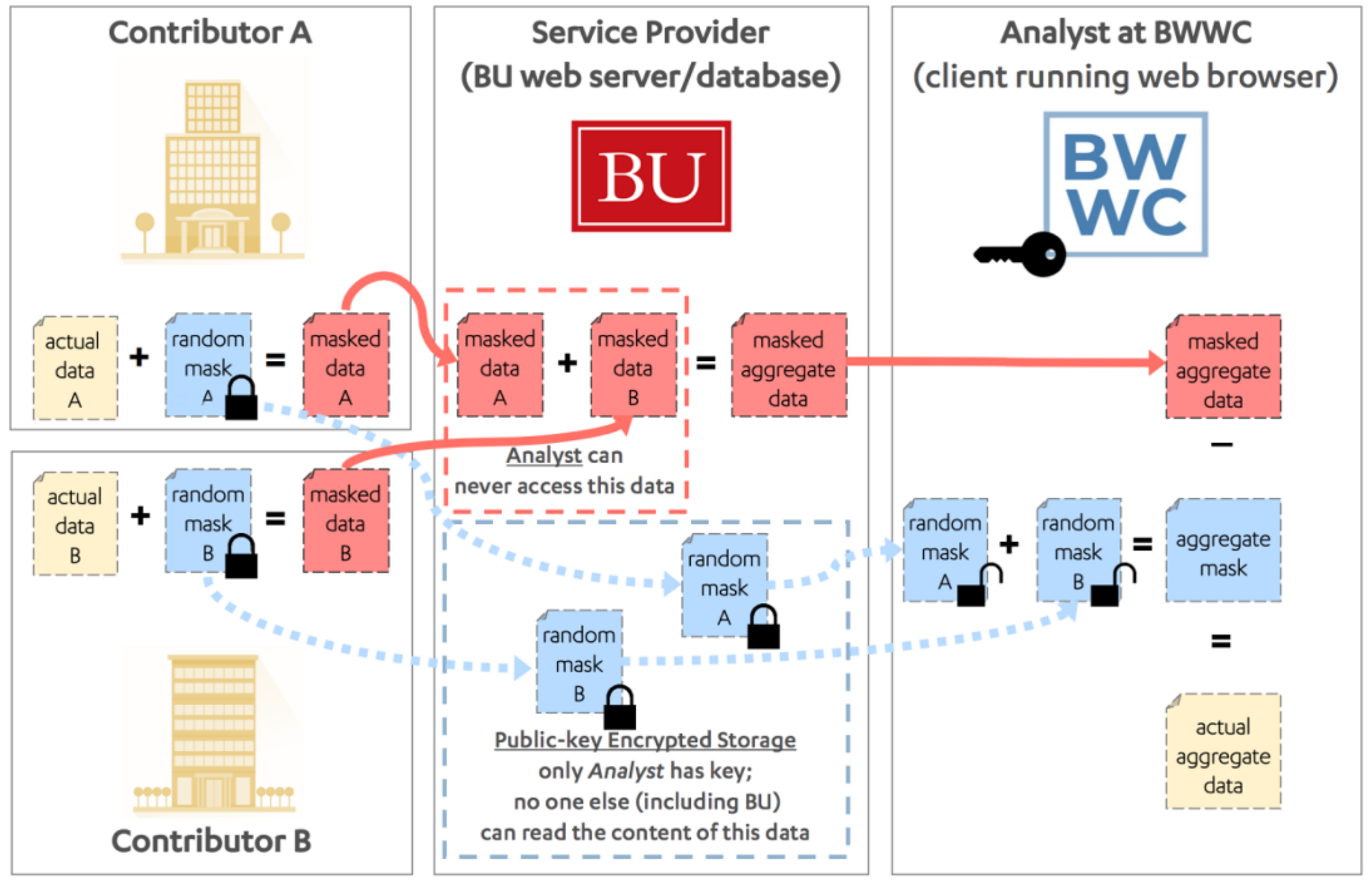


Figure 1: Slide detailing the MPC protocol. This was used in explanations to HR employees, lawyers, CEOs, and so on.

Secure Multiparty Computation (MPC)

20 years ago this was still a purely theoretical idea, but now…

…we use it to protect sensitive cryptographic keys and eliminate single points of failure in the internet infrastructure.

This is my research!

Secure Multiparty Computation (MPC)

20 years ago this was still a purely theoretical idea, but now…

…we we use it to add advanced capabilities and enhanced privacy guarantees to blockchain protocols, and to perform their initial setup steps.

Zcash

ethereum

Filecoin

The Goals of this Course:

1. Understand how to define security for *for interactive protocols.*
2. Understand the theoretical basis for cryptographic protocols. *Focus on general underlying principles.*
3. Understand limitations. *What is possible and what is impossible?*
4. Develop a Cryptographer's Mindset. *How to characterize and reason about unknown adversaries? How to achieve formal guarantees against bad outcomes?*

The Goals of this Course:

This mindset can be very useful in other fields!
Sometimes new fields can be formed by applying cryptographic methodologies to other problems.
e.g. differential privacy, some kinds AI fairness research, some kinds of adversarial ML

4. Develop a Cryptographer's Mindset.
   How to characterize and reason about unknown adversaries? How to achieve formal guarantees against bad outcomes?

Cryptography is Fun!

1. Solve twisty problems

2. Do things that seem impossible!
   (e.g. prove something is true without revealing *why* it's true)

3. Think like an adversary

Syllabus (tentative):

A taxonomy of adversaries; a variety of techniques

(don't worry if you don't understand what everything means yet)

	Part 1: *Information-theoretic* techniques. Adversaries with unbounded power	**Part 2:** *Cryptographic* techniques. Adversaries with bounded power
Semi-honest Adversaries: *follow* the rules of the protocol	Secret Sharing BGW protocol for an honest majority	Oblivious Transfer GMW protocol for a dishonest majority Yao's protocol for two parties Fully Homomorphic Encryption
Malicious Adversaries: *break* the rules of the protocol	Verifiable Secret Sharing BGW protocol for honest supermajority	Coin Tossing Zero-Knowledge Proofs GMW Compiler Byzantine Agreement + Broadcast

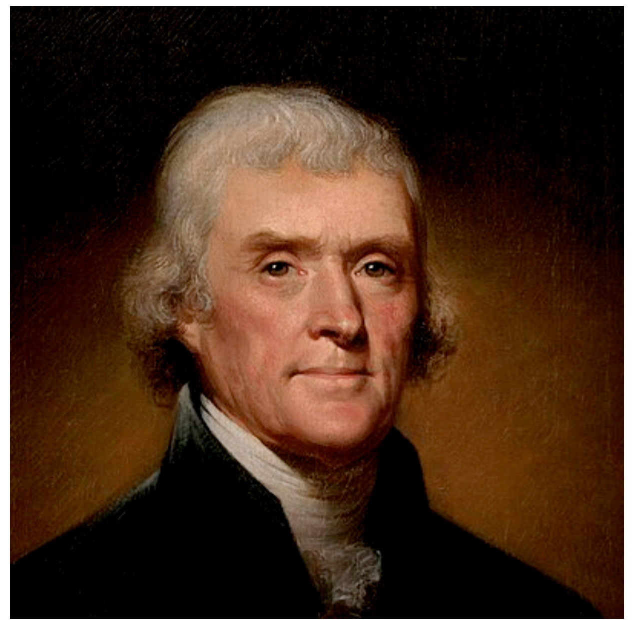
Overarching Questions:

