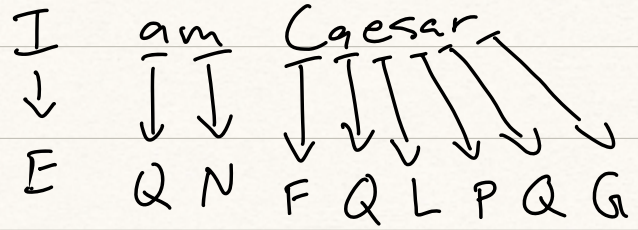
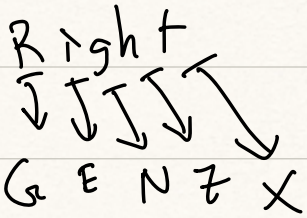
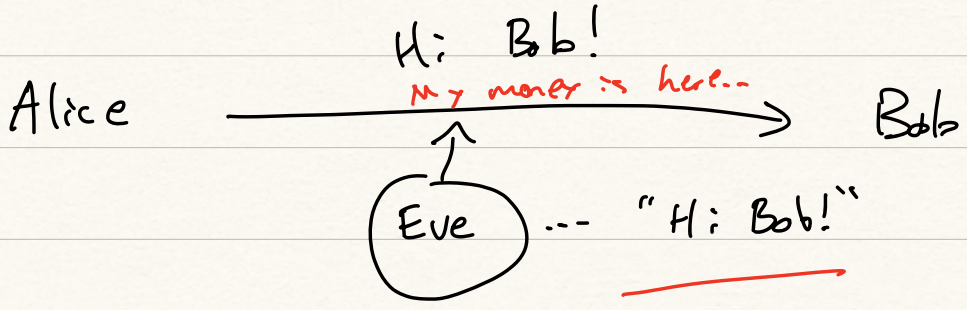
Cryptography and Complexity Theory

Caesar Cipher





Encryption/ Decryption



Modern Cryptography

- Rigorous Security Definitions involving

  Computationally Bounded Adversaries

  ↓

  complexity theory ⟷

  Modern Crypto Security

Crypto Scheme Π

- How to prove Π is "secure"?
- Reductions

  * If Π is not secure,
    * we can use adversary A^* breaking Π to solve some underlying computational problem

Some complexity theory problem that is "hard"

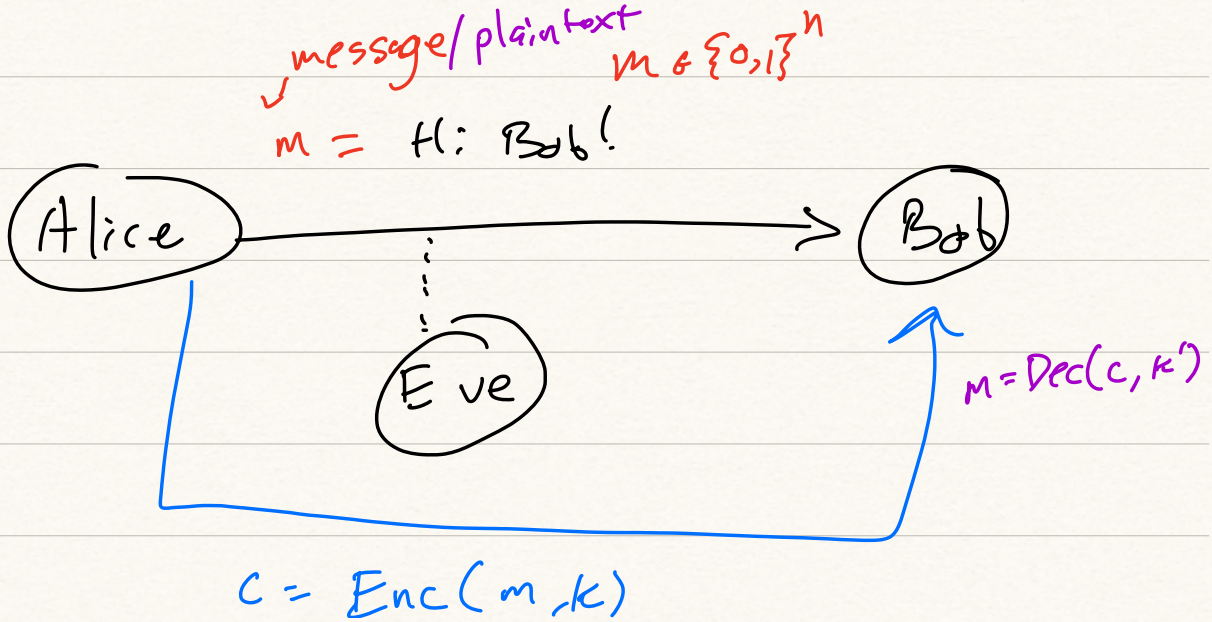
↳ A^*'s resources are much less than we assume to be required to solve C → contradiction.

