UTN-FRBA

GESTION DE DATOS

INTEGRIDAD Y ENCRIPTACION

DIRECTOR CATEDRA: ING. ENRIQUE REINOSA

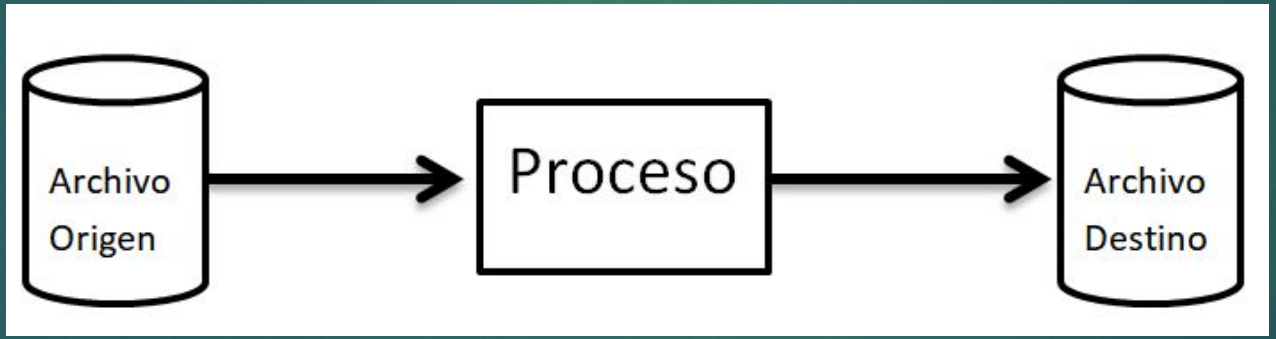
INTEGRIDAD Y ENCRIPTACION

- ***Integridad***: Es el proceso que permite identificar la integridad de un archivo, o sea, poder validar que un archivo no a sufrido cambios ante un proceso que lo modifico, sea este por una transferencia o un proceso de cambio como la compresión y descompresión del mismo.

- ***Encriptación***: es el proceso que permite modificar el contenido de un archivo para que mantenga el contenido establecido, pero que el mismo no pueda ser visible en un formato tradicional del mismo.

PROBLEMÁTICA

Cuando se produce una modificación en un archivo *a través de un proceso modificatorio* ***se requiere conocer la integridad del nuevo archivo generado.***



CONTROL DE INTEGRIDAD

- **Necesidad**: Cuando se produce una modificación sobre un archivo, por ejemplo cuando se transmite mediante una red o se copia o se le realiza un proceso de compresiòn y descompresiòn aparece la necesidad de verificar si el nuevo archivo obtenido es igual al original.

- ***Procedimiento***: si bien es cierto que la ùnica forma de verificar que un archivo es igual a otro se requeriría una lectura de ambos archivos con una comparación de uno a uno, existen medios tecnologicos para poder validar la veracidad del archivo sin necesidad de contar con el archivo original.

CHECKSUM, CRC

CONTROL DE INTEGRIDAD

Que se debe controlar para establecer la igualdad de los archivo Origen y Destino

- **Tamaño:** Ambos archivos deben tener el mismo tamaño en cantidad de caracteres.
- **Contenido:** Ambos archivos deben contener los mismos caracteres
- **Posición:** Considerando que ambos archivos contienen los mismos caracteres, dichos caracteres deben estar en la misma posición

CKECKSUM

- El método se basa en el uso de un polinomio, dado que el mismo evalúa tamaño, contenido y posición.
- El tamaño esta dado por el grado del polinomio.
- El contenido esta dado por los coeficientes del polinomio.
- La posición esta dada por el grado que acompaña a la x del polinomio.

