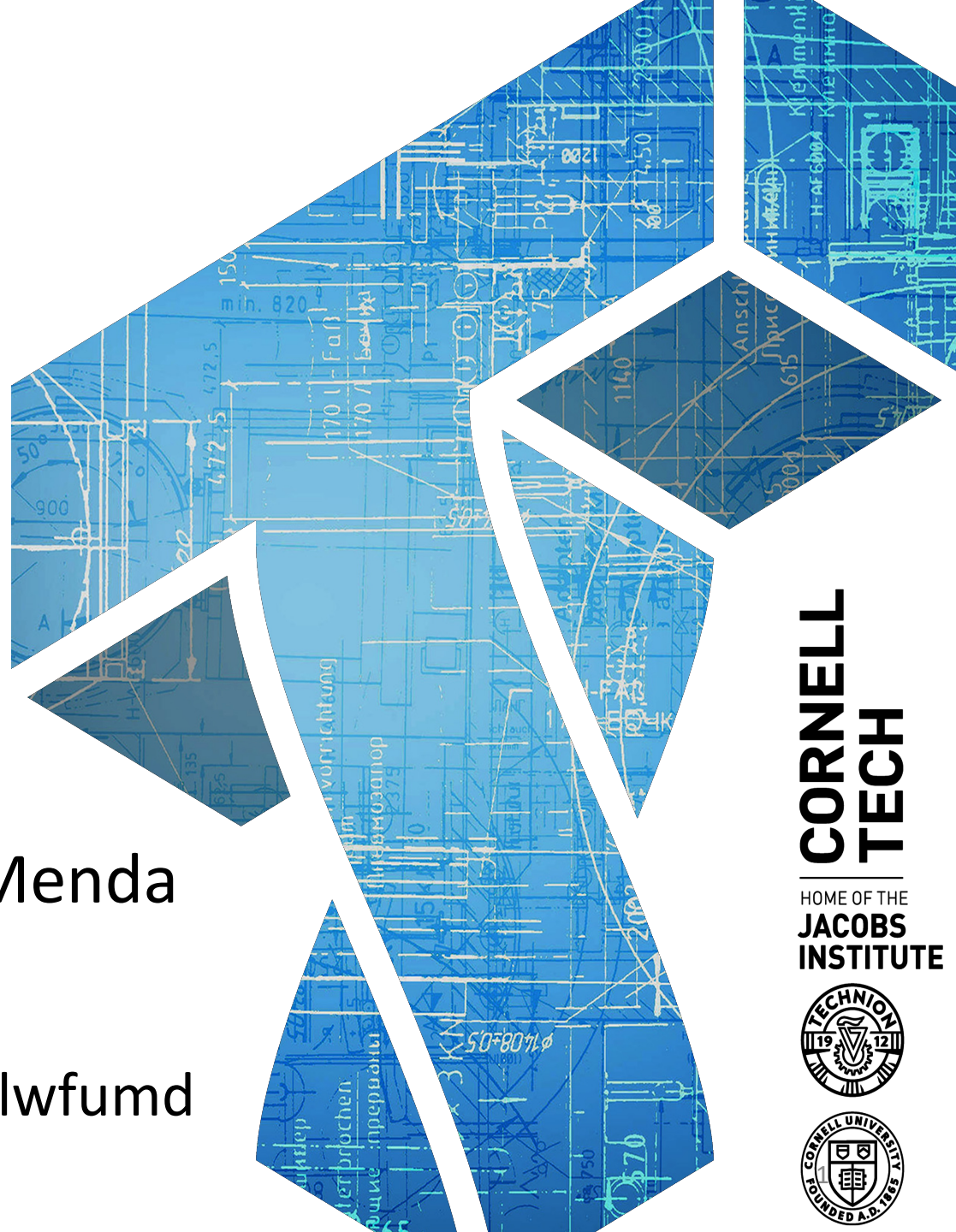
# CS 5830
# Cryptography

Instructor: Tom Ristenpart
TAs: Andres Fabrega, Sanketh Menda

bjqhtrj yt ymj jchnynsl btwqi tk hwduytlwfumd



CORNELL TECH
HOME OF THE JACOBS INSTITUTE

# Who am I? https://rist.tech.cornell.edu


- Academic computer security researcher
  - ~7 years of grad school at UC Davis & UC San Diego
  - 4.5 years as professor at University of Wisconsin-Madison
  - 7 years as professor at Cornell Tech
  - Smaller stints in industry
    - Skyhigh Networks, Cloudflare, Microsoft,


- Applied & theoretical cryptography, cloud computing security, machine learning privacy, user authentication, tech abuse

# Computer security

understanding and improving the behavior of
computing technologies in the presence of adversaries

Attackers
(aka adversaries)

Target/victim
computing
systems

Defenders
(designers, engineers,
lawyers, etc.)

