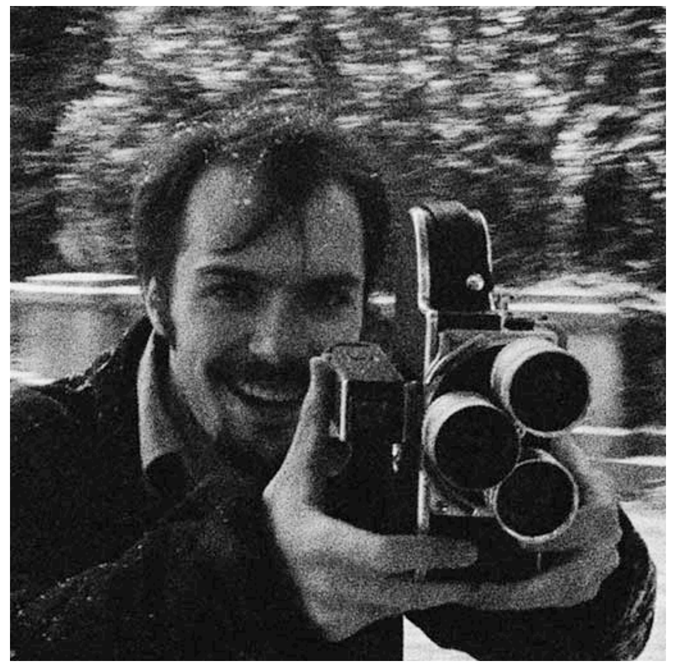
CS6600: Grad Cryptography

Instructor
Jack Doerner
jhd3pa@virginia.edu
Rice 106



TA
Jinye He (Clara)
qfn5bh@virginia.edu

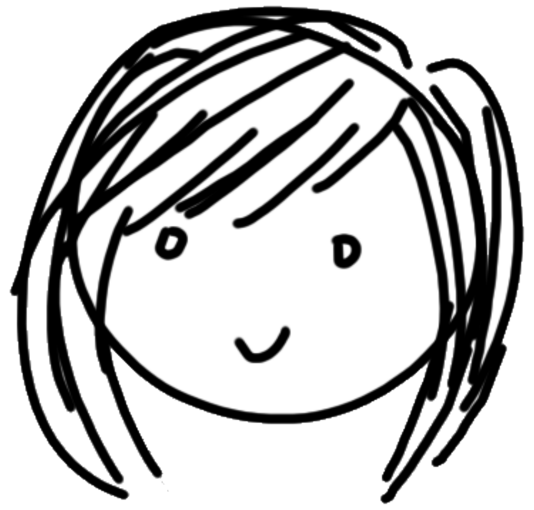
https://jackdoerner.net/teaching/2025/Fall/CS6222

☝ All Course Details Here ☝

♣♥ Matchmaking ♥♣
(how to go on a date with a cryptographer)

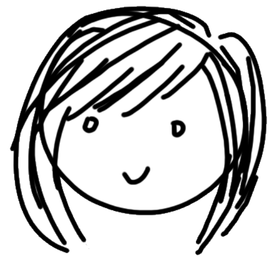
I promise the rest of the class won't be like this

Does this protocol produce a correct result?

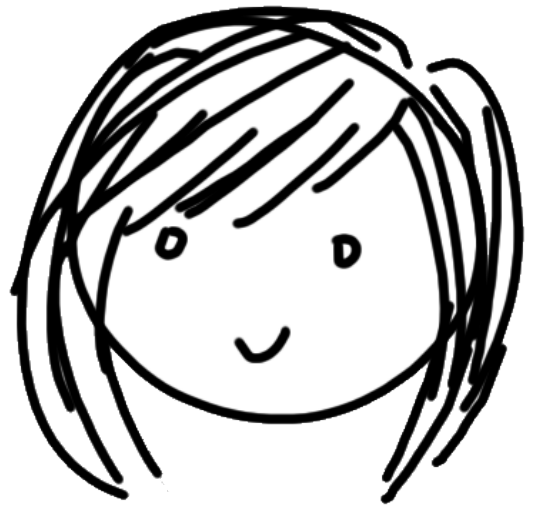


Yes = ♣♥
No = ♥♣

Yes = ♥♣
No = ♣♥

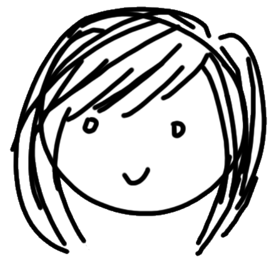
		Cards (Before Random Rotation)	Result
Yes	Yes	♣♥♥♥♣	
Yes	No	♣♥♥♣♥	
No	Yes	♥♣♥♥♣	
No	No	♥♣♥♣♥	

Does this protocol produce a correct result?



