Types of attacks

Attack models

Let

- $m, k, c \in \mathcal{M}, \mathcal{K}, \mathcal{C}$
- $E : \mathcal{M} \times \mathcal{K} \longrightarrow \mathcal{C}$ be an encryption function
- $c = E(m, k)$

Ciphertext only attack - COA

https://en.wikipedia.org/wiki/Ciphertext-only_attack
Given

- One or a set of ciphertexts c_i

Task:

- Find the key k or
- Find the message m_i
- Find the next message m_{i+1}

Known plaintext attack - KPA

https://en.wikipedia.org/wiki/Known-plaintext_attack
Given:

- a bit or the whole message m_i (the crib)
- the ciphertext c_i

Task:

- Find the key k or
- Find the next message m_{i+1} from c_{i+1}

Chosen plaintext attack - CPA

https://en.wikipedia.org/wiki/Chosen-plaintext_attack

Given:

- Choose some m_i messages
- Gen their encryption c_i

Task:

- Find the key k or

- Find the next message m_{i+1} from c_{i+1}

Intuiton:

- more control to the attacker
- The attacker can explore vulnerabilities of $\mathcal{C} \times \mathcal{M} \times \mathcal{K}$ and nonrandom behaviour
- Attacks **ciphertext indistinguishability**
    - Given c and a random string r An attacker *must not* be able to distinguish between them

Remark:

- CPA-security is stronger than KPA and COA security
- A cipher CPA-secure is KPA and COA secure

Adaptive CPA - CPA2

- The attacker can request another set of messages m_j after seeing the first set
- This enables him to modify the message choice depending on the results of the previous encryption

Chosen Ciphertext attack - CCA

https://en.wikipedia.org/wiki/Chosen-ciphertext_attack

Given:

- Choose some c_i ciphertexts
- Gen their decryption m_i

Task:

- Find the key k

Adaptive CCA - CCA2

- The attacker can request another set of ciphertexts c_j to be decrypted after seeing the first set
- This enables him to modify the ciphertext choice depending on the results of the previous decryptions

Open key model attacks

Given:

- Some knowledge about the key
    - Related-key
    - For a chosen key he can distinguish from random

Task:

- Decrypt the message m

Attack types

Weak algorithm

Implementation attack

- Mistakes in the implementation / software of the protocol / encryption algorithm

Statistical attacks

- Not enough randomness
- Can exploit the indistinguishability propriety

Mathematical attack

- Small dataset of keys => Weak encryptions

Analytic attack

- Use algebraic proprieties to weaken the attack (Ex: LLL)

Brute Force

Try every possible combination to search for the key / password

- — Slow

Dictionary attacks

Brute force using a dictionary of common words (think of the language too)
An attacker can combine them too

- + Faster than brute force
- — Useless against good passwords

Rainbow tables

Optimized, precomputed table for caching the output of cryptographic hash functions

- + Fast
  - Used for cracking password hashes
  - Given hash -> lookup rainbow table -> get password
- — You need a rainbow table for each hash type

Birthday attack

Hash collision - The same hash value for two different messages

- Find collision through brute force
- + Based on the birthday paradox, there is a high chance to find a collision on small hashes
- — Weak against big hashes

Meet-in-the-middle

https://en.wikipedia.org/wiki/Meet-in-the-middle_attack

- KPA-like
- space-time tradeoff
- Use both $\mathcal{C}$ and $\mathcal{M}$ spaces

Differential Cryptanalysis

- applicable primarily to block ciphers, stream ciphers and cryptographic hash functions.
- In the broadest sense: how differences in information input can affect the resultant difference at the output.
- The attacker follows several messages of plaintext into their transformed ciphertext. He observes the changes form plaintext to the ciphertext and deduces the key.
- CPA-like attack

Linear Cryptanalysis

- KPA against messages encrypted with the same key
- Get insight into the probability of a particular key
- If more messages are attacked, there is a higher possibility of finding the particular “key"

Side-Channel attacks

https://en.wikipedia.org/wiki/Side-channel_attack
Based on the faulty implementation of a system rather than software bugs

- Timing information
- Power consumption - Power analysis attacks
- Electromagnetic leaks
- Sound