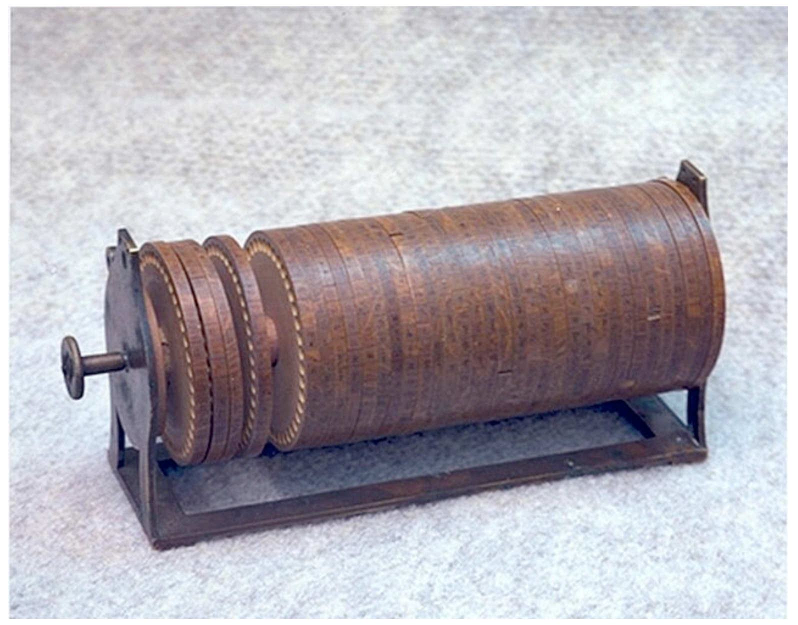
How do we characterize unknown adversaries? How do we formalize intuitive security notions?
What kinds computation can we perform securely in each setting?

We will not talk about:

Historical Cryptography

Arent you tired of that guy?





Foundations of Modern Cryptography

(e.g. how do we get *Encryption* from *One-way Functions;*
how are *One-way Functions* and *Hash Functions* different?)

If you want to learn that kind of thing, join CS6222 in the fall!

We will not talk about:

Implementations

(probably no programming assignments)

Systems Security Techniques

“Cybersecurity”

(as far as I can tell that just means “security and computers” so I guess we will technically talk about it but I refuse to call it “cyber”)

Blockchains, Cryptocurrency*

Secure or private AI/ML*

Quantum Computing

Post-quantum Security†

*however, the techniques we will learn about are critical building blocks for these things!

†actually, some things in the course will be post-quantum secure, but we won’t discuss *why*.

Prerequisites/Background

Mathematical maturity: understanding definitions, reading and writing proofs, using mathematical notation. You'll get a lot of practice if you haven't had it already.

Topics you should understand: discrete math, reductions, polynomial time, modular arithmetic, basic probability theory. If you haven't taken classes on these you might have to study them on your own to keep up. If a lot of people need to review something, we'll review it!

Not expected but useful: groups, fields, linear algebra.

You do not need to know: cryptography, networks, distributed computing!

Class Format:

Mix of slides and white-board proofs.

No recordings or zoom rooms by default. If you want to organize these yourselves, you *must* ask beforehand, and they may not be posted online.

No attendance or participation tracking. This is an elective on an advanced topic - you should be here because you *want* to be here.

This class is a collaboration between me and you. I encourage you to interrupt me with questions. If you are confused, other people are too!

Coursework (tentative):

4-5 Homeworks: 60% of final grade
Mostly mathematical/proof-based. Solved collaboratively (see course website).

1-2 Scribe Notes: 20% of final grade
Everyone must scribe. Sign up online. We need someone for next class!

Final Exam: *20% of final grade*
In person, no collaboration. Should be much easier than Homework.

About this Class

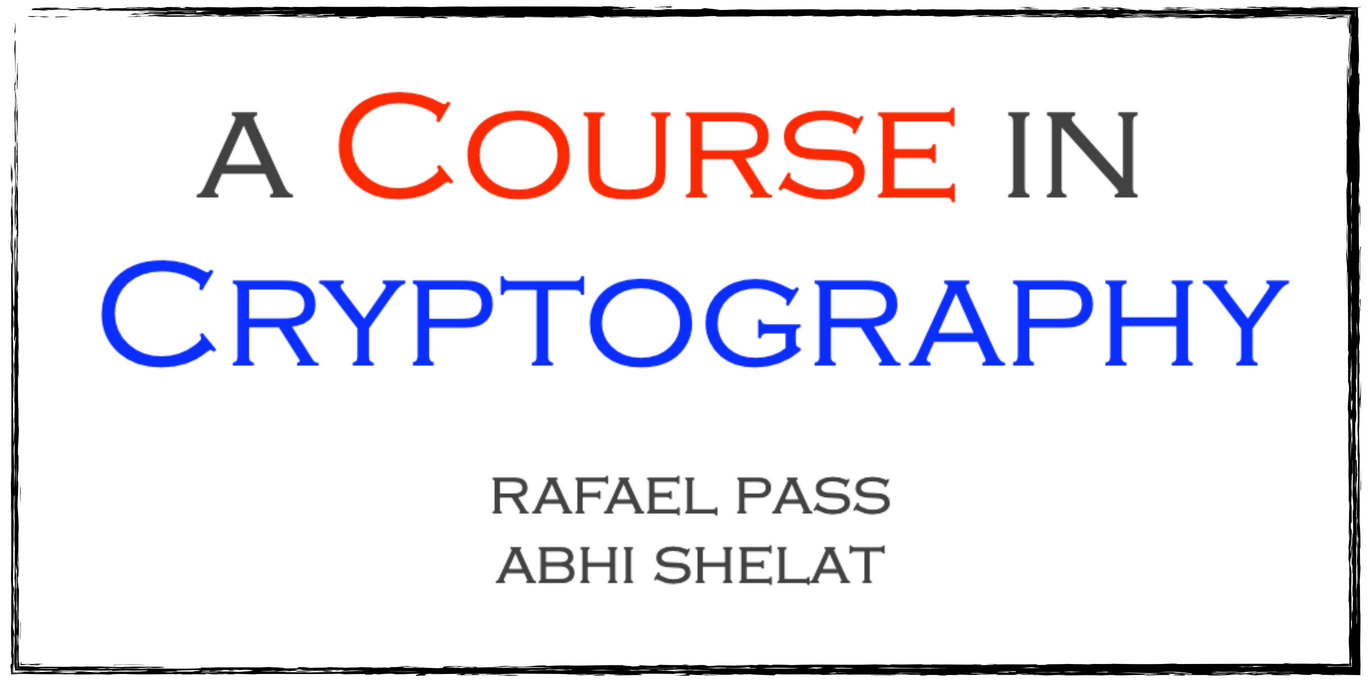
Based on a course developed by Ran Cohen 

New to me and UVa. Not many classes anywhere like it!

$\Longrightarrow$ Things might change as we go along. We will debug together.

No textbook exists

$\Longrightarrow$ You will write one together by scribing the lectures!



The textbook for CS6222 started as scribe notes for the MIT cryptography course in the early '00s

Someday, maybe somebody will use *your* scribe notes to teach a course!

Resources!

Your friends (but not ChatGPT)!

Office hours (Instructor + TA)

Many resources linked from course website

Go here

UVa CS4501: Cryptographic Protocols (Spring 2026)

Status

Welcome to class! This website will be your definitive resource. It was last updated on January 11, 2026.

Table of Contents

People, Places, Materials, Communication

- **Instructor:** Jack Doerner. **Email:** jhd3pa at virgina dot edu. **Office:** Rice Hall 106. **Office Hours:** Fr 2:00pm - 3:00pm and by appointment.
- **TA:** Jinye He (Clara). **Email:** qfn5bh at virginia dot edu. **Office Hours:** Fr 4:00PM - 5:00PM in Rice Hall 442.
- **Lecture Location:** Olsson Hall 009.
- **Lecture Time:** TuTh 3:30pm - 4:45pm.
- **Primary Course Materials:** None.
- **Coursework Submissions:** generally via Gradesope (on Canvas).
- **Online Discussions:** Piazza.
- **Contact Email:** cs4501s26 at jackdoerner dot net. This email forwards to the instructor and the TA(s), who will do their best to respond in a timely fashion. Please try to use descriptive subject lines, and include the assignment number for any email that concerns a specific assignment. Especially urgent messages can be marked as such.

