CSE107: Intro to Modern Cryptography

https://cseweb.ucsd.edu/classes/sp22/cse107-a/

Emmanuel Thomé

March 29, 2022

Lecture 1

Introduction

Course Information

CSE 107 — Introduction to Modern Cryptography

Instructor: Emmanuel Thomé
TAs: Yuka Chu, Rishabh Ranjan, Laura Shea

Website: https://cseweb.ucsd.edu/classes/sp22/cse107-a/

Course Mechanics

40% ≈ 8 Homeworks

- Write up solutions and code yourself
- General discussion in small groups with classmates is encouraged
- Credit the people you talk with
- Usually, homeworks will be available after class every Thursday, and due at 12pm on the next Thursday.

20% Midterm

- Tue, Apr 26 12:30pm (in class)
- No collaboration
- In person

40% Final exam

- Mon, Jun 6 11:30am - 2:30pm (room TBA).
- In person

Course Policies

Late days:

- Get four late days (1 day is one 24 hour period)
- Can use one late day per assignment
- No credit for late work after late days are used up
- These should cover all your normal extension needs
- Contact us for other arrangements if you're in the hospital etc.

Regrade policy:

- Regrades should be the exception not the norm
- Incorrect regrade request $\Rightarrow$ negative points

Academic integrity:

- Please don't cheat, it's a huge pain for everyone
- UC San Diego policy: https://academicintegrity.ucsd.edu
- We *have* to report suspected cases
- If you are not sure if something is cheating, ask

Logistics

Participation and attendance:

- We are still in a pandemic.
- Do *not* show up to class if you have any cold symptoms.
- For this reason, I am not going to police class attendance.
- However, I will be sad if lectures are entirely empty.
- Please do not structure your life so that you cannot ever attend any of the class in person. This is not a remote class.
- *Everything* is subject to change. We cannot predict the course of the pandemic any better than you can.

Course Resources

- No official textbook.

Course Resources

- No official textbook.
- Web site:

  https://cseweb.ucsd.edu/classes/sp22/cse107-a/

  - Various policies
  - Lecture and assignment schedule
  - Lecture slides
  - Schedule updates

Course Resources

- No official textbook.
- Web site:

  https://cseweb.ucsd.edu/classes/sp22/cse107-a/

  - Various policies
  - Lecture and assignment schedule
  - Lecture slides
  - Schedule updates
- Canvas: Links to Zoom, Gradescope, Piazza.

Course Resources

- No official textbook.
- Web site:

  https://cseweb.ucsd.edu/classes/sp22/cse107-a/
  - Various policies
  - Lecture and assignment schedule
  - Lecture slides
  - Schedule updates
- Canvas: Links to Zoom, Gradescope, Piazza.
- Gradescope: Submit assignments

Course Resources

- No official textbook.
- Web site:

  https://cseweb.ucsd.edu/classes/sp22/cse107-a/
  - Various policies
  - Lecture and assignment schedule
  - Lecture slides
  - Schedule updates
- Canvas: Links to Zoom, Gradescope, Piazza.
- Gradescope: Submit assignments
- Zoom + Podcast
  - Zoom links in Canvas
  - Recordings posted to Canvas + Podcast

Course Resources

- No official textbook.
- Web site:

  https://cseweb.ucsd.edu/classes/sp22/cse107-a/
  - Various policies
  - Lecture and assignment schedule
  - Lecture slides
  - Schedule updates
- Canvas: Links to Zoom, Gradescope, Piazza.
- Gradescope: Submit assignments
- Zoom + Podcast
  - Zoom links in Canvas
  - Recordings posted to Canvas + Podcast
- Piazza: Asynchronous Q&A

Homework 0 will be available on Gradescope after lecture.

HW0 is an exception

- It is a Gradescope online form.
- Due on Tue 4/5 12pm (in 1 week).

Homework 1 will be available after lecture on Thursday and due on Thursday 4/7 12pm.

About me

PhD in France some decades ago. Spent a year in Chicago as a visitor while I was beginning my PhD (previous millenium).