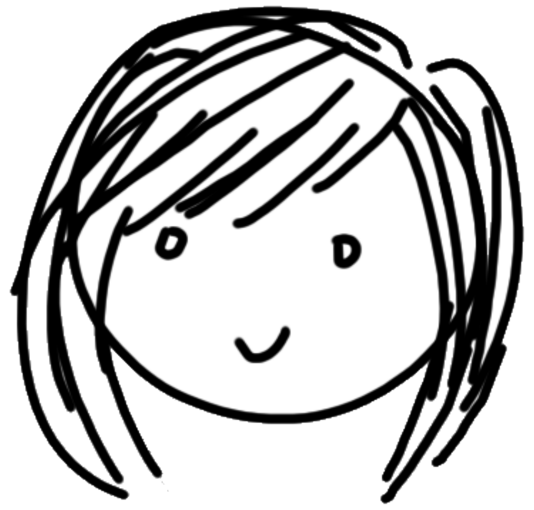
Yes = ♣♥
No = ♥♣

Yes = ♥♣
No = ♣♥

		Cards (Before Random Rotation)	Result
Yes	Yes	♣♥♥♥♣	Yes
Yes	No	♣♥♥♣♥	
No	Yes	♥♣♥♥♣	
No	No	♥♣♥♣♥	

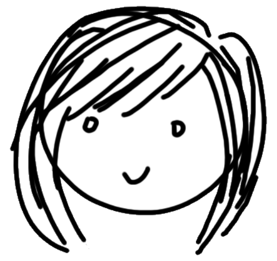
3 ♥ together

Does this protocol produce a correct result?

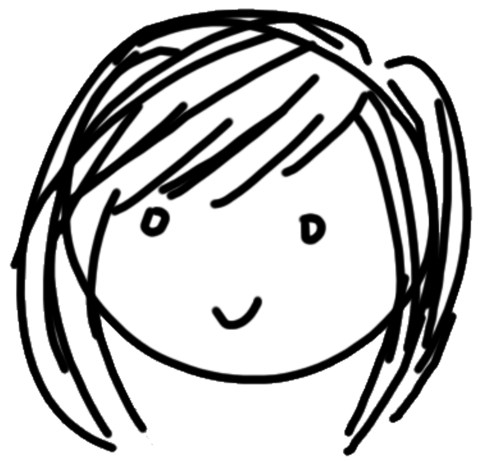


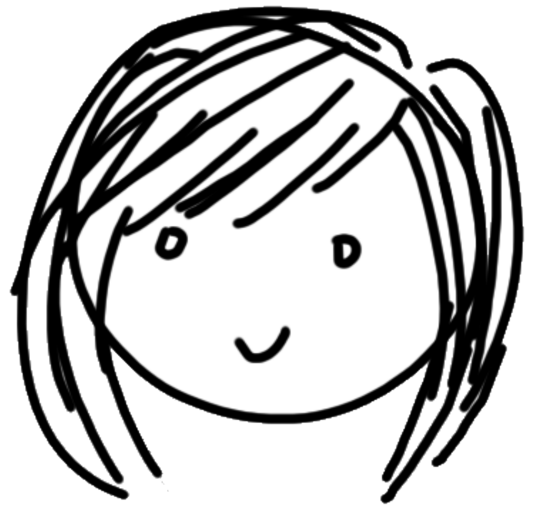
Yes = ♣♥
No = ♥♣

Yes = ♥♣
No = ♣♥

		Cards (Before Random Rotation)	Result
Yes	Yes	♣♥♥♥♣	Yes
Yes	No	♣♥♥♣♥	No
No	Yes	♥♣♥♥♣	No
No	No	♥♣♥♣♥	No

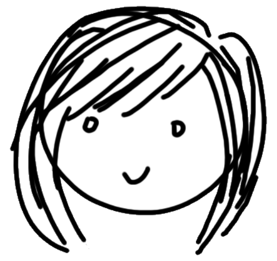
2 ♥ together

What can  and  learn from this?

