Reminders:

Mon = 66-110; Wed = 56-114
https://courses.csail.mit.edu

Admin:

Pset #1 due Friday. (Submit electronically.)
No recitations planned for remainder of course.

Today:

Finish OTP security proof
generating random bits
Pseudo-random pads; RC4
Block ciphers: DES, AES
"Modes of operation"

did

to

Notes:

3 talks this week:
Tue: Certifiable Quantum Dice (see your email) 4:15 32-155
Thu: 0-day economics 10:30 32-463
Thu: Verisign 4:00 326-882
**One-Time Pad (Security proof)**

\[
\begin{align*}
M &= 10110000 \cdots \quad (\lambda\text{-bit string}) \\
K &= 01101000 \cdots \quad (\text{xor } \lambda\text{-bit "pad" (key)}) \\
C &= 11011000 \cdots \quad (\lambda\text{-bit ciphertext}) \\
M &= 10110000 \cdots \\
\end{align*}
\]

\[(M \oplus K) \oplus K = M \oplus (K \oplus K) = M \oplus 0^n = M\]

**OTP is information-theoretically secure = Eve**

- can not break scheme, even with unlimited computing power

(Compare to **computationally secure**: requires assumption that Eve has limited computing power (e.g. can’t factor large numbers.))

**Model Eve’s uncertainty via probabilities**

\[
P(M) = \text{Eve's prior probability that message is } M
\]

\[
P(M | C) = \text{Eve's posterior probability that message is } M, \\
\text{after having seen cipher-text } C.
\]

**Theorem:** For OTP, \( P(M) = P(M | C) \)

\[\implies \text{"Eve learns nothing by seeing } C\)"
Proof:

Assume $|M| = |K| = |C| = \lambda$.

$$P(k) = 2^{-\lambda} \quad \text{(all } \lambda\text{-bit keys equally likely)}$$

**Lemma:** $P(C|M) = 2^{-\lambda}$

$$P(C|M) = \text{Prob of } C, \text{ given } M$$

$$= \text{Prob that } K = C \oplus M$$

$$= 2^{-\lambda}.$$  

$$P(C) = \text{Probability of seeing ciphertext } C$$

$$= \sum_{m} P(C|M) \cdot P(M)$$

$$= \sum_{m} 2^{-\lambda} \cdot P(M)$$

$$= 2^{-\lambda} \cdot \sum_{m} P(M)$$

$$= 2^{-\lambda} \cdot 1 = 2^{-\lambda}, \quad \text{(uniform)}$$

$$P(M|C) = \text{Prob of } M, \text{ after seeing } C \text{ (posterior)}$$

$$= \frac{P(C|M) \cdot P(M)}{P(C)} \quad \text{(Bayes' Rule)}$$

$$= \frac{2^{-\lambda} \cdot P(M)}{2^{-\lambda}}$$

$$= P(M) \quad \text{QED}$$

This is perfect secrecy (except for length $\lambda$ of $M$).
Notes:

Users need to:

- generate large secrets
- share them securely
- keep them secret
- avoid re-using them (google "Venona")

$$C_1 \oplus C_2 = (M_1 \oplus K) \oplus (M_2 \oplus K)$$

$$= M_1 \oplus M_2$$

from which you can derive

$$M_1, M_2$$ often.

Theorem: OTP is malleable.

(That is, changing ciphertext bits causes corresponding bits of decrypted message to change.)

OTP does not provide any authentication of message contents or protection against modification ("mauling").
How to generate a random pad?

- Coins
- Dice
- Radioactive sources (old memory chips were susceptible to alpha particles)
- Microphone, camera
- Hard disk speed variations
- Intel 82802 chip set
- User typing or mouse movements
- Lava lamp → camera
- Alpern & Schneider:

  \[ A \text{ - Eve} \rightarrow \text{B} \]

  Eve can't tell who transmits. A & B randomly transmit beeps. They can derive shared secret.

- Quantum Key Distribution

  Polarized light: \[ \uparrow \leftrightarrow \downarrow \leftrightarrow \downarrow \]

  Filters (example filter):

  \[ \bigotimes \leftrightarrow \bigotimes \leftrightarrow \bigotimes \leftrightarrow \bigotimes \]

  result: \[ \uparrow \leftrightarrow \downarrow \leftrightarrow \uparrow \leftrightarrow \downarrow \leftrightarrow \downarrow \leftrightarrow \downarrow \]

  A sends single photons, polarized randomly. B publicly announces filter choices. Then they know which bits they should have in common.

- ref today's lecture on Certifiable Quantum Dice

- "Noise diodes" 5V

  \[ \uparrow \rightarrow \frac{A/D}{\rightarrow \text{bits}} \]
• **Satellite based:**
  - Broadcast bit
  - A & B can derive key that Eve doesn't know if errors to A, B, Eve are independent.
  - Or: Eve may have limited memory.

• **Book-based pad:** Take 4 or 5 randomly chosen starting points in book, XOR texts together.

---

"Pseudo-random pad": "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin."

(John von Neumann, 1951)

Pseudo-random generator

<table>
<thead>
<tr>
<th>seed</th>
<th>[\rightarrow]</th>
<th>PR &quot;pad&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>\approx 256 bits</td>
<td></td>
<td>arbitrarily long</td>
</tr>
</tbody>
</table>

computational security: if adversary can't tell pad from truly random pads of same length.
"Ron's code 4"

Array \( S[0..255] \), permutation of 0..255 initialized from seed (key)

\( i, j \quad 0 \leq i \leq 255, \quad 0 \leq j \leq 255 \) (pointers into \( S \))

Generate PR pnd:

\[
\begin{align*}
\text{i} &= 0 \\
\text{j} &= 0 \\
\text{while True :} & \\
& \quad \begin{cases} \\
& \quad \text{i} = (i+1) \mod 256 \\
& \quad \text{j} = (j + S[i]) \mod 256 \\
& \quad \text{swap } S[i] \leftrightarrow S[j] \\
& \quad \text{output byte } S[(S[i] + S[j]) \mod 256] \\
& \end{cases}
\end{align*}
\]

Widely used.

Setup of \( S \) from seed is weak; good idea to discard first 1024 bytes of output...

Note bug if you implement \( A \leftrightarrow B \) (swap) as

\[
\begin{align*}
A &= A \oplus B \\
B &= B \oplus A \\
A &= A \oplus B
\end{align*}
\]

(doesn't work if \( A, B \) are same memory location; sets table \( S \) to 0's slowly... !)