Course Introduction

COMPSCI 466 - Spring 2020

Adam O’Neill

Class Logistics

Instructor: Adam O'Neill; adamo@cs.mass.edu

TAs: Ojaswi Acharya, Weiqi Feng

Course Website: https://people.cs.umass.edu/~adamo/sp20466/

Office Hours: TBD. Fill out poll on website!

Grades are based on 50% problem sets, 25% midterm, 25% final.

Details in syllabus.

Rules in Brief

On problem sets:

Rules in Brief

On problem sets:

- List your collaborators on each problem.

Rules in Brief

On problem sets:

- List your collaborators on each problem.
- Write (type-up!) your final solutions by yourself as if you are taking an exam.

Rules in Brief

On problem sets:

- List your collaborators on each problem.
- Write (type-up!) your final solutions by yourself as if you are taking an exam.

Rules in Brief

On problem sets:

- List your collaborators on each problem.
- Write (type-up!) your final solutions by yourself as if you are taking an exam.

Do not look at any solutions to assignments/exams from previous courses or material other than what is approved on website.

Rules in Brief

On problem sets:

- List your collaborators on each problem.
- Write (type-up!) your final solutions by yourself as if you are taking an exam.

Do not look at any solutions to assignments/exams from previous courses or material other than what is approved on website.

Rules in Brief

On problem sets:

- List your collaborators on each problem.
- Write (type-up!) your final solutions by yourself as if you are taking an exam.

Do not look at any solutions to assignments/exams from previous courses or material other than what is approved on website.

Details in syllabus.

Rules in Brief

On problem sets:

- List your collaborators on each problem.
- Write (type-up!) your final solutions by yourself as if you are taking an exam.

Do not look at any solutions to assignments/exams from previous courses or material other than what is approved on website.

Details in syllabus.

Failure to adhere to these rules will be handled accordingly.

Some advice

- This is going to be a hard class…
- Understanding practical cryptography requires understanding the theory behind it. If you haven't done well in theory/math courses, that's a bad sign.
- It is not enough to be hardworking. You may work hard on a homework and get a zero. This reflects what you understand, not how hard you worked. How can you understand better? Some advice at http://www.math.ucsd.edu/~ebender/Supplements/proofs.html
- Strive for concision in all your solutions. Concision and understanding are intimately related. Typically, we are looking for pseudocode, claims of inequalities and their proofs.

What’s Cryptography?

What’s Cryptography?

Broadly, how to communicate and compute in the presence of an adversary.

What's Cryptography?

Broadly, how to communicate and compute in the presence of an adversary.

You may also think of a storage medium (e.g. Gmail server) as the communication channel.

What’s Cryptography?

Broadly, how to communicate and compute in the presence of an adversary.

You may also think of a storage medium (e.g. Gmail server) as the communication channel.

Cryptography is full of counter-intuitive solutions to seemingly impossible problems!

Who Uses Crypto?

Who Uses Crypto?

You probably did, today!

Who Uses Crypto?

You probably did, today!

- TLS/SSL protocol (used for Gmail, Facebook, YouTube, etc.)

Who Uses Crypto?

You probably did, today!

- TLS/SSL protocol (used for Gmail, Facebook, YouTube, etc.)
- Tens of thousands of apps use crypto (many incorrectly...)

Does it matter to you?

If you have something that you don't want anyone to know, maybe you shouldn't be doing it in the first place.

Eric Schmidt, CEO
Google, 2009

Glenn Greenwald

If you have nothing to hide then give me the passwords to ALL your email accounts, your text and chat histories, ...