Preamble

Suppose that Alice, Bob, and Carol each know a secret, and they want to perform a computation together using their secrets, but they do not trust one another. In this class, we will start from the basic question of how they can communicate securely, and build our way up to a protocol that allows them to jointly compute any function on their secrets without revealing those secrets to one another. Along the way, we will explore how to define 'security' not just for data, but for computations, we will determine when secure computation is possible and when it is impossible, and we will learn all of the cryptographic tools that we need to achieve our goal, including digital signatures, zero-knowledge proofs, and consensus protocols.

Additional Resources

These extra materials might help you on your way. Note that there may be discrepancies in notation, ordering of concepts, and even definitions! Note that if you consult these in order to solve a specific homework problem, you should cite them.

Free Online Resources (Basic Level)

- Mathematics for Computer Science, a textbook by Eric Lehman, F Thomson Leighton, and Albert R Meyer. If you find your math background lacking in some respect, there is a good chance you'll find the information you need in this book.
- A Pragmatic Introduction to Secure Multi-Party Computation, a textbook by David Evans, Vladimir Kolesnikov, and Mike Rosulek.
- The Joy of Cryptography, a textbook by Mike Rosulek.

Free Online Resources (Advanced Level)

- The First, Fifth, and Twelfth BIU winter schools dealt with topics in multi-party computation. Video recordings of these lectures are available online.
- How To Simulate It - A Tutorial on the Simulation Proof Technique, by Yehuda Lindell.
- A Course in Cryptography by Rafael Pass and abhi shelat.
- MIT Cryptography Course Notes by Vinod Vaikuntanathan.
- Harvard Cryptography Course Notes by Boaz Barak.
- A Graduate Course in Applied Cryptography, a textbook by Dan Boneh and Victor Shoup.
- Pseudorandomness, a textbook by Salil Vadhan.

Physical Textbooks (Available Online via UVa)

- Introduction to Modern Cryptography by Jonathan Katz and Yehuda Lindell. The UVa Library provies free online access.
- Foundations of Cryptography by Oded Goldreich. A very thorough work that takes no shortcuts, considered by many to be the standard among reference materials for cryptography. The UVa library provides online access to Volumes One and Two.
- Tutorials on the Foundations of Cryptography, edited by Yehuda Lindell. A gift from eight authors to Oded Goldreich (and the wider community) on the occasion of his 60th birthday. The UVa library provides free online access.

Interesting Readings of Debatable Relevance to this Course

- Wittgenstein's Lectures on the Foundations of Mathematics, Cambridge, 1939 by Cora Diamond. Philosophy is like unravelling a ball of wool.
- How to Explain Zero Knowledge Protocols to Your Children, by the Quisquater and Guillou families, and translated by Tom Berson. A mini-lesson for your inner child.
- Goldreich's Essays on various topics. Of particular note: why you should learn to prove theorems in the age of AI theorem provers.

About Me

I was a Student Here

(Olsson 009 has always sucked)

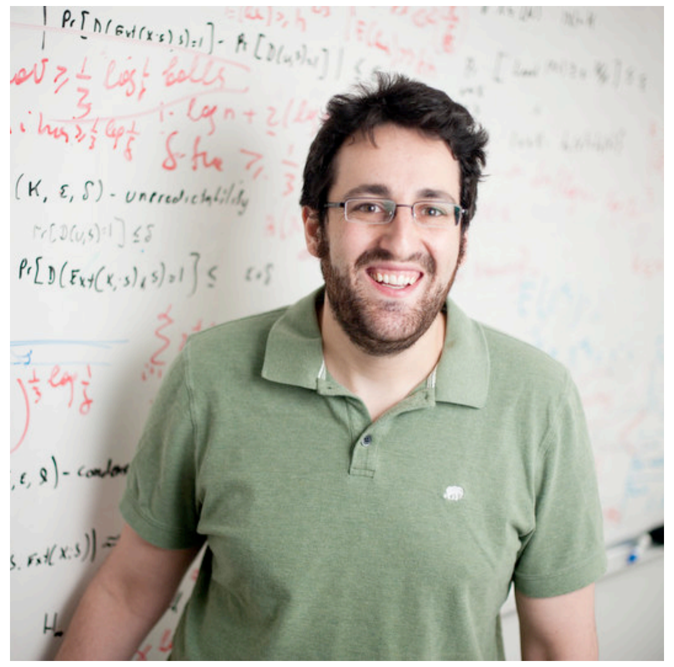
My Research:

TCS ➤ Cryptography ➤ Multiparty Computation ➤ Threshold Crypto ➤ Practical

Most Importantly:

I am a new professor and this is the second class I have taught!
I want your feedback!

A short story about my first crypto class



Instructor
Daniel Wichs
Northeastern University
Fall 2017

Any Questions?
And now, let’s begin!

https://jackdoerner.net/teaching/#2026/Spring/CS4501

☝ All Course Details Here ☝

Lecture 1: An intuitive notion of security (by example)

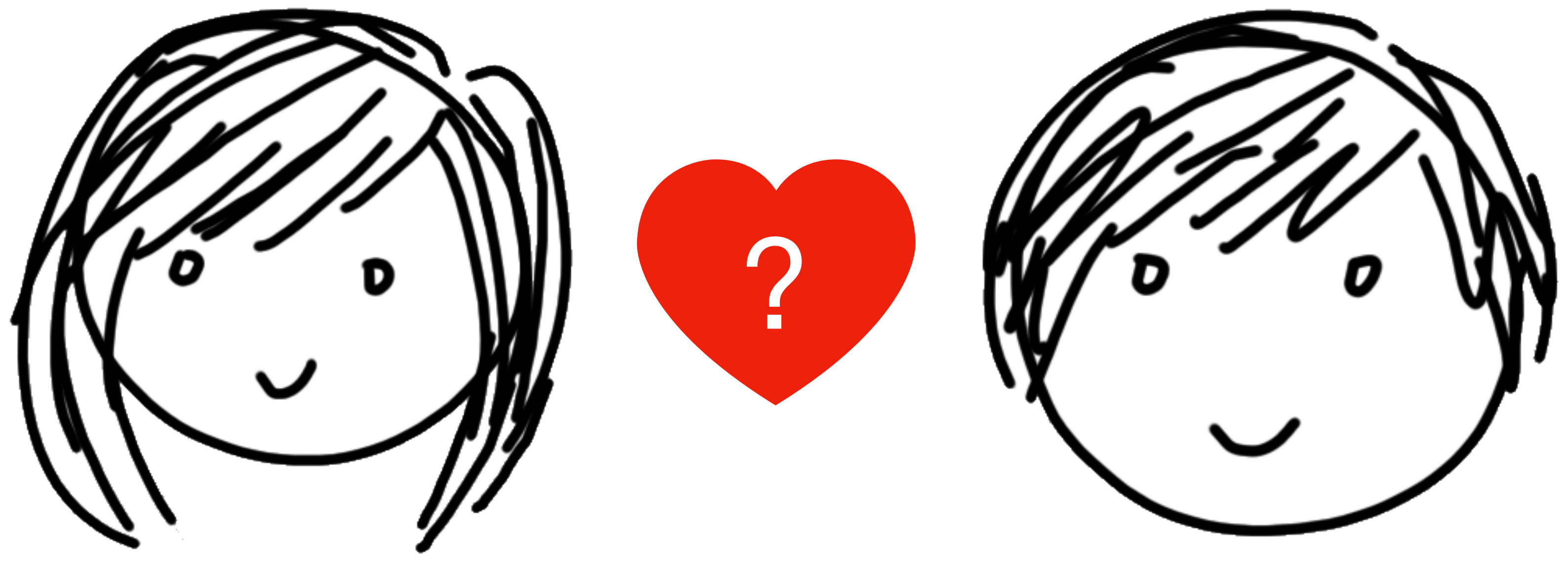
Scenario 1: Dating

Alice and Bob meet in a bar:

- If they both want to go on a date, they should find out!
- If Alice doesn't want to date, Bob doesn't want her to know he is interested
- If Bob doesn't want to date, Alice doesn't want him to know she is interested

How to make a match without revealing secrets?