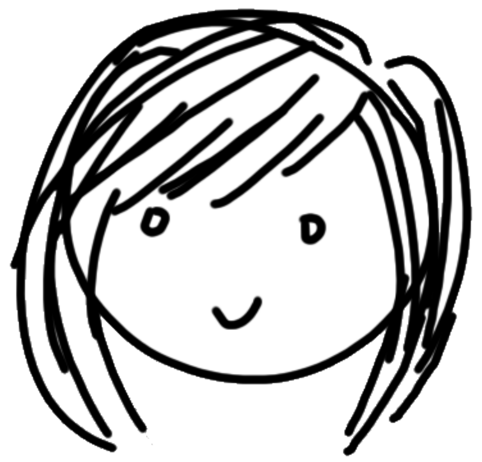


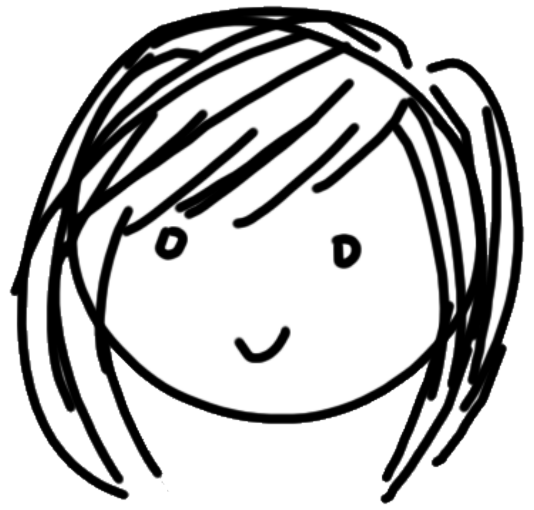
Yes = ♣♥
No = ♥♣

Yes = ♥♣
No = ♣♥

		Cards (Before Random Rotation)	Result
Yes	Yes	♣♥♥♥♣	Yes
Yes	No	♣♥♥♣♥	No
No	Yes	♥♣♥♥♣	No
No	No	♥♣♥♣♥	No

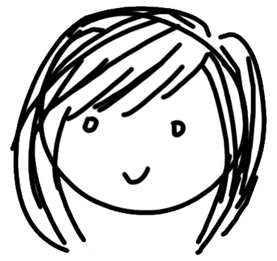
Rotations of Each Other
Indistinguishable after random rotation!

What can  and  learn from this?

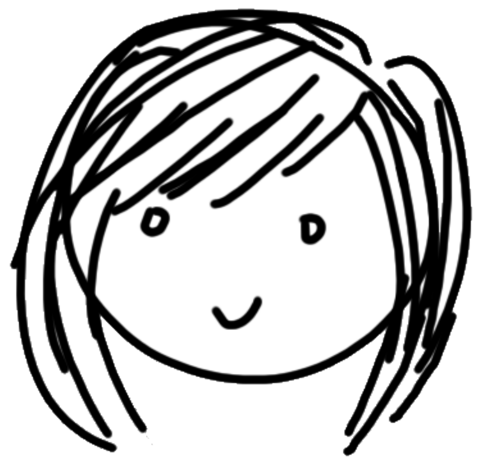


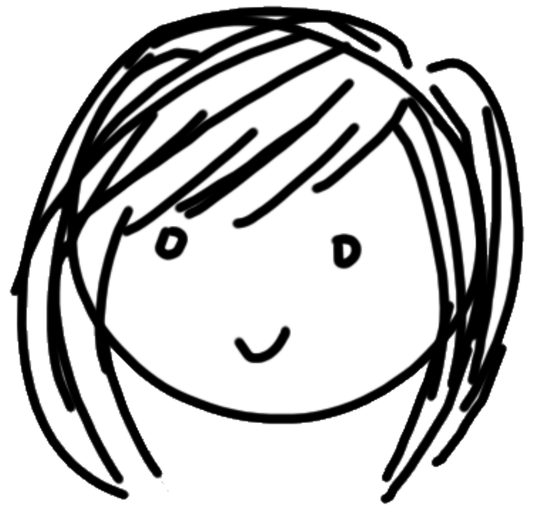
Yes = ♣♥
No = ♥♣

Yes = ♥♣
No = ♣♥

		Cards (Before Random Rotation)	Result
Yes	Yes	♣♥♥♥♣	Yes
Yes	No	♣♥♥♣♥	No
No	Yes	♥♣♥♥♣	No
No	No	♥♣♥♣♥	No

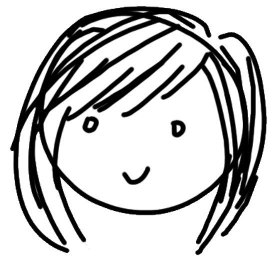
As promised, they learn only the result.
We proved that the protocol is *secure.*

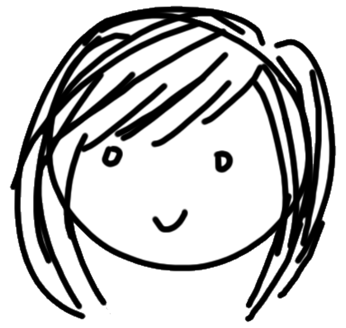
What can  and  learn from this?



