# Pseudorandom Permutations (aka block ciphers)

## Conceptual: PRF vs PRP

PRF: like having a huge, randomly initialized key → value store (associative array)

PRP: same, but no value chosen twice

(range(T) = set of "values")

"Example"

cryptogram

A → F
B → Q
C → R
⋮

permutation of {A, ..., Z}

better cryptogram

AA → QR
AB → FW
AC → AM

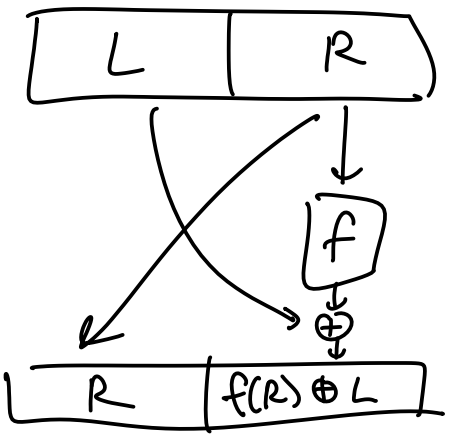
permutation over {AA, ..., ZZ}

even better permutation

$\{0,1\}^{128} \rightarrow \{0,1\}^{128}$

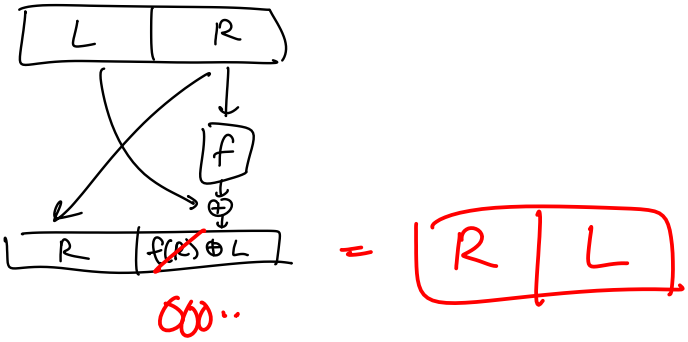
---

## Feistel ciphers:

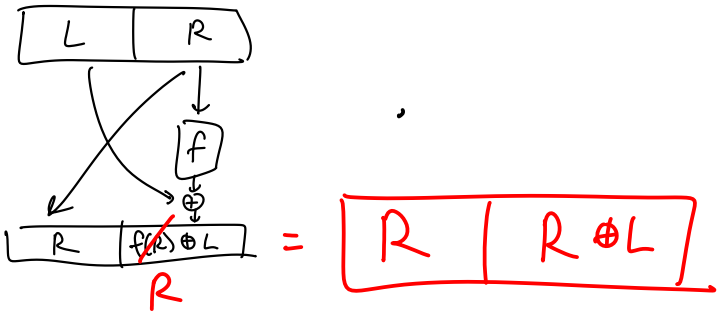


What happens when round function is "trivial"?

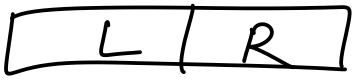
- $f(X) = 000\ldots0$




- $f(X) = X$




Example feistel




16 + 16 bytes

## Attack 1/2-round Feistel:

1 round: $(L, R) \longmapsto (R, f(R) \oplus L)$

want to distinguish

```
QUERY(LR):
    return (R, f(k,R) ⊕ L)
```

```
QUERY(LR):
    random stuff
```

Distinguisher:

```
(A,B) = QUERY(0^n 0^n)
return A ?= 0^n
```

In presence of left lib, $Pr[\text{output } 1] = 1$

In presence of right lib, $Pr[\text{output } 1] = 1/2^n$

## 2-round Feistel:

$(L, R) \longrightarrow (R, f(R) \oplus L)$

$\longrightarrow (f(R) \oplus L, f(f(R) \oplus L) \oplus R)$

want to distinguish

```
QUERY(LR):
    return (f(R) ⊕ L,
            f(f(R) ⊕ L) ⊕ R)
```

```
QUERY(LR):
    random stuff
```

Obs: $f(R) \oplus L$ looks weak with respect to $L$

Idea: Call twice

$$(L, R) \longrightarrow (f(R) \oplus L,\ f(f(R) \oplus L) \oplus R)$$
$$(L', R') \longrightarrow (f(R') \oplus L',\ f(f(R') \oplus L') \oplus R')$$

If $R = R'$: same $f(R)$ term $\Rightarrow$ XOR these 2 things to get $L \oplus L'$

Calling prog:

pick arbitrary $L, R, L'$

$(A, B) = \text{QUERY}(L, R)$

$(C, D) = \text{QUERY}(L', R)$

return $A \oplus C \stackrel{?}{=} L \oplus L'$

In left lib:
always outputs 1

In right lib:
at time of 2nd query
$L, L', A$ fixed,
$C$ to be chosen randomly