CIS 5560

Cryptography
Lecture 1

Course website:
pratyushmishra.com/classes/cis-5560-s25/

Slides adapted from Dan Boneh and Vinod Vaikuntanathan

Course Staff

Instructor: Pratyush Mishra (me!)

prat@upenn.edu

TAs:

Anubhav Baweja (abaweja@upenn.edu)

Tushar Mopuri (tmopuri@upenn.edu)

Bharath Namboothiry (namboo@upenn.edu)

Course Format

- **Lecture:** Tues/Thurs 1:45-3:15PM Fagin Hall 118
- **Grading:**
  - Participation: 5%
  - HW: 40%
  - Midterm 1: 27.5%
  - Midterm 2: 27.5%
- **Important dates:**
  - Midterm 1: TBD
  - Midterm 2: TBD

Homeworks

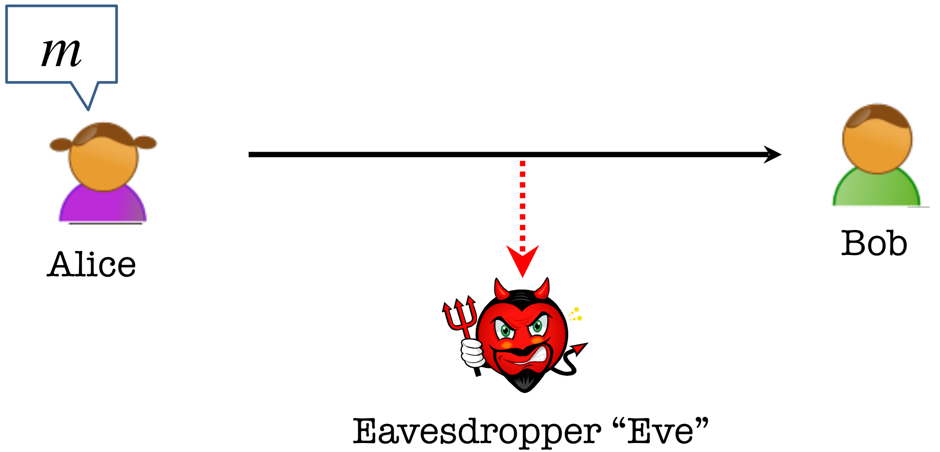
- Usually, 1 per week
- Released on Wednesdays
- Due Friday 5PM
- Drop 2 lowest scores
- Mostly proof-based, with perhaps one programming oriented homework

Important Links

- Class website (WIP): pratyushmishra.com/classes/cis-5560-s25
- EdStem: edstem.org/us/courses/74092/
- Canvas: canvas.upenn.edu/courses/1843458
- Gradescope: gradescope.com/courses/956279

What is Cryptography?

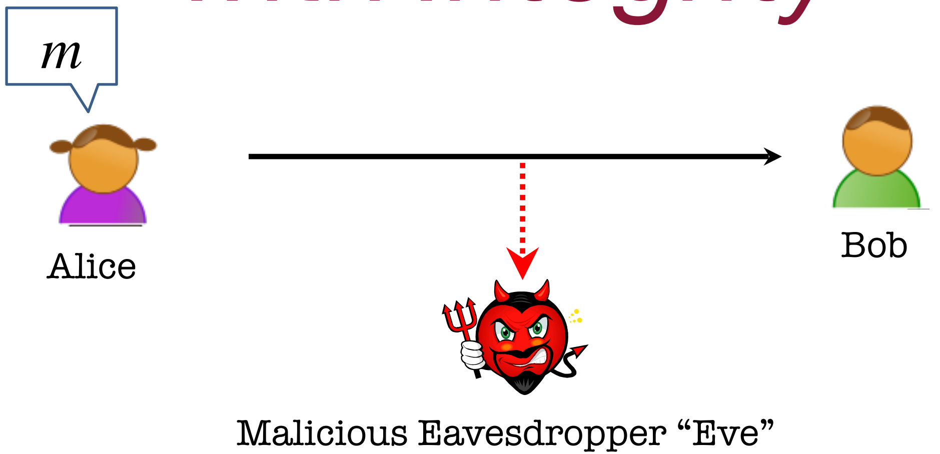
Confidential Communication



Alice wants to send a message m to Bob without revealing it to Eve.

