Last time

• Pseudorandom Functions
• PRFs → CPA secure encryption

Today

• Authentication (MAC) using shared keys
• Getting MACs from PRFs
• Security against active attacks (CCA security)
PS 2 extension

• Due end (10pm) of 28th (Wed).
PS2 clarification for problem 3

\[
\begin{align*}
\mathbf{m} &= \mathbf{m}_1 \mathbf{m}_2 \mathbf{m}_3 - \mathbf{m}_e \\
&= \mathbf{m}_1 \mathbf{m}_2 - \mathbf{m}_e = \mathbf{m}_1 || \mathbf{m}_2 - || \mathbf{m}_e \\
\end{align*}
\]

\[\text{Enc(key, } m; r) = \left[ \text{Enc(key, } m_1; r_1) || \ldots || \text{Enc(key, } m_\ell; r_\ell) \right] \]
\[= (r_1, r_2 - r_\ell)\]

**CPA Security Loss:** Single-message security does NOT hold.

\[
\text{Enc} \left( k \left[ m_1, m_2, m_3 \right] \right) = \left( \text{Enc} \left( k, m_1 \right), \text{Enc} \left( k, m_2 \right), \text{Enc} \left( k, m_3 \right) \right)
\]
\[
\mathbf{m} = [m_1, m_2, m_3]
\]
\[
\mathbf{m} = [m_1 \pm m_2]
\]
Review: randomness in encryption

- Encryption’s own randomness is usually *not* revealed (even though we did reveal it in our specific construction last time)

\[
\text{Enc}(k, m, r) = \text{Enc}(m) \rightarrow C
\]

\[
f_k(x) \rightarrow y
\]

\[
\text{secret key}
\]

\[
\text{to encrypt } |m| = l.
\]

\[
P\text{ick } r \text{ of length } n
\]

\[
\text{output } c = [r, m \oplus f_k(r)]
\]

\[
\text{secret key}
\]

\[
\text{output}
\]

\[
m = f_k(r) \oplus y
\]
What CPA security guarantees

• It guarantees multi-message security (passive attacker)

• It also guarantees a semi-active attacker (somehow obtaining encryptions of messages that they choose.)

• It does not say anything about “active” attacks. What are they?
What could go wrong with a CPA secure scheme?

1. Resending a message.

2. Resending fresh encryptions?

3. Modify the message: flip the last bit of ciphertext.

\[ c = [r, m \oplus F_k(r)] \]

\[ y \rightarrow \text{flip the last bit of } y \]

\[ c' \text{ if Bob decrypts } c' \rightarrow m' = ? \text{ but last bit flipped.} \]
Authentication:
How would Bob know Alice sent this message? ...
... if Eve is not passive anymore...
Authentication

• Could be applied to ciphertexts, but it is a meaningful notion on its own, even for plaintexts without any encryption involved...

• In the private-key (i.e. symmetric-key) setting it is called: **Message Authentication Code (MAC)**

• There is a “public-key” version of the same thing known as: “Digital Signatures”. We will talk about it later.

• If combined with CPA-secure encryption **properly**, gives rise to a more secure encryption that handles “active” attacks as well..
Message Authentication Code

- Alice and Bob share key $k$.
- Alice generates $\text{MAC}_k(m) \rightarrow \text{tag}_m$ and sends: $[m, \text{tag}_m]$
- Bob receives $[m, \text{tag}_m]$ runs $\text{Verify}_k(m, \text{tag}_m)$ and accepts or rejects

**How to define security?**

- Infeasible for Adv to generate a valid $[m, \text{tag}_m]$
- Adv gets to see $[m, \text{tag}_m]$ for many chosen $m$'s before forging for a new $m$
Formal definition of security

The message authentication experiment $\text{Mac-forge}_{A, \Pi}(n)$:

1. A key $k$ is generated by running $\text{Gen}(1^n)$.

2. The adversary $A$ is given input $1^n$ and oracle access to $\text{Mac}_k(\cdot)$. The adversary eventually outputs $(m, t)$. Let $Q$ denote the set of all queries that $A$ asked to its oracle.

3. $A$ succeeds if and only if (1) $\text{Vrfy}_k(m, t) = 1$ and (2) $m \notin Q$. In that case the output of the experiment is defined to be 1.

**DEFINITION 4.2** A message authentication code $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$ is existentially unforgeable under an adaptive chosen-message attack, or just secure, if for all probabilistic polynomial-time adversaries $A$, there is a negligible function $\text{negl}$ such that:

$$\Pr[\text{Mac-forge}_{A, \Pi}(n) = 1] \leq \text{negl}(n).$$
Constructing MACs using PRFs

• Suppose $F_k(\cdot)$ is a PRF with key, input, output lengths $n,*,\ell = n$
• How do we generate MAC tags for messages?

\[ \text{tag} = F_k(m) \]

\[ \text{Vrf}_k(m, t_k) : \begin{cases} 1 & \text{if } F_k(m) = t_k \\ 0 & \text{otherwise} \end{cases} \]
Proof of Security

Start by assuming

\[\exists \text{poly}(n) \text{ Adv. A} \]

E - breaking our scheme M.

\[\frac{1}{\text{poly}(n)} < E(n) \]

then we show

\[\text{PRF F secure } \Rightarrow \text{our } \text{MAC}_{\text{K}_f} \text{ is secure.} \]

if M is NOT secure.

\[\Rightarrow F \text{ is also not secure.} \]

B breaks PRF in poly time by \(\frac{1}{\text{poly}}\) by \(\epsilon\) "chance"
Proof of Security

Ideal World:

- $F_k(\cdot)$ is substituted with $R(\cdot)$
- $R$ is a function from all \{set of queries\} of $\text{ADVR}$
- $R$ has no pre-chosen answer. Given any $x$, it picks $R(x)$ at random and never seen it again.

Goal 1: $P_{\text{I}} \leq \text{neg.}$

Goal 2: $|P_{\text{I}} - P_{\text{R}}| \leq \text{neg.}$

Formally: attacker $\text{APRF}$

First runs $\text{ADVR}\sim\text{mit}$

Then $\text{APRF}$ asks $m$ from Oracle $O(m) = t$

If $P_{\text{I}} - P_{\text{R}} > \text{non-neg.}$, then $\text{APRF}$ wins in breaking PRF $F_k(\cdot)$
Chosen cipher-text security:

• combining CPA security with MACs to handle active attacks.
Password verification example