- Before Modern Crypto, people believed you could have security by hiding the internals of the enc. alg.

- Modern Crypto Approach:
  - security only based on hidden randomness

Perfect Security

* Basic Problem: Encryption / Decryption



↑ key $\quad k \xleftarrow{\$} \{0,1\}^n$

- Alice and Bob share $k \xleftarrow{\$} \{0,1\}^n$
- One-time Pad

$$Enc(m,k) = m \oplus k$$

$$Dec(c,k) = c \oplus k$$

$$= (m \oplus k) \oplus k$$

$$= m \quad \checkmark$$

Perfect Security: Let (Enc, Dec) be an encryption scheme.

It is perfectly secure if

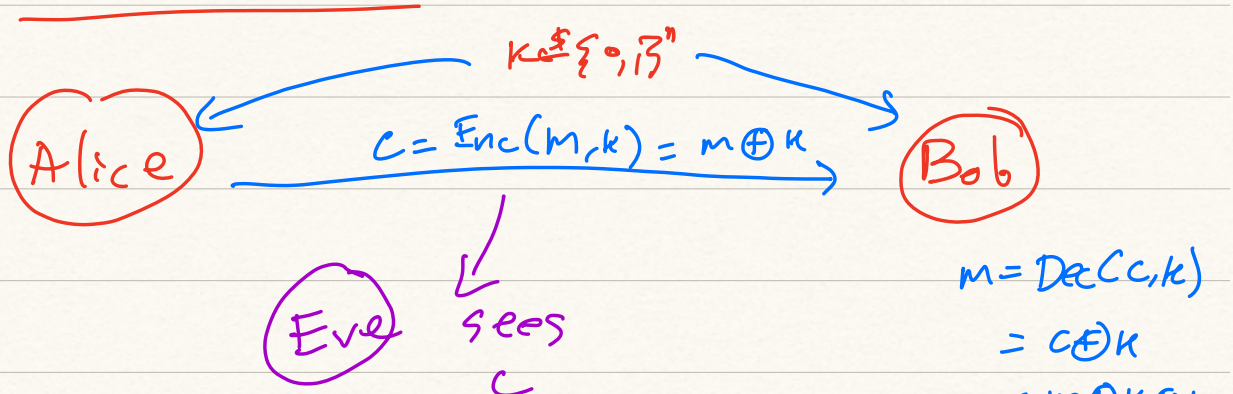
(1) $\forall\, m \in M$ (message space, $\{0,1\}^n$)
$\forall\, k \in K$ (key space)

$$Dec(Enc(m,k), k) = m$$

(2) $\forall\, m, m' \in M$, $\forall\, c \in C$ (ciphertext space)

