# Cryptography

Lecture 15

# Announcements

- HW6 due Monday 4/10

# Agenda

- Last time
  - Practical Constructions of Block-Ciphers: SPN (K/L 6.2)

- This time
  - Feistel Transform (K/L 6.2)
  - Details of AES/DES (K/L 6.2)
  - Practical constructions of CRHF (K/L 6.3)

# Feistel Networks
# An alternative approach to Block Cipher Design

# Feistel Networks

- The underlying round functions do not need to be invertible.
- Feistel network allows us to construct an invertible function from non-invertible components.
- With enough rounds, can construct a PRP from a PRF.

# (Balanced) Feistel Network

- The $i$th round function $\hat{f}_i$ takes as input a sub-key $k_i$ and an $\ell/2$-bit string and outputs an $\ell/2$-bit string.
- Master key $k$ is used to derive sub-keys for each round.
- Note that the round functions $\hat{f}_i$ are fixed and publicly known, but the $f_i(R) \coloneqq \hat{f}_i(k_i, R)$ depend on the master key and are not known to the attacker.

# $i$-th Feistel Round

- If the block length of the cipher is $\ell$ bits, then $L_{i-1}$ and $R_{i-1}$ each has length $\ell/2$.
- The output $(L_i, R_i)$ of the round is:

$L_i \coloneqq R_{i-1}$ and $R_i \ \coloneqq L_{i-1} \oplus f_i(R_{i-1})$

# A three-round Feistel Network

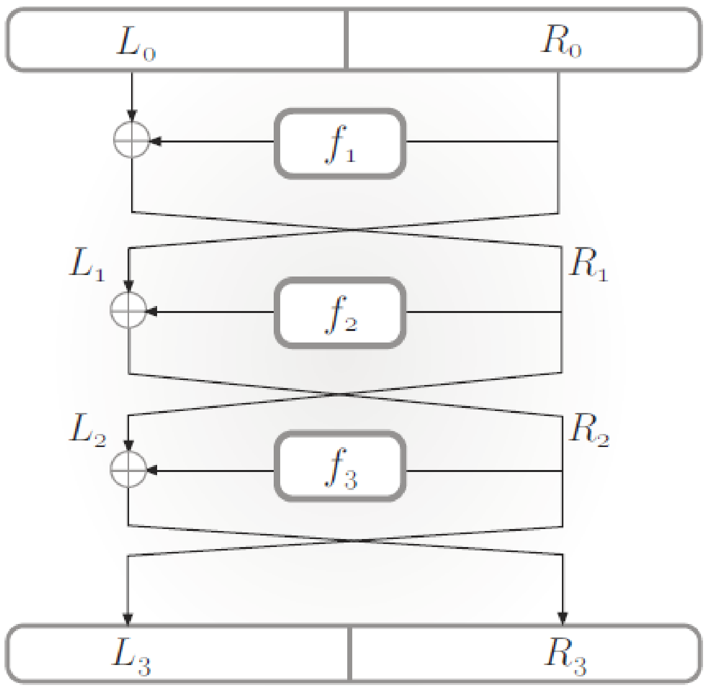
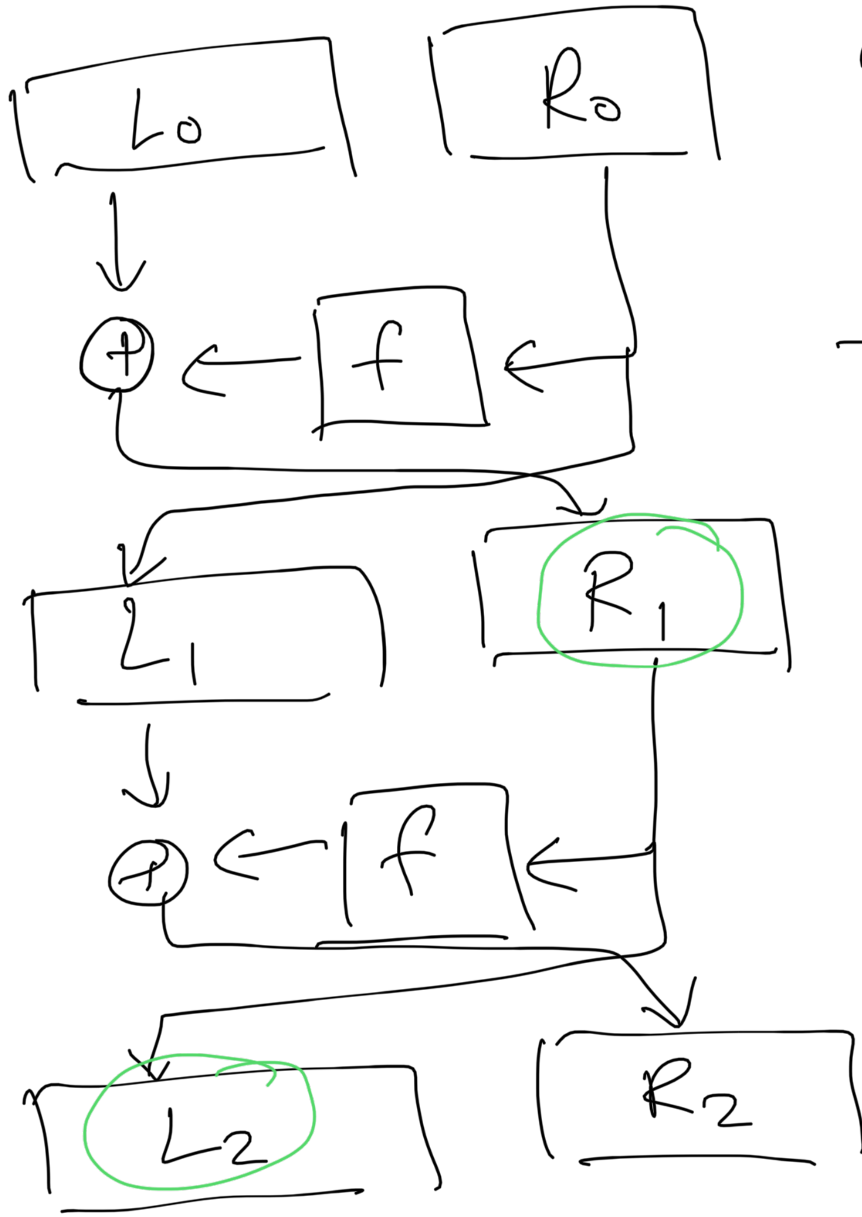