Yes = ♣♥
No = ♥♣

Yes = ♥♣
No = ♣♥

		Cards (Before Random Rotation)	Result
Yes	Yes	♣♥♥♥♣	Yes
Yes	No	♣♥♥♣♥	No
No	Yes	♥♣♥♥♣	No
No	No	♥♣♥♣♥	No

Question: what happens if  doesn't rotate the deck randomly?

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.

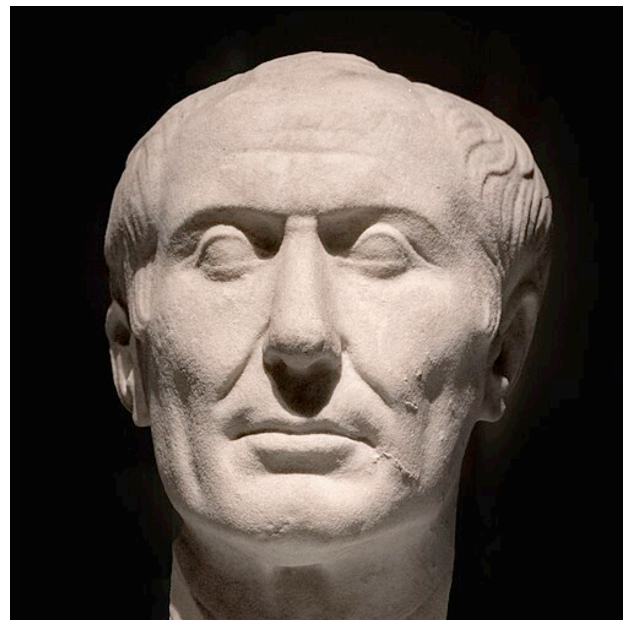
the art of writing or solving codes

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

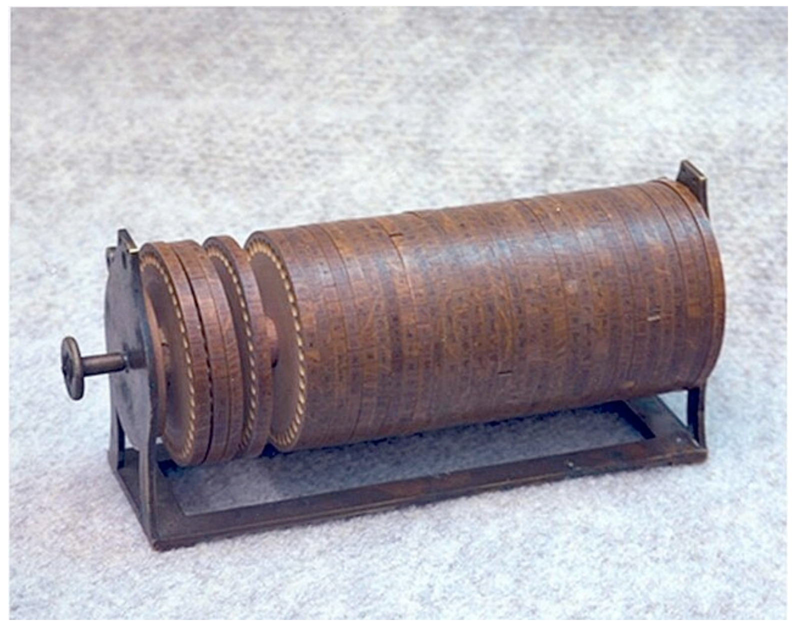



Governments and Militaries. Two-Party Communication with pre-agreed participants.

(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)!

What kind of things can we do?

Who uses Cryptography and for What?

Now:

Everyone (including you)!

Secure communication (many parties, maybe no pre-agreement)
Authentication
Anonymous communication + authentication
Computing on secret data without revealing it (*Homomorphic Encryption*)
Computing with people you don't trust (*Multiparty Computation*)
Currency without centralized authority (*Blockchain/E-Cash*)
Verifying computation efficiently
Secure elections
Derandomization
Consensus

Who uses Cryptography and for What?