This class is on cryptography, class of mechanisms for helping achieve security

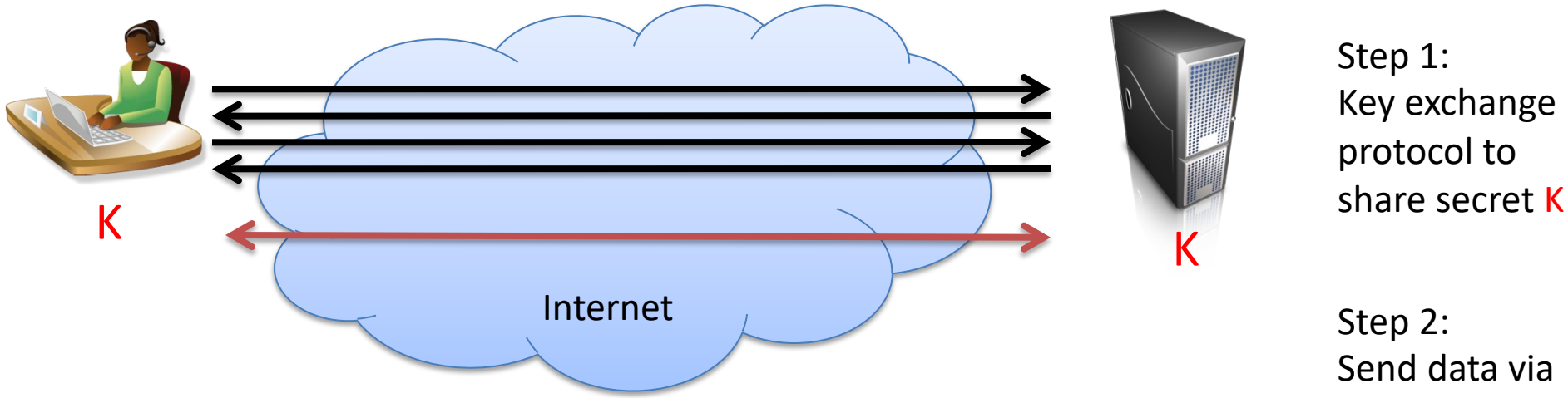
# Cryptography: “Hidden writing”

- Study and practice of building security protocols that resist adversarial behavior
  - Not ***just*** encryption
- Blend of mathematics, engineering, computer science

- Cryptanalysis: breaking cryptography

Crypto = cryptography (not cryptocurrencies)

