Cryptography

Lecture 15

Announcements

- HW4 due Wednesday 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

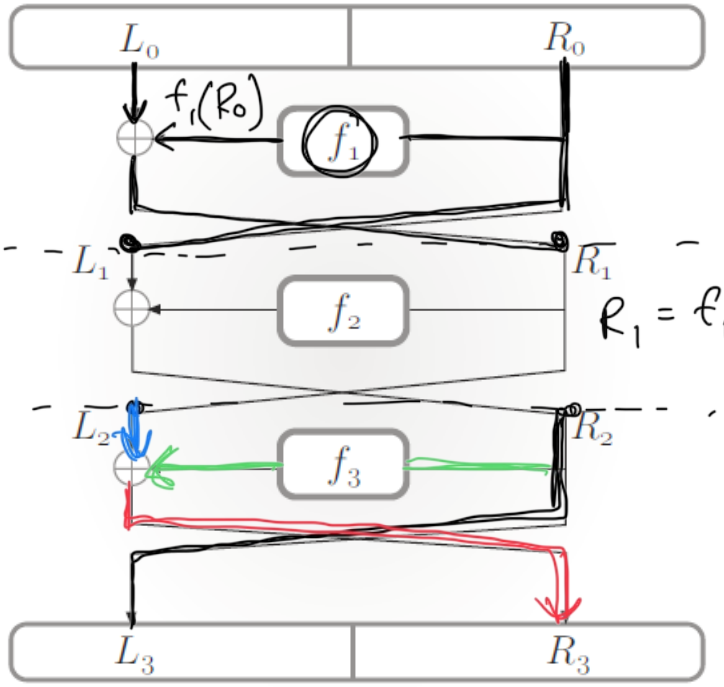
- The ith round function $\hat{f}_i$ takes as input a sub-key k_i and an $\ell/2$-bit string and outputs an $\ell/2$-bit string.
- Master key k is used to derive sub-keys for each round.
- Note that the round functions $\hat{f}_i$ are fixed and publicly known, but the $f_i(R) \coloneqq \hat{f}_i(k_i,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

i-th Feistel Round

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

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

A three-round Feistel Network



$L_1 = R_0$

$R_1 = f_1(R_0) \oplus L_0$

$R_2 = L_3$

$L_2 =$

$f_3(L_3) \oplus R_3$

FIGURE 6.4: A 3-round Feistel network.

"balanced"

$f_1(R_0) = \hat{f}_1(K_1, R_0)$

↑ round key.

Feistel Networks are invertible

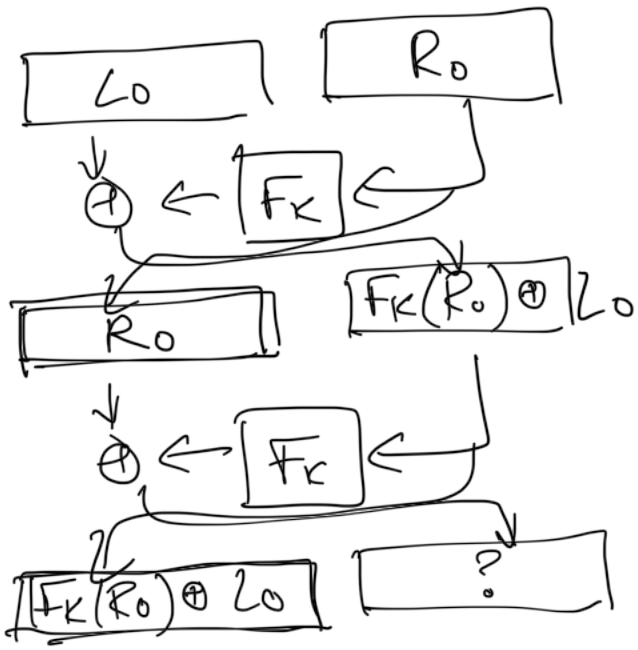
Proposition: Let F be a keyed function defined by a Feistel network. Then regardless of the round functions $\{\hat{f}_i\}$ and the number of rounds, F_k is an efficiently invertible permutation for all k.

Luby-Rackoff: Feistel Networks

- Start out w/ round functions that are PRFs
- Run Feistel w/ ↑ for 3 rounds ⇒ PRP
- Run Feistel w/ ↑ for 4 rounds ⇒ strong PRP

Challenge: Prove that 3-round Feistel is not a strong PRP

Today: Prove that 2-round Feistel is not a PRP.