Full-time researcher at Inria, France, since 2003, where I've been leading a research group since 2015. (group = 8 permanent full-time researchers + 2 permanent faculty. Small number of students).

Currently in San Diego on a Fulbright grant, which is a fantastic way to visit the world. Works both **to** and **from** the US, both for students and researchers.

Visiting professor in the CSE department this year.

My work

My research is mostly about cryptanalysis and computational number theory, in the context of public-key cryptography.

- What algorithms can we design to deal with the supposedly hard mathematical problems that underpin public-key cryptography?
- What algorithms can we design to create new public-key cryptographic primitives?
- What can we do in practice?
- What is the impact of this in terms of computer security.

Part of my work is of mathematical nature. Another part of it is the implementation of complicated algorithms, and lots of computer code.

Researchers reveal a method the NSA may use to spy on Web traffic

By Sean Sposito | October 21, 2015 | Updated: October 21, 2015 5:05pm

ars TECHNICA BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE FORUMS SIGN IN

RISK ASSESSMENT —

NSA could put undetectable “trapdoors” in millions of crypto keys

Technique allows attackers to passively decrypt Diffie-Hellman protected data.

DAN GOODIN - 10/11/2016, 7:30 AM

Many amazing folks at UCSD working on crypto/security

Russell Impagliazzo	Daniele Micciancio	Mihir Bellare	Nadia Heninger	Deian Stefan	Aaron Schulman	Alex Snoeren	Stefan Savage	Geoff Voelker
								

Theory — Applied

Crypto

Systems

Nadia Polikarpova	Ranjit Jhala	Sorin Lerner	kc Claffy	Lawrence Saul	Ryan Kastner
					

PL & Verification — Networking — ML — Embedded

Topics Covered

- Provable Security
  - Security models and reductions
- Symmetric cryptography
  - Block ciphers, symmetric encryption, hash functions, message authentication, authenticated encryption
- Asymmetric cryptography
  - Key distribution, RSA and discrete logarithm based systems, digital signatures
- Protocols and Applications
  - TLS, zero-knowledge proofs, PKI, etc.

Course Goals

- Critical thinking
  - How to reason about cryptographic security
  - Better understanding of cryptographic solutions

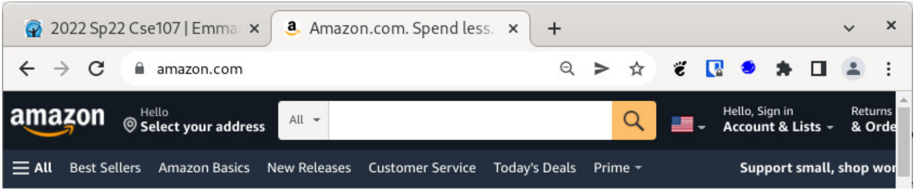
Course Goals

- Critical thinking
  - How to reason about cryptographic security
  - Better understanding of cryptographic solutions
- CS skills
  - Proofs, reductions, CS theory
  - Applied math and number theory

Cryptography usage

Did you use any cryptography today?

Cryptography usage



- https invokes the TLS protocol
- TLS uses cryptography
- TLS is in ubiquitous use for secure communication: shopping, banking, Netflix, gmail, Facebook, ...

Secure messaging apps

WhatsApp, Signal, iMessage/FaceTime, Viber, Telegram, LINE, Threema, ChatSecure, KakaoTalk, ...

Which ones do you use?

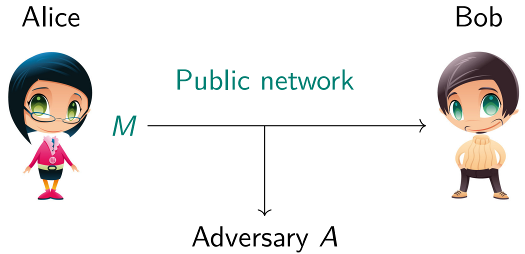
Cryptography usage

Other uses of cryptography

- ATMs
- Bitcoin or other cryptocurrencies
- Tor: Anonymous web browsing
- Google authenticator, Duo
- ...