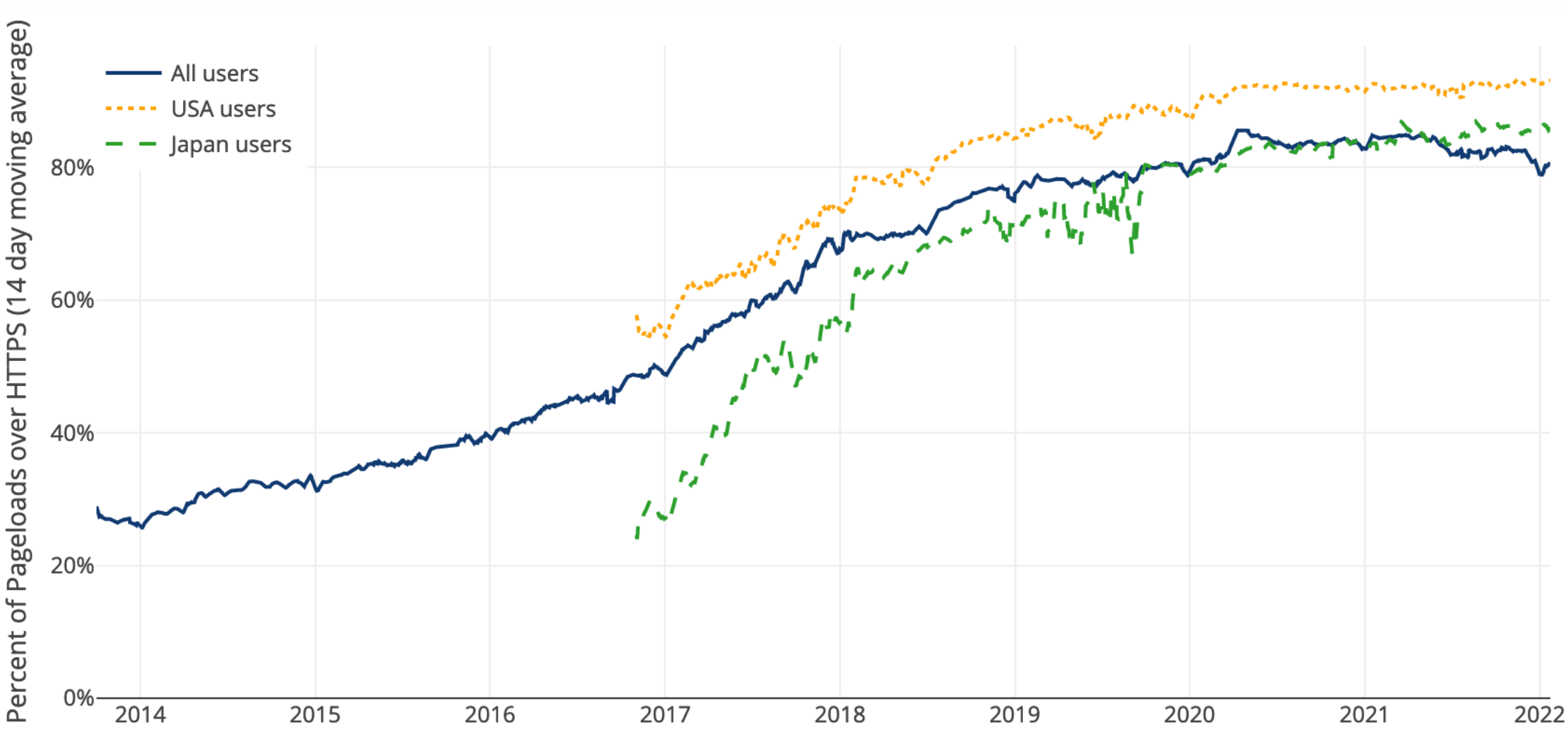
# Cryptography use cases




TLS (Transport Layer Security) protects web traffic “in flight”

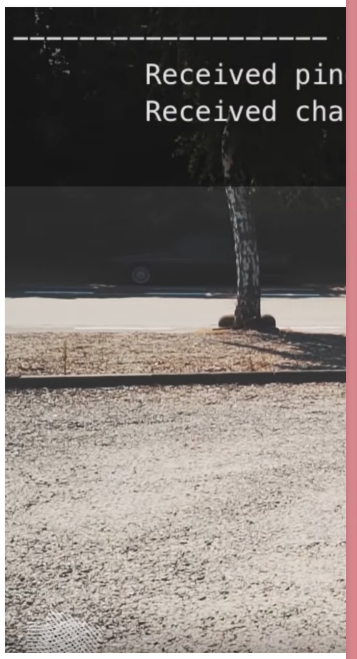
- Passwords
- Credit card numbers
- Personal information
- …

