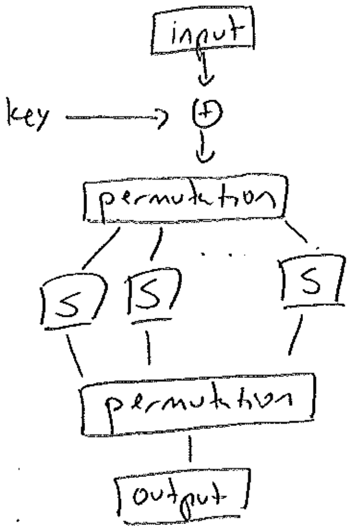
CIS 700 Lecture 5: Attacks on Block Ciphers

"Theoretical" example: Single-round substitution-permutation network



known plaintext: single pair $(x, y = F_k(x))$

x

$\downarrow$

known, deterministic y' $\qquad x \oplus y' = k$

=> key recovery from single input/output pair

$\uparrow$ "

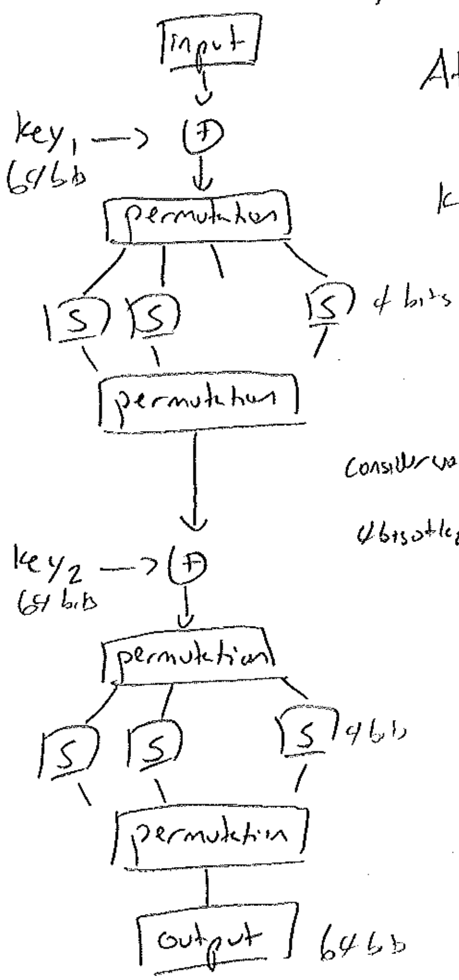
$\uparrow$ "

$\uparrow$ "

$\uparrow$ deterministic, known

y start here

Two-round substitution-permutation network



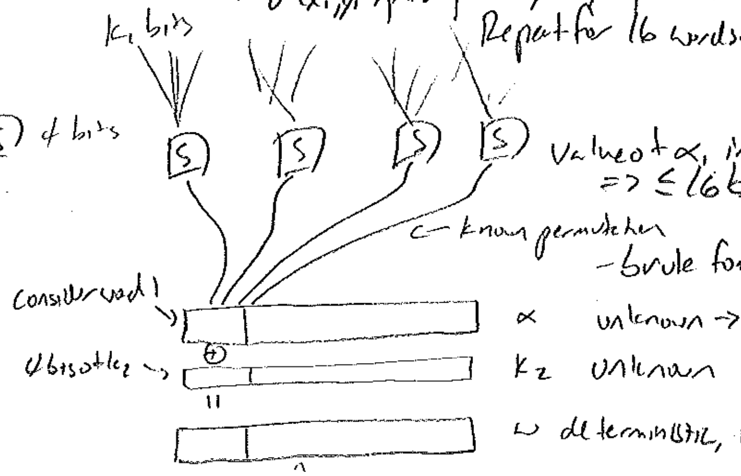
Known plaintext: $(x_1, y_1 = F_k(x)), (x_2, y_2 = F_k(x)), \ldots$

Attack: Do 2^{20} work to get 2^{16} possible (k_1, k_2) pairs

Use (x_i, y_i) pairs to narrow: 2^{20} work $\sim$ Pr(pass) $\sim 2^{-4}$

=> 8 (x_i, y_i) pairs: probably only 1 pair left, work: $8 \cdot 2^{20} = 2^{23}$

