CS171: Cryptography

Lecture 1

Sanjam Garg

All Photos are by Unknown Author and are licensed under CC BY-SA

Logistics

- Please join on Piazza
- Google Calendar link
- Use entry code **2PY8Y4** to join on Gradescope
- Homework will be submitted there (released on Tue and due the next week on Th 8:59 pm)
- Homework drop policy: Best x-2 (out of x).
- Office Hours: Online, 1:10-2 pm, on Wednesday or by email
- GSI: Bhaskar Roberts, Jaiden Keith Fairoze, Sriram Sridhar

Exam dates and policies

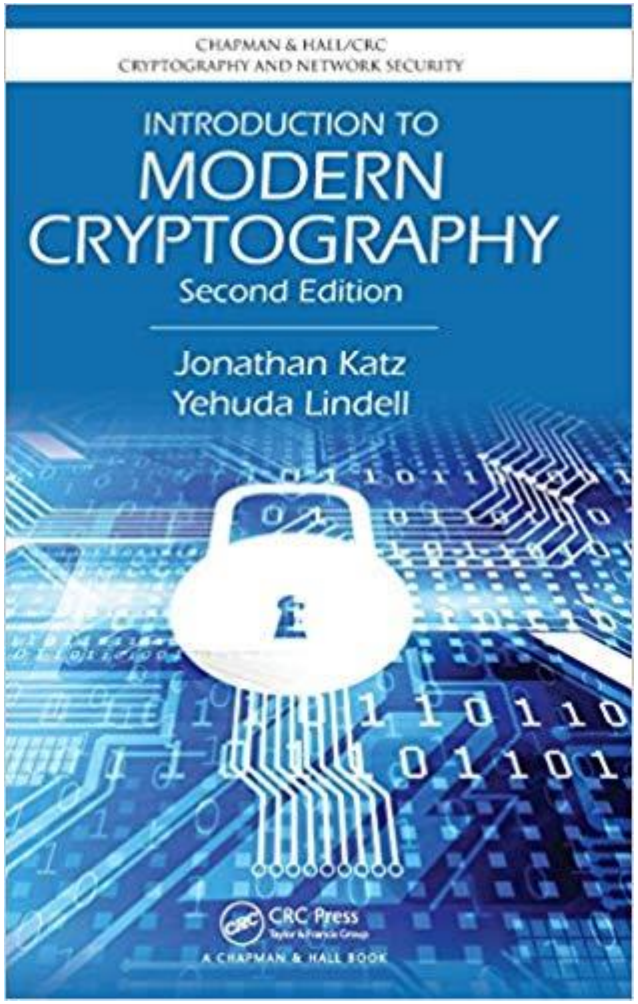
- Midterm I: Feb 14th (During class time)
- Midterm II: Mar 20th (During class time)
- Final: May 7th (7PM – 10PM)
- All exams are mandatory!

Grading

- Homeworks (almost every week) - 20%
- Midterm I - 25%
- Midterm II - 25%
- Final - 30%

- See collaboration policy on Piazza

Book


CHAPMAN & HALL/CRC
CRYPTOGRAPHY AND NETWORK SECURITY
INTRODUCTION TO
MODERN
CRYPTOGRAPHY
Second Edition
Jonathan Katz
Yehuda Lindell
CRC Press
A CHAPMAN & HALL BOOK

- Questions?

- New material! Please ask questions throughout this class!

About the Course

- Cryptography is used everywhere
- Many applications
- Course goals: Learn the theoretical basis of the cryptography used in the real-world
  - Learn about key primitives
  - Understand what security they provide
  - Know how to use them
  - Understand “how things work”
- Not Do: Brew your own crypto!

Crypto mindset!

Cryptography Historically

- Historically, cryptography focused on private communication between two parties using a previously shared secret information
- Lack of clarity on what it means to be secure
- Heuristic, unprincipled design approach
- Break and fix cycle (cat and mouse game)

Modern Cryptography

- More than just confidentiality, e.g., data integrity
- More tasks: public-key cryptography
- Rigorous security definitions
- Sound mathematical principles for arguing/proving security
    - E.g., basing it on factoring

In summary, cryptography has developed from an art to a science.