https://www.youtube.com/watch?v=pcSlowAhvUk

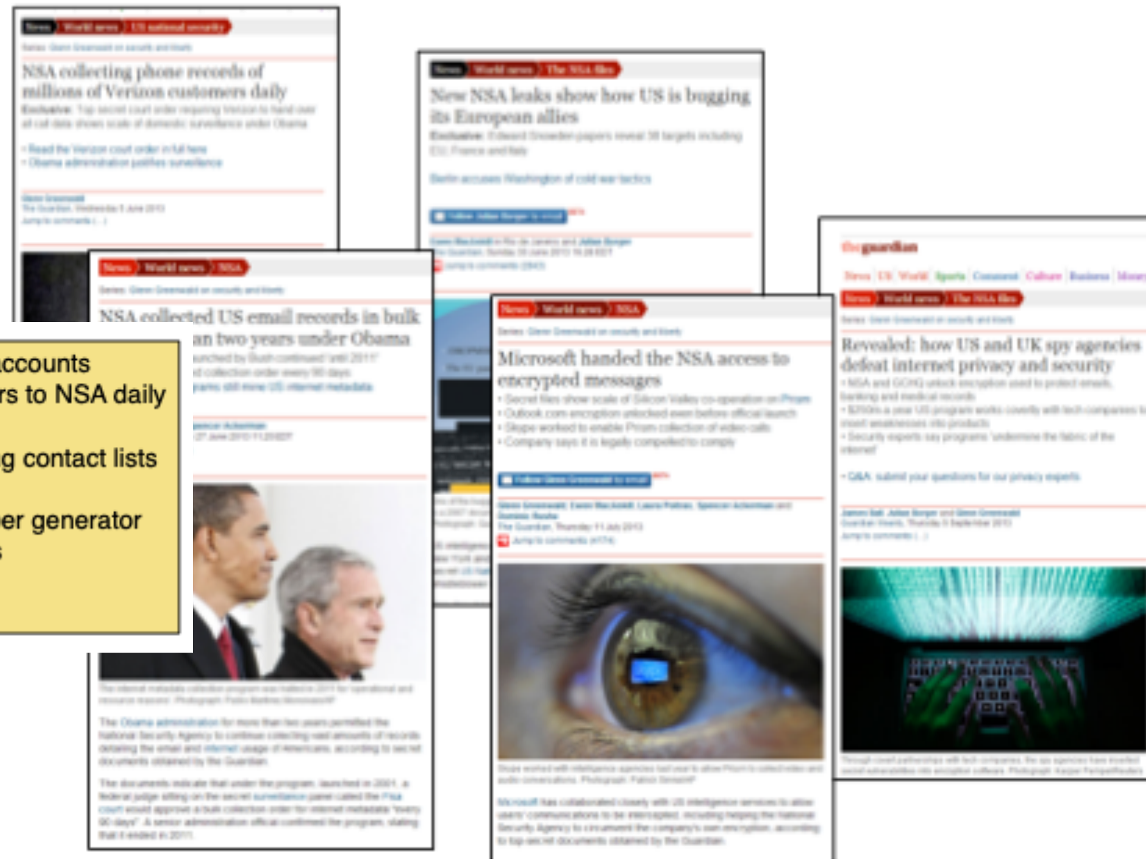
20 minute video of TED talk

The Chronicle of Higher Education
Why privacy matters even if you have nothing to hide
http://chronicle.com/article/Why-Privacy-Matters-Even-if/127461/

Bruce Schneier The Value of Privacy
Privacy protects us from abuses by those in power ... We keep private journals, sing in the shower ... privacy is a basic human need.
https://www.schneier.com/blog/archives/2006/05/the_value_of_pr.html

http://zeroknowledgeprivacy.org/library/why-privacy-matters/

Snowden revelations



Court-approved NSA access to Google and Yahoo accounts
Verizon hands phone records of millions of customers to NSA daily
Extensive wiretapping, tapping undersea cables
Harvesting of millions of email and instant-messaging contact lists
Tracking and mapping location of cellphones
Backdoor planted in Dual_EC_DBRG random-number generator
Paying corporations to adopt NSA-broken standards
Sophisticated malware
...

Laura
Poitras

See the movie!

