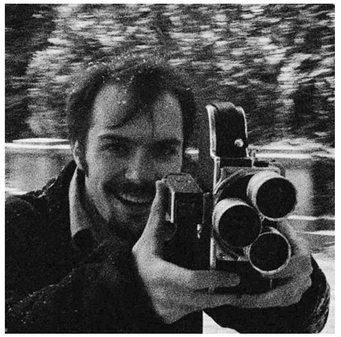
CS6600: Grad Cryptography

Instructor
Jack Doerner
jhd3pa@virginia.edu
Rice 106



TA
Jinye He (Clara)
qfn5bh@virginia.edu

https://jackdoerner.net/teaching/2026/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 = ♥♣

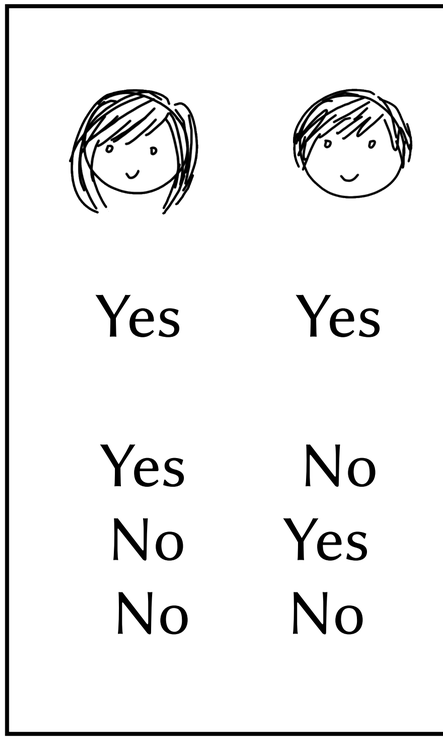
Yes = ♥♣
No = ♣♥

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

Does this protocol produce a correct result?

Yes = ♣♥
No = ♥♣

Yes = ♥♣
No = ♣♥



Does this protocol produce a correct result?

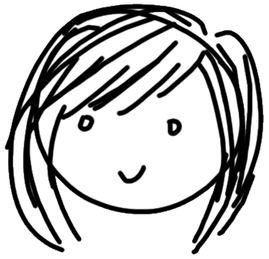
Yes = ♣♥
No = ♥♣

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.

Anchoring the security of many schemes in a common set of assumptions give us an objective basis for comparison: we can know which are risker or safer without reference to any specific attack.

*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.

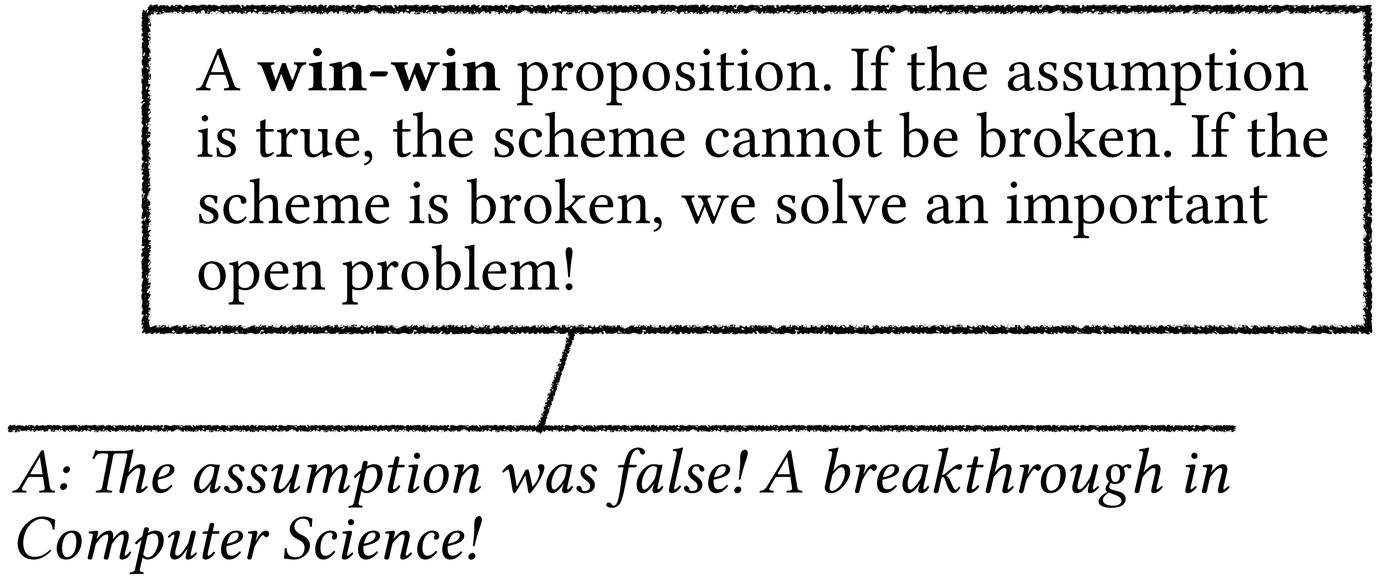
Q: What does it mean when a cryptosystem is broken?