Tool: Encryption schemes
Eg: Caesar Cipher (broken!!), AES, DES, RSA, etc

Confidential Communication with *Integrity*

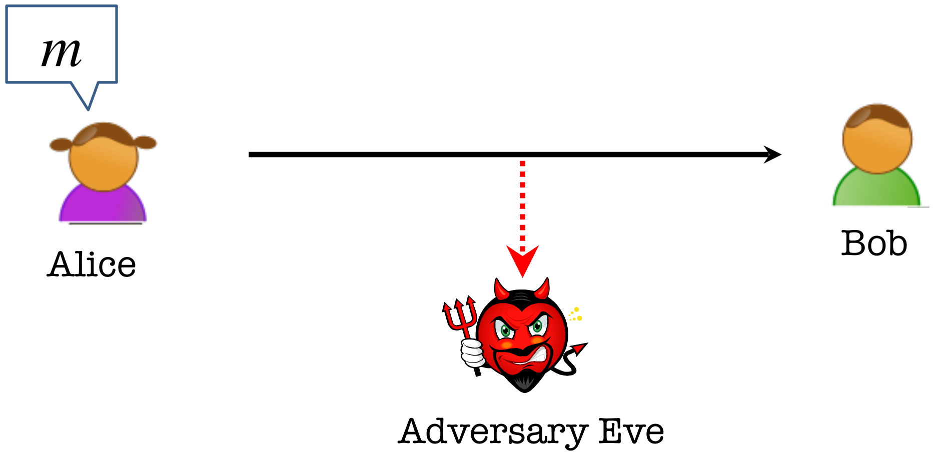


Eve can tamper with messages now

Alice wants to send a message m to Bob without Eve changing it.

Tool: Message Authentication Codes

Communication with *Authenticity*

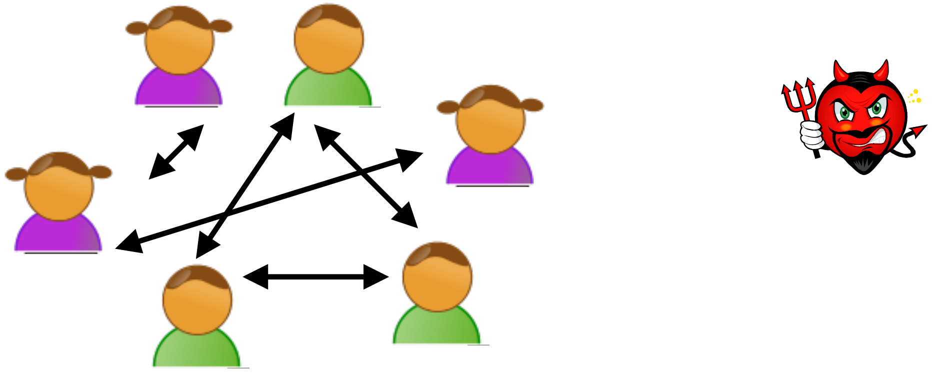


Eve can tamper with messages now

Bob wants guarantee that *only* Alice sent m.