11,748 android apps use cryptography (encryption), and 10,327 get it wrong [EBFK13]

What is cryptography about?



Adversary: clever person with powerful computer

Security goals:

- **Data privacy:** Ensure adversary does not see or obtain the data (message) M.
- **Data integrity and authenticity:** Ensure M really originates with Alice and has not been modified in transit.

Example: Medical databases

Doctor | Database

Get Alice →

Alice	F_A
Bob	F_B

← F_A

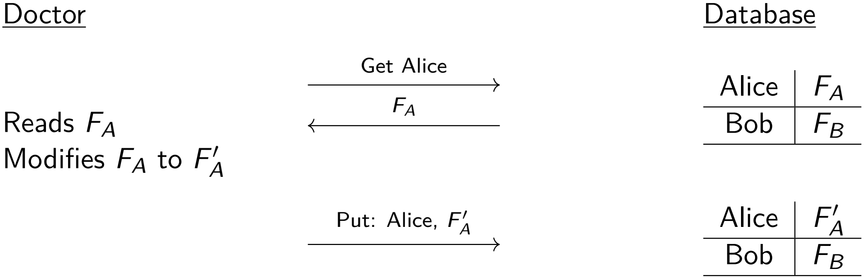
Reads F_A

Modifies F_A to F'_A

Put: Alice, F'_A →

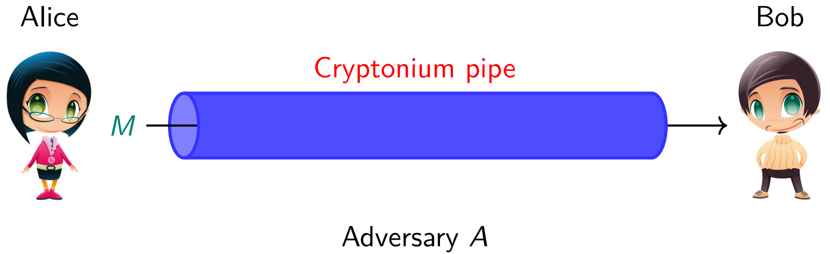
Alice	F'_A
Bob	F_B

Example: Medical databases



- Privacy: F_A, F'_A contain confidential information and we want to ensure the adversary does not obtain them
- Integrity and authenticity: Need to ensure
  - doctor is authorized to get Alice's file
  - F_A, F'_A are not modified in transit
  - F_A is really sent by database
  - F'_A is really sent by (authorized) doctor

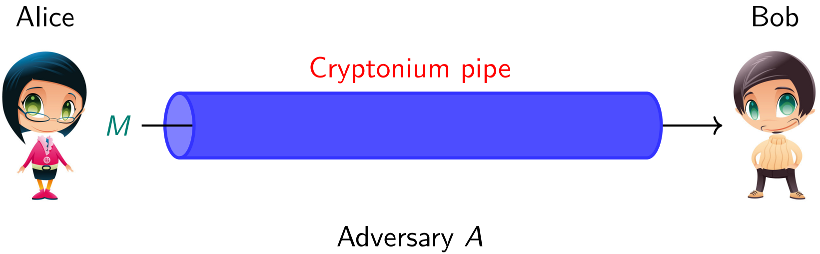
Ideal World



Cryptonium pipe: Cannot see inside or alter content.

All our goals would be achieved!

Ideal World

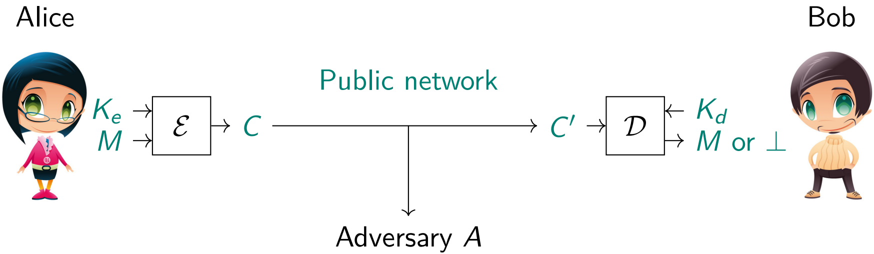


Cryptonium pipe: Cannot see inside or alter content.