**FIGURE 6.4**: A 3-round Feistel network.

# Feistel Networks are invertible

Proposition: Let $F$ be a keyed function defined by a Feistel network. Then regardless of the round functions $\{\hat{f}_i\}$ and the number of rounds, $F_k$ is an efficiently invertible permutation for all $k$.




Show an attack distinguishing from a truly random permutation

A: query $L_0 \| R_0$

$R_1 = f(R_0) \oplus L_0$

query $L_0' \| R_0$

$R_1' = f(R_0) \oplus L_0'$

Check:

$R_1 \oplus R_1' \stackrel{?}{=} L_0 \oplus L_0'$

if yes, output 1 o/w output 0.

# Details on DES

- The Data Encryption Standard was developed in the 1970s by IBM (with help from the National Security Agency), and adopted in 1977 as a Federal Information Processing Standard for the US.
- DES is no longer considered secure due to its short key length of 56 bits which makes it vulnerable to brute-force attacks.
- It remains in wide use today in the strengthened form of triple-DES, described in Section 6.2.4.
- DES is of great historical significance. It has undergone intensive scrutiny within the cryptographic community, arguably more than any other cryptographic algorithm in history. The common consensus is that, relative to its key length, DES is an extremely well designed cipher.
  - To date, the best known attack on DES in practice is an exhaustive search over all $2^{56}$ possible keys.

# Details on DES

- The DES block cipher is a 16-round Feistel network with a block length of 64 bits and a key length of 56 bits. The same round function $\hat{f}$ is used in each of the 16 rounds.
- Round function takes a 48-bit sub-key and, as in a (balanced) Feistel network, a 32-bit input
- The key schedule of DES is used to derive a sequence of 48-bit sub-keys $k_1, \ldots, k_{16}$ from the 56-bit master key.

# Details on DES

- The DES round function $\hat{f}$—the DES mangler function—is constructed using a 1-round substitution-permutation network
- S-boxes are not permutations!!
  - Map 6-bit inputs to 4-bit outputs.

