Permutation (review from CS 255)

Questions: note cards

A permutation on \{0,1\}^n is a (one-to-one) mapping

\[ f : \{0,1\}^n \rightarrow \{0,1\}^n. \]

In crypto terms, \( f \) takes \( n \)-bit strings to \( n \)-bit strings.

A "block cipher" is defined over sets \((\mathcal{X}, \mathcal{K})\) and consists of algs \((E, D)\):

\[ E : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{X} \]
\[ D : \mathcal{X} \times \mathcal{K} \rightarrow \mathcal{X} \]

Idea: Both \( E, D \) are efficient

* implement permutation on \( \mathcal{X} \)
* are inverses of each other for given \( \mathcal{K} \)
* satisfy some security notion

("pseudo-random permutation" ... defined in 1sec)

Compare with PRF

\( \rightarrow \) They are hard to construct!

- Any sufficiently crazy circuit might give you a PRF
  (plausibly, at least)
  Very hard to build a PRF! A random function won't do it.
- Can do everything we need with PRF in CTR Mode
  + TLS etc.
- Old days: used them for reasons we can discuss...
- One reason to care about PRFs: Format-preserving encryption

\[ \text{Credit card} \quad \rightarrow \quad \text{credit card} \]
\[ \text{Zip code} \quad \rightarrow \quad \text{Zip code} \]
\[ \text{Phone #} \quad \rightarrow \quad \text{Phone #} \]

Uses: Small, backwards compatible, valuable

\( \rightarrow \) C.S. Voltage
Pseudo Random Permutation (PRP) ("Block Cipher")

Formalizes one security notion for a block cipher

\[ \begin{align*}
\text{Challenger} & \quad \begin{cases} 
\sigma \in \mathcal{K} \\
\sigma^\prime = 0 \in \mathcal{K} \\
\sigma^\prime = 1 \in \mathcal{K} \\
\mathcal{F}(\cdot) \leftarrow \text{Perms}[\mathcal{K}]
\end{cases} \\
\text{Adv} & \quad \begin{cases} 
\mathcal{X} \\
\mathcal{F}(\cdot) \leftarrow \mathcal{E}(\cdot)
\end{cases} \\
\text{Solver} & \quad \begin{cases} 
\mathcal{X} \\
\mathcal{F}(\cdot) \leftarrow \text{Perms}[\mathcal{K}]
\end{cases} \\
\end{align*} \]

Intuitively:

\[ \begin{align*}
\forall \sigma \in \mathcal{K} \\
\exists \mathcal{F}(\cdot) \leftarrow \mathcal{E}(\cdot) \\
\mathcal{F}(\cdot) \leftarrow \text{Perms}[\mathcal{K}]
\end{align*} \]

[Note: we didn't say anything about \( \mathcal{D}(\cdot) \).
Can define a stronger notion of security "strong PRP" allowing \( \mathcal{D}(\cdot) \) non-]

\[ \forall \mathcal{K} \in \{0, 1\}^n \quad \text{consider a family } \{\mathcal{E}_n\}_{n=1}^\infty \]

Then \( \mathcal{E}_n \) must run in time \( \text{poly}(n) \).

We say \( \mathcal{E}_n \) is a PRP if, for all ppt. algorithms \( \mathcal{A} \),
\( \mathcal{A} \) (running in time \( \text{poly}(n) \)), there exists a negl function \( \text{negl}(n) \) s.t.
\[ \text{PRPAdv[}\mathcal{A}, \mathcal{E}] := |\Pr[\mathcal{A} \text{ outputs } 1] - \Pr[\mathcal{A} \text{ outputs } 1]| \leq \text{negl}(n) \]
for all \( n \) sufficiently large.

\[ \Rightarrow \text{From CS255: Remember to never use a PRP directly } \]
\[ \text{for encryption!} \]
Building a PRF from a PRP \( (\text{PRP} \Rightarrow \text{PRF}) \)

(Theorem 4.3 in Boneh Shoup)

A good PRP is also a good PRF.

Intuition: The only difference between PRF and PRP is collisions. If you see a collision, you can't distinguish.

PRF Switching Lemma. Let \( E \) be a PRP over \((X, X)\), let \( |X| = 2^n \).

Then, if an adversary makes \( q \) queries, then

\[
|\text{PRPAdv}[A, E] - \text{PRFAdv}[A, E]| \leq \frac{q^2}{2^{2n}}.
\]

Note: If \( A \) is efficient/ppt, then \( q \in \text{poly}(n) \), so

\[
\frac{q^2}{2^{2n}} \in \text{negl}(n) \Rightarrow E \text{ is a secure PRF}.
\]

Note: To be completely formal, we need to define a family of PRPs \( \{E_n\}_{n=1}^{\infty} \). We omit this notation BUT ASK IS UNCLEAR.