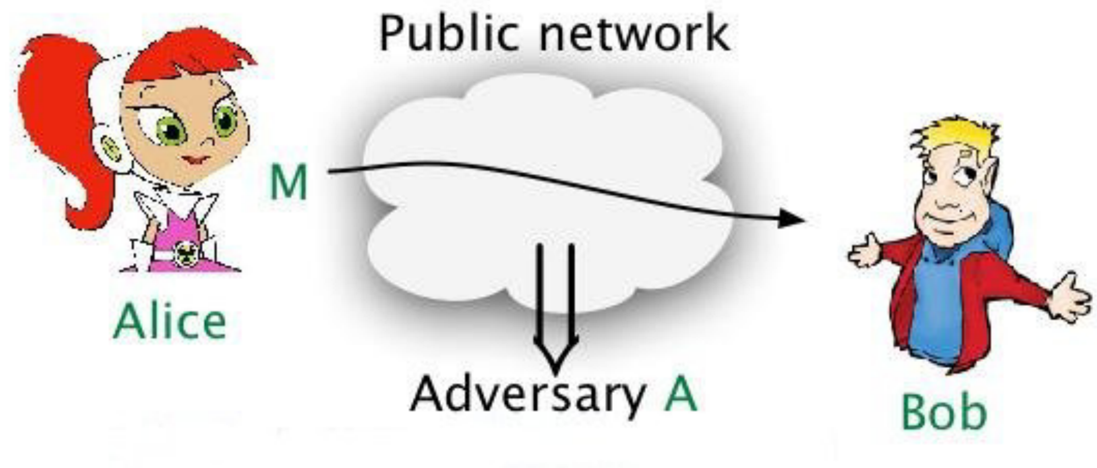
Read the news!

Secure Messaging Apps

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

Use them!

Basic Task: Secure Channels



Adversary: clever person with powerful computer

Desired Properties

Desired Properties

Privacy: Adversary does not learn anything about the message.

Desired Properties

Privacy: Adversary does not learn anything about the message.

Integrity and Authenticity: Bob is assured the message “really came from” Alice and was not modified.

Desired Properties

Privacy: Adversary does not learn anything about the message.

Integrity and Authenticity: Bob is assured the message "really came from" Alice and was not modified.

Desired Properties

Privacy: Adversary does not learn anything about the message.

Integrity and Authenticity: Bob is assured the message "really came from" Alice and was not modified.

These definitions are informal.

“If a clear explanation is provided, then no examples are needed; otherwise, no examples will help” —Oded Goldreich

Example: Medical Databases

<u>Doctor</u> | <u>Database</u>

Get Alice $\longrightarrow$

$\xleftarrow{F_A}$

Alice	F_A
Bob	F_B

Reads F_A

Modifies F_A to F'_A

Put: Alice, F'_A $\longrightarrow$

Alice	F'_A
Bob	F_B

- 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

Modern Cryptography

Modern Cryptography

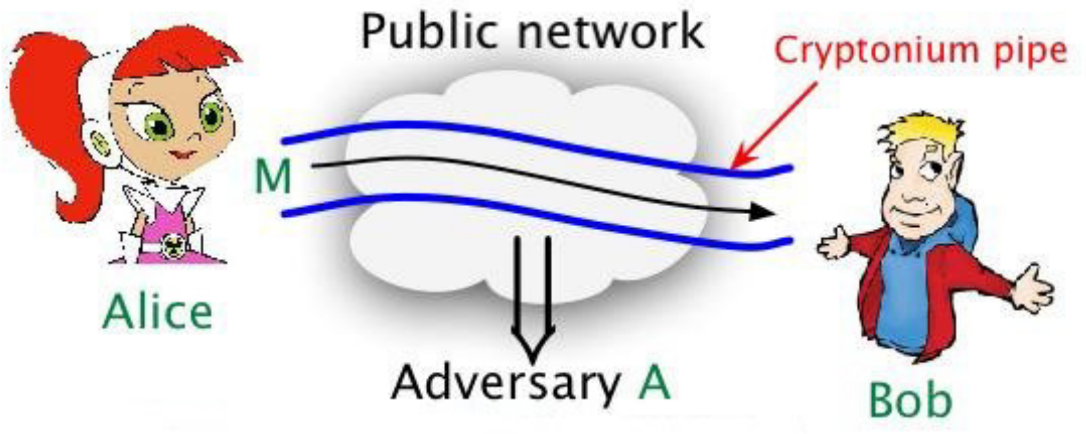
Modern cryptography deals with how to formalize and provably achieve such properties.

Modern Cryptography

Modern cryptography deals with how to formalize and provably achieve such properties.

Grew out of theoretical computer science in the 1980s, and has seen a surge of interest due to the Internet.