Classical Cryptography

- Focused on secret-key/shared-key/private-key/symmetric-key cryptography
- Start with simple schemes used for centuries
- Demonstrate dangers of an unprincipled approach
- Shows why simple tricks are unlikely to work

Private-key encryption

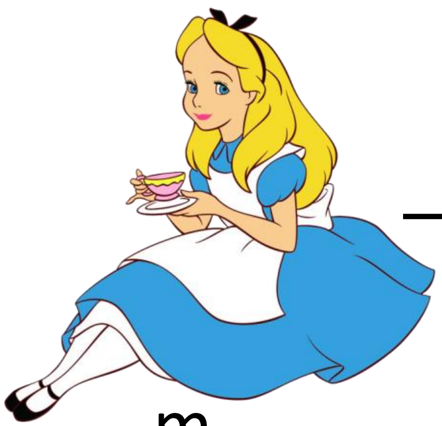
ciphertext

key

key

c

k



k

m

$c \leftarrow \mathrm{Enc}_k(m)$

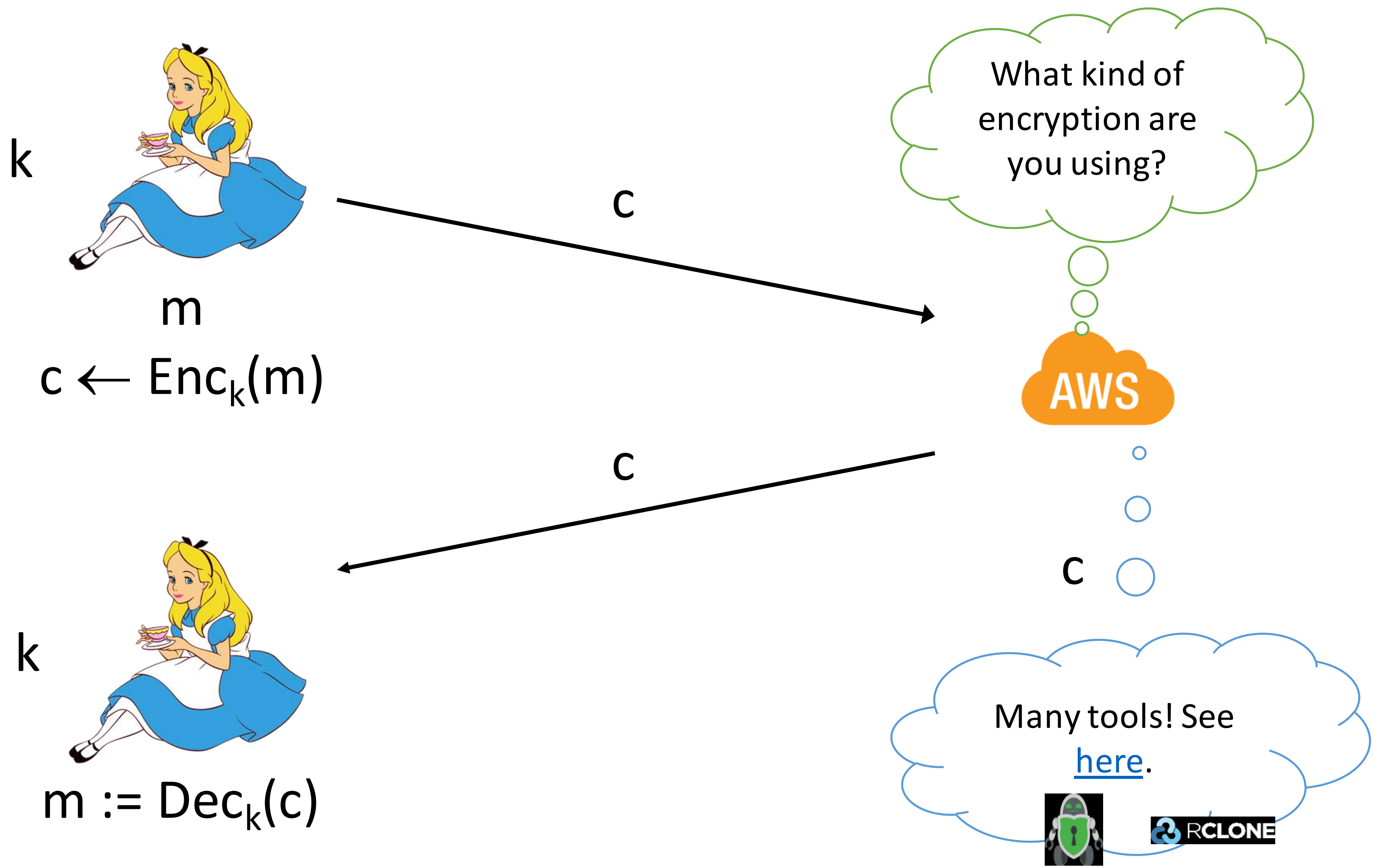
message/plaintext

$m := \mathrm{Dec}_k(c)$

decryption

encryption

Private-key encryption



Private-key Encryption (syntax)

- A *private-key encryption scheme* is defined by a message space $\mathcal{M}$, a key space $\mathcal{K}$, and algorithms (Gen, Enc, Dec):
  - Gen (key-generation algorithm): outputs $k \in \mathcal{K}$
