More Birthday Bounds

Basic bound: Hash function w/ n bits output
⇒ need to evaluate H on ~2^{n/2} inputs
before seeing a collision w/ good
probability (90, 95, 99 %)

Limitation: collision can be between any 2
values that we consider ⇒ no control!

Birthday bound for Meaningful - Collisions

Idea: have m "good" messages, m "bad" messages

Ex:

Dear {Dr, Prof} Rosulek

Congrats! Your {position, appointment}
has been {extended, renewed}.
{Sincerely, Yours truly}, the Provost

Dear {Dr, Prof} Rosulek

{I'm sorry, I regret to inform you}

that Your {position, appointment}
has been {terminated, made redundant}.
{Sincerely, Yours truly}, the Provost
**Claim:** need \( m \sim 2^{n/2} \) messages in each category to have good probability of collision between "good" msg & "bad" msg.

**Note:** birthday bounds show up in other places too! (not just with hash functions)

**Ex:** ECB mode:

\[
\begin{array}{ccc}
F & F & F \\
\downarrow & \downarrow & \downarrow \\
\end{array}
\]

Same input blocks \( \Rightarrow \) same output block.

CBC encryption:

\[
\begin{array}{ccc}
\text{IV} & F & b \\
\downarrow & \downarrow & \downarrow \\
\end{array}
\]

\[
\begin{array}{ccc}
a & c & F \\
\downarrow & \downarrow & \downarrow \\
\end{array}
\]

\( F \) has \( n \) bits of output.

\( \Rightarrow \) security breaks down after \( \sim 2^{n/2} \) plaintext blocks.

\( \Rightarrow \) collisions lead to duplicated inputs to \( F \Rightarrow \) duplicated outputs \( \Rightarrow \) leaked information.

**Password Hashing:** (use salt)

**Scenario:** user database, collection of (username, passwd) entries

Store \( H(\text{passwd}) \) instead of passwd in clear.
Problems:
1. same password $\Rightarrow$ same hash
2. vulnerable to pre-computed dictionary attack
   $\Rightarrow$ using hash chains & rainbow table

**Dictionary Attack**
most people pick passwords badly
compute $H(p)$ for all $p$ in large dictionary
(mapping of $p \leftrightarrow H(p)$)

Feasible?

Dictionary = English words
$\sim 1$ million entries $\times 16$ bytes $\Rightarrow 16$ MB

Dictionary = 10-char strings of upper/lower letters
$52^{10} \times 16$ bytes $\Rightarrow 2^{61}$ bytes = exabyte
$= 1$ billion GB