Distinguishing Attack:

Pick whatever for R_0

Pick $L_0 = 0^{\ell/2}$

Output: $F_K(R_0) \| ?$

Pick same R_0

Pick $L_0 = 1^{\ell/2}$

Output: $F_K(R_0) \oplus 1^{\ell/2} \| \tilde{?}$

XOR the LHS of both outputs and check whether it is equal to $1^{\ell/2}$. 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

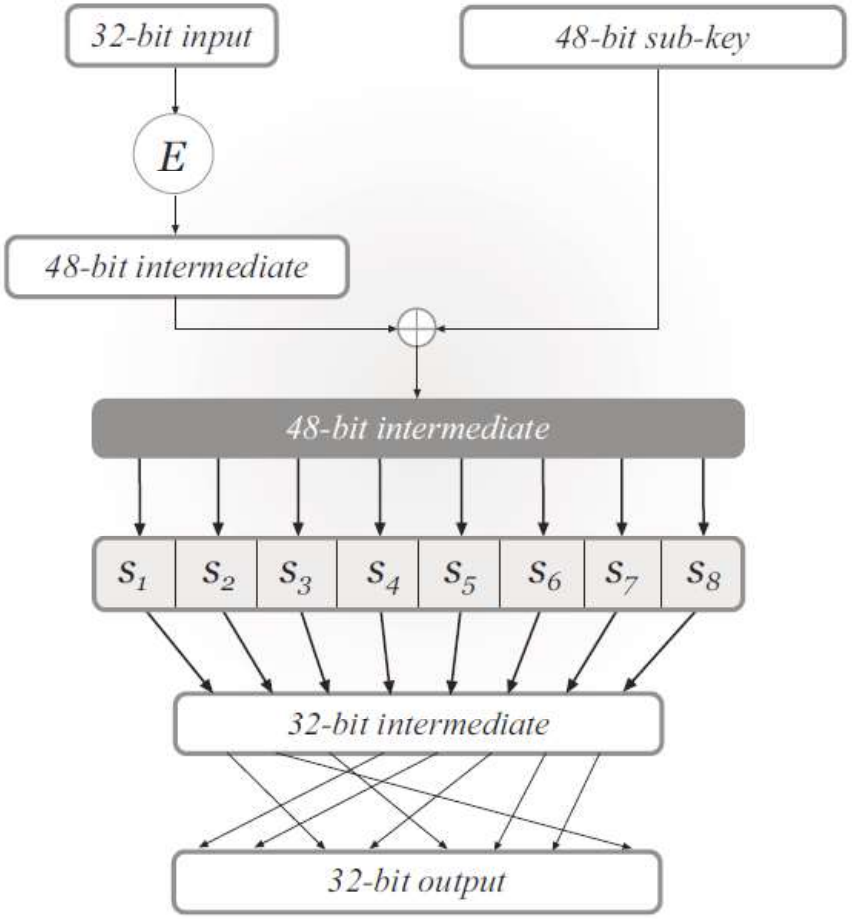


FIGURE 6.5: The DES mangler function.

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 ℓ-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

Meet in the Middle Attack on Double DES

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.

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) .

Triple DES

Two variants:

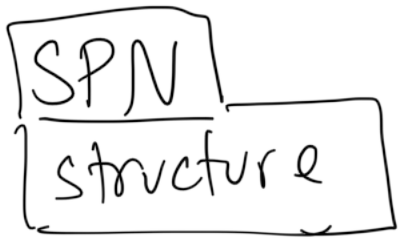
- $F'_{k_1,k_2,k_3}(x) \coloneqq F_{k_3}(F_{k_2}^{-1}(F_{k_1}(x)))$
- $F'_{k_1,k_2}(x) \coloneqq F_{k_1}(F_{k_2}^{-1}(F_{k_1}(x)))$

- 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)

128 192 256

rounds 10 12 14

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?