Tool: Digital signatures

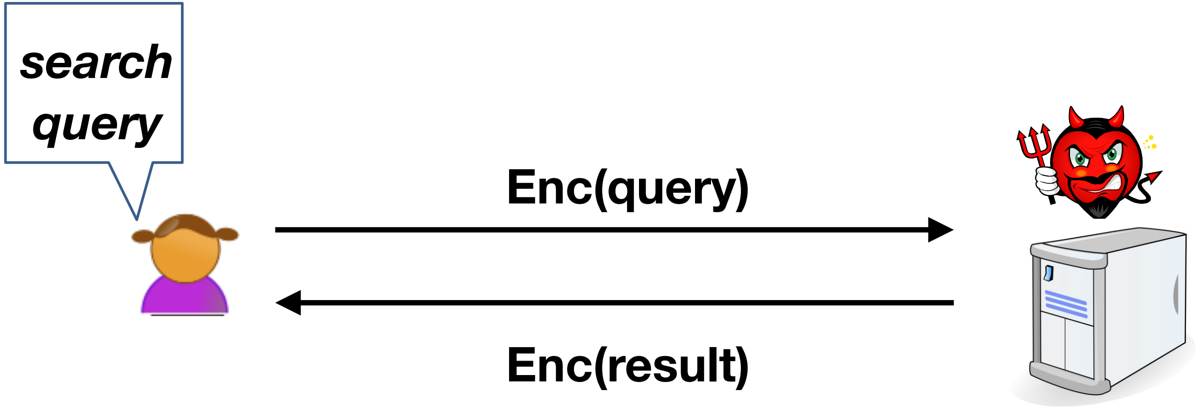
Anonymous Communication



Eve should not be able to tell who is talking to whom

Tool: dining cryptographer networks, onion encryption, etc

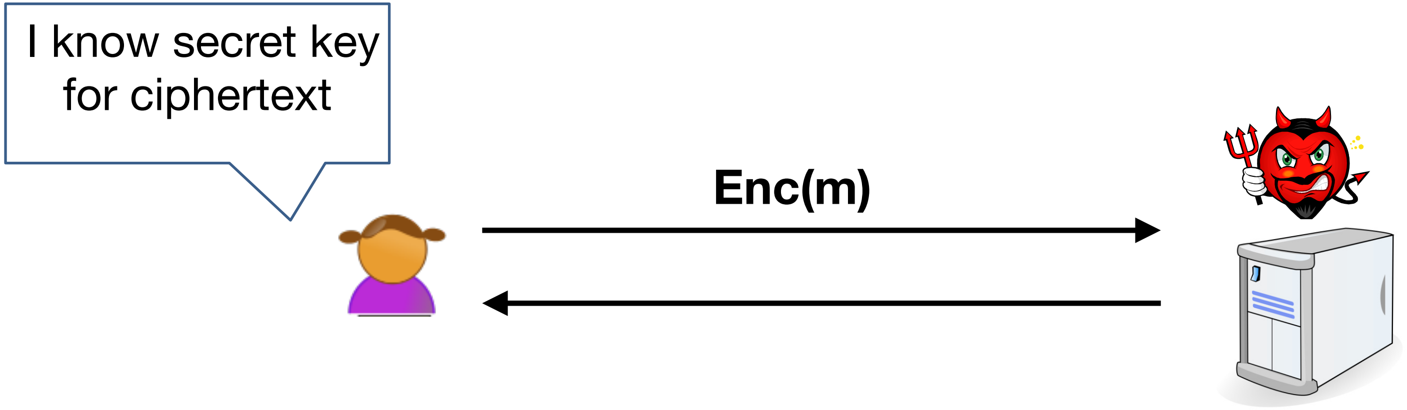
Computation on Secret Data



Eve's server should run computation without learning Alice's data

Tool: Homomorphic encryption, multiparty computation

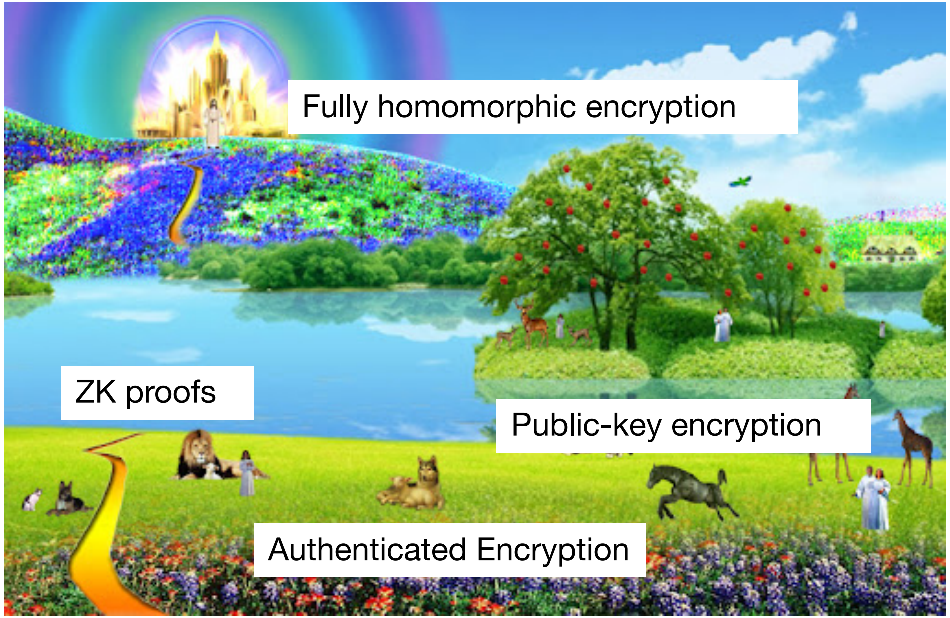
Proofs about Secret Data



Eve's server should be convinced about Alice's claim without learning Alice's secrets.

Tool: Zero knowledge proofs

Crypto is a magical land!