Ideal world

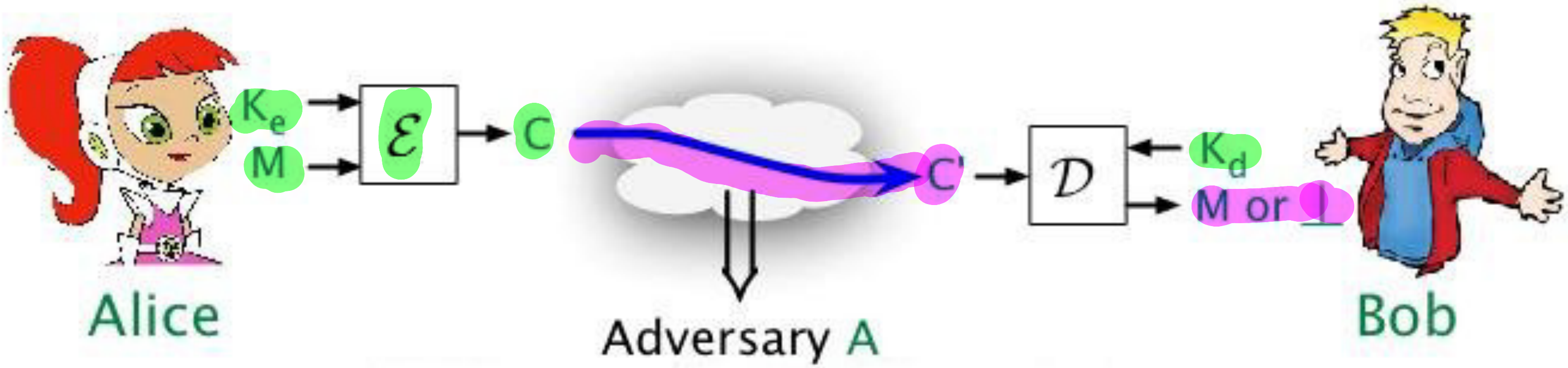


Cryptonium pipe: Cannot see inside or alter content.

All our goals would be achieved!

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

Real World



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

Algorithms: standardized, implemented, public!

Our Concerns

- How to define security goals?

Our Concerns

- How to define security goals?
- How to design encryption and decryption algorithms?

Our Concerns

- How to define security goals?
- How to design encryption and decryption algorithms?
- How to gain confidence our designs achieve our security goals?

Early History

Substitution ciphers/Caesar ciphers:

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

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

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

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

Our Def of Substitution Cipher

The encryption and decryption algorithms are able to be written as

$\mathcal{K}$ outputs some $\pi : \Sigma \to \Sigma$

Algorithm $\mathcal{E}_\pi(M)$	Algorithm $\mathcal{D}_\pi(C)$
For $i = 1, \ldots, \lvert M \rvert$ do	For $i = 1, \ldots, \lvert C \rvert$ do
$C[i] \leftarrow \pi(M[i])$	$M[i] \leftarrow \pi^{-1}(C[i])$
Return C	Return M

assume the scheme is punctuation-preserving.

Cryptanalysis

how many possible keys? 26!

Suppose adversary has ciphertext:

COXBX TBX CVK CDGXR DI T GTI'R ADHX VOXI OX ROKQAU
IKC RNXPQATCX: VOXI OX PTI'C THHKBU DC, TIU VOXI OX
PTI.

Frequency

```
  E E   E        E          ’     E   E   E
COXBX TBX CVK CDGXR DI T GTI’R ADHX VOXI OX ROKQAU

      E     E:   E   E    ’           ,       E
IKC RNXPQATCX: VOXI OX PTI’C THHKBU DC, TIU VOXI

 E    .
OX PTI.
```