# Cryptography use cases




Statistics from Firefox
https://letsencrypt.org/stats/

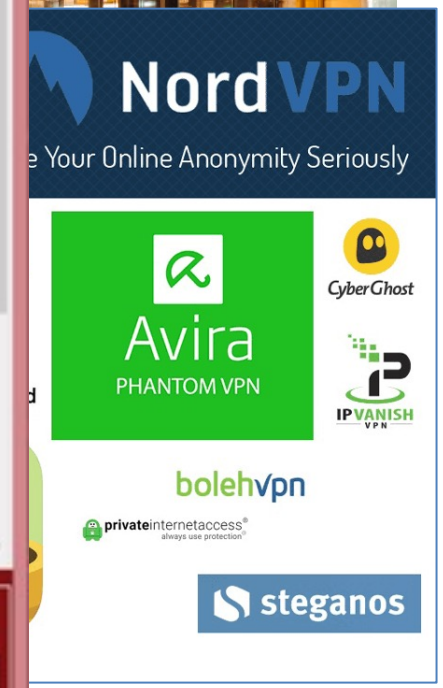
# Cryptography use cases




https://www








.com/blog/vpn-beginners-guide/

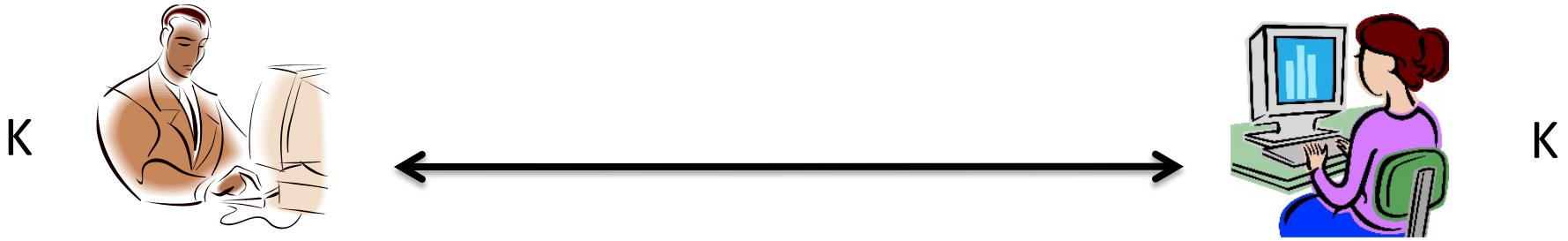
# Crypto changes power relationships

- WW2 cryptanalysis
- CryptoAG backdoor
- Crypto wars of 1990s through today (United States)
  - RSA invented by intelligence agencies (GCHQ) before RSA
  - Late 1970s, early 1980s surge in public-sector cryptography
  - Clipper Chip (1990s)
  - Export restrictions on cryptography (treated as “munition”)
  - US intelligence efforts on surreptitious backdoors
- Law enforcement and lawful access debate
  - Child sexual abuse media (CSAM), encrypted phones

# Some goals for course

- **Learning to "speak" crypto**
  - What different primitives are for
  - What are the security goals associated with them
  - How do cryptographers reason about security
- **Current best practices** for cryptographic constructions you are likely to encounter
  - Understand why they are considered best
  - Know how to break some inferior choices

# Symmetric encryption uses shared secret key K




Scheme SE = (Kg,Enc,Dec) has three algorithms:

- Key generation (Kg), Encryption (Enc), Decryption (Dec)

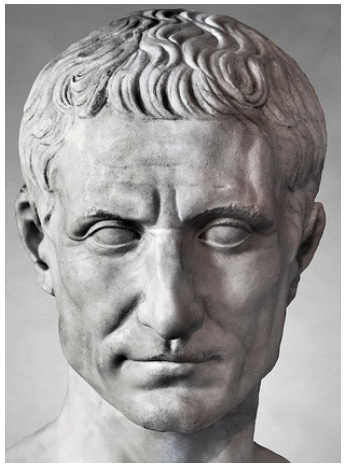
Functionality (correctness)

- Decryption should reverse encryption (for same key)
- Should be efficient to run all three algorithms

Security

- Capabilities and goals of attacker

# Substitution cipher (aka Caeser cipher)



Kg():
K <-$ {a,b,c,…,z}

Pick a random English letter from [a-z]

Enc(K,M):
Split M into characters $M_1,M_2,\ldots,M_m$
For i = 1 to m do
$C_i$ <- $M_i$ + K mod 26
Return $C_1||C_2||\ldots||C_m$

Assume M is string of English lower-case letters
Plus '+' interprets letters as numbers (a = 0, b = 1, …), adds mod 26, then converts back to letter

Dec(K,C):
Split C into characters $C_1,C_2,\ldots,C_m$
For i = 1 to m do
$C_i$ <- $M_i$ - K mod 26
Return $M_1||M_2||\ldots||M_m$

Assume C is string of English lower-case letters

# Threat models for symmetric encryption

C

C <- Enc(K,M)

M <- Dec(K,C)

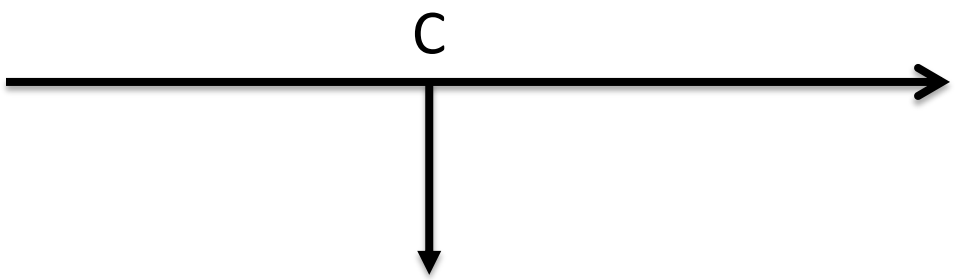
Threat model captures:

1. Adversary's goals
2. Adversary's capabilities

What are example adversarial goals?
What are potential capabilities?

