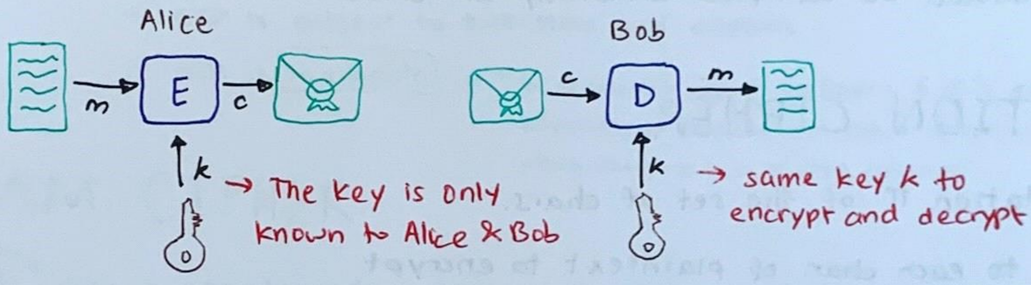
SYMMETRIC ENCRYPTION

→ A symmetric cipher consists of 2 algos:

1) Encryption algo. E
2) Decryption algo. D



→ An encryption scheme is secure if an adversary cannot:

- recover key k
- recover the plaintext m underlying a ciphertext c
- recover any bits of the plaintext m underlying a ciphertext c

KERCKHOFF'S PRINCIPLE

→ The architecture and design of a security mechanism should be made public

→ The E and D algos are public; the security relies entirely on the secrecy of key

ATTACK MODEL

→ specifies the kind of access an attacker has to a system

1) Ciphertext-only attack (COA)
2) Known-plaintext attack (KPA)
3) Chosen-plaintext attack (CPA)
4) Chosen-ciphertext attack (CCA)

BRUTE FORCE ATTACK

→ Try all possible keys $k \in \mathcal{K}$ - requires some knowledge abt. the struct. of plaintext

→ To make exhaustive search unfeasible:

- keys should be sufficiently long
- keys should be sampled uniformly at random from $\mathcal{K}$

SUBSTITUTION CIPHER

1) A permutation π of the set of chars.
2) Apply π to each char. of plaintext to encrypt
3) Apply π^{-1} to each char. of plaintext to decrypt

BREAKING THE CIPHER

- Key space size: $|\mathcal{K}| = 26! \simeq 2^{88}$ ⇒ Brute-force infeasible!
- Use frequency analysis → exploits regularities of the lang.
  - ↳ freq. of letters, digrams, trigrams, expected words...
    - ↳ the > and > ing.

THE ONE-TIME PAD (OTP)

1) $M = C = K = \{0, 1\}^n$
2) ENCRYPT: $\forall k \in K.\ \forall m \in M.$ $\boxed{E(k, m) = k \oplus m}$

$$
\begin{array}{rcccccccc}
k = & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
m = & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
\hline
c = & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

3) DECRYPT: $\forall k \in K.\ \forall c \in C.$ $\boxed{D(k, c) = k \oplus c}$
4) check CONSISTENCY: $D(k, E(k, m)) = k \oplus (k \oplus m) = m$

↳ The OTP satisfies perfect secrecy

A cipher (E, D) over (M, C, K) satisfies perfect secrecy if for all messages $m_1, m_2 \in M$ of same length, and for all ciphertexts $c \in C$

$$| Pr(E(k, m_1) = c) - Pr(E(k, m_2) = c)| \leq \underbrace{\varepsilon}_{\text{negligible qty.}}$$

↳ LIMITATIONS:

1) The key should be as long as the plaintext

2) Getting true randomness

- The key shld not be guessable from an attacker

3) Perfect secrecy does not capture all possible attacks

- OTP is subject to two-time pad attacks
- OTP is malleable → "An encryption algo. is malleable if it is possible for an adversary to transform a ciphertxt into another ciphertxt which decrypts to a related plaintext"

