Announcements:
- PSet 2 out; due March 23
- Feedback form

Plan for today:
- Quick recap: goals and constructions
  - “Create” randomness: PRF
  - Hide a message: Encryption
  - Authenticate a message: MAC
  - CPA/CCA: clarification of MACs
- Review hash functions
  - Random Oracle Model (ROM)
  - Families and salts
  - Properties
- Birthday problem
“Create randomness”:

...actually, “create random-looking” strings.

Tool: Pseudo-Random Function (PRF)

Idea: take \( n \) bits of randomness \( k \leftarrow \{0,1\}^n \).

Then a PRF \( F \) lets us create \( \text{poly}(n) \) “random-looking” strings from \( k \).

Hide a message \( m \): Encryption scheme \((\text{gen, enc, dec})\)

“Confidentiality”

Authenticate a message \( m \): Message Authentication Codes (MACs)

“integrity”

To try to clarify some confusion in lecture:

\( \rightarrow \) MACs are defined independent of encryption, so we wrote message-tag pairs as

\[ (m, \text{MAC}(k, m)) \]

for an arbitrary input \( m \) to authenticate.

\( \rightarrow \) For CCA security, we authenticate a ciphertext:

\[ \text{shared } k \leftarrow \text{Gen}(1^n), \ c \leftarrow \text{Enc}(k, m) \]

send \( (c, \text{MAC}(k, c)) \).
Hash Functions

Why PRFs? Why is randomness “expensive”?

→ A truly random function is not efficiently computable. Call it H

In our context, we have a compression:

\[ H : \{0, 1\}^* \rightarrow \{0, 1\}^d \]

map is fixed for H

so \( H(x) = y \)

(Consistently! We don’t write \( y \leftarrow H(x) \).)

But map is random between distinct \( x, x' \in \{0, 1\}^* \)

ie, \( H(x) \) and \( H(x') \) are independent

(One thinks of \( H \) itself as sampled from the set of random maps onto \( \{0, 1\}^d \))
Back to efficiency: We said $H$ is not efficiently computable. Suppose we built a circuit for $H$, $C_H$.

What is the size of $C_H$?

Recall range of $H$ is $\{0,1\}^d$, size $2^d$ so $C_H$ needs to be able to compute $\approx 2^d$ different conditions on its input.

Therefore the size of $C_H$ is exponential and can’t be computed in poly time.

$\Rightarrow$ The random oracle model (ROM) is a useful, but controversial model!

In the ROM, $H$ itself can be used to compute PRFs, even if $P=NP$!

(Why? $H$ is a OWF even if $3SAT \in P$, $\delta$ OWF $\Rightarrow$ PRE).

See: Katz-Lindell book, section called “Is the Random-Oracle Methodology Sound?”
ROM gives us a theoretical arena.

In practice we don’t have oracle access to \( H \), so we use SHA-256 and pray...

**Notation:**

Function families: \( \{h_s\} \) indexed with \( s \)

- Each \( h_s: \{0,1\}^* \rightarrow \{0,1\}^d \)

With salts: \( h_s(x) = h(s \| x) \)

…again, commonly SHA-256

What properties of \( h_s \) might we want?
Properties of hash functions

One way:

given $s, y$ (where $y = h_s$(something)),

infeasible to find an input $x$ in $h_s^{-1}(y)$.

Note: this is a set:

$\left\{ h_s \right\}$

Collision resistance:

given $s,$

infeasible to find distinct $x, x'$ s.t. $h_s(x) = h_s(x')$

Target Collision resistance:

given $s, x,$

infeasible to find $x'$ s.t. $x' \neq x$ and $h_s(x') = h_s(x)$

CR/TCR distinction:

TCR says "it is hard in general"  

CR says "there are hard cases" (i.e., I tell you a target $x$)

\[ \therefore \text{CR is stronger} \]

\[ \text{CR} \Rightarrow \text{TCR} \Rightarrow \text{OW} \]

\[ \neg \text{OW} \Rightarrow \neg \text{TCR} \quad \text{bc can take given} \]

\[ x \quad \text{and compute} \quad y \quad \text{as} \quad h_s(x) \]
Pseudo random:

given $s$, infeasible to distinguish between

\[
\begin{cases}
    \{ h_{s} \} \\
    \{ A \}
\end{cases}
\]  $\text{VS}$  \[
\begin{cases}
    \{ H \} \\
    \{ A \}
\end{cases}
\]

Non-malleable:

given $s$, $h_{s}(x)$ for some random $x$
infeasible to find $h_{s}(x')$ for a related $x'$

\[x' = f(x)\] for known $f$, not nec. known $x$

(this should remind you of the discussion from lecture for motivating CCA security — $(r, F(k,r)\oplus m)$ is CPA, easy to send $(r, F(k,r)\oplus m\oplus 1)$... so this encryption scheme is malleable)
Birthday Paradox

(appendix of Katz-Lindell)

A faux "paradox"

→ w/ 23 people in a room, > 50% chance that 2 people share a birthday!

But there are 365 days!

Generalizing:

Sample set of size N
Take q samples
Probability p of collision

\[ N = 365 \text{ for birthdays} \]
\[ q = 23 \text{ above} \]
\[ p = 0.5 \text{ above} \]

Theorem: \[ p(q, N) = \Theta(\sqrt{N}) \]

Proof: We will show \[ \frac{q(q-1)}{4N} \leq p(q, N) \leq \frac{q^2}{2N} \]
First: \[ \rho \leq \frac{q^2}{2N} \]

Write \( q \) samples as \( x_i \) for \( i = 0, \ldots, q-1 \)

Union bound over all ways to collide:

\[ \rho \leq \sum_{i,j} \Pr[x_i \text{ and } x_j \text{ collide}] \]

The probability of two \( x \)'s colliding is at least \( \frac{1}{N} \)

\[ \Rightarrow \rho \leq \binom{q}{2} \frac{1}{N} \leq \frac{q^2}{2N} \quad \text{because} \quad \binom{q}{2} \leq \frac{q^2}{2} \]
p is probability of collision.

Pairwise over \( x \)'s, the probability two \( x \)'s don't collide is:

\[
\prod_{i=1}^{q-1} \left( 1 - \frac{i}{N} \right) \leq \prod_{i=1}^{q-1} e^{-i/N}
\]

Recall Taylor series of \( e^x = 1 + x + O(x^2) \)

Move \( \prod \) to sum inside exponential:

\[
\Pr[ \text{No collision}] \leq \exp \left[ - \sum_{i=1}^{q-1} \frac{i}{N} \right] = \exp \left[ - \frac{q(q-1)}{2N} \right]
\]

Nice summation result:

\[
\sum_{i=1}^{q-1} i = \frac{q(q-1)}{2}
\]

Either we have a collision or we don't:

\[
p = 1 - \Pr[ \text{No collision}]
\]

\[\Rightarrow p \geq 1 - \exp \left[ - \frac{q(q-1)}{2N} \right]\]

\[\Rightarrow p \geq \frac{q(q-1)}{4N} \quad (\text{back to Taylor series, } e^{-x} \leq 1 - \frac{x}{2} \text{ for } |x| < 1)\]