How do we get there? Not magic, but science!

The three steps in cryptography:

- Precisely specify problem, goal, and threat model
- Propose a construction
- Prove that breaking construction under threat model will solve an underlying hard problem

Things to remember

Cryptography is:

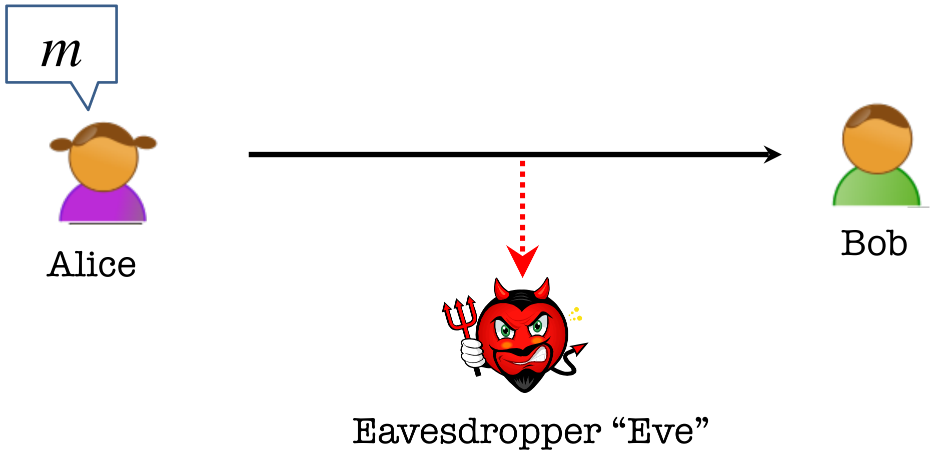
- A tremendous tool
- The basis for many security mechanisms

Cryptography is not:

- The solution to all security problems
- Reliable unless implemented and used properly
- Something you should try to invent yourself
  - many many examples of broken ad-hoc designs