STREAM CIPHER

→ IDEA: Use a pseudorandom key rather than a really random key

- The key will not rlly be random, but will look random
- Key will be generated frm. a key seed using a Pseudo-Random Generator (PRG)

$$G : \{0,1\}^s \to \{0,1\}^n \text{ with } s \ll n$$

→ ENCRYPT: Using PRG G, $E(k,m) = G(k) \oplus m$

→ DECRYPT: Using PRG G, $D(k,m) = G(k) \oplus c$

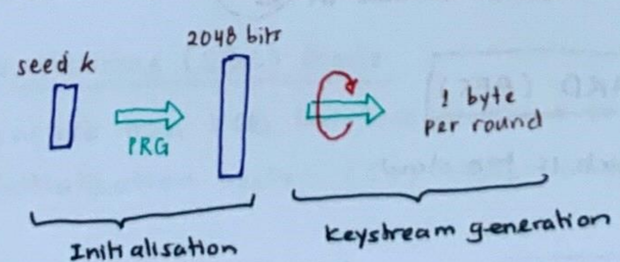
→ Stream ciphers are still subject to two-time pad attacks and are malleable.

↳ do not satisfy perfect secrecy bc. the keys in K are smaller than the messages in M

↳ Use a random IV to prevent this

RC4

→ is a stream cipher, consisting of 2 phases



Main data struct is array S of 256 bytes

→ Used in HTTPS and WEP

→ WEAKNESSES:

1) First bytes are biased → Drop the first 256 generated bytes

2) Subject to related key attacks → Choose randomly generated keys as seeds

BLOCK CIPHER

→ w/ params k and ℓ is a pair of deterministic algos (E, D) s.t.

- ENCRYPT: $E : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$
- DECRYPT: $D : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$

e.g. 3DES : $\ell = 64, \ k = 168$

AES : $\ell = 128, \ k = 128/192/256$

DATA ENCRYPTION STANDARD (DES)

→ Widely deployed in banking (ATM machines)

→ Attacks on DES → Exhaustive search
→ Linear cryptanalysis

→ 3DES — resistant against exhaustive search attacks

↳ Used in bank cards & RFID chips

↳ $E_{3DES}((K_1, K_2, K_3), M) = \underbrace{E_{DES}}(K_1, \underbrace{D_{DES}}(K_2, \underbrace{E_{DES}}(K_3, M)))$

Decrypt so that it'll be backward compatible

$D_{3DES}((K_1, K_2, K_3), C) = D_{DES}(K_3, E_{DES}(K_2, D_{DES}(K_1, C)))$

∴ 3 times as slow as DES

↳ Key-size = $3 \times 56 = 168$ bits

∴ Exhaustive search attack in 2^{168}

Meet-in-the-middle attack can bring this down to 2^{118}

ADVANCED ENCRYPTION STANDARD (AES)

→ Goal is to replace 3DES which is too slow

- m_i : 4×4 byte matrix, K_i: 128-bit key
- m_0: plaintext, m_{11}: ciphertext
- at the last round MixColumns is not applied

USING BLOCK CIPHER

→ Goal is to encrypt M using a block cipher operating on blocks of length ℓ when $|M| \neq \ell$

1) Bit Padding

   - append a set bit ('1') at the end of message, and then append as many reset bits ('0') required

2) ANSI X.923

   - pad w/ zeroes and last byte defines no. of padded bytes

3) PKCS #7

   - value of each added byte is the total no. of padding bytes

   e.g. The padding will be 01 / 02 02 / 03 03 03 / 04 04 04 04 . . . .

→ Electronic Code Book (ECB) mode

- To encrypt a message M under key K using ECB mode:

  1) M is padded
  2) M' is broken into m blocks of length ℓ
  3) Each block is encrypted under key K using the block cipher
  4) Ciphertext is the concatenation of the C_i s

- Weakness:

  ↳ There is 1-to-1 mapping from message blocks to ciphertxt blocks Hence, malleable and weak to freq. analysis → Not used in practice!

→ Cipher-block chaining (CBC) mode

- More secure than ECB but less resilient to packet loss
- Uses initialisation vector (IV) chosen at random

→ Counter (CTR) mode

- More secure than ECB and parallelisable — every step of encryption & decryption can be done in parallel