  - Enc (encryption algorithm): takes key k and message $m \in \mathcal{M}$ as input; outputs ciphertext c

    $$c \leftarrow Enc_k(m)$$

  - Dec (decryption algorithm): takes key k and ciphertext c as input; outputs m or "error"

    $$m := Dec_k(c)$$

Keep Enc and Dec secret as well?

k must be kept secret

Correctness: For all $m \in \mathcal{M}$ and k output by Gen,

$$Dec_k(Enc_k(m)) = m$$

Very important in the context of modern cryptograph. Use open-source tools.

Kerckhoff’s principle

This Photo by Unknown Author is licensed under CC BY-SA

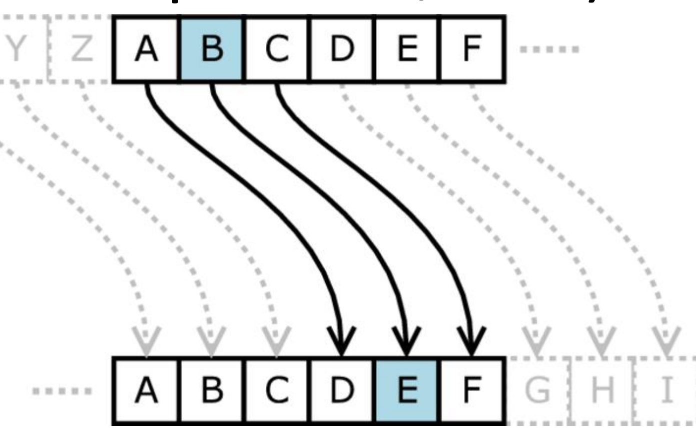
The cipher method must not be required to be secret, and it must be able to fall in the hands of the enemy with inconvenience.