# Threat models for symmetric encryption

C



C <- Enc(K,M)

Adversary

M <- Dec(K,C)

Can the adversary recover M from C? Message/plaintext recovery attack

Can the adversary recover K from C? Key recovery attack

What does adversary know about Enc?

# Auguste Kerckhoffs' (Second) Principle

(circa 1883)

"The system must not require secrecy and can be stolen by the enemy without causing trouble"

A cryptosystem should be secure even if its algorithms, implementations, configuration, etc. is made public --- the only secret should be a key

Why?

# Threat models in cryptography

In general, we want to build cryptography that is secure:

- even for seemingly “weak” goals
- against adversaries with strong capabilities

Crypto design

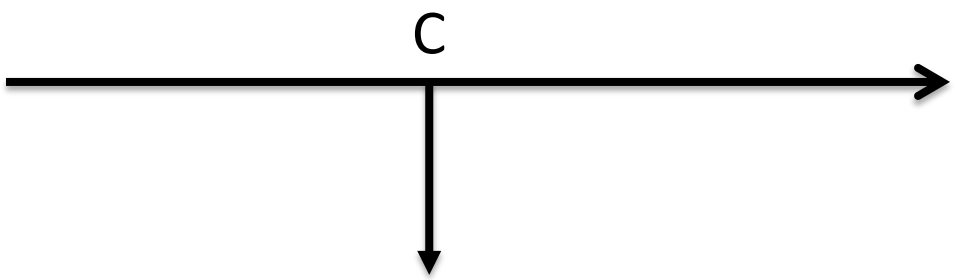
For cryptanalysis we want to build attacks that:

- are as damaging as possible
- assume as few capabilities as possible

Cryptanalysis

# Substitution cipher

C



C <- Enc(K,M)

Adversary

M <- Dec(K,C)

**Partial message recovery:**
Adversary obtains a single ciphertext, attempt to recover one bit of M

**Message recovery:**
Adversary obtains a single ciphertext, attempt to recover all of M

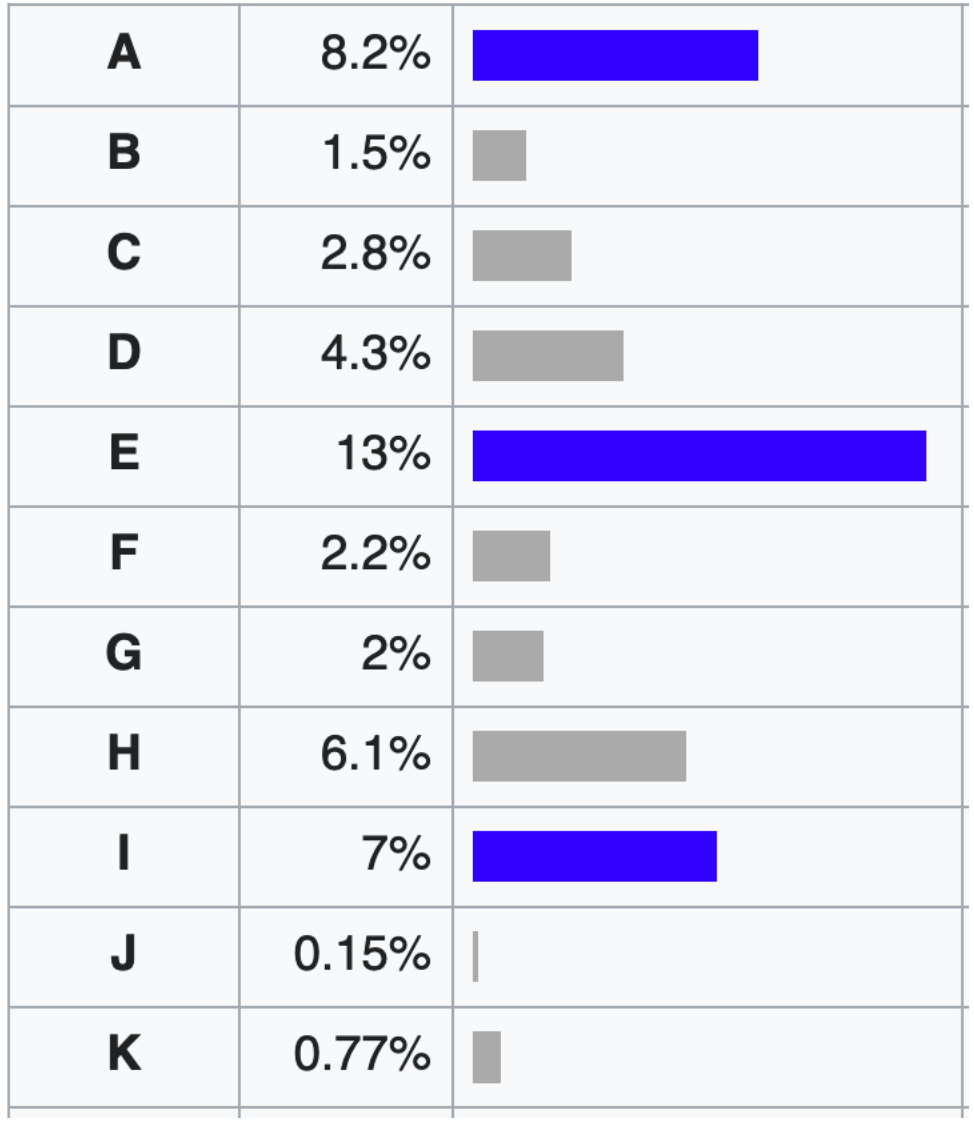
**Key recovery:**
Adversary obtains a single ciphertext, attempts to recover K

