01 - Introduzione alla cybersecurity

Alcune definizioni

- **Plaintext**: messaggio originale che viene criptato dal programma
- **Encryption** algorithm (cipher): trasforma il plaintext e lo rende illeggibile
- **Secret Key**: secondi input all '*Encryption algorithm*' che determina le esatte trasformazioni fatte dall'algoritmo sul plaintext
- **Ciphertext**: plaintext trasformato (output dell'Encryption algorithm)
- **Decryption** algorithm: inverso dell'Encryption algorithm (prende un Ciphertext come input e la secret key per restituire il plaintext originale)
- **Cryptography**: lo studio di sicuri e efficienti ciphers
- **Cryptanalysis**: il processo tramite il quale si cerca di capire il plaintext o la chiave. Le strategie usate dipendono dall'algoritmo di cifratura
- **Cryptology**: Cryptography + Cryptanalysis
- **Secret/Private-key** (simmetric) Crypthografy:
  - Le chiavi usate per la crittazione sono le stesse (e quindi vanno condivise tra chi invia il messaggio e chi lo riceve).
  - Gli algoritmi per di Encryption e Decryption **C** e **D** sono spesso gli stessi
- **Public-key** (asymmetric) Cryptography:
  - Vengono usate chiavi differenti per Encryption e Decryption
  - Le funzioni di Encryption e Decryption **C** e **D** sono diverse

Tipologie di attacco

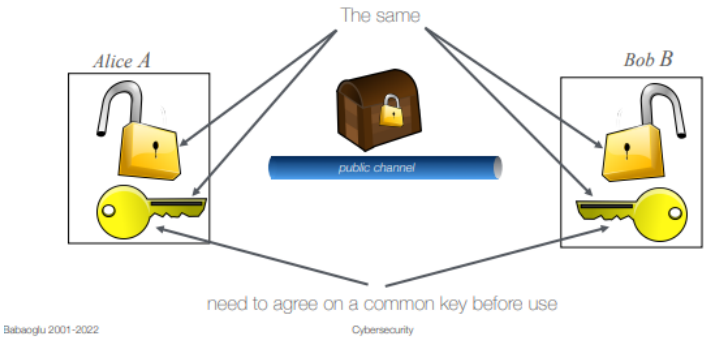
In base a cosa conosce il cryptanalyst oltre l'algoritmo di cifratura/decifratura esso determina la strategia migliore da usare:

- Brute-force attack: non conosce nulla oltre l'algoritmo di cifratura.
- Ciphertext attack: Ciphertext come collezione da $\mathbf{c}_1$ **a** $\mathbf{c}_n$
- Known plaintext attack: Ciphertext più una o più coppie (m_i, c_i)
- Chosen plaintext attack: Ciphertext più una o più coppie (m_i, c_i) dove m_i è scelta dal cryptanalyst

Definizioni

- Encryption function
  - C_k(m) = c “encryption of m with key k”
- Decryption function
  - D_k((c) = m “decryption of c with key k”
- D_k is the mathematical inverse of C_k:
  - $D_k(C_k((m)) = m$
  - Qualche volta è richiesto che siano commutative → $D_k(C_k((m))) = C_k(D_k((m)) = m$

Secret-Key (Symmetric) Cryptography



le due parti devono scambiarsi una secret key comune, questo richiede che:

- concordino di non scambiarla tramite canali pubblici
- deve essere passata su un canale che verrà usato una volta sola (“*out-of-band*" channel)

- l' “*out-of-band*" channel può essere molto costoso e lento
- Lo spazio (dimensione) della secret key deve essere molto grande

Alcuni “secret key” (symmetric) ciphers:

- DES
- Triple-DES
- Blowfish
- International Data Encryption Algorithm (IDEA)
- Advanced Encryption Standard (AES)

Le funzioni per criptare e decriptare sono interscambiabili:
Sender e receiver:
- Conoscono la chiave segreta k
- Possono entrambi criptare e decriptare il messaggio
- Entrambi devono promettere di tenere k segreta
Difetti:

- richiede una chiave segreta comune
- per comunicare tra O(n) parti si necessitano $O(n^2)$ chiavi

Public-Key (Asymmetric) Cryptography

Scopo: interrompere la simmetria tra encrypting e decrypting
Chi conosce come criptare **non** deve conoscere come decriptare
La chiave k è divisa in due parti k[priv] k[public]

- La chiave viene generata dal destinatario
- k[priv] viene tenuta segreta dal destinatario
- k[public] viene resa pubblica dal destinatario e viene usata da chiunque intenda criptare un messaggio da inviare al destinatario
  Concetto di funzione “one-way trap-door”

One-way trap-door
è una funzione che è semplice da computare, difficile da invertire e semplice da invertire se si conoscono alcune informazioni extra (esempio di un lucchetto, facile da chiudere se aprire, difficile da aprire se chiuso tranne che con la chiave)