$$\Pr_k[c = Enc(m,k)] = \Pr_k[c = Enc(m',k)]$$

One-time Pad



$m = Dec(c,k)$
$= c \oplus k$
$= m \oplus k \oplus k$
$= m$

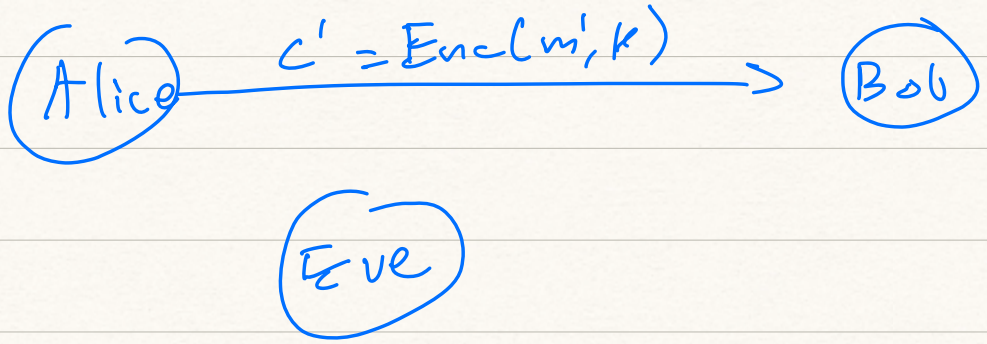
By ~~perfect~~ security

$$\Pr_k[c = Enc(m,k)] = \Pr_k[c = Enc(m',k)]$$

I.e., $c \equiv U_{\{0,1\}^n}$

$c \xleftarrow{\$} \{0,1\}^n$

why one-time?



$c \oplus c' = (m \oplus k) \oplus (m' \oplus k)$
$= m \oplus m'$

where m, m' are different

$c_1, c_2, \ldots, c_\ell$

$m_i \oplus m_j \quad \forall\, i \neq j$

Limitations of Perfect Security

Theorem: If (Enc, Dec) is perfectly secure, then $|K| \geq |M|$

i.e., keys have to be as long as messages.

Modern Crypto Saves the Day

- use complexity theory ideas to get around Perfect Security Limitations.

(1) $|K| \leq |M|$ is adversaries are computationally bounded

↳ computational security, (strict)

↳ i.e., Probabilistic Polytime Algs (i.e., BPP)

↳ Eve is PPT algs

↳ Eve being $\{C_n\}_{n\in\mathbb{N}} \in P/poly$

(X) not in these lectures

(2) Base security on certain problems being hard for BPP algs!

Crypto vs. Complexity Theory

- P=NP is the worst outcome for crypto (or one of the worst)
  - we have to use one-time pads.

Lemma: If P=NP, then for any polynomial-time computable encryption scheme (Enc, Dec) with $|K|<|M|$, there exists poly-time adversary A such that

$\forall n \in \mathbb{N}$, $\exists m_0, m_1 \in \{0,1\}^n \subset M$ s.t.

$$\Pr_{\substack{b \xleftarrow{\$} \{0,1\} \\ k \xleftarrow{\$} K}} \left[A(\mathrm{Enc}(m_b, k)) = b \right] \neq 3/4 .$$

Proof Idea:

- $n' = \log |K| < n = \log |M|$
- $S \subseteq \{0,1\}^*$ be all y s.t. $\exists k \in K$ where $y = \mathrm{Enc}(0^n, k)$, $0^n \in \{0,1\}^n$
- This is an NP statement! $S \in NP$
- $P = NP$, so there is poly-time TM M s.t. on input y, $M(y) = 1 \iff y \in S$.
- If $m_0 = 0^n$, then for any $m_1 \neq 0^n$, $M(m_1) = 0$ and $M(m_0) = 1$ } Lemma follows by showing $\exists$ such $m_1 \neq 0^n$ ∎

Q $P \neq NP$: is this enough?

No

Possible that $P \neq NP$ but we have no crypto! (beyond one time pads).

Negligible Functions

- $\varepsilon : \mathbb{N} \to [0,1] \subset \mathbb{R}$ is negligible if $\varepsilon(n) = n^{-\omega(1)} = \frac{1}{n^{\omega(1)}} = o(n^{-c})$ for any $c = \Theta(1) > 0$.

$\hookrightarrow 1/\varepsilon(n) = \omega(n^c)$

One-way Functions

- $f : \{0,1\}^* \to \{0,1\}^*$ is one-way if
  - $f(x)$ is "easy" to compute for any x; and
  - given y, it is "hard" to find x s.t. $f(x) = y$
    → "hard" to compute $f^{-1}(y)$

Def. A polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ is a one-way function if $\forall$ PPT algorithms A, $\exists$ negligible function ε such that $\forall n \in \mathbb{N}$

$$\Pr_{x \xleftarrow{R} \{0,1\}^n}\left[A(y) = x' \text{ s.t. } f(x') = y \;\middle|\; y = f(x) \right] \leq \varepsilon(n)$$

$x \xleftarrow{R} \{0,1\}^n$: random

$\varepsilon(n)$: if this is $\frac{1}{poly(n)}$

* In other words, to invert f, you need super-polynomial time.
  ↳ not efficient!