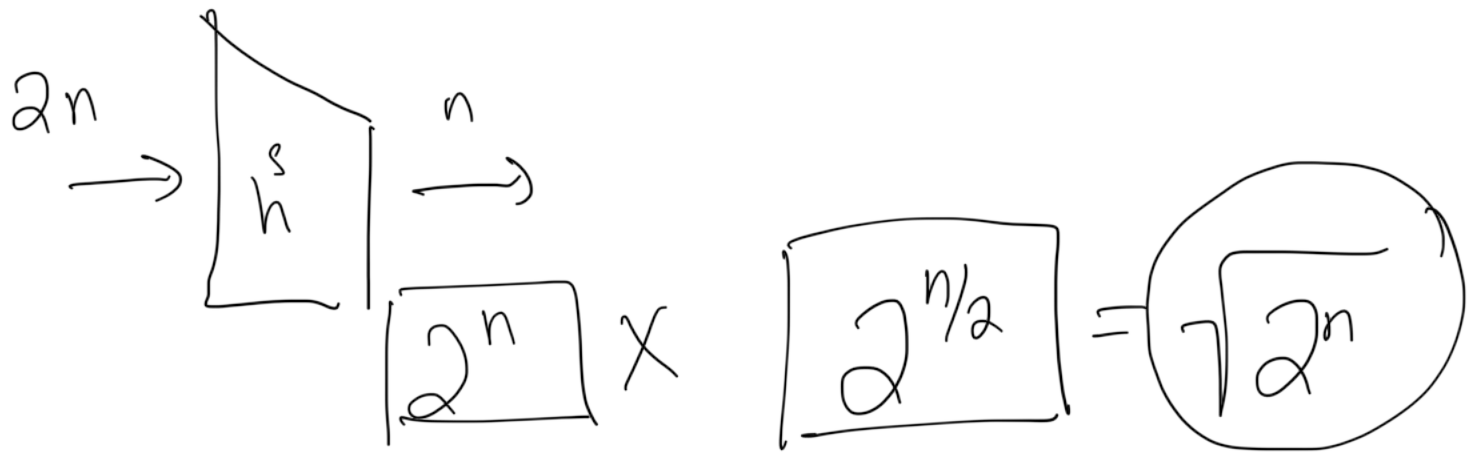
- To date, no practical cryptanalytic attacks significantly better than a exhaustive search.

Agenda

- Practical constructions of Collision-Resistant Hash Functions (K/L 6.3)
- New Unit: Number Theory!

Preliminaries

- How much security can we hope for from a CRHF that outputs ℓ bits?
- Discuss the "**birthday bound**"
  - No matter what function is used, collisions can be found with high probability after making $2^{\ell/2}$ queries.



Show the expected number of collisions after q queries to $\boxed{h^s}$ behaves randomly $\approx \frac{q^2}{2^n}$

Weaker Notions of Security

- Second preimage or target collision resistance: Given s and a uniform x it is infeasible for a ppt adversary to find $x' \neq x$ such that $H^s(x') = H^s(x)$.
- Preimage resistance: Given s and uniform y it is infeasible for a ppt adversary to find a value x such that $H^s(x) = y$.

Hash Functions From Block Ciphers

- Hash functions are generally constructed in two steps:
  - First, a compression function (fixed-length hash function) h is designed
  - Next, some mechanism (e.g. Merkle-Damgard) is used to extend h so as to handle arbitrary input lengths
- We will focus on the first step

Hash Functions From Block Ciphers

- Davies-Meyer construction:
  - F is a block-cipher will n-bit key and ℓ-bit block length.

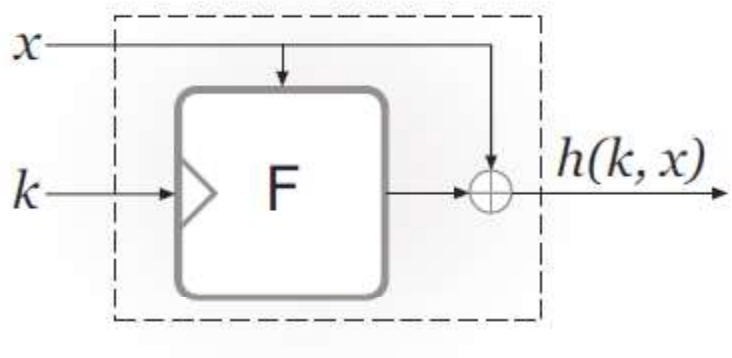
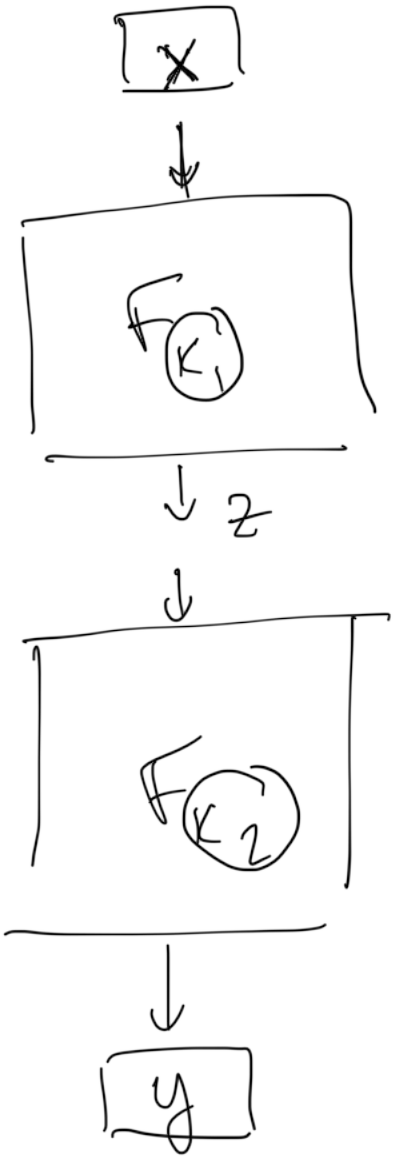


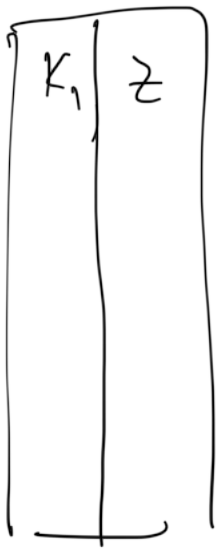
FIGURE 6.10: The Davies-Meyer construction.