Only the key is kept secret

The shift cipher

- Consider encrypting some English text
- Associate ‘a’ with 0, ‘b’ with 1, …, ‘z’ with 25
- Let key-space be $\mathcal{K} = \{0, \ldots, 25\}$

- To encrypt using key k, shift every letter of the plaintext by k positions (with wraparound/mod)
- For example, for k=3
- Decryption reverses this



The shift cipher, formally

- $\mathcal{M}$ = {strings over lowercase English alphabet}
- $\mathcal{K}$ = {English alphabets}
- Gen: choose uniform $k \in \{0, \ldots, 25\}$
- $\text{Enc}_k(m_1 \ldots m_t)$: output $c_1 \ldots c_t$, where
  $c_i := [m_i + k \bmod 26]$
- $\text{Dec}_k(c_1 \ldots c_t)$: output $m_1 \ldots m_t$, where
  $m_i := [c_i - k \bmod 26]$

- Correctness: For each i, $[c_i - k \bmod 26] = [[m_i + k \bmod 26] - k \bmod 26] = [m_i + k - k \bmod 26] = m_i$

Is shift cipher secure?

```
etarvqogcpuetarvqitcrja
```

- **No**, only 26 possible keys!
  - Try decrypting with every possible key
  - Only one possibility will “make sense”
  - Assumes that English language has a sparse structure.
- Example of a “brute-force” or “exhaustive-search” attack

Is shift cipher secure?

```
etarvqogcpuetarvqitcrja
ccccccccccccccccccccccc
-----------------------
cryptomeanscryptography
```

- `etarvqogcpuetarvqitcrja`
- `fub`…

- `cryptomeanscryptography`
- …

Only when ciphertext is large enough (more details later)

Lesson: Any secure encryption scheme must have a *sufficiently large* key space.

So as to make brute-force attack computationally infeasible.

The mono-alphabetic substitution cipher

- $\mathcal{K}$ = {All bijections (or permutations) from {a,..z} to {a,...z}}
- An example permutation/key π would be:

CIPHER ALPHABET

A = B	H = A	O = O	V = L
B = V	I = D	P = Y	W = P
C = G	J = Z	Q = F	X = U
D = Q	K = C	R = J	Y = I
E = K	L = W	S = X	Z = R
F = M	M = S	T = H	
G = N	N = E	U = T	

- How large is the key space now?

$$26! \approx 2^{88}$$

The mono-alphabetic substitution cipher, formally

- Gen: choose uniform k= $\pi \in \mathcal{K}$
- $\text{Enc}_k(m_1 \ldots m_t)$: output $c_1 \ldots c_t$, where
  $c_i := \pi(m_i)$
- $\text{Dec}_k(c_1 \ldots c_t)$: output $m_1 \ldots m_t$, where
  $m_i := \pi^{-1}(c_i)$

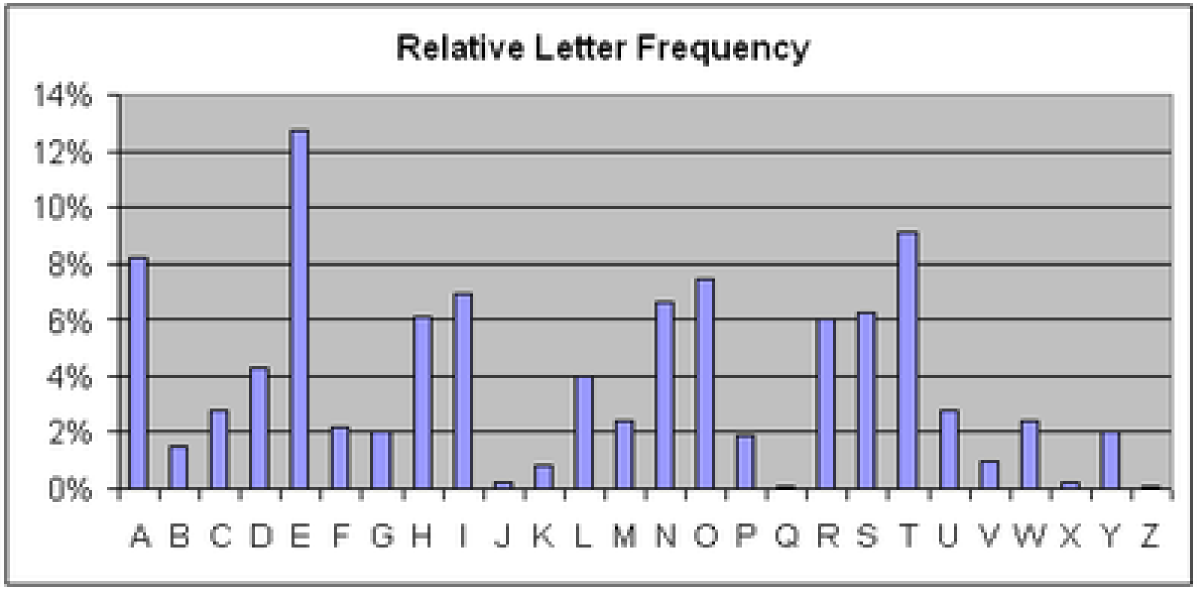
- Correctness: For each i, $\pi^{-1}(c_i) = \pi^{-1}(\pi(m_i)) = m_i$