All our goals would be achieved!

But cryptonium is only available on planet Crypton and is in short supply. ☹

Cryptographic schemes

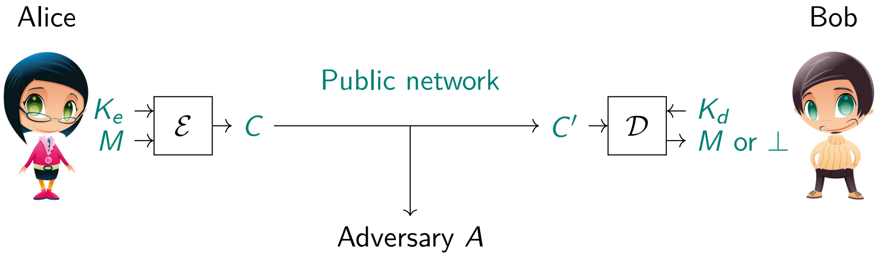


$\mathcal{E}$: encryption algorithm K_e: encryption key

$\mathcal{D}$: decryption algorithm K_d: decryption key

Algorithms: standardized, implemented, public!

Cryptographic schemes



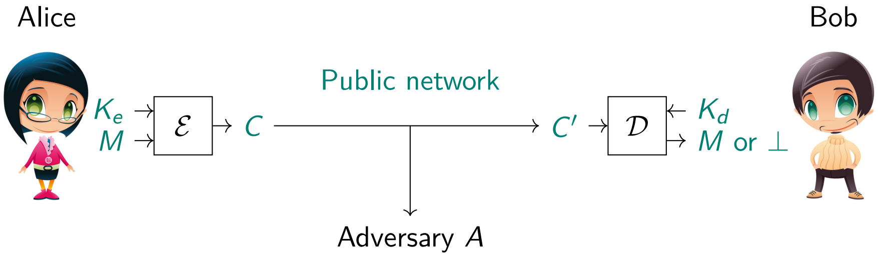
$\mathcal{E}$: encryption algorithm $\quad$ K_e: encryption key

$\mathcal{D}$: decryption algorithm $\quad$ K_d: decryption key

Settings:

- secret-key (symmetric): $K_e = K_d$ secret
- public-key (asymmetric): K_e public, K_d private

Cryptographic schemes

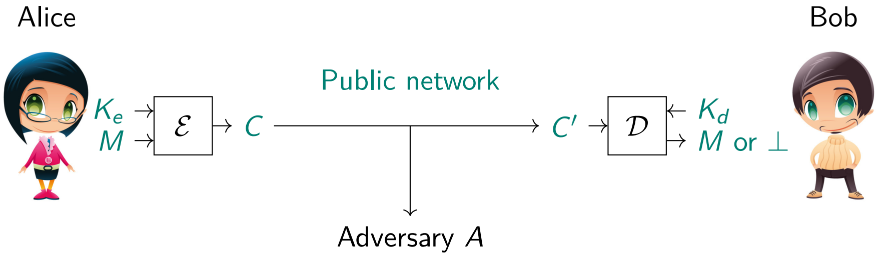


$\mathcal{E}$: encryption algorithm — K_e: encryption key

$\mathcal{D}$: decryption algorithm — K_d: decryption key

How do keys get distributed? Magic, for now!