# Let’s do some cryptanalysis

bjqhtrj yt ymj jchnynsl btwqi tk hwduytlwfumd

# Frequency analysis

- English language has patterns
- Can exploit, why?
  - Substitution cipher ***preserves*** frequency of plaintext
  - Need to not leak this information




https://en.wikipedia.org/wiki/Letter_frequency


ENIGMA

# Shannon's security notion (1949)

Def. A symmetric encryption scheme Enc is perfectly secure if for all messages M, M' and ciphertexts C

$$\Pr[\,Enc(K,M) = C\,] = \Pr[\,Enc(K,M') = C\,]$$

where probabilities are over choice of K

In words:
each message is equally likely to map to a given ciphertext

In other words:
seeing a ciphertext leaks nothing about what message was encrypted

Does a substitution cipher meet this definition? No!

# One-time pad (OTP)

Kg():
K <-$ {0,1}$^L$

Pick a random bit string

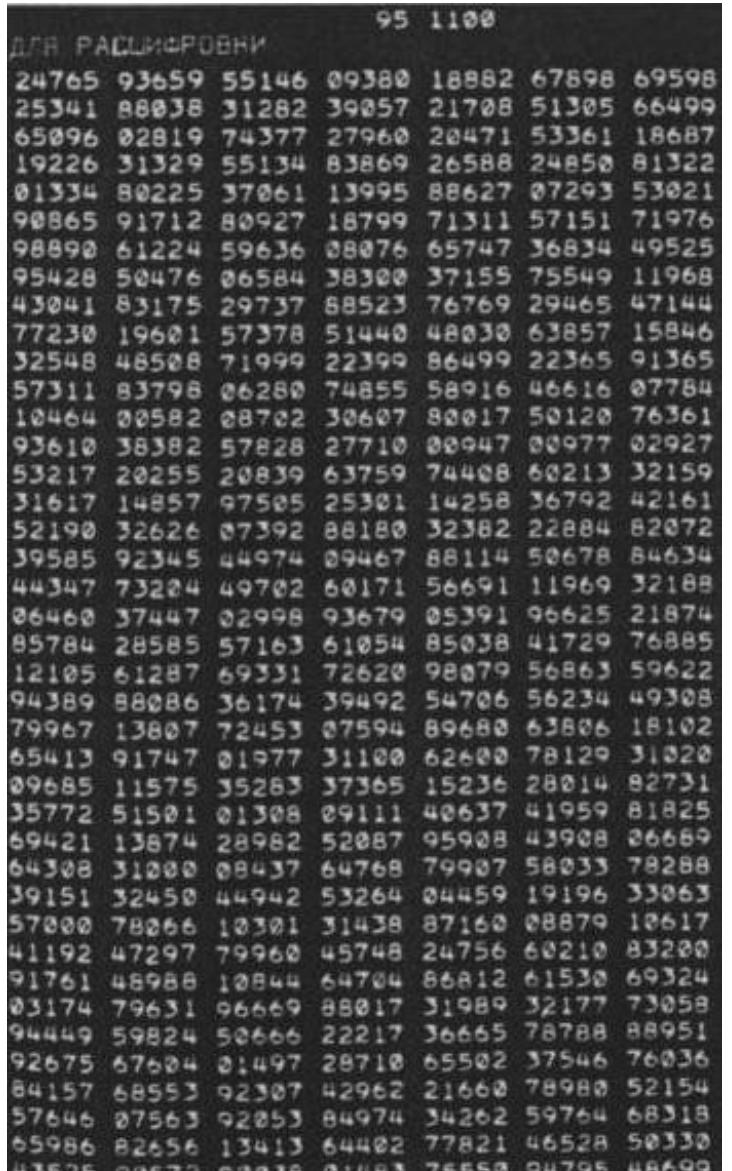
Enc(K,M):
Return M $\oplus$ K

Assume M is L-bit string

Dec(K,C):
Return C $\oplus$ K

Assume C is L-bit string

Part of a CIA OTP used by Soviet diplomat spying for CIA



95 1100

ДЛЯ РАСШИФРОВКИ

24765 93659 55146 09380 18882 67898 69598
25341 88038 31282 39057 21708 51305 66499
65096 02819 74377 27960 20471 53361 18687
19226 31329 55134 83869 26588 24850 81322
01334 80225 37061 13995 88627 07293 53021
90865 91712 80927 18799 71311 57151 71976
98890 61224 59636 08076 65747 36834 49525
95428 50476 06584 38300 37155 75549 11968