Conjecture: One-way functions exist.

- Conjectured OWFs
  - mult : $\{0,1\}^n \to \{0,1\}^{O(n)}$

    mult(x):
    - $(x_0, x_1) \in \{0,1\}^{n/2} \times \{0,1\}^{n/2}$
    - output $x_0 \cdot x_1$
  - RSA: Two large primes p, q
    → easy to compute $p \cdot q = N$
    → hard to find p, q given N. ↳ super-poly in $\log(N)$
  - Quadratic Residues
  - Decoding random error-correcting codes

- Noisy lattice problems
- Hash functions (e.g., SHA256)

Theorem: If OWFs exist, then $P \neq NP$.

- Recall: P / NP (BPP, PSPACE, ...) are all worst-case classes
- E.g., 3SAT on n variables is easy (i.e., poly-time) to solve on 99% of all formulas, but requires exp-time on the other 1%
  $\Rightarrow$ NP worst case
  *this implies $P \neq NP$
- OWFs: # of invertible strings for OWF f is $\leq \varepsilon(n) \cdot 2^n$

Example: $\varepsilon(n) = \frac{1}{n^{\log(n)}}$

$\frac{2^n}{n^{\log(n)}}$ # of strings that are invertible.

→ Can't find these in polynomial time!

- We don't know if $P \neq NP$ is enough to build "modern crypto."
- Fine-grained Crypto
  - Seeks to find out what crypto we have if $P = NP$
  - Example: Problem P of length n needs $\Theta(n^2)$ time to solve in worst-case
    - ↳ then build crypto against any $o(n^2)$ adversaries.

Q: are OWFs enough for the crypto we have today?

* No!

- OWFs only give us secret-key crypto

Crypto we use: Public-key crypto

Trapdoor Functions:

- f is a trapdoor one-way function if

① f is a one-way function

② $\exists T$ and g such that for any x,

$$g(f(x), T) = x$$

and g is poly-time computable.

Example: RSA

- p, q large primes secret
- $N = p \cdot q$ is public, $e \in \mathbb{Z}_N^*$

$m \in \mathbb{Z}$ message — all elements of $\mathbb{Z}_N$ w/ inverses

$c = m^e \bmod N$

Decryption: Need to compute

$\phi(N) = (p-1)(q-1)$

Euler's ~~totient~~ — can't eff. compute w/o p, q.

$c^{e^{-1}} \bmod N$

$(m^e)^{e^{-1}} \bmod N = (m^{e \cdot e^{-1}}) \bmod N$

$= m$

$e^{-1} \bmod \phi(N)$ $\quad \mathbb{Z}_N^*$

Average-case Complexity

- So far, all complexity classes studied in this course are worst-case

  Ex: 3SAT over n variable

  Possible: ① 99% of these formulas $\phi(x_1, \ldots, x_n)$ are solvable in polynomial-time ($poly(|\phi|)$)

  ② 1% -> a single formula require exponential time

  * Under above, $3SAT \notin NP$ and $3SAT \in P$

- Problems that are "easy" on "average"

- Generate n-vertex graph G as follows
- For each edge e of all possible $\binom{n}{2}$ edges
  * Sample $b \xleftarrow{\$} \{0,1\}$
  * If $b=0$, add e to G.
- Then, G:

① $G \in 3COL$ w.h.p. and you can check this in linear time $O(|G|)$

easy

② CLIQUE/INDSET are solvable in time $O(n^{2\log(n)})$ on G

not easy, not difficult

super-poly BUT much less than $2^{n\varepsilon}$ for $\varepsilon > 0$ constant

runtime for best worst-case algorithms

- Also have evidence that there are NP problems which are hard on average.

  * One-way functions!

Distributional Problems/Languages

- **Def.** A **distributional problem** is a pair (L, D) such that
  1. $L \subseteq \{0,1\}^*$ $\Rightarrow$ a language
  2. $D = \{D_n\}_{n \in \mathbb{N}}$ family of distributions over $\{0,1\}^n$.

