COS433/Math 473: Cryptography

Mark Zhandry
Princeton University
Fall 2020
Announcements/Reminders

HW2 due Today
  • Submit through Gradescope

PR1 Due October 6
Previously on COS 433...
Confusion/Diffusion Paradigm
Confusion/Diffusion Paradigm

Third Attempt: Repeat multiple times!
Confusion/Diffusion Paradigm

While single round is insecure, we’ve made progress
• Each bit affects 8 output bits

With repetition, hopefully we will make more and more progress
Confusion/Diffusion Paradigm

With 2 rounds,
  • Each bit affects 64 output bits

With 3 rounds, all 128 bits are affected

Repeat a few more times for good measure
Limitations

Describing subs/perms requires many bits

- Key size for $r$ rounds is approximately $2^g \times \lambda \times r$
- Ideally want key size to be $128$ (or $256$)

Idea: instead, fix subs/perms

- But then what’s the key?
Substitution Permutation Networks

Variant of previous construction

• Fixed public permutations for confusion (called a substitution box, or S-box)

• Fixed public permutation for diffusion (called a permutation box, or P-box)

• XOR “round key” at beginning of each round
Substitution Permutation Networks
Substitution Permutation Networks

Round

$S_1 \quad S_2 \quad S_3 \quad S_4 \quad S_5 \quad S_6$

$\Pi$

$\oplus$

Round key

Potentially different
Substitution Permutation Networks

Round key

Potentially different

Final key mixing

Round

Round key

Potentially different

Final key mixing

\(\pi\)

\(\oplus\)

\(s_1\)

\(s_2\)

\(s_3\)

\(s_4\)

\(s_5\)

\(s_6\)
Substitution Permutation Networks

To specify a network, must:
• Specify S-boxes
• Specify P-box
• Specify key schedule (how round keys are derived from master)

Choice of parameters can greatly affect security
Designing SPNs

Avalanche Affect:
• Need S-boxes and mixing permutations to cause every input bit to ”affect” every output bit

One way to guarantee this:
• Changing any bit of S-box input causes at least 2 bits of output to change
• Mixing permutations send outputs of S-boxes into at least 2 different S-boxes for next round
• Sufficiently many rounds are used
Designing S-Boxes

Random?
• Let $x, x'$ be two distinct 8-bit values
• $\Pr[S(x) \text{ and } S(x')]$ differ on a single bit] = $\frac{8}{255}$
• Very high probability that some pair of inputs will have outputs that differ on a single bit

Therefore, must carefully design S-boxes rather than choose at random
Linearity?

Can S-Boxes be linear?
• That is, $S(x_0) \oplus S(x_1) = S(x_0 \oplus x_1)$?
AES

State = \textbf{4} \times \textbf{4} \text{ grid of bytes}
AES

One fixed S-box, applied to each byte
• Step 1: multiplicative inverse over finite field $\mathbb{F}_8$
• Step 2: fixed affine transformation
• Implemented as a simple lookup table
AES

Diffusion (not exactly a P-box):
• Step 1: shift rows
• Step 2: mix columns
AES

Shift Rows:
AES

Mix Columns
• Each byte interpreted as element of $\mathbb{F}_8$
• Each column is then a length-4 vector
• Apply fixed linear transformation to each column
AES

Number of rounds depends on key size
• 128-bit keys: 10 rounds
• 192-bit keys: 12 rounds
• 256-bit keys: 14 rounds

Key schedule:
• Won’t describe here, but involves more shifting, S-boxes, etc
• Can think of key schedule as a weak PRG
Feistel Networks
Feistel Networks

Designing permutations with good security properties is hard

What if instead we could build a good permutation from a function with good security properties...
Feistel Network

Convert functions into permutations

Can this possibly give a secure PRP?
Feistel Network

Convert functions into permutations

\[ F \oplus k_0, k_1: \text{round keys} \]

\[ F: \text{round function} \]
Feistel Network

Depending on specifics of round function, different number of rounds may be necessary
• Number of rounds must always be at least 3
• Maybe need even more for weaker round functions
Luby-Rackoff

3-round Feistel where round function is a PRF

**Theorem:** If $F$ is a secure PRF, then 3 rounds of Feistel (with independent round keys) give secure PRP.

- Proof non-trivial, won’t be covered in this class
Limitations of Feistel Networks

Turns out Feistel requires block size to be large
• If number of queries \( \sim 2^{\text{block size}/2} \), can attack

Format preserving encryption:
• Encrypted data has same form as original
• E.g. encrypted SSN is an SSN
• Useful for encrypting legacy databases