A	B	C	D	E	F	G	H	I	J	K	L	M
3	3	7	4	0	0	2	3	9	0	4	0	0

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	8	3	2	4	0	8	3	4	0	13	0	0

```
  E E   E        E          '     E   E   E
COXBX TBX CVK CDGXR DI T GTI'R ADHX VOXI OX ROKQAU

      E     E:   E   E    '           ,       E
IKC RNXPQATCX: VOXI OX PTI'C THHKBU DC, TIU VOXI

 E    .
OX PTI.
```

```
  E E   E        E          '     E   E   E
COXBX TBX CVK CDGXR DI T GTI'R ADHX VOXI OX ROKQAU

      E     E:   E   E    '           ,       E
IKC RNXPQATCX: VOXI OX PTI'C THHKBU DC, TIU VOXI

 E    .
OX PTI.
```

OX in ciphertext $\Rightarrow \pi^{-1}(\texttt{O}) \in \{\texttt{B},\texttt{H},\texttt{M},\texttt{W}\}$

Guess $\pi^{-1}(\texttt{O}) = \texttt{H}$ since O has pretty high frequency

```
THE E   E T   T  E          '     E  HE  HE  H
COXBX TBX CVK CDGXR DI T GTI'R ADHX VOXI OX ROKQAU

  T   E    TE:  HE  HE    'T         T,      HE
IKC RNXPQATCX: VOXI OX PTI'C THHKBU DC, TIU VOXI

HE    .
OX PTI.
```

```
*HE*E
COXBX
```

Could be: `THERE`,`THESE`,`WHERE`,...

Guess $\pi^{-1}(C) = T$ since there is no ? in ciphertext so `WHERE` is unlikely.

```
THERE ARE T   T  E     A  A '     E  HE  HE  H
COXBX TBX CVK CDGXR DI T GTI'R ADHX VOXI OX ROKQAU

  T   E   ATE:  HE  HE  A 'T A   R   T, A    HE
IKC RNXPQATCX: VOXI OX PTI'C THHKBU DC, TIU VOXI

HE  A .
OX PTI.
```

T is a single-letter word so $\pi^{-1}(T) \in \{\mathtt{A}, \mathtt{I}\}$

We know $\pi^{-1}(\mathtt{B}) \in \{\mathtt{R}, \mathtt{S}\}$

So TBX could be: ARE,ASE,IRE,ISE

We guess ARE

```
THERE ARE T   T  E    A  A  '    E  HE  HE  H    
COXBX TBX CVK CDGXR DI T GTI'R ADHX VOXI OX ROKQAU

  T   E   ATE:  HE  HE  A 'T A   R   T, A    HE 
IKC RNXPQATCX: VOXI OX PTI'C THHKBU DC, TIU VOXI

HE  A .
OX PTI.
```

```
*T
DC
```

D must be: A or I but T is A so D is I.

THERE ARE TWO TIMES IN A MAN'S LIFE WHEN HE SHOULD
COXBX TBX CVK CDGXR DI T GTI'R ADHX VOXI OX ROKQAU

NOT SPECULATE: WHEN HE CAN'T AFFORD IT, AND WHEN
IKC RNXPQATCX: VOXI OX PTI'C THHKBU DC, TIU VOXI

HE CAN.
OX PTI.

Another Example

Example 1 (Cryptography). Stanford's Statistics Department has a drop-in consulting service. One day, a psychologist from the state prison system showed up with a collection of coded messages. Figure 1 shows part of a typical example.