Solution: Use a trusted bartender.

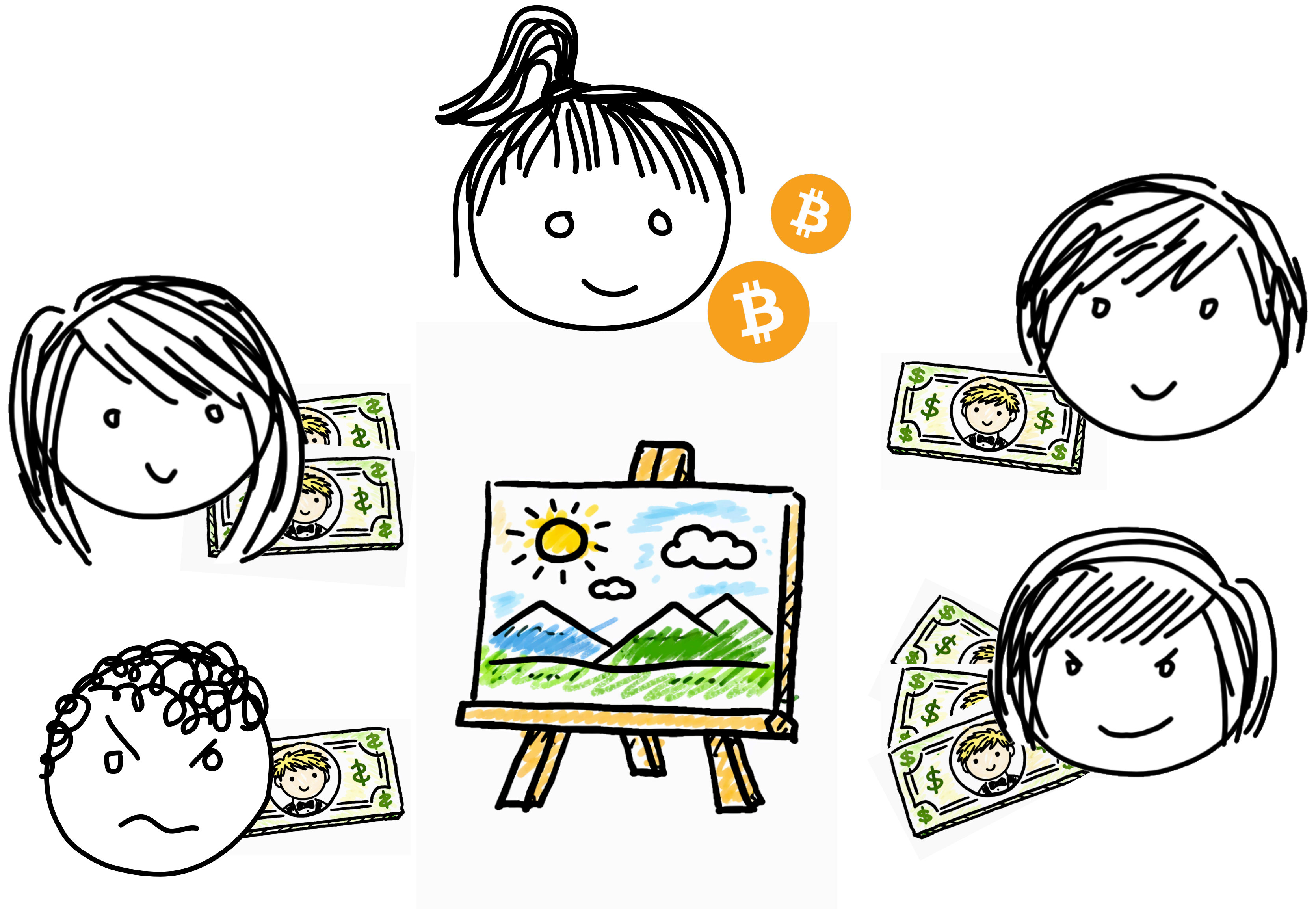


Scenario 2: Private Auction

Many parties wish to run a private auction

- The highest bid wins
- Only the highest bid (and bidder) is revealed

How to conduct the sale fairly?



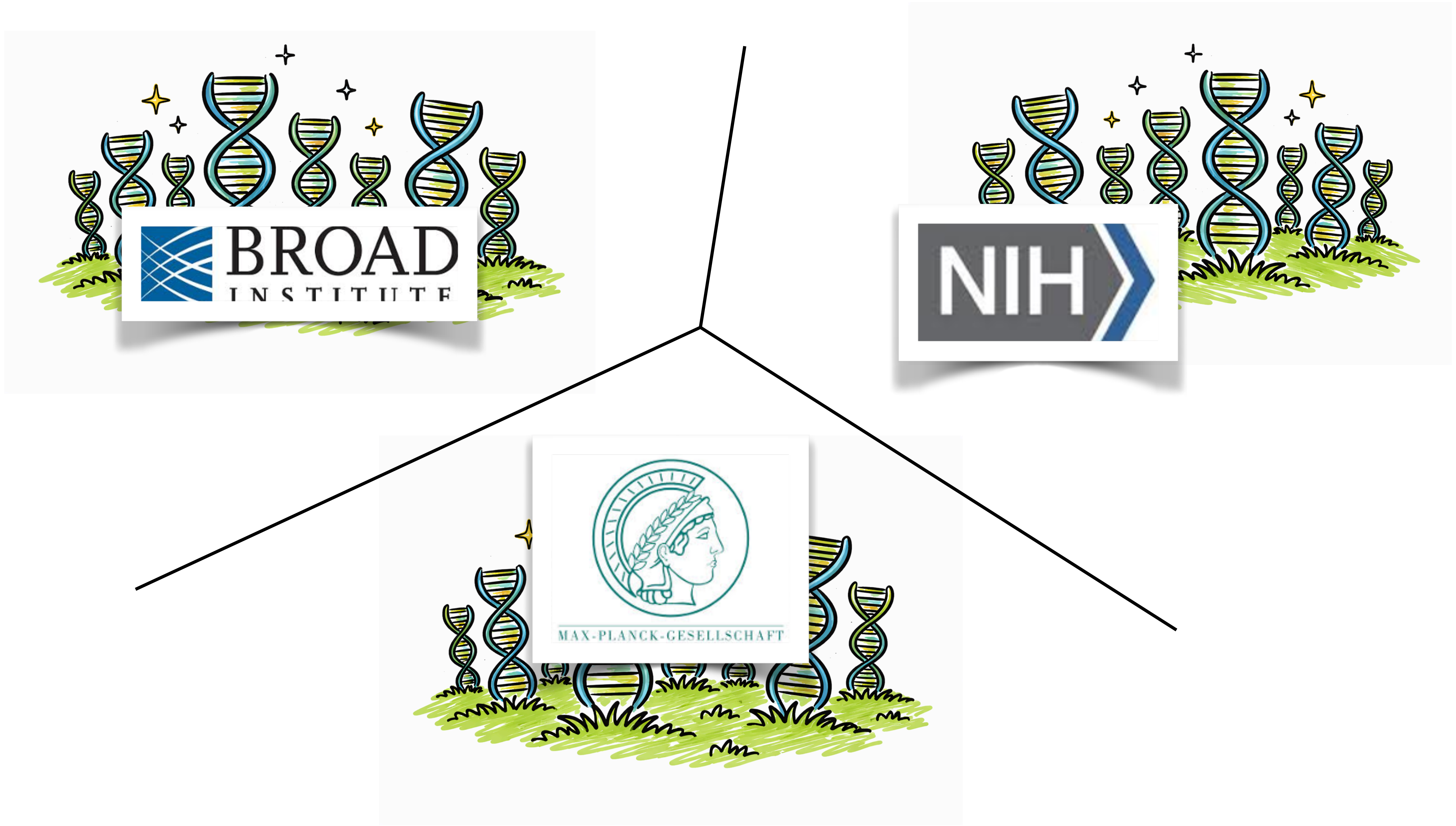
Solution: Use a trusted auctioneer.