Now:

Everyone (including you)!

Secure communication (many parties, maybe no pre-agreement)
Authentication
Anonymous communication + authentication
Computing on secret data without revealing it (*Homomorphic Encryption*)
Computing with people you don't trust (*Multiparty Computation*)
Currency without centralized authority (*Blockchain/E-Cash*)
Verifying computation efficiently
Secure elections
Derandomization
Consensus

Current Research:
Obfuscating programs
Watermarking GenAI outputs

...and much more!

Who uses Cryptography and for What?

Now:

Theoretical Computer Scientists!

Important connections to other fields, e.g.:

Complexity: Cryptography Exists $\implies P \neq NP$

Learning Theory: many recent advancements are based on assumptions about learning problems

The Goals of this Course:

1. Understand the theoretical basis for the real world cryptosystems all around you (now, and in the near future).

2. Be ready to read current cryptography research and maybe even become involved!

3. 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 fairness research, some kinds of adversarial ML

3. 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

Prerequisites and Materials:

Mathematical Maturity: reading and writing proofs, mathematical notation

Topics you should understand: reductions, decision problems, NP-completeness, computational models (e.g. Turing machines), polynomial time, modular arithmetic, basic probability theory, linear algebra.

Also helpful: groups, fields

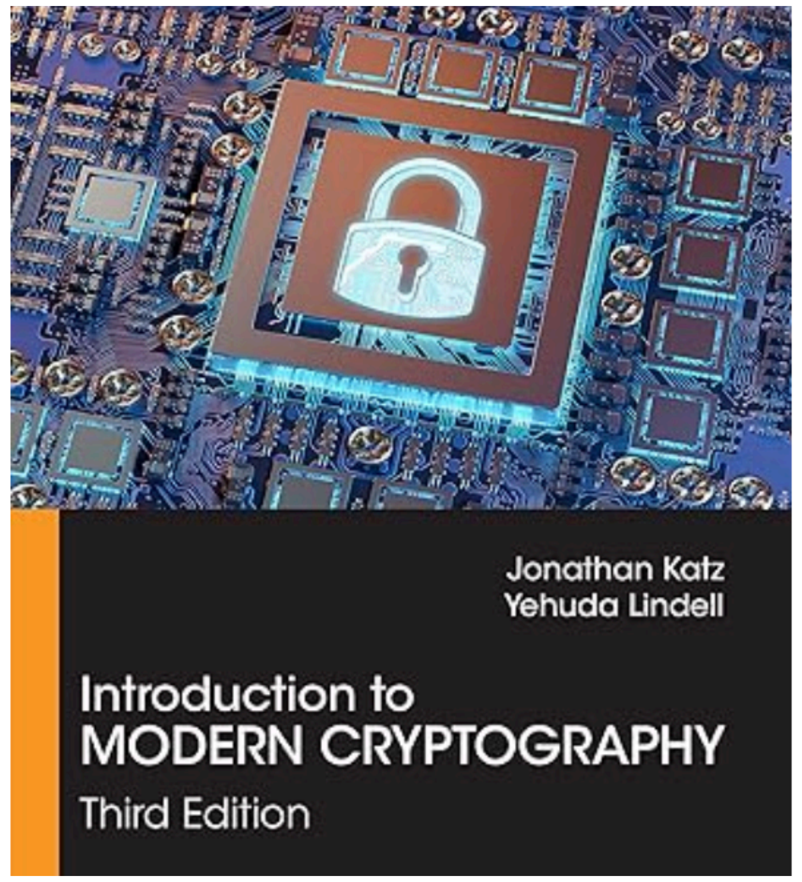
Free Online Textbook:

A COURSE IN CRYPTOGRAPHY

RAFAEL PASS
ABHI SHELAT

He used to work here...

Physical Textbook (free access via UVa):



Coursework (tentative):

4-5 homeworks: 50% of final grade
Solved Collaboratively - see course website

Scribe Notes: 15% of final grade
Everyone must scribe. Sign up online. We need someone for next class!

Final Project: 20% of final grade
Present a research paper in small groups.

Final Exam: 15% of final grade
In person, no collaboration.

Quizzes and misc: 10+% of final grade
Quizzes will be easy, I promise.

Syllabus (tentative):

Part 1: *Foundational Primitives*

One-way Functions (OWF)
Pseudorandom Generators (PRG)
Pseudorandom Functions (PRF)
Symmetric Encryption
Authentication (MAC, Signatures)

How are these things related?
How do they differ?
Why should we believe they exist?
How can we build them using
basic assumptions?