Is mono-alphabetic substitution cipher secure?

JGRMQOYGHMVBJWRWQFPWHGFFDQGFPFZRKBEEBJIZQQOCIBZKLFAFGQVFZFWWE
OGWOPFGFHWOLPHLRLOLFDMFGQWBLWBWQOLKFWBYLBLYLFSFLJGRMQBOLWJVFP
FWQVHQWFFPQOQVFPQOCFPOGFWFJIGFQVHLHLROQVFGWJVFPFOLFHGQVQVFILE
OGQILHQFQGIQVVOSFAFGBWQVHQWIJVWJVFPFWHGFIWIHZZRQGBABHZQOCGFHX

Example from Katz and Lindell.

- **No**, because of frequency analysis attack.



This Photo by Unknown Author is licensed under CC BY-SA

Need a larger ciphertext than what was needed for breaking shift cipher

Lesson: Need to ``smooth out’’ the frequency distribution of characters.

Rule out statistical tests.

Vigenère Cipher (multiple shift cipher)

- $\mathcal{K}$ = $\{\{a, .. z\}^{t}\}$ (description of scheme by example)

```
tellhimaboutme
cafecafecafeca
--------------
veqpjiredozxoe
```

Example from Katz and Lindell.

- How large is the key space?

  26^{t} ($\approx 2^{70}$ for t = 15)

- Smooth out the distribution: The two ‘l’s in ‘tell’ are encrypted differently

Remained unbroken for centuries.

Is Vigenère Cipher secure?

Attack (when t is known)

- For each $j \in \{1 \ldots t\}$, consider $c_j, c_{j+t}, c_{j+2t} \ldots$
- Just an example of a shift cipher with the same shift
- Can do frequency analysis. Call the most frequent alphabet `e' and recover the key k_j
- Next: a more methodical attack (removing the guess work)

Better attack on Shift Cipher

- Let p_i $(0 \leq i \leq 25)$ denote the frequency of the i^th^ English alphabet in normal English plaintext
  - One can compute $\sum_i p_i^2 \approx 0.065$
- Let q_i $(0 \leq i \leq 25)$ denote the frequency of the i^th^ English alphabet in the given ciphertext
- Find j in {0… 25} such that $I_j = \sum p_i q_{i+j}$ is close to 0.065.

Apply the same attack to the stream c_j, c_{j+t}, c_{j+2t} … in the Vigenère Cipher

What if t is unknown?

- Simple Strategy: Try for all possible choices of t! (Only a few possibilities, given an upper bound on t.)
- More efficient: For a given τ, let r_i $(0 \leq i \leq 25)$ denote the frequency of the i^th English alphabet in the stream $c_1, c_{1+\tau}, c_{1+2\tau}$ ... compute

$$S_\tau = \sum_i r_i^2$$

- If $t = \tau$, then $S_\tau \approx 0.065$. Otherwise, $S_\tau \approx 26 \times \frac{1}{26^2} = 0.038$ (heuristically).
- Iterate over all possible choices of τ to find the correct one.

Lesson: Ad hoc fixes are likely to break.

Thank You!