Sometimes, want a very small block size
Constructing Round Functions

Ideally, ”random looking” functions

Similar ideas to constructing PRPs
• Confusion/diffusion
• SPNs, S-boxes, etc

Key advantage is that we no longer need the functions to be permutations
• S-boxes can be non-permutations
DES

Block size: 64 bits
Key size: 56 bits
Rounds: 16

!!!
DES

Key Schedule:
• Round keys are just 48-bit subsets of master key

Round function:
• Essentially an SPN network
DES S-Boxes

8 different S-boxes, each
• 6-bit input, 4-bit output
• Table lookup: 2 bits specify row, 4 specify column

• Each row contains every possible 4-bit output
• Changing one bit of input changes at least 2 bits of output
DES History

Designed in the 1970’s
• At IBM, with the help of the NSA
• At the time, many in academia were suspicious of NSA’s involvement
  • Mysterious S-boxes
  • Short key length
• Turns out, S-box probably designed well
  • Resistant to “differential cryptanalysis”
  • Known to IBM and NSA in 1970’s, but kept secret
• Main weakness is the short key length
  • Maybe secure in the 1970’s, definitely not today
DES Security Today

Seems like a good cipher, except for its key length and block size

What’s wrong with a small block size?

• Remember for e.g. CTR mode, IV is one block
• If two identical IV’s seen, attack possible
• After seeing q ciphertext, probability of repeat IV is roughly $q^2/2^{\text{block length}}$
• Attack after seeing $\approx$ billion messages
3DES: Increasing Key Length

3DES key = Apply DES three times with different keys

Why three times?
- Later: “meet in the middle attack” renders 2DES no more secure than 3DES

Why inverted second permutation?
Attacks on block ciphers
Brute Force Attacks

Suppose attacker is given a few input/output pairs

Likely only one key could be consistent with this input/output

Brute force search: try every key in the key space, and check for consistency

Attack time: $2^{\text{key length}}$
Insecurity of 2DES

DES key length: 56 bits
2DES key length: 112 bits
Brute force attack running time: $2^{112}$
Meet In The Middle Attacks

For 2DES, can actually find key in $2^{56}$ time
• Also $\approx 2^{56}$ space
Meet In The Middle Attacks

\[
d = \text{DES}(k_0, m)
\]

\[
d = \text{DES}^{-1}(k_1, m)
\]

<table>
<thead>
<tr>
<th>$k_0$</th>
<th>$d = \text{DES}(k_0, m)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>1</td>
<td>93</td>
</tr>
<tr>
<td>2</td>
<td>03</td>
</tr>
<tr>
<td>3</td>
<td>96</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>49</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$k_1$</th>
<th>$d = \text{DES}^{-1}(k_1, m)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>69</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>86</td>
</tr>
<tr>
<td>3</td>
<td>49</td>
</tr>
<tr>
<td>4</td>
<td>99</td>
</tr>
<tr>
<td>5</td>
<td>08</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Meet In The Middle Attacks

Complexity of meet in the middle attack:
• Computing two tables: time, space $2 \times 2^{\text{key length}}$
• Slight optimization: don’t need to actually store second table

On 2DES, roughly same time complexity as brute force on DES
MITM Attacks on 3DES

MITM attacks also apply to 3DES...

\[
m \xrightarrow{k_0} \text{DES} \xrightarrow{k_1} \text{DES} \xrightarrow{k_2} c
\]
MITM for 3DES

\[ m \rightarrow \text{DES}(k_0,m) \rightarrow \text{DES}(k_1,m) \rightarrow \text{DES}^{-1}(k_2,c) \rightarrow c \]

<table>
<thead>
<tr>
<th>( k_0 )</th>
<th>( k_1 )</th>
<th>( d = \text{DES}(k_0,m) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>93</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>03</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>96</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>49</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( k_2 )</th>
<th>( d = \text{DES}^{-1}(k_2,m) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>69</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>86</td>
</tr>
<tr>
<td>3</td>
<td>49</td>
</tr>
<tr>
<td>4</td>
<td>99</td>
</tr>
<tr>
<td>5</td>
<td>08</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
MITM for 3DES

No matter where “middle” is, need to have two keys on one side
• Must go over $2^{112}$ different keys

Space?

While 3DES has 168 bit keys, effective security is 112 bits
Generalizing MITM

In general, given $r$ rounds of a block cipher with $t$-bit keys,

- Attack time: $2^{t\lceil r/2 \rceil}$
- Attack space: $2^{t\lfloor r/2 \rfloor}$
Announcements/Reminders

HW2 due today
• Submit through Gradescope

PR1 Due October 6