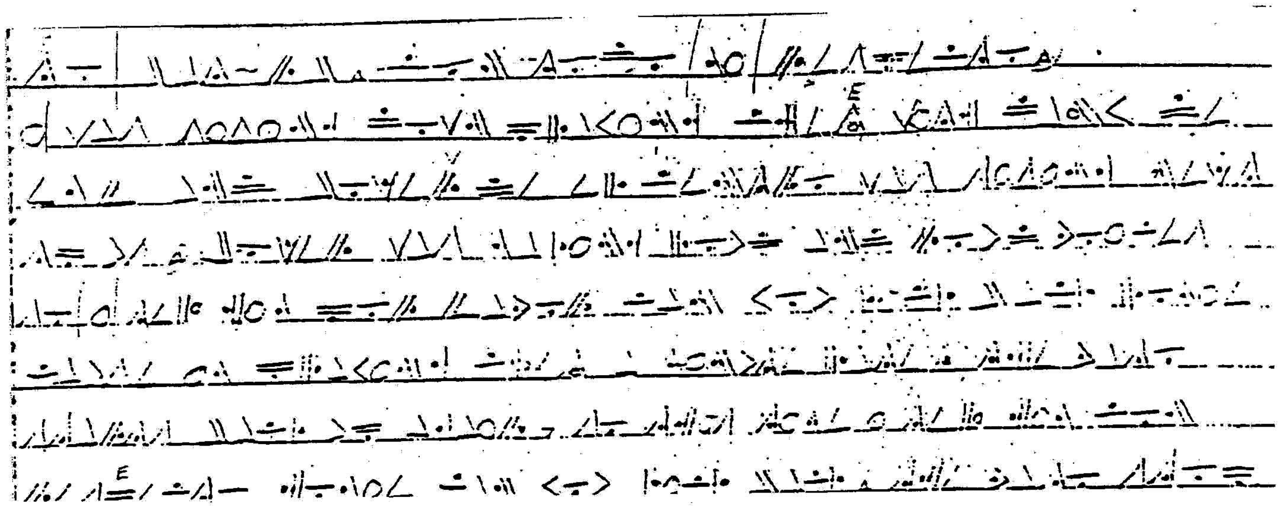


Figure 1

The General Setting

$$f : \{\text{code space}\} \longrightarrow \{\text{usual alphabet}\}.$$

To get the statistics, Marc downloaded a standard text (e.g., *War and Peace*) and recorded the first-order transitions: the proportion of consecutive text symbols from x to y. This gives a matrix $M(x, y)$ of transitions. One may then associate a plausibility to f via

$$\mathrm{Pl}(f) = \prod_i M\left(f(s_i), f(s_{i+1})\right)$$

Markov Chain Monte Carlo

The Metropolis Algorithm

- Start with a preliminary guess, say f.
- Compute $\mathrm{Pl}(f)$.
- Change to f_* by making a random transposition of the values f assigns to two symbols.
- Compute $\mathrm{Pl}(f_*)$; if this is larger than $\mathrm{Pl}(f)$, accept f_*.
- If not, flip a $\mathrm{Pl}(f_*)/\mathrm{Pl}(f)$ coin; if it comes up heads, accept f_*.
- If the coin toss comes up tails, stay at f.

Output of the Algorithm

to bat-rb. con todo mi respeto. i was sitting down playing chess with danny de emf and boxer de el centro was sitting next to us. boxer was making loud and loud voices so i tell him por favor can you kick back homie cause im playing chess a minute later the vato starts back up again so this time i tell him con respecto homie can you kick back. the vato stop for a minute and he starts up again so i tell him check this out shut the f**k up cause im tired of your voice and if you got a problem with it we can go to celda and handle it. i really felt disrespected thats why i told him. anyways after i tell him that the next thing I know that vato slashes me and leaves. dy the time i figure im hit i try to get away but the c.o. is walking in my direction and he gets me right dy a celda. so i go to the hole. when im in the hole my home boys hit doxer so now "b" is also in the hole. while im in the hole im getting schoold wrong and

XOR	0	1
0	0	1
1	1	0

1940s

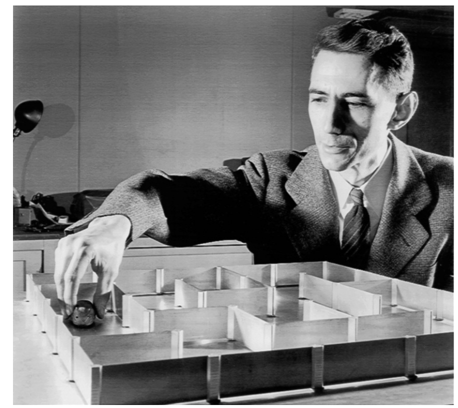
Shannon and OTP

one time pad

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

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

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



M bit-wise XOR w/ K.

$\Downarrow$

ciphertext

Shannon and OTP

OTP is perfectly secure

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.