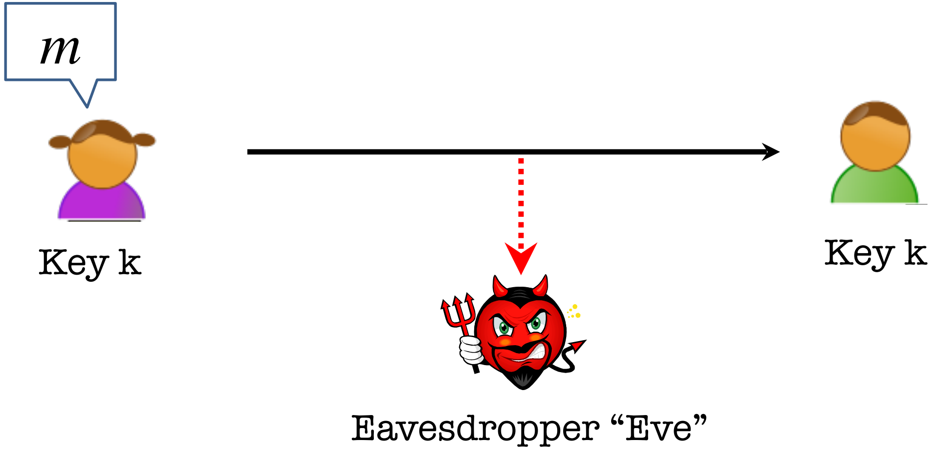
Our First Definition: Symmetric Key Encryption

Secure Communication



Alice wants to send a message m to Bob without revealing it to Eve.

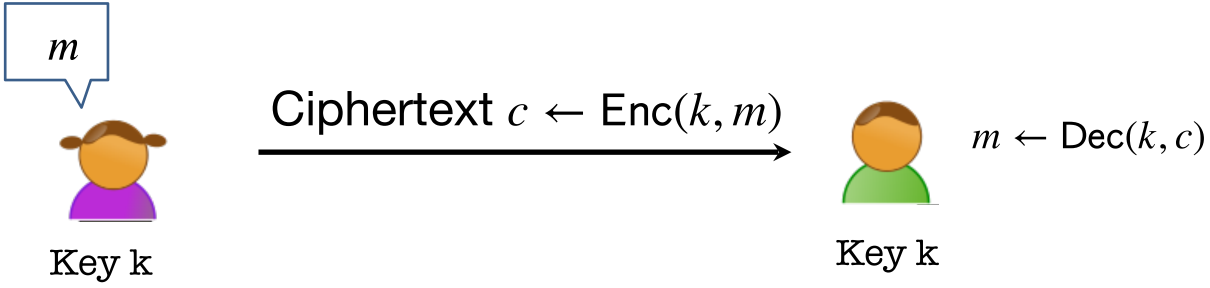
Secure Communication



SETUP: Alice and Bob meet beforehand to agree on a secret key $\mathtt{k}$.

Key Notion: Secret-key Encryption

(or Symmetric-key Encryption)



Three (possibly randomized) polynomial-time algorithms:

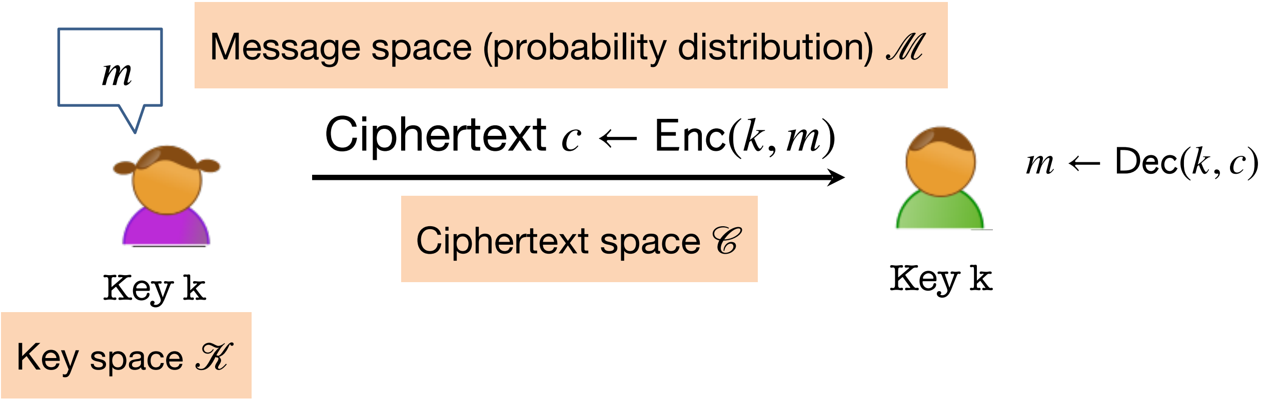
- **Key Generation Algorithm:** $\text{Gen}(1^k) \rightarrow k$

  Has to be randomized (why?)