- $\forall n \in \mathbb{N}$, $x \xleftarrow{\$} D_n$

  $$P_{x \xleftarrow{\$} D_n}[x \in L] = \_\_\_$$

Ex: Random 3SAT

- Let n be # of variables
  m be # of clauses
- Let k be the number of bits needed to represent all possible n-variable, m-clause 3CNF formulas
- Distribution D_k:
  - For each $i \in [m]$
    - Sample $j_1, j_2, j_3 \xleftarrow{\$} [2n]$
    - Define $\phi_i = (y_{j_1} \vee y_{j_2} \vee y_{j_3})$
  - Output $\phi = \phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_m$

$y_j \in \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$

- Known
  ① as m increases, $\Pr_{\phi \leftarrow D_k}[\phi \in 3SAT]$ decreases.

(2) $\exists$ constants $c_1 < c_2$ such that

- If $m < c_1 \cdot n$, then
  $\phi \leftarrow D_k$ is satisfiable w.h.p.
- If $m > c_2 \cdot n$, then
  $\phi \leftarrow D_k$ is unsatisfiable w.h.p.

$m < an$ | $m > an$

3SAT w.h.p | 3SAT

$\alpha = .4\ldots$

(3) $\exists$ f s.t. $c_1 n \le f(n) \le c_2 n$ such that $\forall \varepsilon > 0$, if
$m < (1-\varepsilon) f(n)$, then
$\phi \leftarrow D_k$ is SAT w.h.p.
$m > (1+\varepsilon) f(n)$, $\phi \leftarrow D_k$ is UNSAT w.h.p.

distP : Efficiently Solvable distribution problems.

Def. distP := all pairs (L, D) such that $\exists$ alg A for L and constants $C, \epsilon > 0$ such that $\forall n \in \mathbb{N}$

$$\underset{x \leftarrow D_n}{\mathbb{E}}\left[\frac{time_A(x)^{\epsilon}}{n}\right] \leq C$$

(how to sample: $x \leftarrow D_n$)

* $P \subseteq distP$ for any distribution
  - $L \in P$, D is any family of distributions
  - Set $\epsilon = 1/c$

$time_A(x)$ = runtime of TM in P solving L.

$$time_A(x)^{1/c} = O(|x|^c)^{1/c}$$
$$= O(|x|) = O(n)$$

Computable / Sampleable Distributions

P-computable Distributions

- $D = \{D_n\}_n$ is P-computable if $\exists$ DTM M running in polynomial time such that

$\forall n, x \in \{0,1\}^n$

$$M(x) = \mu_{D_n}(x)$$

$$= \sum_{y \leq x} \Pr_{y \leftarrow D_n}[y]$$

lexicographic ordering

$$\Rightarrow \Pr_{D_n}[x] = \mu_{D_n}(x) - \mu_{D_n}(x-1)$$

$\uparrow$ poly-time computable if μ_{D_n} is

* converse is false if $P \neq NP$

i.e., if $P_{D_n}[x]$ is poly-time computable, it is not true that μ_{D_n} is.

* Uniform dist. is P-computable

P-sampleable

- $D = \{D_n\}_n$ is P-Sampleable if $\exists$ PPT algorithm (or TM) A s.t. $\forall n \in \mathbb{N}$

$x \leftarrow A(1^n)$ is identical to $x \leftarrow D_n$

$SD(A(1^n), D_n) = 0$

$A(1^n) \equiv D_n$

P-computable $\Rightarrow$ P-sampleable

But under plausible complexity theory conjectures,

P-sampleable $\not\Rightarrow$ P-computable.

