Permutation (Review from CS255)

⇒ Questions + note cards

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

$$f : \{0,1\}^n \longrightarrow \{0,1\}^n.$$

In crypto terms, f takes n-bit strs to n-bit strings.

A "block ciper" is defined over sets $(\mathcal{X}, \mathcal{K})$ and Consists of algs (E, D).

$\mathcal{X}$ ← message space, $\mathcal{K}$ ← key space

$E : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$.

$D :$ " " "

Idea: Both E, D
* are efficient
* implement permutations on $\mathcal{X}$
* are inverses of each other for given $k \in \mathcal{K}$
* satisfies some security notion
  ↳ "pseudo-random permutation" ... defined in 1 sec

Compare with PRF

→ They are hard to construct!
  ↳ Any sufficiently crazy circuit might give you a PRF (plausibly, at least)
  Very hard to build a PRP! 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...

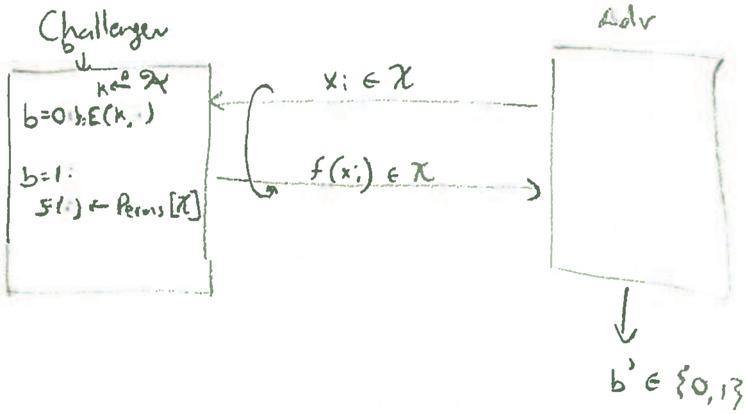
→ One reason to care about PRPs: Format-preserving enc

Credit card # → credit card #
Zip code → zip code
phone # → phone #

Useful! Backwards compat is valuable
↳ c.f. Voltage inc

PseudoRandom Permutation (PRP) ("Block Cipher")

Formalizes one security notion for a block cipher



Intuitively:

$$\left\{ \begin{array}{l} k \xleftarrow{R} \mathcal{K} \\ f(\cdot) = E(k, \cdot) \end{array} \right\} \approx_c \left\{ f(\,) \xleftarrow{R} \text{Perms}[\mathcal{X}] \right\}$$

[Note: we didn't say anything about $D(\,)$.
Can define a stronger notion of security "strong PRP" allowing $D(\,)$ queries]

What $\mathcal{K} = \{0,1\}^\lambda (= n)$, $\mathcal{X} = \{0,1\}^n$, consider a family $\{E_n\}_{n=1}^{\infty}$

Then E_n must run in time poly(n).

We say E is a PRP if, for all p.p.t. algorithms A, A (running in time poly(n)), there exists a negl function $\text{negl}_A(n)$ s.t.

$$\text{PRPAdv}[A, E] := \left| \Pr\left[\begin{smallmatrix} A \text{ outputs } 1 \\ \text{in exp } 0 \end{smallmatrix}\right] - \Pr\left[\begin{smallmatrix} A \text{ outputs } 1 \\ \text{in exp } 1 \end{smallmatrix}\right] \right| \le \text{negl}_A(n)$$

for all n sufficiently large.

$\Rightarrow$ From CS255: Remember to never use a PRP directly for encryption!

Building a PRF from a PRP ($PRP \Rightarrow PRF$)

(Theorem 4.3 in Boneh Shoup)

A good PRP is also a good PRF.

Intuition: The only difference b/w PRF and PRP is collisions.
↳ until you see a collision you can't distinguish.

PRF Switching Lemma. Let E be a PRP over $(\mathcal{K}, \mathcal{X})$, let $|\mathcal{X}| = 2^n$.
Then, if an ppt adversary makes q queries, then

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

[Note: If A is efficient/ppt, then $q \in poly(n)$, so
$\frac{q^2}{2 \cdot 2^n} \in negl(n) \Rightarrow E$ is a secure PRF.]

[Note: To be completely formal, we need to define a family of PRPs $\{E_n\}_{n=1}^{\infty}$. We elide this notation BUT ASK IF UNCLEAR.]