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.



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

It really does go both ways!

Quasipolynomial Cryptanalysis of the McEliece Cryptosystem
(or: PIR Meets McEliece)

Ashrujit Ghoshal* Yuval Ishai† Aayush Jain‡ Nuozhou Sun§

August 7, 2026

Abstract

The McEliece code-based cryptosystem, utilizing binary Goppa codes, is the earliest public-key encryption scheme that is still considered post-quantum secure. We present a simple, classical quasipolynomial-time distinguisher for Goppa–McEliece in the asymptotic "Classic McEliece" regime: for code length n, extension degree $m = \Theta(\log n)$, Goppa degree $t = \Theta(n/\log n)$, and public-code dimension $k = \Theta(n)$, the algorithm runs in time $n^{\mathcal{O}(\log n)}$ and distinguishes the McEliece public key from the uniform distribution over $\mathbb{F}_2^{k\times n}$ with advantage $1 - o(1)$. The distinguisher is not merely asymptotic: it applies to all Classic McEliece parameter sets considered in the NIST process and yields improved (though not yet practical) concrete attack estimates.

Our distinguishing attack originated from a failed attempt to construct doubly efficient private information retrieval (PIR) protocols from algebraic locally decodable codes, and can be intuitively explained from the PIR perspective. We extend this provable algorithm to a heuristic $n^{\mathcal{O}(\log n)}$-time ciphertext-decryption attack that recovers the message from a noisy codeword.

The authors of this paper tried to construct a new cryptosystem from a 50-year-old assumption.

Many people have previously studied the assumption, nobody could figure out if it was true.

These authors noticed an attack on their new cryptosystem. This means they invalidated the assumption and solved a longstanding problem!

What happened? The construction contains the insight needed to perform the attack. Constructing cryptography and breaking it are two sides of the same coin.

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.

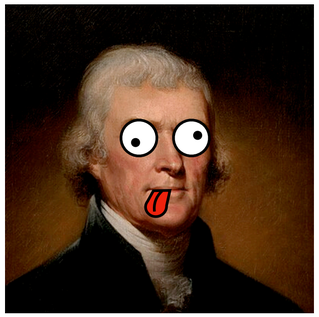
the art of writing or solving codes

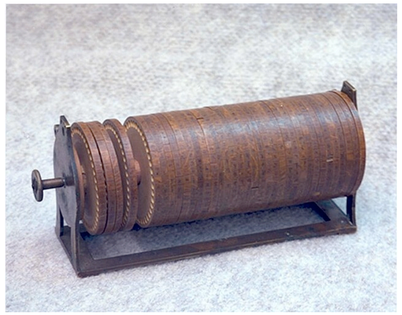
(above: some historical cryptographers)

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

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
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*)
Proofs that can be verified more efficiently than reading the witness
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
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*)
Proofs that can be verified more efficiently than reading the witness
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 $\Longrightarrow$ $P \neq NP$

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).

   You will be able to rigorously assess the truth of a security claim, without trusting the person who made it.

   AI might be able to implement things, but it can also lie. You must be able to distinguish truth from falsehood *on your own*.

   Theory is a necessary anchor for your understanding.