  - Above forms a compression function from $n + \ell$ bits to n bits.

Idea of Meet-in-the-Middle Attack



2^{56} (K_1, z)



2^{56}

2^{56} (K_2, z)

2^{56}

time so far

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

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

Security Analysis

- We do not know how to prove collision-resistance of the compression function based on the assumption that F is a strong PRP.
- Requires stronger assumption that F behaves like an **ideal cipher**.
  - Like a truly random permutation, except can query oracle on **different keys**.
  - Each key $k \in \{0,1\}^n$ specifies an independent, uniform permutation $F(k,\cdot)$ on ℓ-bit strings.

Security Analysis

- Theorem: If F is modeled as an ideal cipher, then the Davies-Meyer construction yields a collision-resistant compression function. Concretely, any attacker making $q < 2^{\ell/2}$ queries to its ideal-cipher oracles finds a collision with probability at most $q^2/2^{\ell}$.

MD5

- 128-bit output length.
- Designed in 1991, and for several years was believed to be collision-resistant. Over a period of several years, various weaknesses began to be found in MD5 but
- these did not appear to lead to any easy way to find collisions.
- In 2004 a team of Chinese cryptanalysts presented a new method for finding
- collisions in MD5 and were able to demonstrate an explicit collision!
- Since then, the attack has been improved—collisions can be found in under a minute on a desktop PC—and extended so that even "controlled collisions" (e.g., two postscript files generating arbitrary viewable content) can be found.
- Due to these attacks, MD5 should no longer be used today for any application requiring cryptographic security.

SHA-0, SHA-1, SHA-2

- The Secure Hash Algorithm (SHA) refers to a series of cryptographic hash functions standardized by NIST.
- SHA-1, was introduced in 1995. This algorithm has a 160-bit output length, and supplanted a predecessor called SHA-0 which was withdrawn due to unspecified flaws discovered in that algorithm.
- Theoretical analysis over the past few years indicates that collisions in SHA-1 can be found using significantly fewer than the 280 hash function evaluations that would be necessary using a birthday attack.
- Recently an explicit collision has been found.
- It is therefore recommended to migrate to SHA-2, which does not currently appear to have the same weaknesses.
- SHA-2 comprises two related functions: SHA-256 and SHA-512, with 256- and 512-bit output lengths, respectively.

SHA-0, SHA-1, SHA-2

- All hash functions in the SHA family are constructed using the same basic design:
  - A compression function is first defined using the Davies-Meyer construction as applied to some block cipher
  - Extended to support arbitrary length inputs using the Merkle-Damg°ard transform.
- The block cipher in each case was designed specifically for building the compression function.
  - Block ciphers SHACAL-1 (for SHA-1) and SHACAL-2 (for SHA-2). Have large block lengths (160 and 256 bits respectively) and 512-bit key lengths.

SHA-3 (Keccak)

- NIST announced in late 2007 a public competition to design a new cryptographic hash function to be called SHA-3.
- Submitted algorithms were required to support both 256- and 512-bit output lengths.
- 51 first-round candidates were narrowed down to 14 in December, 2008, and these were further reduced to five finalists in 2010. The remaining candidates were subject to intense scrutiny by the cryptographic community over the next two years.
- In October, 2012, NIST announced the selection of Keccak as the winner of the competition.
- This algorithm is currently undergoing standardization as the next-generation replacement for SHA-2.

SHA-3 (Keccak)

- Keccak is unusual in several respects.
  - One of the reasons Keccak was chosen is because its structure is very different from that of SHA-1 and SHA-2.
- It is based on an **unkeyed** permutation f with a large block length of 1600 bits; this is radically different from, e.g., the Davies-Meyer construction which relies on a keyed permutation.
- Keccak does not use the Merkle-Damgard transform to handle arbitrary input lengths. Instead, it uses a newer approach called the **sponge** construction.
- Keccak—and the sponge construction more generally—can be analyzed in the random-permutation model
  - Here parties have access to an oracle for a random permutation $f : \{0,1\}^{\ell} \rightarrow \{0,1\}^{\ell}$ (and possibly its inverse).
  - This is weaker than the ideal-cipher model.