Cryptographic schemes



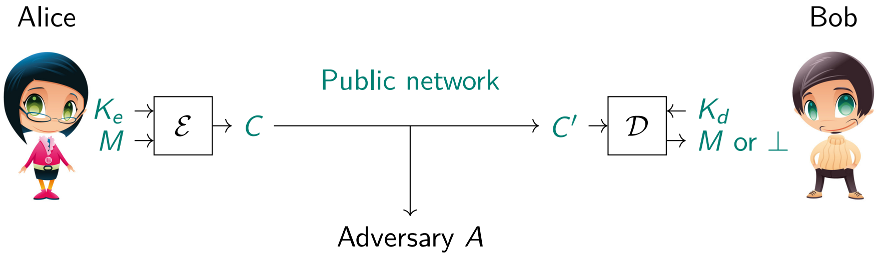
$\mathcal{E}$: encryption algorithm $\quad$ K_e: encryption key

$\mathcal{D}$: decryption algorithm $\quad$ K_d: decryption key

Our concerns:

- How to define security goals?
- How to design $\mathcal{E}$, $\mathcal{D}$?
- How to gain confidence that $\mathcal{E}$, $\mathcal{D}$ achieve our goals?

Cryptographic schemes



$\mathcal{E}$: encryption algorithm K_e: encryption key
$\mathcal{D}$: decryption algorithm K_d: decryption key

Computer Security: How does the computer/system protect K_e/K_d from attack (bugs, memory safety, OS vulnerabilities, ...)? (CSE 127,227)

Cryptography: How do we use K_e, K_d to ensure security of communication over an insecure network? (CSE 107,207)

Why is cryptography hard?

- One cannot anticipate an adversary strategy in advance; number of possibilities is infinite.
- “Testing” is not possible in this setting.

Early history

Substitution ciphers/Caesar ciphers:

$$K_e = K_d = \pi \colon \Sigma \to \Sigma, \text{a secret permutation}$$

e.g., $\Sigma = \{A, B, C, \ldots\}$ and π is as follows:

σ	A	B	C	D	$\cdots$
$\pi(\sigma)$	E	A	Z	U	$\cdots$

$$\begin{aligned} \mathcal{E}_\pi(CAB) &= \pi(C)\pi(A)\pi(B) \\ &= Z\ \ E\ \ A \\ \mathcal{D}_\pi(ZEA) &= \pi^{-1}(Z)\pi^{-1}(E)\pi^{-1}(A) \\ &= C\ \ A\ \ B \end{aligned}$$

Early history

Substitution ciphers/Caesar ciphers:

$$K_e = K_d = \pi \colon \Sigma \to \Sigma, \text{a secret permutation}$$

e.g., $\Sigma = \{A, B, C, \ldots\}$ and π is as follows:

σ	A	B	C	D	$\cdots$
$\pi(\sigma)$	E	A	Z	U	$\cdots$

$$\begin{aligned} \mathcal{E}_\pi(CAB) &= \pi(C)\pi(A)\pi(B) \\ &= Z\ E\ A \\ \mathcal{D}_\pi(ZEA) &= \pi^{-1}(Z)\pi^{-1}(E)\pi^{-1}(A) \\ &= C\ A\ B \end{aligned}$$

Not very secure! (Common newspaper puzzle)

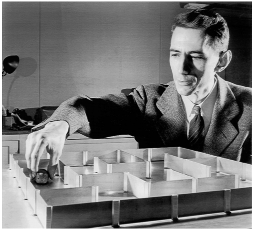
The age of machines

Enigma: German World War II machine

Broken by British in an effort led by Turing

Shannon and One-Time-Pad (OTP) Encryption

$$K_e = K_d = \underbrace{K \xleftarrow{\$} \{0,1\}^k}_{K \textit{ chosen at random from } \{0,1\}^k}$$

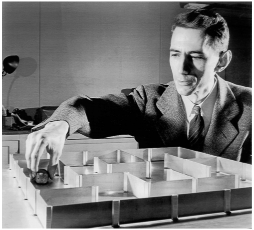


For any $M \in \{0,1\}^k$

- $\mathcal{E}_K(M) = K \oplus M$
- $\mathcal{D}_K(C) = K \oplus C$

Shannon and One-Time-Pad (OTP) Encryption

$$K_e = K_d = \underbrace{K \stackrel{\$}{\leftarrow} \{0,1\}^k}_{K \textit{ chosen at random from } \{0,1\}^k}$$



For any $M \in \{0,1\}^k$

- $\mathcal{E}_K(M) = K \oplus M$
- $\mathcal{D}_K(C) = K \oplus C$

Theorem (Shannon): OTP is perfectly secure as long as only one message encrypted.