The Goals of this Course:

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

   Maybe you've heard the cybersecurity dogma "don't roll your own crypto."

   The goal in this class is to give you the (the beginnings of) the mental framework you need to design new cryptography and achieve new security properties *without* taking new risks or enabling new attacks.

The Goals of this Course:

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

   This way of thinking is valuable elsewhere.

   e.g. taking a similar approach to problems in machine learning gave us the field of Differential Privacy and helped us make progress on problems in fairness, adversarial ML, etc.

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

Helpful but not required: abstract algebra, number theory

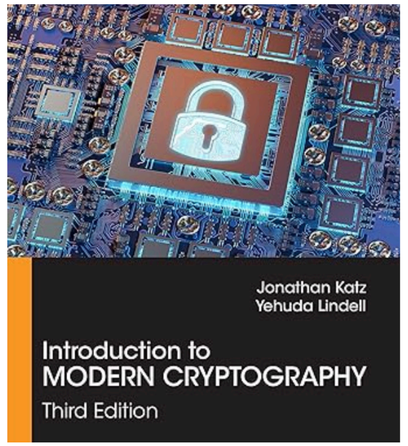
Free Online Textbook:

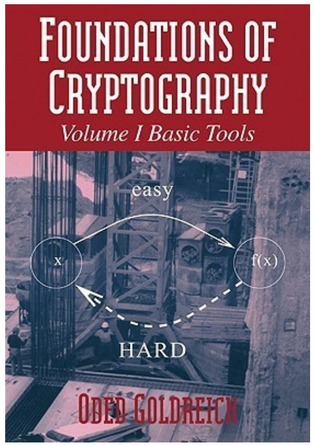
A COURSE IN CRYPTOGRAPHY

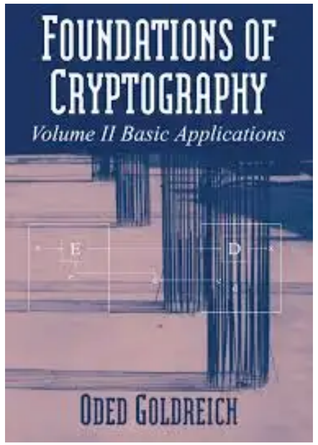
RAFAEL PASS
ABHI SHELAT

He used to work here...

Physical Textbooks (free access via UVa):







Coursework (tentative):

4 Homeworks: 60% of final grade
Solved Collaboratively - see course website

1-3 Scribe Notes: 20% 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.

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?
Non-malleable Encryption?
Obfuscation?
Succinct Proofs?
Multiparty Computation?

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?

Part 2: *Advanced Cryptography*

Zero-knowledge Proofs
Public-Key Encryption
(Fully) Homomorphic Encryption?
Non-malleable Encryption?
Obfuscation?
Succinct Proofs?
Multiparty Computation?

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?

Not on the syllabus:
(but ask about them anyway!)

Historical Cryptosystems
Cryptanalysis*
“Mathematical” crypto beyond the basics
(e.g. Elliptic Curves, Class Groups, Isogenies, Lattices, Number Theory Stuff)
Implementations
Systems security, “Cybersecurity”
Blockchains, Cryptocurrency
Quantum Computing
Post-quantum Cryptography†
Secure or private AI/ML

*actually, we will spend most of the course constructing elaborate ways to break basic assumptions, but we will not succeed

†many things in the course will be post-quantum, but we won’t prove it

About Me

I was a Student Here

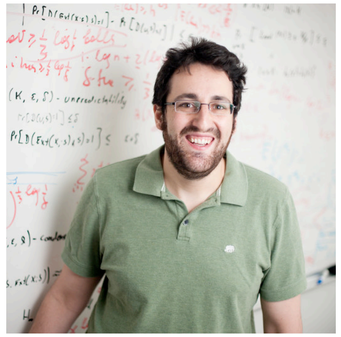
My Research:

TCS ➤ Cryptography ➤
Multiparty Computation ➤
Threshold Crypto ➤ Practical

Most Importantly:

I am a new-ish professor and this is the third 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/Fall/CS6222

☝ All Course Details Here ☝