Scenario 3: Collaborative Science

Each medical facility knows the genomes of its patients

- We want to run medical studies using all of them at once
- The law strictly limits how patient data can be shared

Can we simultaneously protect the privacy of the patients *and* learn from their records?



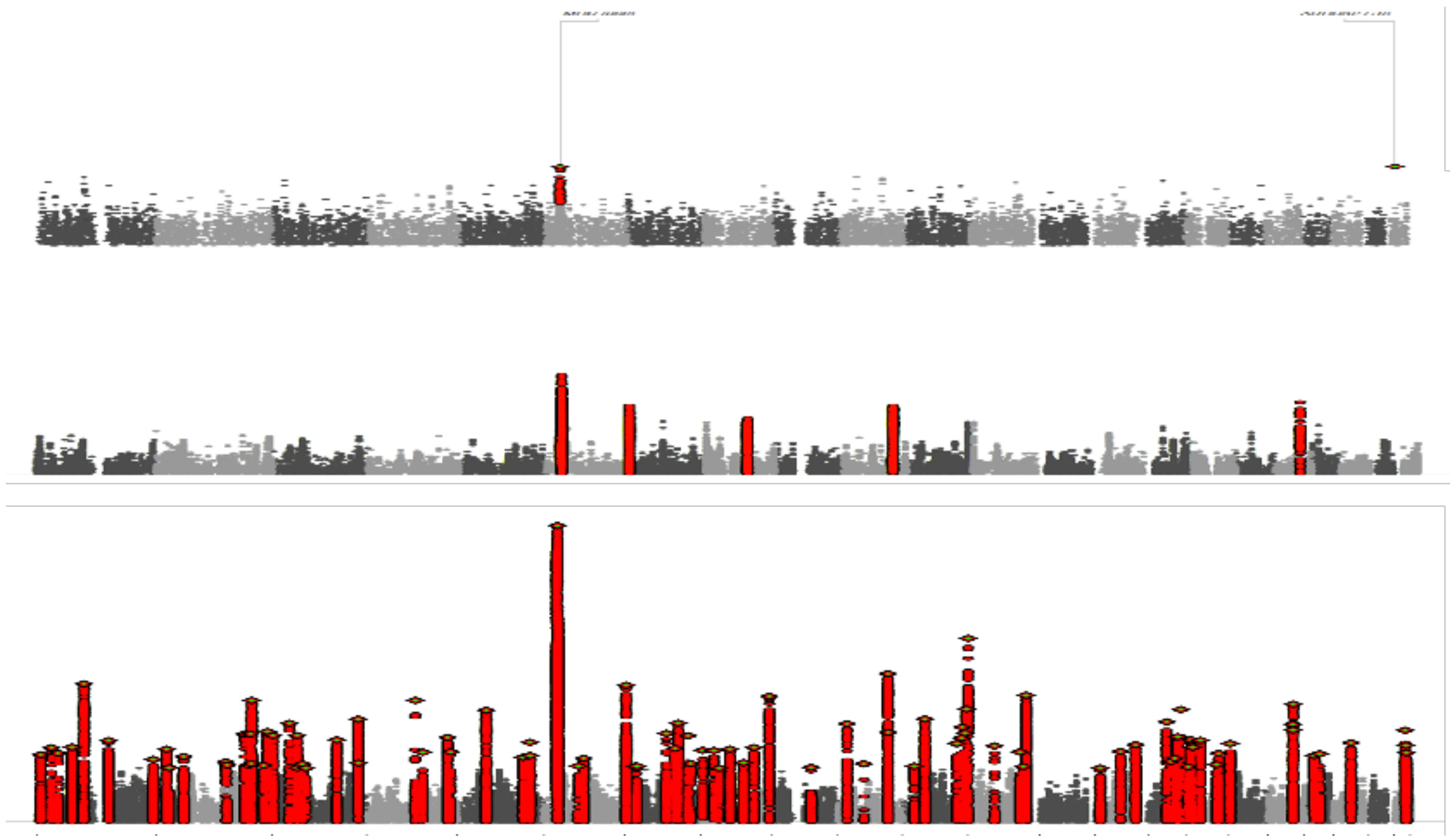
Solution: Use a single trusted scientist.
Note: it can be hard to get legal permission!

Aside: big data yields important results!

3500 cases
⟹ 0 loci

10000 cases
⟹ 5 loci

35000 cases
⟹ 62 loci



Data courtesy of Manolis Kellis

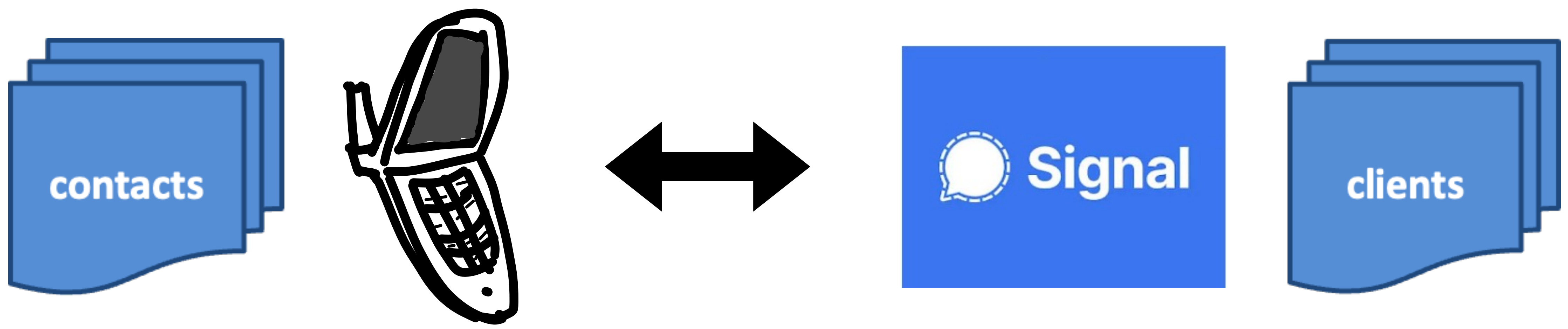
Increasing sample sizes for schizophrenia association studies
has led to increases in the number of risk genes discovered

Scenario 4: Private Contact Discovery

Your phone has a list of contacts, and Signal has a list of clients

- You want to find out which of your contact use Signal
- Don't want to reveal your contacts to Signal
- Signal does not want to reveal any clients who aren't in your contacts

Solution: Someone you both trust can compute the intersection of the lists. But who?



What Secure Multiparty Computation Achieves

- In all of these scenario, an *ideal* trusted third party solves the problem!
- Trusting a third party is a very strong assumption. By the nature of the problem, nobody is allowed to check their work! How many people really behave ideally?
- In high-stakes real-world situations, agreement upon a trusted entity might be impossible.