43041 83175 29737 88523 76769 29465 47144
77230 19601 57378 51440 48030 63857 15846
32548 48508 71999 22399 86499 22365 91365
57311 83798 06280 74855 58916 46616 07784
10464 00582 08702 30607 80017 50120 76361
93610 38382 57828 27710 00947 00977 02927
53217 20255 20839 63759 74408 60213 32159
31617 14857 97505 25301 14258 36792 42161
52190 32626 07392 88180 32382 22884 82072
39585 92345 44974 09467 88114 50678 84634
44347 73204 49702 60171 56691 11969 32188
06460 37447 02998 93679 05391 90625 21874
85784 28585 57163 61054 85038 41729 76885
12105 61287 69331 72620 98079 56863 59622
94389 88086 36174 39492 54706 56234 49308
79967 13807 72453 07594 89680 63806 18102
65413 91747 01977 31100 62600 78129 31020
09685 11575 35283 37365 15236 28014 82731
35772 51501 01308 09111 40637 41959 81825
69421 13874 28982 52087 95908 43908 06669
64308 31000 08437 64768 79907 58033 78288
39151 32450 44942 53264 04459 19196 33063
57000 78066 10301 31438 87160 08879 10617
41192 47297 79960 45748 24756 60210 83200
91761 48988 10844 64704 86812 61530 69324
03174 79631 96669 88017 31989 32177 73058
94449 59824 50666 22217 36665 78788 88951
92675 67604 01497 28710 65502 37546 76036
84157 68553 92307 42962 21660 78980 52154
57646 07563 92053 84974 34262 59764 68318
65986 82656 13413 64402 77821 46528 50330

# Shannon's security notion (1949)

Def. A symmetric encryption scheme Enc is perfectly secure if for all messages M, M' and ciphertexts C