Syllabus (tentative):

Part 1: *Foundational Primitives*

One-way Functions (OWF)
Pseudorandom Generators (PRG)
Pseudorandom Functions (PRF)
Symmetric Encryption
Authentication (MAC, Signatures)

How are these things related?
How do they differ?
Why should we believe they exist?
How can we build them using basic assumptions?

Part 2: *Advanced Cryptography*

Zero-knowledge Proofs
Public-Key Encryption
(Fully) Homomorphic Encryption
Secure Two-Party Computation
Multi-Party Computation
Private Information Retrieval
Oblivious RAM

We will not get to all of this!
Some of these require stronger and more specific assumptions.

Syllabus (tentative):

Part 1: *Foundational Primitives*

One-way Functions (OWF)
Pseudorandom Generators (PRG)
Pseudorandom Functions (PRF)
Symmetric Encryption
Authentication (MAC, Signatures)

How are these things related?
How do they differ?
Why should we believe they exist?
How can we build them using basic assumptions?

My Research

Part 2: *Advanced Cryptography*

Zero-knowledge Proofs
Public-Key Encryption
(Fully) Homomorphic Encryption
Secure Two-Party Computation
Multi-Party Computation
Private Information Retrieval
Oblivious RAM

We will not get to all of this!
Some of these require stronger and more specific assumptions.

Wei-Kai Lin's Research

David Evans's Research

Syllabus (tentative):

Part 1: *Foundational Primitives*

One-way Functions (OWF)
Pseudorandom Generators (PRG)
Pseudorandom Functions (PRF)
Symmetric Encryption
Authentication (MAC, Signatures)

How are these things related?
How do they differ?
Why should we believe they exist?
How can we build them using basic assumptions?

Part 2: *Advanced Cryptography*

Zero-knowledge Proofs
Public-Key Encryption
(Fully) Homomorphic Encryption
Secure Two-Party Computation
Multi-Party Computation
Private Information Retrieval
Oblivious RAM

We will not get to all of this!
Some of these require stronger and more specific assumptions.

Other Kinds of Question:

How do we characterize adversaries? How do we formalize intuitive security notions?
How do we know when a particular assumption or primitive isn't powerful enough?

We will not talk about:

Historical Cryptosystems*
Cryptanalysis (historical or modern)
"Mathematical" Crypto
(e.g. Elliptic Curves, Class Groups, Isogenies, Number Theory Stuff)
Implementations
Systems security, Cybersecurity
Blockchains, Cryptocurrency
Quantum Computing
Post-quantum Cryptography†
Secure or private AI/ML

*there might be some homework problems though

†some things in the course will be post-quantum, but we won't discuss *why* this is the case

About Me

I was a Student Here

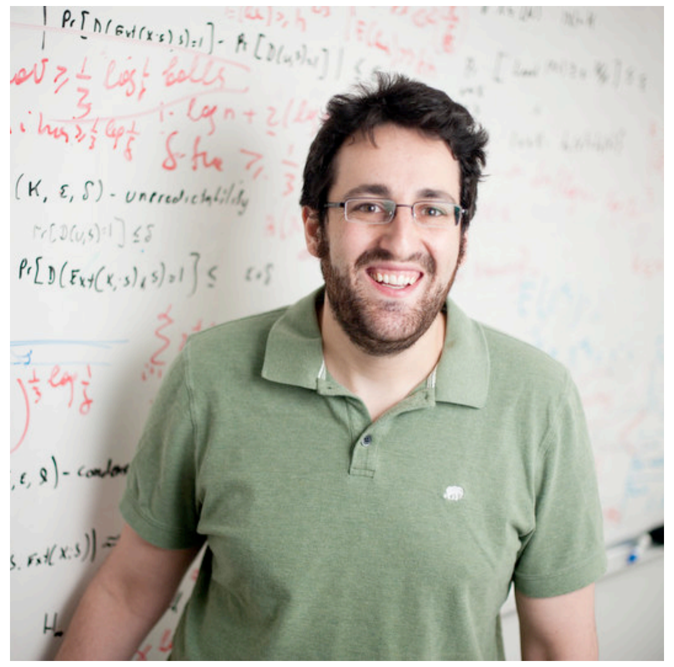
My Research:

TCS ➤ Cryptography ➤ Multiparty Computation ➤ Threshold Crypto ➤ Practical

Most Importantly:

I am a new professor and this is the first 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/2025/Fall/CS6222

☝ All Course Details Here ☝