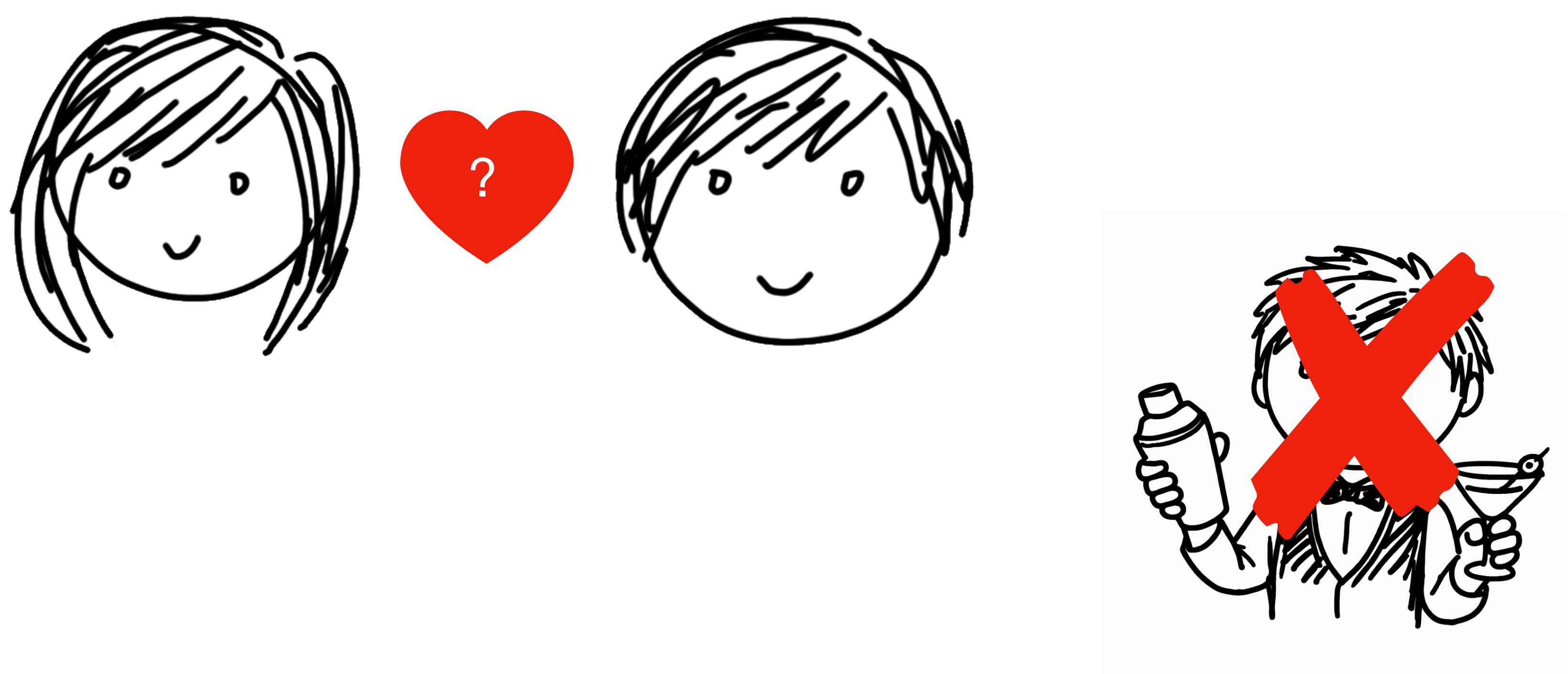
Can we do better?

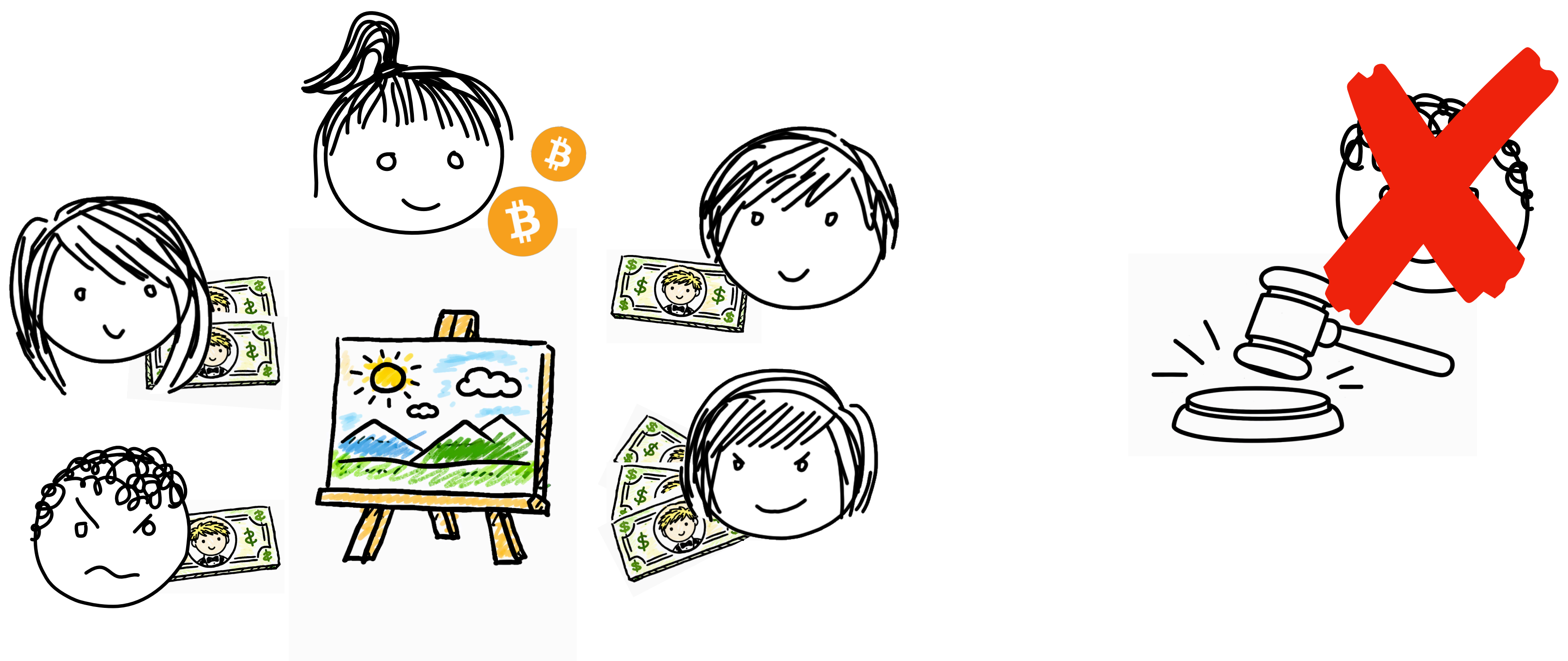
- We would like a solution with the same security guarantees as a trusted helper, but **without** actually using a trusted helper.