- **Encryption Algorithm:** $\text{Enc}(k, m) \rightarrow c$

- **Decryption Algorithm:** $\text{Dec}(k, c) \rightarrow m$

Key Property 1: Correctness



- $\forall k \in \mathscr{K}, \forall m \in \mathscr{M}, \mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m$
- **Most basic property: if Bob gets incorrect answer, scheme is useless!**

The Worst-case Adversary

- An arbitrary computationally *unbounded* algorithm **EVE**.*

- Knows Alice and Bob's algorithms Gen, Enc and Dec but does not know the key nor their internal randomness.

  (*Kerckhoff's principle or Shannon's maxim*)

- Can see the ciphertexts going through the channel
  (but cannot modify them… we will come to that later)

Security Definition: What is she trying to learn?

Attempt 1: Caesar cipher

- Idea: shift each letter over by a specific amount N.
- Example: A → D, B → E, …, Z → C
  Encrypt "HELLO CLASS" → "KHOOR FODVV"

- Keyspace $\mathscr{K} =$?
  - Answer: "shifts by $N \in \{0,\ldots,25\}$"
- Gen: Sample $k = N \leftarrow \{0,\ldots,25\}$
- $\text{Enc}(k, m)$: replace each character ch in m with $\text{ch} + N$
- $\text{Dec}(k, c)$: replace each character ch in c with $\text{ch} - N$

Attempt 1: Caesar cipher

- Question: Is this secure? Can adversary recover message?
- Answer: Yes!
  - Just iterate over 26 possible keys, and see which one decrypts!
- Example: “KHOOR FODVV”
  - Try with shift 1 → “LIPPS GPEWW”
  - Try with shift 2 → “IFMMP DMBTT”
  - Try with shift 3 → “HELLO CLASS”