# Details on DES

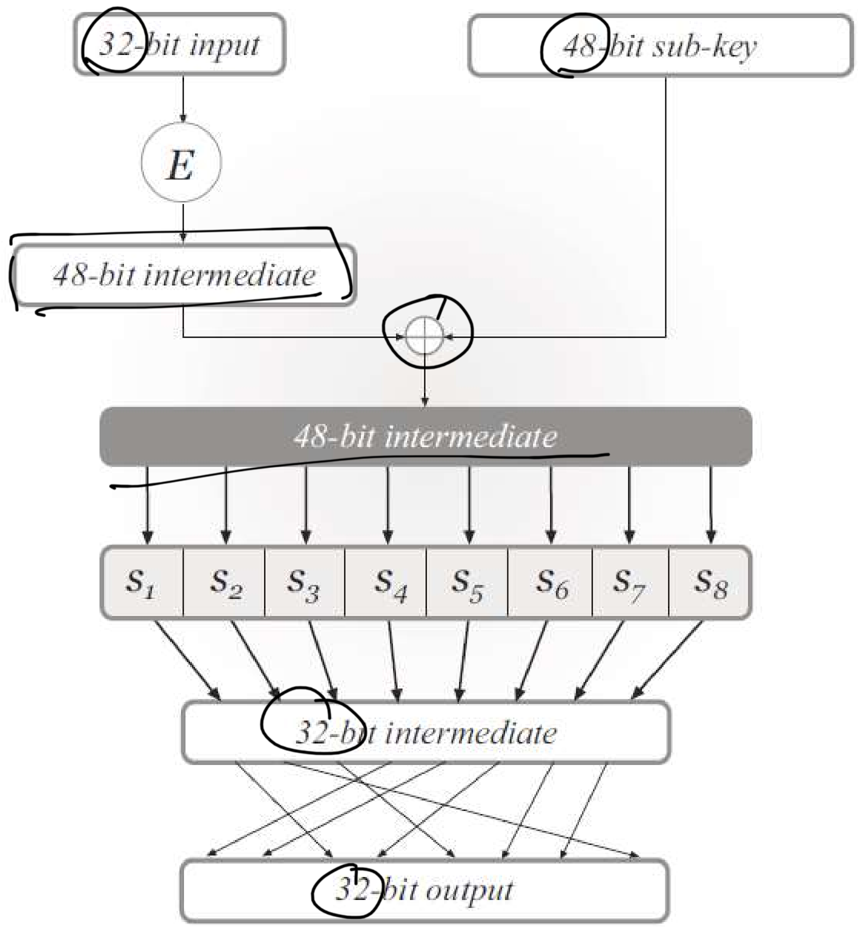
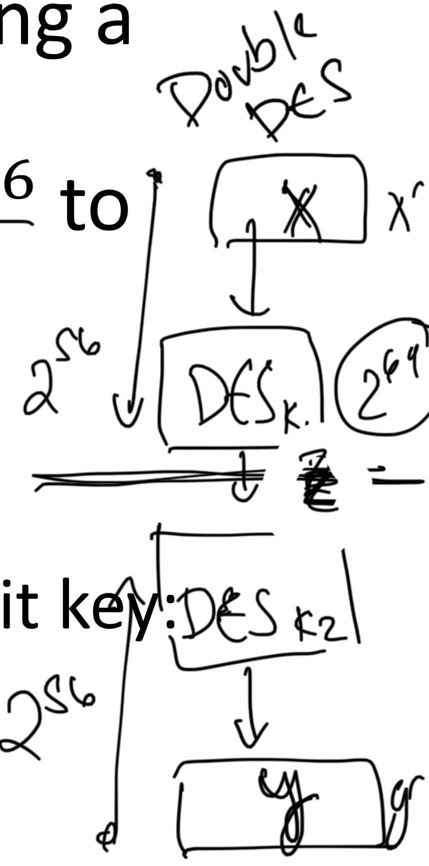



FIGURE 6.5: The DES mangler function.

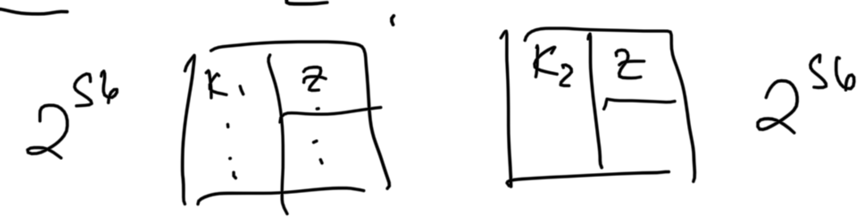
similar to 1-round SPN

# 3DES (Triple Encryption)

- First Idea: increase the key length by doing a double-encryption, thereby increasing complexity of brute-force attack from $2^{56}$ to $2^{112}$.
- Let $F$ be a block-cipher with an $n$-bit key length and $\ell$-bit block length.
  - Define the following block cipher with $2n$-bit key: $F'_{k_1,k_2}(x) := F_{k_2}(F_{k_1}(x))$
- Problem: Meet in the middle attack




time just slightly more than $2^{56}$



$56 \cdot 2^{56}$ $56 \cdot 2^{56}$

$O(n \cdot 2^n)$

# Meet in the Middle Attack on Double DES

$2^{112}/2^{64} = 2^{48}$

| $k_1$ | $k_2$ | $z$ |
|---|---|---|
| | | |

Adversary is given a single input/output pair $(x, y)$ where $y = F_{{k^*}_1,{k^*}_2}(x)$ for unknown $k_1, k_2$. The adversary does the following:

- For each $k_1 \in \{0,1\}^n$, compute $z := F_{k_1}(x)$ and store $(z, k_1)$ in a list $L$.
- For each $k_2 \in \{0,1\}^n$, compute $z := {F_{k_2}}^{-1}(y)$ and store $(z, k_2)$ in a list $L'$.
- Sort $L$ and $L'$, respectively, by their first components.
- Entries $(z_1, k_1) \in L$ and $(z_2, k_2) \in L'$ are a match if $z_1 = z_2$. For each match of this sort, add $(k_1, k_2)$ to a set $S$.

$2^{2n-\ell} < 2^n$

Expected number of elements in $S$ is $2^{2n-\ell}$. Can use a few more input/output pairs to reduce to a single $(k_1, k_2)$ .

$\ell \geq n$

64 128

$O(n2^n)$ $\Omega(2^{2n})$

# Triple DES

Two variants:

- $F'_{\underbrace{k_1,k_2,k_3}}(x) \coloneqq F_{k_3}({F_{k_2}}^{-1}(F_{k_1}(x)))$
- $F'_{\underbrace{k_1,k_2}_{112}}(x) \coloneqq F_{k_1}({F_{k_2}}^{-1}(F_{k_1}(x)))$
- Middle cipher is reversed for backwards compatibility: setting $k_1 = k_2 = k_3$ results in a single invocation of $F$ using key $k_1$.

# Security of Triple-DES

- Security of the first variant: The cipher is susceptible to a meet-in-the-middle attack just as in the case of double encryption, though the attack now takes time $2^{2n}$. This is the best known attack.
- Security of the second variant. There is no known attack with time complexity better than $2^{2n}$ when the adversary is given only a small number of input/output pairs. Thus, two-key triple encryption is a reasonable choice in practice.

Disadvantage of both Triple-DES variants: Fairly slow since it requires 3 invocations of DES.

# Details on AES

- In January 1997, the United States National Institute of Standards and Technology (NIST) announced a competition to select a new block cipher—to be called the Advanced Encryption Standard, or AES
- 15 submissions from all over the world. Each team's candidate cipher was intensively analyzed by members of NIST, the public, and (especially) the other teams. Two workshops were held ('98, '99) to analyze the various submissions. Following the second workshop, NIST narrowed the field down to 5 "finalists" and the second round of the competition began. A third AES workshop was held in April 2000, inviting additional scrutiny on the five finalists.
- In October 2000, NIST announced that the winning algorithm was Rijndael (a block cipher designed by Belgian cryptographers Vincent Rijmen and Joan Daemen)

# Details on AES

A 4-by-4 array of bytes called the **state** is modified in a series of rounds. The state is initialized to the input to the cipher (128 bits = 16 bytes). The following operations are then applied in each round:

1. Stage 1 – AddRoundKey: A 128-bit sub-key is derived from the master key, and is interpreted as a 4-by-4 array of bytes. **state** updated by XORing it with this sub-key.
2. Stage 2 – SubBytes: Each byte of **state** is replaced by another byte according to a single fixed lookup table S. This substitution table (or S-box) is a bijection over $\{0,1\}^8$.
3. Stage 3 – ShiftRows: The bytes in each row of **state** are cyclically shifted to the left as follows: the first row of the array is untouched, the second row is shifted one place to the left, the third row is shifted two places to the left, and the fourth row is shifted three places to the left. All shifts are cyclic so that, e.g., in the second row the first byte becomes the fourth byte.
4. Stage 4 – MixColumns: An invertible transformation is applied to the four bytes in each column. (linear transformation—i.e., matrix multiplication—over an appropriate field.)

   If two inputs differ in $b > 0$ bytes, then transformation yields two outputs differing in at least $5 - b$ bytes.

   In the final round, MixColumns is replaced with AddRoundKey. Why?

- To date, no practical cryptanalytic attacks significantly better than a exhaustive search.