Triada de seguridad

1. Confidencialidad: Resguardar el acceso y privacidad de información de los usuarios y asociada, sean personas o empresas, etc.
   - Autorización: Damos acceso a distintas acciones a ciertos usuarios
   - Control de acceso
   - Cifrado: Transformación de la información para que no sea legile usando llaves de cifrado.
     - No solo por medio de la criptografía se mantiene la seguridad de la información, si no por medio de distintos mecanismos.
     - Protege la información de accesos no autorizados.
   - Autenticación: Verificación de la identidad de los entes que acceden a la información. Sea por medio de contrasenas, datos biometricos (huella), etc.
2. Integridad: Protege la información de modificaciones no autorizadas
   - Hashing
   - Firma digital
3. Disponibilidad: La información sea accesible, siempre, en el momento en el que se necesita.
   - Redundancia y tolerancia a fallos
   - Planes de recuperación ante desastres
   - Mantenimiento y actualizaciones regulares: De forma preventiva
     - DRP: Hay ciertos servicios que son más importantes que deben estar siempre disponibles.
       - Esto implica que tengamos prioridad de servicios a recuperar ante un desastre.

3 Estados de la información

- En reposo
- En uso
- En transito

Criptografía

Muy importante para los principios de seguridad de la información.

- Por ejemplo por medio de la Autenticación con TLS y LDAP (redes)
- No repudio:
  - Por medio de firmas digitales y sellos de tiempo, podemos tener evidencia de una acción especifica realizada por un ususario particular
  - La seguridad en redes implementan sistemas de registro y auditoría para mantener un historial de acciones realizadas por la red
  - Eso si, puede que aunque la firma y sellos que hay indiquen que un usuario lo hizo en realidad haya sido otra persona (por ejemplo por medio de acceso remoto), lo que requiere técnicas forenses para verificarlo
- Autorización:
- Privacidad: La criptografía nos ayuda a emplear distintos métodos de cifrado para controlar la privacidad de los datos
- Seguridad en capas
- criptoanalisis: Cómo rompemos el algoritmo para poder obtener la llave.
  - Estegoanalisis: Ver cómo se pudo haber ocultado

Mensaje

- Texto plano: El mensaje original
- Texto cifrado: Dependiendo del algoritmo y llaves, esta cifrado
- Algoritmos de cifrado: Funciones matemáticas que convierten de texto plano a cifrado

El objetivo de lal ciclo de la información es poder hacer la difusion, es decir el poder hacer algo con lo que se difunde

Cifrado:

Queremos transmitir un mensaje de forma que lo haga sin pérdida, de forma secreta, que solo lo pueda recibir la persona objetivo.

- Permutación: Se permutan en grupos las letras hacia la izquierda-derecha
- Número de llaves:
  - Simétrico: 1 sola llave
  - Asimétrico: 2 llaves. Una publica y otra privada
- Cómo se procesa el algoritmo:
  - Por bloques de N bytes?
  - Por flujo?

Cesar

Nombre por el emperador romano Julio Cesar. Shift de las letras del mensajes. Categorías:

- Sencillo: Un mismo shift para todas las letras
- Monoalfabética: Un shift arbitrario para cada letra en el mensaje

Criptoanalisis si es monoalfabético: Se puede hacer fuerza bruta porque solo son 25 o 26 llaves posibles.

- Tenemos 27! llaves de cifrado cuando es monoalfabética (que no hay un solo corrimiento sino que cada letra tiene un shift aleatorio asignado) por el numero de permutaciones posibles.
- Podemos ver la frecuencia de las letras para compararlo contra la frecuencia de letras en oraciones normales
- Podemos usar bigramas para ir viendo qué onda
  - (t, h) es el más común

Playfare

Hill Cipher

- Lester Hill 1929. Es un cifrado multi letra.
- A cada letra se le asigna un valor del 0 al 26
- Es una multiplicación de matrices $\text{Cifrado} = \text{Plano}\ \text{Key} \bmod 26$

Con:

- $\text{Plano} = \text{Pay more money}$
- $\text{Key} = [[17, 17, 5], [21, 18, 21], [2, 2, 19]]$

```
import numpy as np

P = "Pay more money"

Key = np.array([[17, 17, 5], [21, 18, 21], [2, 2, 19]])
C = np.array([ord(c) - ord('a') for c in P.lower() if c != ' '])

Cp = C.reshape(-1, 3)
```

```
" ".join(["".join([chr(v + ord('a')) for v in np.matmul(c, Key) % 26]) for c in Cp])
```