Average-case NP

① $\text{distNP} = \{(L, D)$ s.t. $L \in NP$ and D is P-computable$\}$

② $\text{sampNP} = \{(L, D)$ s.t. $L \in NP$ and D is P-sampleable$\}$

Average-case Reductions

Def. We say (L, D) **average-case reduces** to (L', D'), $(L, D) \leq (L', D')$ if $\exists$ poly-time computable f

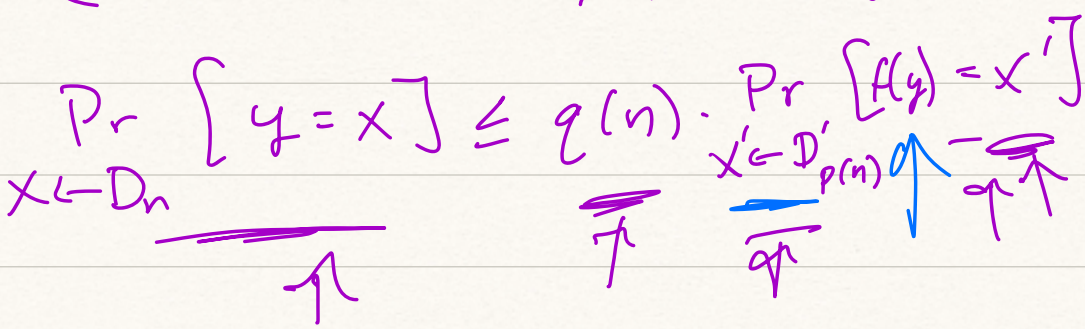
and polynomials p, q such that

(1) (Correctness) $\forall x,\ x \in L \leftrightarrow f(x) \in L'$

(2) (Length-regularity) $\forall x,\ |f(x)| = p(|x|)$

(3) (Domination) $\forall n,\ y \in \{0,1\}^n$

$$\Pr_{x \leftarrow D_n}[y = x] \le q(n) \cdot \Pr_{x' \leftarrow D'_{p(n)}}[f(y) = x']$$



Conjecture (NP is hard on average)

$\exists (L, D) \in \text{samp}NP$ such that $\forall$ poly-time A, $\exists n$ s.t.

$$\Pr_{x \leftarrow D_n}[A(x) = L(x)] < 0.51$$

some constant

* above tells us that $\forall$ NP-hard L' (i.e., 3SAT),

$\exists$ eff. sampleable (or P-samplable) $D' = \{D'_n\}_{n \in \mathbb{N}}$ such that

(L', D') is hard on average (above def.).

$\forall (L, D) \in$ Samp NP

(1) $L \leq_p L'$ (Cook-Levin)

(2) D'_n: ① Sample x from $\rightarrow D_n$ eff-samp.
② apply reduction $f(x) = y$ P poly-time
③ output y

eff. samp.

Open: Does $P \neq NP$ imply NP is hard on average?

Evidence/nice to haves

- Some worst-case NP problems that are equivalent to their average-case formulations
  - Ex: Discrete-log (crypto thing)
  - lattice problems ↗

* If it is easy to solve the problem on average (e.g., on 10% of all instances), then you can easily solve on all instances (i.e., poly-time)

Connections w/ Crypto

- If OWFs exist, then NP is hard on average
  - * Stronger we believe than just $P \neq NP$

Lemma: OWFs exist $\iff$ there exists a "fast" (i.e., poly-time) way to generate "hard" (i.e., super-poly time solvable/decidable) random 3SAT instances w/ "planted solutions."

That is, $\exists$ P-samp. $D = \{D_n\}_n$ over (ϕ, x) where $\phi \leftarrow 3SAT$ and $\phi(x) = 1$ such that $\forall$ poly-time A $\exists$ negligible ε such that for suff. large n

$$\Pr_{(\phi,x)\leftarrow D_n}\left[\phi(A(\phi))=1\right] \leq \varepsilon(n)$$

* Trapdoor OWFs are even stronger

Impagliazzo's Five Worlds

1. Algorithmica
   - $P = NP$ or some morally Equivalent ($NP \subseteq BPP$)
   - Complex things are easy to solve on computers
   - Modern crypto does not exist.

   * Fine-grained crypto

(2) Heuristica

- $P \neq NP$ but $distNP, sampNP \subseteq distP$
- NP easy on average, but hard in the worst case
- $\neq PH$ we don't know if it collapses
- ∄ OWFs don't exist

(3) Pessiland

- $P \neq NP$, $distNP, sampNP \not\subseteq distP$
- OWFs don't exist
- hard problems
- sample them easily, but we can't solve them

(4) Minicrypt

- OWFs exist
- public-key crypto doesn't exist

(5) Cryptomania

- Public-key crypto
- Highly structured NP problems are hard to solve
  - RSA, DL, Lattices