Perfect Security/Secrecy

ignore

~~**Definition 0.1.** A cryptosystem $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ is *perfectly secure* if for all distributions $\mathcal{D}$ on messages and every message g and every ciphertext c,~~

$$\cancel{\Pr[\, g = m \mid \mathcal{E}(K, m) = c \,] = \Pr[\, m = g \,]}$$

~~where the probability is over $K \leftarrow_{\$} \mathcal{K}$ and $m \leftarrow_{\$} \mathcal{D}$.~~

$\mathcal{M}$ is message space, $\mathcal{C}$ is ciphertext space

Def. $\forall m_0, m_1 \in \mathcal{M} \quad \forall c \in \mathcal{C}$

$$\Pr_{K}[\, \mathcal{E}_K(m_0) = c \,] = \Pr_{K}[\, \mathcal{E}_K(m_1) = c \,]$$

sample space $\rightarrow K$

~~for a given ciphertext, each msg is equally likely~~

Modern Crypto: 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, $$$

Example Source of Computational Hardness: Factoring

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 Efficiently?

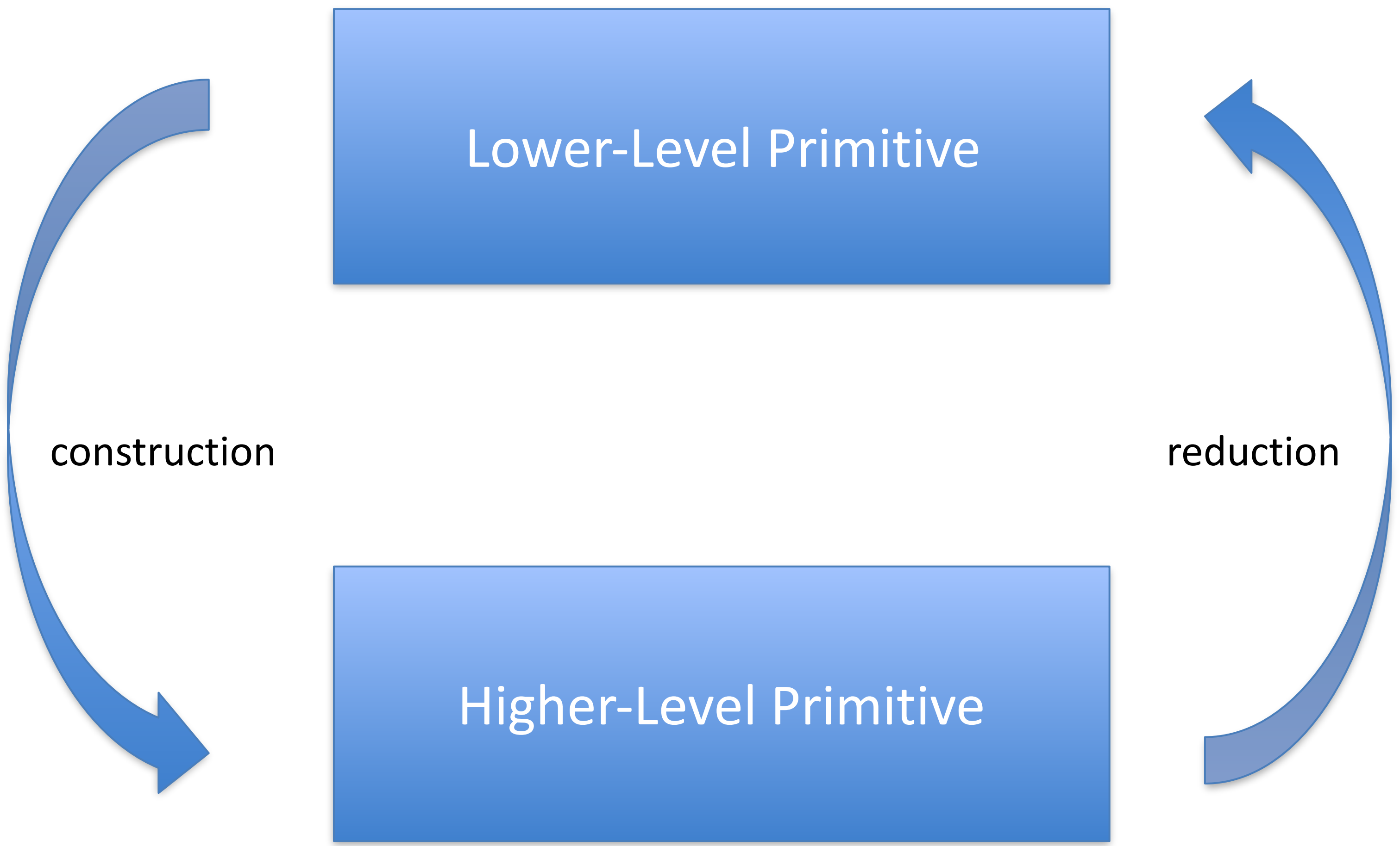
- Gauss couldn't figure out how
- Today there is no known algorithm to factor a 400 digit number in a practical amount of time.