Proof. By a hybrid argument:

Game 0: Challenger uses \( S_i = E(k, 0) \) for \( k \in \mathbb{R}^n \)

Game 1: \( S_i \in \mathcal{R} \text{ Perm}[X] \)

Game 2: \( S_i \in \mathcal{R} \text{ Funcs}[X, X] \subseteq \text{Random Function} \)

By definition

\[
|\text{Pr}[A \text{ outputs } 1 \text{ in Game 0}] - \text{Pr}[A \text{ outputs } 1 \text{ in Game 1}]| \leq \text{PRPAdv}[A, E] \leq \text{negl}(n)
\]

\[
|\text{Pr}[A \text{ outputs } 1 \text{ in Game 1}] - \text{Pr}[A \text{ outputs } 1 \text{ in Game 1}]| \leq \frac{q^2}{2^{2n}} \text{ Want to show}
\]
To bound $|P_1 - P_2|$, we observe that, conditioned on all responses to the queries being distinct, there is no way for the adversary to distinguish $P_R$ from a PRF (almost by definition).

So $|P_1 - P_2| = \Pr\left[ \text{A sees a query response} \right]$.

$\Pr[\text{collision}] \leq \Pr_{x,y} \left[ \exists x, y \text{ s.t. } S(x) = S(y) \text{ in game } 1 \right]$ by union bound.

\[
\leq \left( \frac{\text{# of pairs}}{\text{pairs}} \right) \Pr_{x,y} \left[ S(x) = S(y) \right]
\]

\[
\leq \left( \frac{\text{# of pairs}}{\text{pairs}} \right) \sum_{x \neq x^*} \Pr_{x,y} \left[ S(x) = x^* \land S(y) = x^* \right]
\]

\[
\leq \left( \frac{q^n}{2} \right) 2^n \left( \frac{1}{2^n} \right)^2
\]

\[
\leq \frac{q(q-1)}{2} \frac{1}{2^n}
\]

\[
\leq \frac{q^2}{2^{2^n}}
\]

So, $|P_1 - P_2| \leq \frac{q^2}{2^{2^n}}$. This completes the proof.

$P_{\text{Adv}} = \Pr[\text{P sees a query}] - \Pr[\text{P sees a query}]$ of $P_R$ is all we need to do.

Ask a question now!

Remark: This PRE is "secure up to the Birthday Bound".

Can we get a cipher that is indistinguishable from PRE? Its advantage is

\[
\frac{q^2}{2^n}
\]

For AES-128, $q = 2^{64}$ queries is not that many.

Yes. We sometimes care about this.

See "Sweet32" attack for a different place that the Birthday Bound comes up.
Union Bound

Let \( E_1, E_2, \ldots, E_n \) be events defined over some space. Then \( \Pr[ E_1 \lor E_2 \lor \cdots \lor E_n ] \leq \sum_i \Pr[ E_i ] \).

\[ \Rightarrow \text{One of the most useful tools in security analysis.} \]

\[ \Pr[ \text{Bad}_1 \lor \text{Bad}_2 \lor \cdots \lor \text{Bad}_n ] \leq \sum_i \Pr[ \text{Bad}_i ] \]
The Most Interesting Direction \( (PRF \Rightarrow PRP) \)

A priori, it's not clear that this is even possible. Think about it.

Idea: Use a Feistel network.

- Invented by Horst Feistel. (German-American Cryptographer)
- Designer of "Lucifer" cipher \( \Rightarrow \) DES.
- Feistel was one of Hellman's influences, ran the IBM Yorktown Heights crypto group... very influential (hence Craig Gentry)
- Motivated by ATM's!
- It's classic! DES was the standard for ~30 yrs. Still not bad!

**Feistel Network.**

Let \( f : X \rightarrow X \) be a function.

We construct a permutation \( \Pi : X^2 \rightarrow X^2 \) as

\[
\begin{align*}
\Pi_1(x,y) &:= (y, x \oplus f(y)) \\
\Pi_2(u,v) &:= (v \oplus f(u), u)
\end{align*}
\]

Amazingly simple!

Use many rounds of \( \Pi \) with independently keyed \( f's \).

In a classic paper, Michael Luby and Charles Rackoff show that 3-round Feistel network instantiated w/ 3 indep. keyed \( PRF's \Rightarrow PRP. \)
Thm (Luby-Rackoff)

Let $F: \mathcal{K} \times X \to X$ be a PRP with $|X| = 2^n$. Then the 3-round Feistel network $E$ is a PRP. That is, for any ppp $A$ that attacks $E$, there exists a ppp $B$ attacking $F$ s.t.,

$$\text{PRFAdv}[A, E] \leq 3 \cdot \text{PRFAdv}[B, F] + \frac{q^2}{2^n} + \frac{q^3}{2 \cdot 2^{2n}}.$$ 

Proof Idea: Show that $E$ is a PRP.

Game 0: Real attack game.

Game 1: Replace PRFs $F$ with real random functions.

Game 2: Adv interacts with a real random permutation.

Only tricky step is 1 → 2. Show that an input $(u_i, v_i)$

$$u_i \leftarrow u, \oplus S(v_i)$$

$$x_i \leftarrow v_i \oplus S_2(w_i)$$

$$y_i \leftarrow w_i \oplus S_3(x_i)$$

output $(x_i, y_i)$

This is just a statement about probability/distributions.

Idea: Say no two $w_i$'s are the same after $q$ queries. Then all $S_2$ outputs are indep $\&$ random. Then all $S_3$ inputs will likely be distinct $\Rightarrow$ Everything looks random.

Note: After $q = \sqrt[3]{2^n}$ queries, security (and proof) breaks down.