$$\Pr[\,Enc(K,M) = C\,] = \Pr[\,Enc(K,M') = C\,]$$

where probabilities are over choice of K

Thm. OTP is perfectly secure

For any C and M of length L bits

$$\Pr[\,K \oplus M = C\,] = 1 / 2^L$$

$$\Pr[\,K \oplus M = C\,] = \Pr[\,K \oplus M' = C\,]$$

# Shannon's security notion (1949)

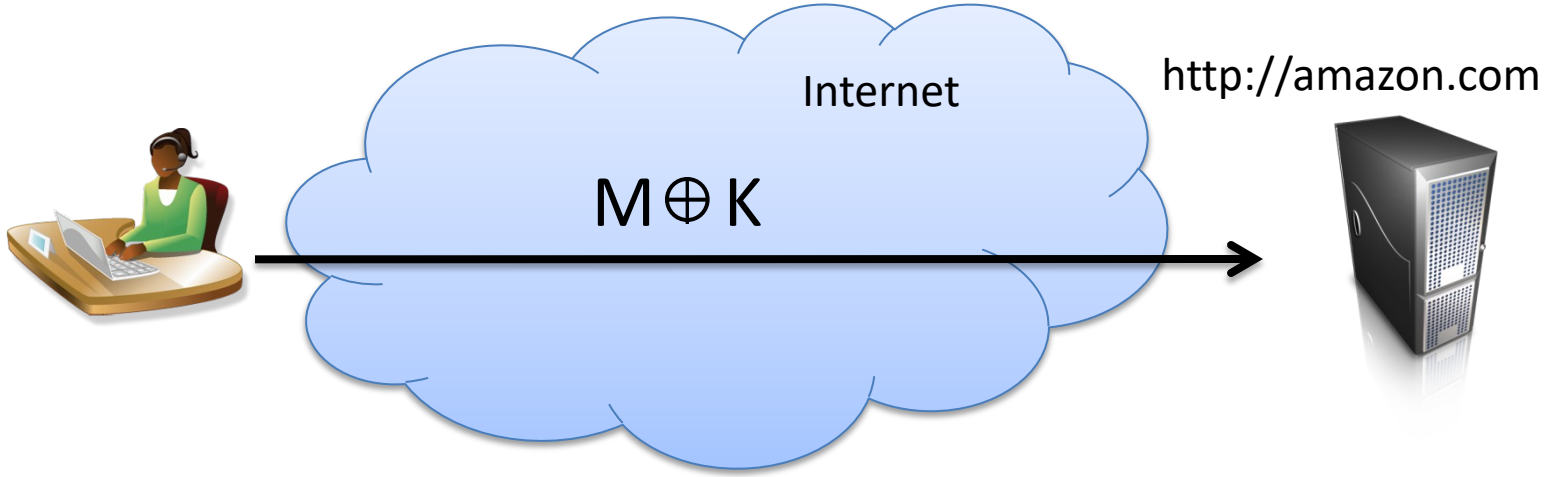
Def. A symmetric encryption scheme Enc is perfectly secure if for all messages M, M' and ciphertexts C

$$\Pr[\,Enc(K,M) = C\,] = \Pr[\,Enc(K,M') = C\,]$$

where probabilities are over choice of K

Thm. OTP is perfectly secure

Thm. Perfectly secure encryption requires $|K| \geq |M|$




Does OTP suffice for securing communications online?

Integrity easily violated

Reuse of K for messages M,M’ leaks M ⊕ M’

Encrypting same message twice under K leaks the message equality

K must be as large as message

Message length revealed

# Formal security notions are hard to get right

Simplifying abstraction to allow rigorous analysis


Amount of deployment details needed to capture attacks

# How do cryptographers *evaluate* crypto?

- **Cryptanalysis**
  - Try to break it. Best attacks are implementable
  - Certificational weakness: attack that works on paper, but can't be actually executed
- **Formal analyses by hand**
  - Give rigorous definition of security, prove manually that scheme meets it (usually via reduction to underlying hard problem)
- **Automated protocol analysis tools**
  - Build software tools to analyze protocols for bugs
- **Implementation analysis tools**
  - Build software tools to analyze implementations

# How do cryptographers *design* crypto?

- **Avoid security through obscurity:**
  - Public designs and evaluation
- **Public competitions**
  - Set out requirements document, solicit submissions and then have several years of people trying to break stuff
- **Standardization processes (IETF / ISO)**
  - Write down protocols as RFCs (Request for Comments)
  - Sometimes this is design-by-committee
- ***No single person can design good crypto. Community effort***

# The game plan

- Planned topics
  - Stream ciphers, block ciphers
  - Symmetric encryption (block cipher modes of operation)
  - Authenticated encryption & hashing
  - TLS overview and public key encryption (RSA)
  - PKI, Diffie-Hellman, Elliptic curve cryptography
  - Encrypted messaging & “backdoors”
  - Random number generation
  - Misc. topics

# The game plan

- **Targeting 5 homework assignments (90%, equal weight)**
  - Mix of Python programming, short answer questions
  - Hope to get full homework schedule up this week
  - Will have some extra credit opportunities in homeworks
- **Participation (10%)**
  - Want to have a vibrant class environment in lecture
  - By default everyone gets full 10%
  - If attendance gets too low, same handful of people participating with questions or comments, etc., then I may start taking attendance, calling on people, giving quizzes, etc. Will give warnings

# Remote versus in-person

- Default is to deliver lecture in person
  - In some rare cases I may have to deliver remotely, and I will warn you about this and it will be via Zoom
- Remote attendance is possible and I will record lectures (only for distribution within class):
  - Stay home if you are feeling sick, attend if you like (please let us know)
  - Other reasonable excuses (unavoidable travel, etc.)
  - Remember if in-person attendance gets too low, I will change participation grading strategy

# Class logistics

- Use Slack for announcements, support
  - Ask via direct messages to TAs, myself, or in #office-hours
  - We will try to get back to you as quickly as possible
- Canvas has zoom links
- In-person office hours TBD
- Some suggested other readings, but we won't be following a specific textbook

# Questions?