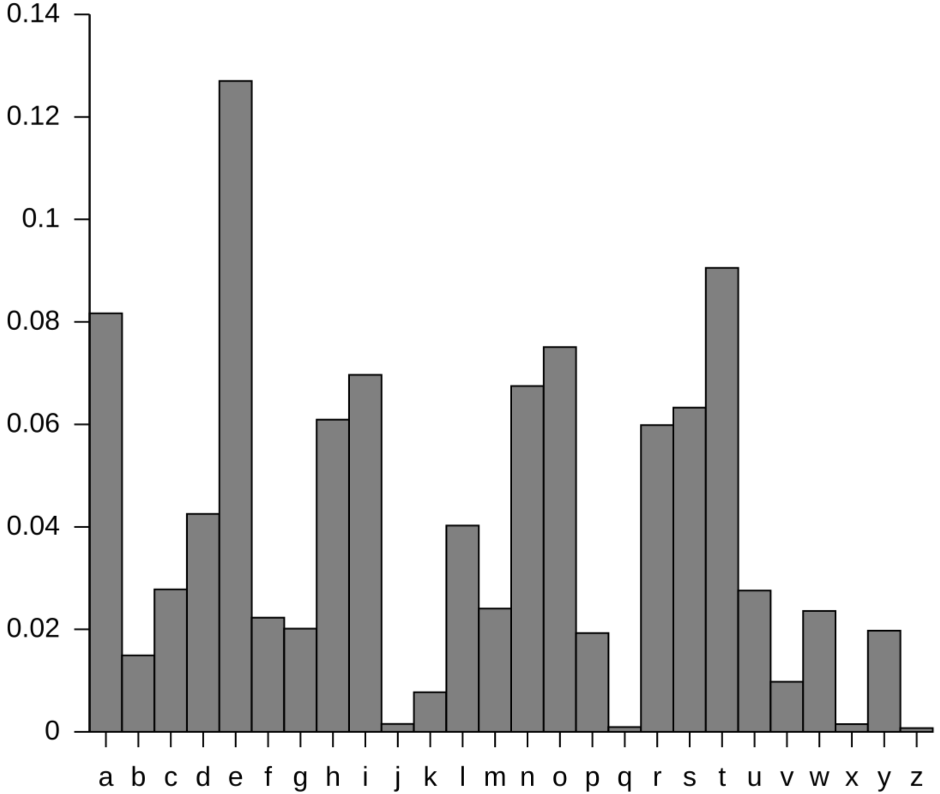
Attempt 2: Substitution cipher

- Idea: Caesar cipher maps letters to other letters in a simple way (shifts)
- Can we use an arbitrary mapping?
- Example: A → E, B → C, …, Z → D

- Keyspace $\mathscr{K} =$?
  - Answer: "all permutations over $\{0,\ldots,25\}$"
- Gen: Sample a random permutation $k = \pi$
- Enc(k, m) : replace each character ch in m with π(ch)
- Dec(k, c) : replace each character ch in c with π^{-1}(ch)

Attempt 2: Substitution cipher

- Question: Does the old attack work?
- Answer: No!
  - Number of permutations = $26! \approx 2^{88}$, can’t try each one!
- Question: Is this secure?
- Answer: Also no!
  - Idea: how many times does “X” show up in a message?
  - How many times does “E” show up in a message?
  - E is much more common!



Can count number of times letters shows up in ciphertext, match with frequency table

What is a secure encryption scheme?

Attacker's abilities: **CT only attack** (for now)

Possible security requirements:

attempt #1: **attacker cannot recover secret key**

$\text{Enc}(k, m) = m$ would be secure

attempt #2: **attacker cannot recover all of plaintext**

$\text{Enc}(k, (m_1, m_2)) = \text{Enc}(k, m_1) \mid\mid m_2$ would be secure

Shannon's idea: **CT should reveal no "info" about PT**

Discrete Probability Primer

- **Probability distribution** P over a finite set S is a function $P : S \to [0,1]$ such that $\sum_{x \in S} P(x) = 1$
- **An event** is a set $A \subseteq S$; $\Pr[A] = \sum_{x \in A} P(x) \in [0,1]$
- **Union bound:** For events A_1 and A_2, $\Pr[A_1 \cup A_2] \leq \Pr[A_1] + \Pr[A_2]$
- A **random variable** X is a fn $X : S \to V$ that induces a dist. on V
- Events A and B are **independent** if $\Pr[A \text{ and } B] = \Pr[A] \cdot \Pr[B]$
- RVs X and Y are **ind.** if $\Pr[X = a \text{ and } Y = b] = \Pr[X = a] \cdot \Pr[Y = b]$

- $S = \{0,1\}^2$
- **Example distribution:** Uniform: for all $x \in S, P(x) = 1/|S|$
- **Example event:** $A = \{x \in S \mid \text{lsb}(x) = 1\}$. $\Pr[A] = 1/2$
- **Example RV:** $X = \text{lsb}$. Here $V = \{0,1\}$, and induced distribution is $\Pr[X = 0] = 1/2 \;;\; \Pr[X = 1] = 1/2$
- **Example independent RVs:** $X = \text{lsb}$ and $Y = \text{msb}$ $\Pr[X(x) = 0 \text{ and } Y(x) = 0] = \Pr[x = 00] = \frac{1}{4} = Pr[X(x) = 0]\Pr[Y(x) = 0]$

Uniform RV

- A **Uniform RV** is $R : S \to S$ that induces a uniform dist on S.
- That is, for all $x \in S$, $\Pr[R = x] = 1/|S|$

Randomized algorithms

- Deterministic algorithm: $y \leftarrow A(m)$
- Randomized algorithm: $y \leftarrow A(m; R)$ where $R \xleftarrow{\$} \{0,1\}^n$
  - Output is a random variable $y \xleftarrow{\$} A(m)$