Proof By a hybrid argument

Game 0: Challenger uses $f := E(k, \cdot)$ for $k \xleftarrow{R} \mathcal{K}$

Game 1: " " $f \xleftarrow{R} \text{Perms}[\mathcal{X}]$

Game 2: " " $f \xleftarrow{R} \text{Funs}[\mathcal{X}, \mathcal{X}]$ ← Random Function

By definition

$$\left| \Pr\left[\begin{smallmatrix} A \text{ outputs } 1 \\ \text{in Game } 0 \end{smallmatrix}\right] - \Pr\left[\begin{smallmatrix} A \text{ outputs } 1 \\ \text{in game } 1 \end{smallmatrix}\right] \right| = \text{PRPAdv}[A, E] \leq negl(n)$$

$$\left| \Pr\left[\begin{smallmatrix} A \text{ outputs } 1 \\ \text{in Game } 1 \end{smallmatrix}\right] - \Pr\left[\begin{smallmatrix} A \text{ outputs } 1 \\ \text{in game } 1 \end{smallmatrix}\right] \right| \leq \frac{q^2}{2 \cdot 2^n} \Bigg\} \text{Want to show}$$

Pf (cont'd)

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 a PRP from a PRF (almost by def'n).

So $|P_1 - P_2| = \Pr[\text{A sees a collision in query responses}]$.

$$\Pr[\text{collision}] \leq \Pr[\exists\, x_i, x_j \text{ s.t. } f(x_i) = f(x_j) \text{ in game 1}]$$

By union bound

$$\leq \binom{\#\text{ of } (i,j)}{\text{pairs}} \Pr[f(x_i) = f(x_j)]$$

$$\leq \binom{\#\text{ of}(i,j)}{\text{pairs}} \sum_{x^* \in \mathcal{X}} \Pr[f(x_i) = x^* \wedge f(x_j) = x^*]$$

$$\leq \binom{q}{2} |2^n| \left(\frac{1}{2^n}\right)^2$$

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

$$\leq \frac{q^2}{2 \cdot 2^n}$$

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

PrPAdv $= |P_1 - P_0|$, PRFAdv $= |P_2 - P_0|$... Showing $|P_2 - P_1|$ small is all needed to do.

Ask a question now!

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

Can we get a cipher that is indist from PRF s.t. advantage is $\frac{q}{2^n}$? 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 $\Pr[E_1 \text{ or } E_2 \text{ or} \cdots \text{ or } E_n] \leq \sum_i \Pr[E_i]$.

$\Rightarrow$ One of the most useful tools in security analysis.

$$\Pr[\mathrm{Bad}_1 \text{ or } \mathrm{Bad}_2 \text{ or} \cdots \text{ or } \mathrm{Bad}_n] \leq \sum_i \Pr[\mathrm{Bad}_i].$$

The More 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 ⟹ DES.
- Feistel was one of Hellman's influences, ran the IBM Yorktown Heights crypto group... very influential (home to Craig Gentry)
- Motivated by ATMs!
- It's classic! DES was the standard for ~ 30 yrs. Still not bad**

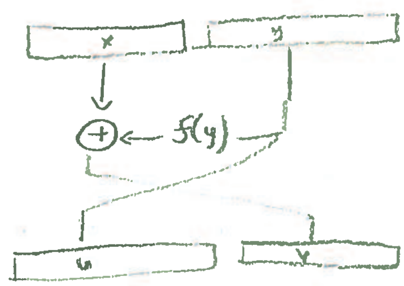
Feistel Network.

Let $f: \mathcal{X} \to \mathcal{X}$ be a function

We construct a permutation $\pi: \mathcal{X}^2 \to \mathcal{X}^2$ as

$$\pi(x,y) := (y,\ x \oplus f(y))$$

$$\pi^{-1}(u,v) := (v \oplus f(u),\ u)$$



One round of Feistel network.

Amazingly simple!

Use many rounds of π 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 PRFs ⟹ PRP.

Thm (Luby - Rackoff)

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

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

Pf Idea: Show that E is a PRP.

Game 0: Real attack game.

Game 1: Replace PRFs F with real random functions

$|p_0 - p_1| \leq 3\, \text{PRFAdv}[B, F]$

Game 2: Adv interacts with a real random permutation.

Only tricky step is $1 \to 2$. Show that on input (u_i, v_i)

Π^3:

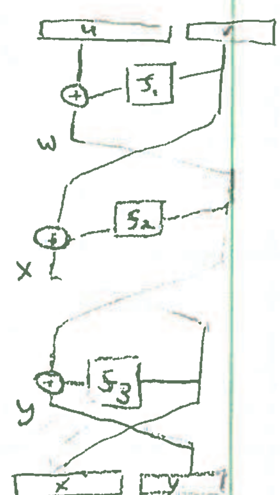
$w_i \leftarrow u_i \oplus f_1(v_i)$

$x_i \leftarrow v_i \oplus f_2(w_i)$

$y_i \leftarrow w_i \oplus f_3(x_i)$

output (x_i, y_i)

Really random fs



This is just a statement about probability / distributions.

Idea: Say no two w_i's are the same after q queries. Then all f_2 outputs are indep & random. Then all f_3 inputs will likely be distinct $\Rightarrow$ Everything looks random.

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