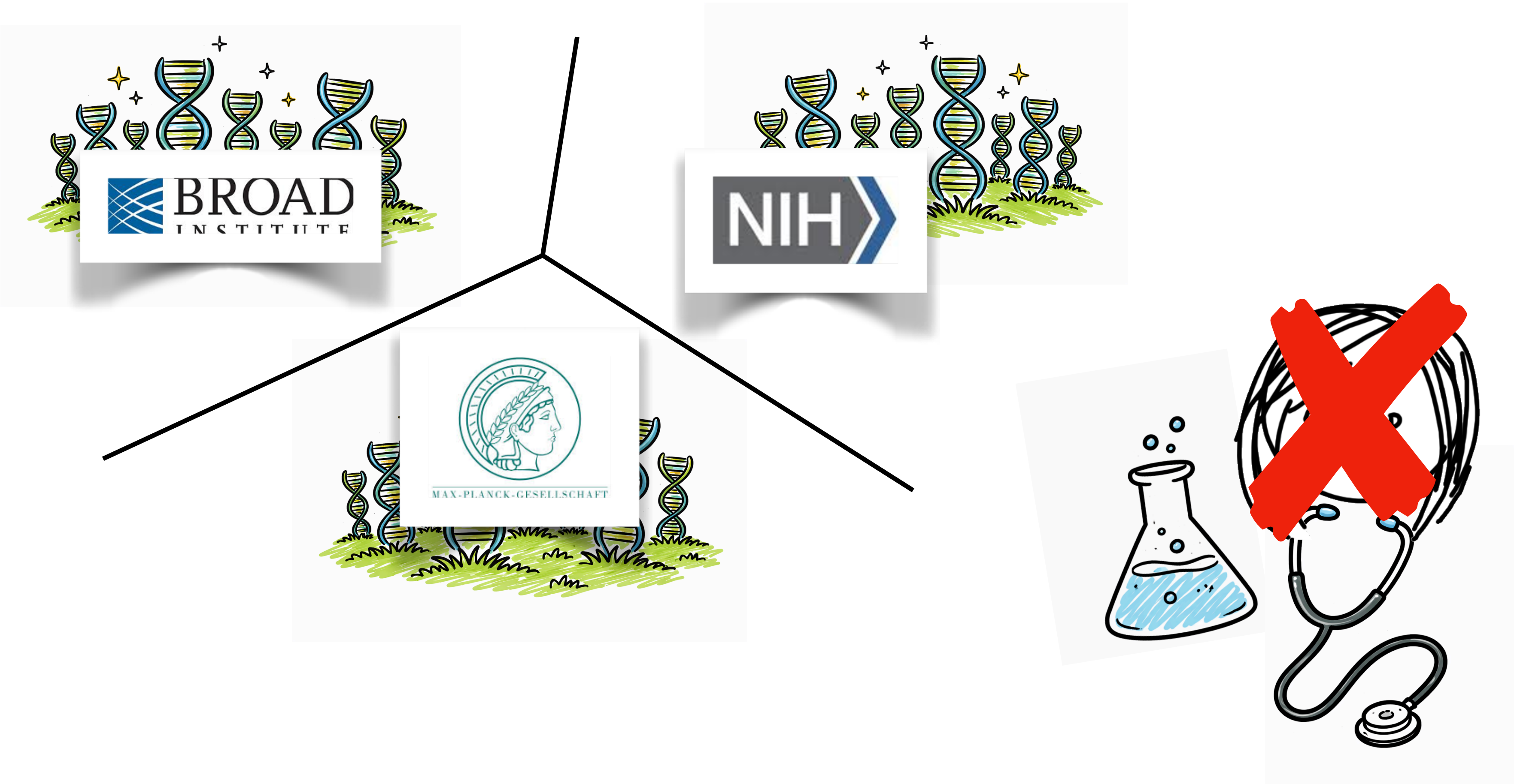


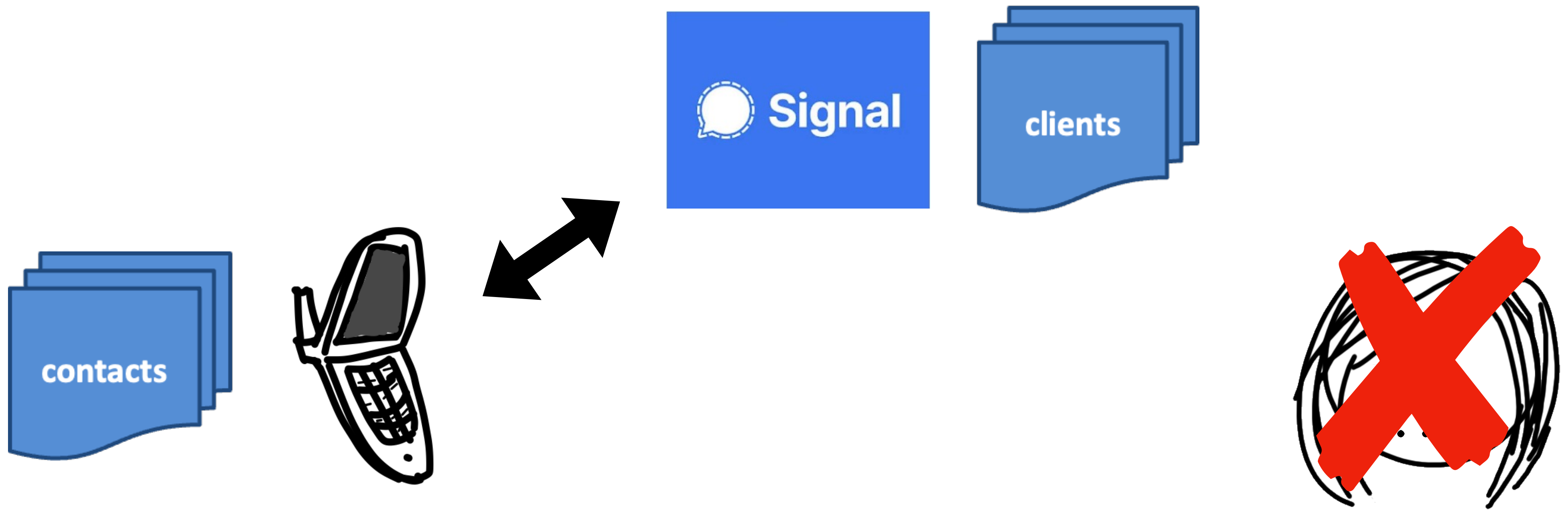

What Secure Multiparty Computation Achieves

We can design protocols that provably *emulate* trusted third parties
This enables the other participants to achieve their goals *without* an extra trusted entity