"Perfect" secrecy, a notion Shannon defines, captures mathematical impossibility of breaking an encryption scheme.

Fact: if $|M| > |K|$, then no scheme is perfectly secure.

Modern Cryptography: A Computational Science

Security of a "practical" system must rely not on the impossibility but on the computational difficulty of breaking the system.

("Practical" = more message bits than key bits)

Rather than:

"It is impossible to break the scheme"

We might be able to say:

"No attack using $\leq 2^{160}$ time succeeds with probability $\geq 2^{-20}$"

I.e., Attacks can exist as long as cost to mount them is prohibitive, where Cost = computing time/memory, $$$

Modern Cryptography: A Computational Science

Security of a "practical" system must rely not on the impossibility but on the computational difficulty of breaking the system.

Cryptography is now not just mathematics; it needs to draw on computer science

- Computational complexity theory (CSE 105,200)
- Algorithm design (CSE 101,202)

The factoring problem

Input: Composite integer N
Desired output: prime factors of N

Example:
Input: 85
Output:

The factoring problem

Input: Composite integer N

Desired output: prime factors of N

Example:

Input: 85

Output: $17, 5$

The factoring problem

Input: Composite integer N
Desired output: prime factors of N

Example:
Input: 85
Output: $17, 5$

Can we write a factoring program?

The factoring problem

Input: Composite integer N
Desired output: prime factors of N

Example:
Input: 85
Output: $17, 5$

Can we write a factoring program? Easy!

Alg Factor(N) // N a product of 2 primes
For $i = 2, 3, \ldots, \lceil \sqrt{N} \rceil$ do
If $N \bmod i = 0$ then return i

The factoring problem

Input: Composite integer N
Desired output: prime factors of N

Example:
Input: 85
Output: $17, 5$

Can we write a factoring program? Easy!

Alg Factor(N) // N a product of 2 primes
For $i = 2, 3, \ldots, \lceil \sqrt{N} \rceil$ do
If $N \bmod i = 0$ then return i

But this is very slow ... 
Prohibitive if N is large (e.g., 400 digits)

Can we factor fast?

- Gauss couldn't figure out how
- Today there is no known algorithm to factor a 400 digit number in a practical amount of time.

Factoring is an example of a problem believed to be computationally hard.

Note 1: A fast algorithm MAY exist.

Note 2: A quantum computer can factor fast! One has not yet been built but efforts are underway ...

State of the art

$$\begin{aligned}\text{RSA-250} &= 214032465024074\ldots(\text{250 digits in total})\ldots494975937497937\\ &= 6413528947\ldots(\text{125 digits in total})\ldots9798853367\\ &\quad\times 3337202759\ldots(\text{125 digits in total})\ldots6932062711\end{aligned}$$

This required 2700 core-years of computation.

"We used computer resources of the Grid'5000 experimental testbed in France (INRIA, CNRS, and partner institutions), of the EXPLOR computing center at Université de Lorraine, Nancy, France, an allocation of computing hours on the PRACE research infrastructure using resources at the Juelich supercomputing center in Germany, as well as computer equipment gifted by Cisco Systems, Inc. at UCSD."

Atomic Primitives or Problems

Examples:

- Factoring: Given large $N = pq$, find p, q
- Block cipher primitives: DES, AES, ...
- Hash functions: MD5, SHA1, SHA3, ...

Atomic Primitives or Problems

Examples:

- Factoring: Given large $N = pq$, find p, q
- Block cipher primitives: DES, AES, ...
- Hash functions: MD5, SHA1, SHA3, ...

Features:

- Few such primitives
- Design an art, confidence by history.

