Pseudorandom Permutations

HW due Wednesday

But First: Recap PRFs:

Def: $F : \{0,1\}^\lambda \times \{0,1\}^{in} \to \{0,1\}^{out}$ is secure PRF if:

$k \leftarrow \{0,1\}^\lambda$ QUERY($x \in \{0,1\}^{in}$): return $F(k,x)$	$\approx$	cache = empty dict. QUERY(x): if cache[x] undef: cache[x] $\leftarrow \{0,1\}^{out}$ return cache[x]
Adv can query PRF		Adv can query ideal, random function

Idea: when k chosen uniformly, $F(k,\cdot) : \{0,1\}^{in} \to \{0,1\}^{out}$ should "look like" randomly chosen function

Attacks:

let F be a secure PRF

define $F'(k,x) = F(k,x) \oplus F(k,\overline{x})$ (flip every bit)

Claim: F' is NOT a secure PRF, even if F is

Hint: Don't try to break F (ie, distinguish outputs of F from random)

✡ outputs of F on distinct inputs look random
→ try to get same input into F twice (by querying F')

Adv:

pick any x
$z_1 = \text{QUERY}(x)$
$z_2 = \text{QUERY}(\bar{x})$
return $z_1 \stackrel{?}{=} z_2$

In "real" PRF world

$z_1 = F(k,x) \oplus F(k,\bar{x})$
$z_2 = F(k,\bar{x}) \oplus F(k,x)$
$z_1 = z_2$ always

$\Pr[\text{output true}] = 1$

In "ideal" rand. func world

$z_1 \leftarrow \{0,1\}^{out}$
$z_2 \leftarrow \{0,1\}^{out}$

$\Pr[\text{output true}] = \frac{1}{2^{out}}$

$\Rightarrow$ Advantage is $\left| 1 - \frac{1}{2^{out}} \right|$: not negligible

$\geq \frac{1}{2}$

Pseudorandom Permutations (PRPs)

("block cipher")
e.g. AES, DES, ...

basically a PRF, but

$in = out = blen$ (block length)

and there exists F^{-1}:

for each k : $F^{-1}(k, F(k,x)) = x$

$F(k,*)$, $F^{-1}(k,*)$ are inverses

"you can invert if you know the key k"

Security: Changes to make to PRF definition (above):

```
k ← {0,1}^λ
QUERY(x ∈ {0,1}^in):
    return F(k,x)
```

≈≈≈

```
cache = empty dict.

QUERY(x):
    if cache[x] undef:
        cache[x] ← {0,1}^out \ range(cache)
    return cache[x]
```

Adv can query ~~PRF~~ PRP

Adv can query ideal, random ~~function~~ permutation

Idea: when k chosen uniformly, $F(k,\cdot): \{0,1\}^{in} \to \{0,1\}^{out}$ should "look like" randomly chosen ~~function~~ permutation

PRP switching lemma: If $b_{len} \geq \lambda$, then ideal permutation is indistinguishable from ideal rand. func

(only diff. is sampling with/without replacement)

(constructions that require PRF can use PRP)

Challenge:

a PRF $F(k,x)$ somehow "scrambles" x

a PRP $F(k,x)$ "scrambles" x but in a way that x is still recoverable via $F^{-1}(k,*)$

Feistel Cipher (way to convert PRF ⇝ PRP)

(e.g. DES uses this idea)

Simplest idea: Suppose $F : \{0,1\}^n \to \{0,1\}^n$

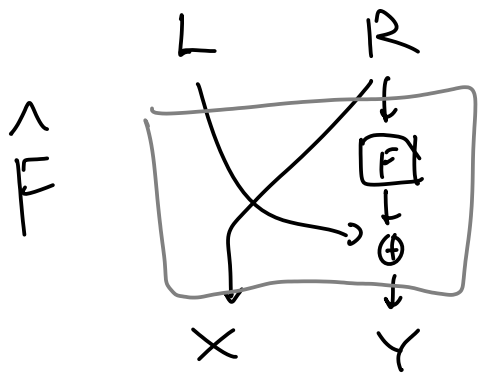
F may not have inverse

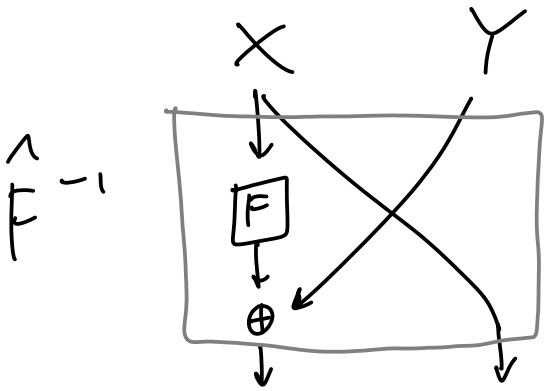
Def $\hat{F} : \{0,1\}^{2n} \to \{0,1\}^{2n}$

$$\hat{F}(L,R) = (R,\ F(R) \oplus L)$$
$$= (X,\ Y)$$

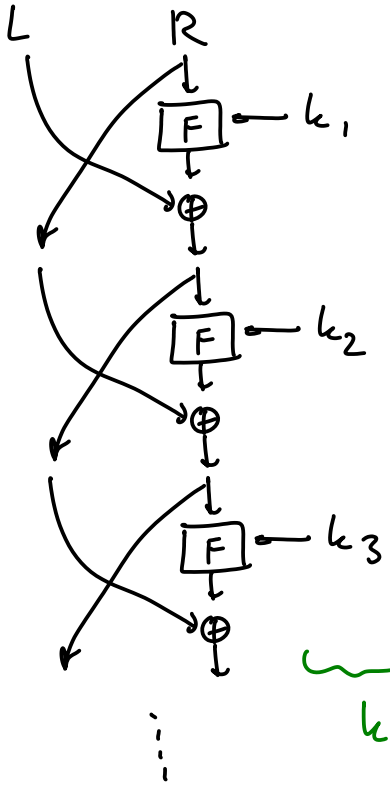
$$\hat{F}^{-1}(X,Y) = (F(X) \oplus Y,\ X)$$
$$= (F(R) \oplus (F(R) \oplus L),\ R)$$
$$= (L,\ R)$$

$\hat{F}$ is invertible





Feistel Cipher: with F a PRF



F is the round function with round key k_1

$k_1 \ldots k_i \ldots$ is the key schedule

Claim: 1-2 round Feistel cipher $\Rightarrow$ NOT a PRP

but 3-round Feistel cipher (with PRF round func) $\Rightarrow$ result is a PRP