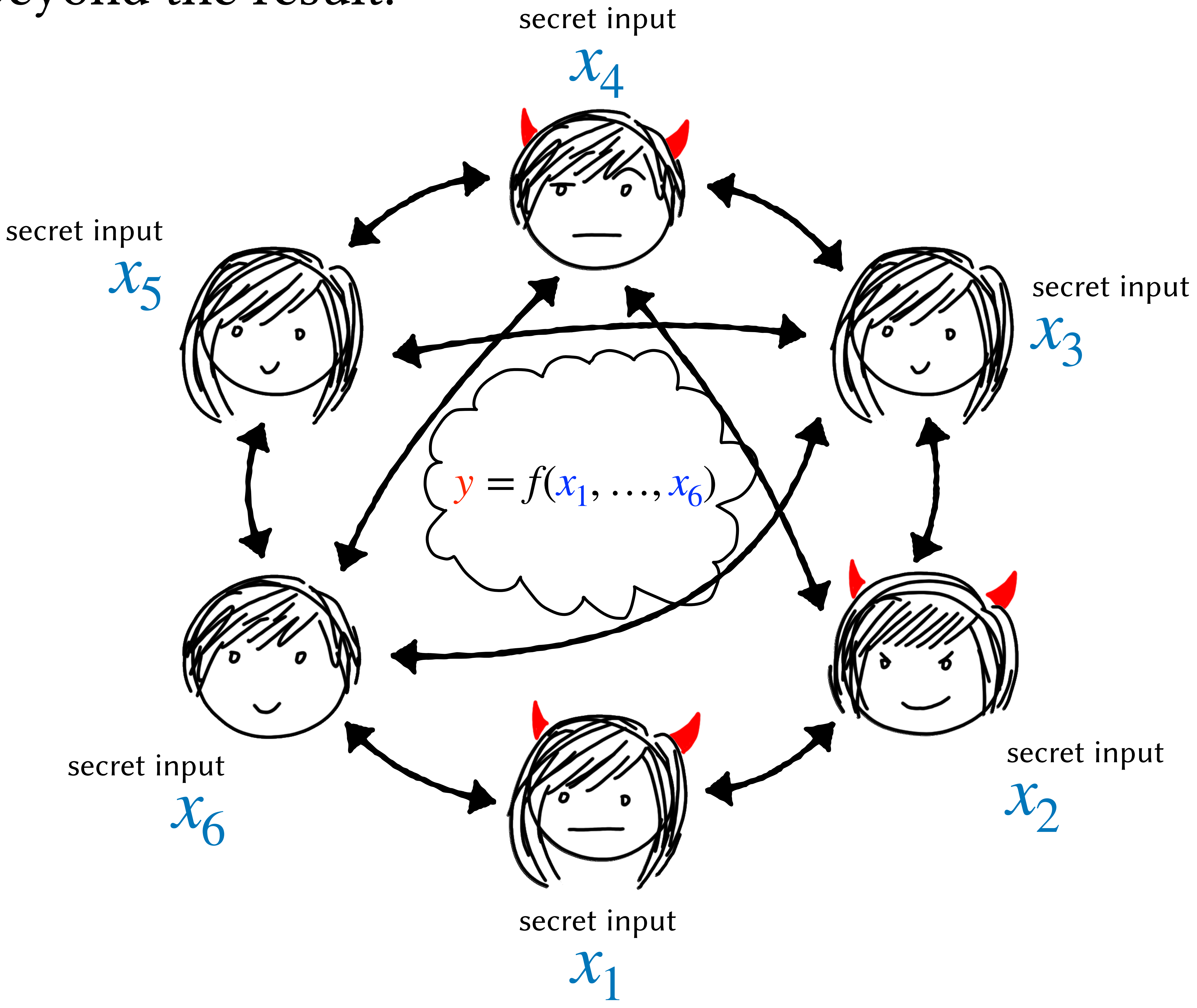


Let's look at MPC another way

Secure Multiparty Computation means **jointly computing** on secret data, while **revealing nothing** about the data beyond the result.

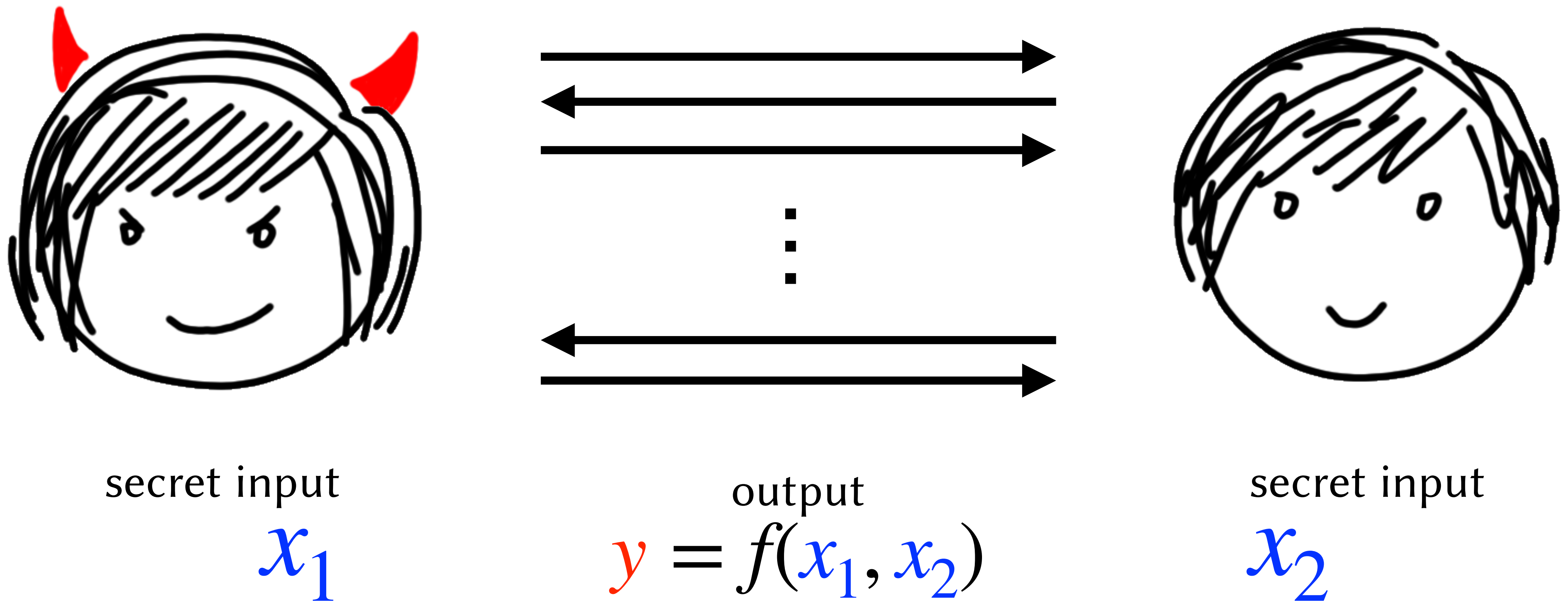
In this example, only y is revealed
The internal workings of f and the secret inputs $\{x_1, \ldots, x_6\}$ remain hidden

This must be true even if some parties misbehave in an attempt to learn more than they should!



What does it mean to “reveal nothing”

Remember, a protocol includes local instructions and communication. That is, *messages.*



Q: What is revealed to Alice about Bob’s input x_2?

A: Whatever information the messages and y convey!

Q: What does it mean that the messages reveal nothing about x_2, beyond y?

A: Using only x_1 and y, Alice could generate messages by herself that are *indistinguishable* from the messages she generated interacting with Bob.

Does this seem counterintuitive?

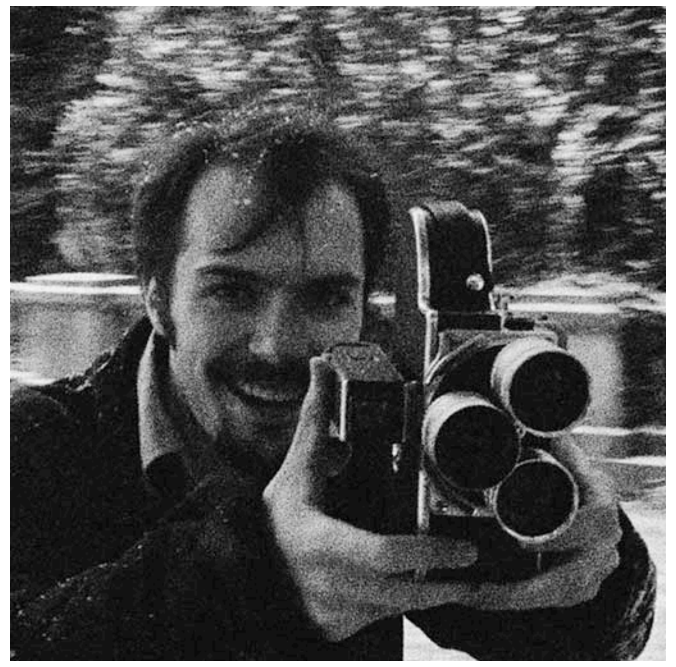
A: Using only x_1 and y, Alice could generate messages by herself that are *indistinguishable* from the messages she generated interacting with Bob.

Next Time: What Can an Adversary Do?
Towards Formalizing these Intuitions

(And some simple protocols too!)

CS4501: Cryptographic Protocols

Instructor
Jack Doerner
jhd3pa@virginia.edu
Rice 106



TA
Jinye He (Clara)
qfn5bh@virginia.edu

https://jackdoerner.net/teaching/#2026/Spring/CS4501

☝ All Course Details Here ☝