Atomic Primitives or Problems

Examples:

- Factoring: Given large $N = pq$, find p, q
- Block cipher primitives: DES, AES, ...
- Hash functions: MD5, SHA1, SHA3, ...

Features:

- Few such primitives
- Design an art, confidence by history.

Drawback: Don't directly solve any security problem.

Higher Level Primitives

Goal: Solve security problem of direct interest.

Examples: encryption, authentication, digital signatures, key distribution, . . .

Higher Level Primitives

Goal: Solve security problem of direct interest.

Examples: encryption, authentication, digital signatures, key distribution, ...

Features:

- Lots of them

Lego Approach

We typically design high-level primitives from atomic ones

Atomic primitive

↓

Transformer

↓

High-level primitive

Defining security

A great deal of design tries to produces schemes without first asking:

"What exactly is the security goal?"

This leads to schemes that are complex, unclear, and wrong.

Being able to precisely state what is the security goal of a design is challenging but important.

We will spend a lot of time developing and justifying strong, precise notions of security.

Thinking in terms of these precise goals and understanding the need for them may be the most important thing you get from this course!

Defining Security

What does it mean for an encryption scheme to provide privacy?

Defining Security

What does it mean for an encryption scheme to provide privacy?

Does it mean that given $C = \mathcal{E}_{K_e}(M)$, adversary cannot

- recover M?
- recover the first bit of M?
- recover the XOR of the first and the last bits of M?
- ...

Defining Security

What does it mean for an encryption scheme to provide privacy?

Does it mean that given $C = \mathcal{E}_{K_e}(M)$, adversary cannot

- recover M?
- recover the first bit of M?
- recover the XOR of the first and the last bits of M?
- ...

We will provide a formal definition for privacy, justify it, and show it implies the above (and more).

Cryptography in practice

Schemes designed via the principles we will study are in use (TLS, SSH, IPSec, . . .): HMAC, RSA-OAEP, ECIES, Ed25519, CMAC, GCM, . . .

New uses for old mathematics

Cryptography uses

- Number theory
- Combinatorics
- Modern algebra
- Probability theory

Modern Cryptography: Esoteric mathematics?

Hardy, in his essay A Mathematician's Apology writes:

> *"Both Gauss and lesser mathematicians may be justified in rejoicing that there is one such science [number theory] at any rate, and that their own, whose very remoteness from ordinary human activities should keep it gentle and clean"*

No longer: Number theory is the basis of modern public-key systems such as RSA.

Security today

- Server breaches, malware
- Compromise of people's private information leading to identity theft, credit-card fraud, ...
- Lack of privacy: Information about us is collected and harvested
- Mass surveillance: Snowden Revelations

2017 Equifax breach exposed 143 million social security numbers.

Cryptography is a central tool in getting more security and privacy.

Cryptography on the horizon

Computing on encrypted data

- Searchable encryption
- Homomorphic encryption
- multi-party computation
- garbled circuits
- ...

What you can get from this course

Be able to

- Identify threats
- Evaluate security solutions and technologies
- Design high-quality solutions
- Develop next-generation privacy tools
- ...

If nothing else, develop a healthy sense of paranoia!

How to do well in CSE 107

Characteristics of the successful 107 student:

- More interested in learning than grades
- Likes challenges, does not give up easily
- Tries to understand *all* the material, not just some of it
- Questions are more often about the material (slides) than about how to do the homework.
- Understands theory behind examples.

If you take the course with the view that you only want to pass, you increase the risk of not passing. If you take it aiming to get an A and are willing to work for it, you may very well get one.

How to do well in CSE 107

Doesn't work too well: Random access mode, in which you look at homework problem, then try to find something in slides that "matches" it.

Works well: Sequential mode, where you first go through all the slides, sequentially, and make sure you understand the material, and THEN attempt homework.

Some students expect a recipe for success: "I am willing to work hard. Just tell me what to do!"

We are not aware of any such recipe. Different people understand things in different ways and have different paths to success. You will find your own!