Repeat for 16 words of k_2 => $16 \cdot 8 \cdot 2^{20} = 2^{27} < 2^{128}$



Value of α, influenced by ≤ 4 S-boxes

=> ≤ 16 bits of k_1 influence value

- brute force 16 bits of k_1 + 4 bits of k_2

verify solution. 2^{20} work

Pr(incorrect soln. works) $\sim \frac{1}{2^4} = \frac{1}{16}$

$\Rightarrow \frac{2^{20}}{2^4} = 2^{16}$ possibilities

$\uparrow$ deterministic

$\uparrow$ deterministic

deterministic

$\uparrow$

$\uparrow$ deterministic

y start here

=> not enough diffusion in 2 rounds

Attacking 2DES

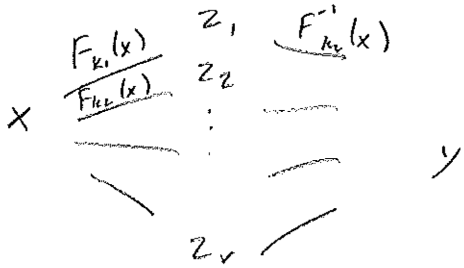
$F'_{k_1,k_2}(x) = F_{k_2}(F_{k_1}(x))$

DES => $k_1 k_2$ = 112 bits = Ok from exhaustive search

Meet-in-the-middle attack

$(x, y = F'_{k_1,k_2}(x))$

1. For each $k_1 \in \{0,1\}^n$, compute $z = F_{k_1}(x)$ and store (z, k_1)
2. For each $k_2 \in \{0,1\}^n$, compute $z = F^{-1}_{k_2}(y)$ and store (z, k_2)
3. Find matches $(z_i, x_i)(z_j, y_j)$ $(z_i = z_j)$ => (x_i, y_j) is possible key.



F has key, block length n
- $Pr(k_1, k_2 \text{ matches}) \sim 2^{-n}$
=> $E[\# \text{ matches}] \sim \frac{2^{2n}}{2^{-n}} = 2^n$

=> Given more input-output pairs => narrow down to 1 pair

Running Time:

step 1: 2^n time

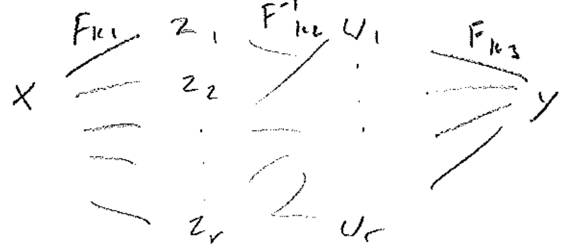
step 2: 2^n time

step 3: $2 \times O(n \cdot 2^n)$ time to sort, 2^n time to match => $O(n \cdot 2^n)$ total time

$O(2^n)$ space

Meet-in-the-middle for 3DES

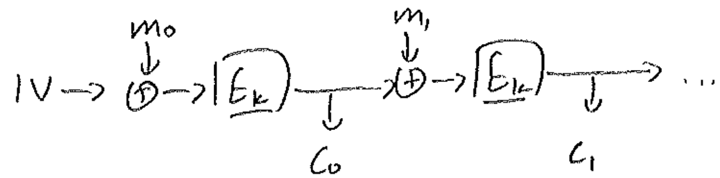
$F_{k_1}(F^{-1}_{k_2}(F_{k_3}(x))) = F'_{k_1 k_2 k_3}$ $\qquad 2^{2n}$ time



Chosen Plaintext attacks against CBC mode

③

CBC mode



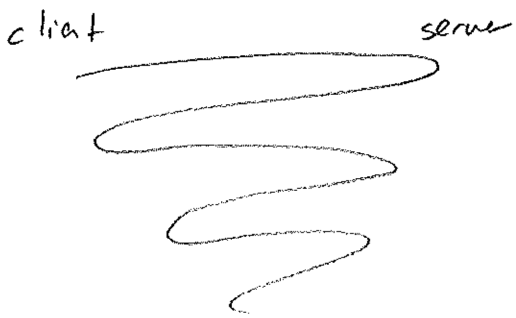
Rogaway 1995:

CBC mode not secure against chosen plaintext if attacker can see IV or previous block before choosing m

1. Attacker observes $c_1\ c_2 \ldots c_{i-1}\ c_i \ldots \quad c_j$
   c_i: wants to decrypt; c_j: we're here in stream
2. Attacker guesses c_i plaintext is p.
3. Attacker causes victim to encrypt $c_j \oplus c_{i-1} \oplus p$
4. Victim sends $E_k(c_j \oplus (c_j \oplus c_{i-1} \oplus p)) = E_k(c_{i-1} \oplus p)$
5. Attacker compares $E_k(c_{i-1} \oplus p)$ to c_i, match if p correct.

Dai 2002: SSH2 chains ciphertext between client & server

SSLv3, TLS 1.0 also chain between requests



Problem: 128-bit block size is a lot to brute force

Solution: In many use cases we know almost all of the text already

Problem: How does real adversary request encryptions?

Solution: Duong + Rizzo 2011

Javascript, Java etc. allow browsers to make cookie-bearing requests across domains

Websockets, HTML5

BEAST (Browser EXploit Against SSL/TLS)

BEAST continued: (4)

Threat model: Attacker can make arbitrary requests from victim client
Attacker has network access to see encrypted traffic

Contents of HTTPS encrypted messages:

HTTP requests: known, attacker-controls

GET /index.html HTTP/1.1

Cookie: sessionid = ...

want to steal

(body)

Duong + Rizzo: Pad plaintext so 1 byte of unknown text is in a block.

1. Alice HTTPS POST → http://bob.com/AAAAAA...

block boundary

$\underbrace{\text{POST /AAAAAA HTTP/1.1\r\n Cookie: sessionid = a}}_{\text{chosen by attacker}}$bcd

1 unknown character

2. Mallory captures ciphertext blocks $C_1 \ldots C_n$

3. Mallory requests $C_1 \oplus C_n \oplus P$ ← brute forcing 256 chars until match

4. Mallory shifts As until new block exposed.

Fixes: - Implement TLS 1.1 (many sites don't since 2006)
- Prepend empty records (fails due to bugs)
- Send one byte of data in 1st CBC-encrypted data record to randomize IV (fails due to bugs)

Adam Langley "The internet is vast and filled with bugs."

- Use RC4 (see Lecture 3)

CRIME Attack (Compression Ratio Info-leak Made Easy)

Kelsey 2002: Compression is a side channel if compress then encrypt and leak length

Duong + Rizzo 2012:
- Attacker controls body of HTTP requests

POST / HTTP/1.1
Host: bob.com
...
Cookie: secret = 1234
...
Cookie: secret = 0 } body controlled by attacker

Compressed length differs if attacker correctly guesses byte of cookie

LZ 77 Compression algorithm:

[window] [↑] [lookahead buffer]
coding position

1. Start at beginning of stream
Repeat until done:
2. Find longest match in window for lookahead buffer.
3. Output ((B, L), C)
move B chars back; copy L chars to output; next non-match character
4. move coding position L+1 chars forward

Chosen Ciphertext Attack Security

CCA indistinguishability experiment:

A		V
oracle access to $\mathrm{Enc}_k(\cdot)$, $\mathrm{Dec}_k(\cdot)$		$k = \mathrm{Gen}(1^n)$
	$\xrightarrow{m_0, m_1}$	$b \in_R \{0,1\}$
	$\xleftarrow{c}$	$c \leftarrow \mathrm{Enc}_k(m_b)$
- Can't query $\mathrm{Dec}_k(c)$	$\xrightarrow{b'}$	A succeeds if $b' = b$

Def: "indistinguishable encryptions under a chosen ciphertext attack" (CCA-secure)

$\forall$ p.p.t adversaries A $\quad\quad$ ε negligible

$\Pr[A \text{ succeeds}] \leq \frac{1}{2} + \varepsilon(n)$