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 ...

Provable Security



Why is this the Right Approach?

Why is this the Right Approach?

Ad-hoc design is subject to bug-then-patch cycle. Very dangerous and costly.

Why is this the Right Approach?

Ad-hoc design is subject to bug-then-patch cycle. Very dangerous and costly.

Doesn’t make sense to try to design a secure encryption scheme without first asking what “secure” means.

Lower-level Primitives

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, ...

Features:

- Lots of them

Definitional Phase vs Constructive Phrase

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?

Cryptography in Practice

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

Cryptography Beyond Communications Security

- Homomorphic encryption
- Functional Encryption
- Cryptocurrency
- Anti-Surveillance Techniques
- Multiparty Computation
- Zero-knowledge Proofs
- …

New uses for old mathematics

Cryptography uses

- Number theory
- Combinatorics
- Modern algebra
- Probability theory

Keep a Security Mindset

Keep a Security Mindset

Put on your adversary hat!

Keep a Security Mindset

Put on your adversary hat!

Ask what the risks and threats are. How might the system be attacked?

Keep a Security Mindset

Put on your adversary hat!

Ask what the risks and threats are. How might the system be attacked?

Be critical of security claims.

Keep a Security Mindset

Put on your adversary hat!

Ask what the risks and threats are. How might the system be attacked?

Be critical of security claims.

Our system is secure because it uses 128-bit keys!

How are they being used? What is the threat model? Do you have a security proof?

Beware of Human Fallibility

Beware of Human Fallibility

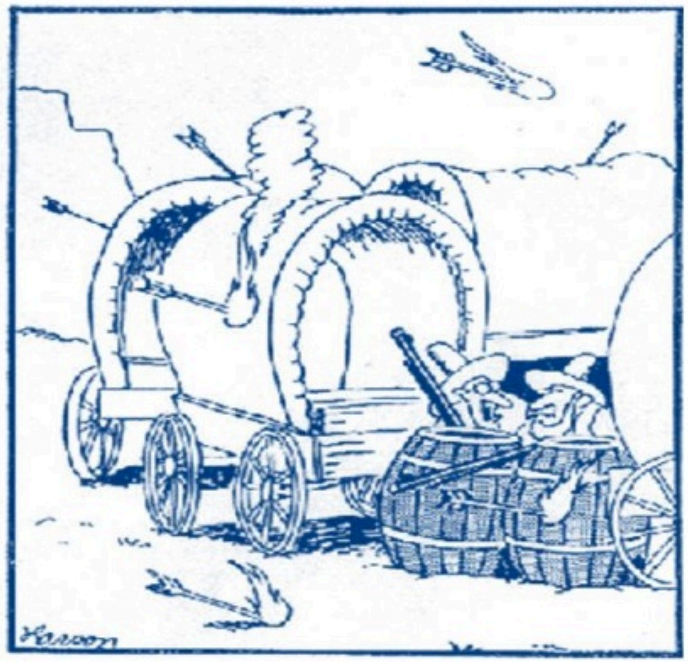
Edgar Allen Poe (an amateur cryptographer!) said “human ingenuity cannot concoct a cipher which human ingenuity cannot resolve.”

Beware of Human Fallibility

Edgar Allen Poe (an amateur cryptographer!) said “human ingenuity cannot concoct a cipher which human ingenuity cannot resolve.”

Richard Feynman said, “the first principle is to not fool yourself – And you are the easiest person to fool.”

Coloring outside the lines



"Hey! They're lighting their arrows!
Can they do that?"

What To 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!