Block Ciphers

(last time: one-time pad)

Q: If I flip 1st bit of OTP ciphertext, how does that change the result of decryption?
A: flips 1st bit of plaintext

What are DES, AES, etc.??

Q: How many bits required to represent a choice of among 500 items? \(\frac{9}{\text{bits}}\)

How about \(N\) items? \(\frac{\log_2 N}{\text{bits}}\)

Idea: I only care about \(n\)-bit msgs ("blocks")

I choose permutation \(p\) of the set \(\{0, 1\}^n\)

\(p: \{0, 1\}^n \rightarrow \{0, 1\}^n\) set of \(n\)-bit strings

I "encrypt" \(m \in \{0, 1\}^n\) as \(p(m)\)

Must be permutation so I can "decrypt" \(p^{-1}(c)\)

Permutation itself is the key

\(\rightarrow \) # of possible keys = \((2^n)!\)

\(\rightarrow \) # of bits needed to store key = \(\log((2^n)!))\)

\(~ n \cdot 2^n\)
Modern Crypto:

What I wanted:
- pick a random permutation as my key

What I would settle for:
- pick a permutation from some small set
  (if permutations from small set "look like")
  permutations from big set

Note: In crypto, block ciphers are called "pseudo random permutations"

Block Cipher: \( F \) takes 2 arguments: key, plaintext block

\[ F(k, m) \text{ or } F_k(m) \]

We also have \( F^{-1} \) function

\[ F^{-1}(k, F(k, m)) = m \]

So \( F^{-1}(k, \cdot) \) and \( F(k, \cdot) \) are inverse permutations.
Assume that length of key = length of block then only $2^n$ possible permutations of the form $F(k, \cdot)$ (compared to $(2^n)!$, total permutations)

**Security Def:**

game between “Adversary” & “challenger”

Challenger flips a coin
  - if heads: choose random key $k$
    - set $p = F(k, \cdot)$
  - if tails choose random permutation $p$
    - from set of all permutations

repeatedly:
  - Adv picks block $x$
  - gets back $p(x)$
  - Adv guesses the coin toss

**Note:** can always get it right w/ prob 0.5

So, “success” = $\Pr[\text{guessed right}] - \frac{1}{2}$

Block cipher is secure if all attacks require exponential effort.
OpenSSL usage:

enc: openssl enc -aes-128-ecb -nosalt -k "passwd"

dec: --- enc -d ---

base64 to encode printer-friendly

Relating block-cipher security to concrete attacks:

Key-recovery attack:

Adv chooses $x$, gets back $F(k, x)$
repeat
Adv outputs guess for $k$

Claim: If $F$ satisfies my security def then key recovery attacks not possible

Contrapositive: If I can do key recovery, then I can distinguish block cipher from random permutation