CKECKSUM - EJEMPLO

Tomemos a modo de ejemplo que se quiere validar un archivo que contiene la expresión 'HOLA'

Para validar esta situación se puede armar un polinomio donde se utilicen los caracteres que componen el archiivo como coeficientes del polinomio:

$$Hx^0 + Ox^1 + Lx^2 + Ax^3$$

CKECKSUM - EJEMPLO

$$Hx^0 + Ox^1 + Lx^2 + Ax^3$$

- Antes de realizar el procedimiento del archivo se calcula el polinomio y se resuelve aplicando una raíz específica.
- Luego el resultado se agrega al archivo destino generado vía el proceso de transformación.
- Luego se vuelve a generar el polinomio con el contenido del archivo destino.
- Se resuelve dicho polinomio y se compara el resultado obetnido con el almacenado en el archivo, si es igual se puede afirmar que los archivos son iguales.

CKECKSUM - EJEMPLO

$$Hx^0 + Ox^1 + Lx^2 + Ax^3$$

- En la realidad no se arma un polinomio con caracteres como coeficientes, sino que se toman los bits que componen dicho carácter de forma tal de iluminar o apagar potencias para evitar las fallas en el método.

1 char = 1 byte = 8 bits

- Si H se representara como 00110011 el polinomio se armaría así:

$$0x^0 + 0x^1 + 1x^2 + 1x^3 + 0x^4 + 0x^5 + 1x^6 + 1x^7$$

CKECKSUM - EJEMPLO

$$0x^0 + 0x^1 + 1x^2 + 1x^3 + 0x^4 + 0x^5 + 1x^6 + 1x^7$$

- De esta forma la posibilidad de error de la función es nula porque las diferencias con incompensables intercambiando potencias.

- El valor de raíz que se toma es un valor en el intervalo comprendido en el rango ***(0:1)***

ENCRIPTACION

- El objetivo es ocultar la información contenida en el archivo para que no pueda ser legible.
- El archivo debe ser modificado sin cambiar su tamaño y espacio ocupado a tal fin.
- Para encriptar existen innumerables métodos para realizarlo, nos concentraremos en las diferentes formas de encriptar que existen

ENCRIPTACION

Procesos de Encriptación

- ***Desplazamiento***: los procesos de encriptación se basa en el desplazamiento de los caracteres en función de algún patrón
- ***Reemplazo***: los procesos de encriptación se basan en el reemplazo de determinados caracteres en función del algún patrón donde ese reemplazo puede ser fijo o variable, con lo sin intervención del usuario
- ***Mixto***: se aplican ambos procesos en cualquier orden, reemplazo y desplazamiento.

ENCRIPTACION POR REEMPLAZO

- ***Reemplazo Fijo***: se toma un valor por el cual se van reemplazando a determinados caracteres por un valor de acuerdo a un patrón, por ejemplo reemplazar todas las posiciones pares por su contenido por un valor Ascii preestablecido.

- ***Reemplazo Variable***: Se encripta el archiivo con una clave dispuesta por el usuario y dicha clave se copia al contenido del archivo o valor a encriptar.

ENCRIPTACION POR DESPLAZAMIENTO

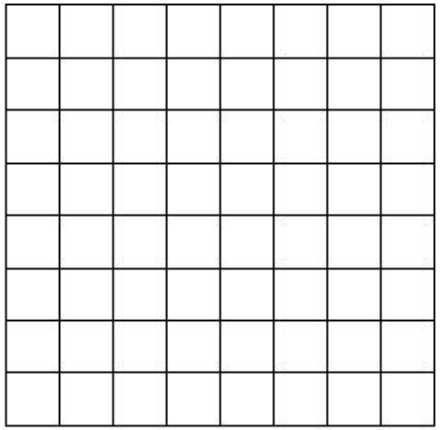
- ***Desplazamiento***: se desplazan valores en forma similar a lo que se conoce como "sopa de letras" pero en lugar de realizarlo por caracteres se realiza por bits, modificando totalmente el contenido del mismo.

ENCRIPTACION MIXTA

- ***Mixta***: se aplican ambas formas de encriptación una primero y otra después, en cualquier orden de ejecución, de forma tal de dar mayor seguridad a la encriptación.

EL SALTO DEL CABALLO

Existe un tablero de 8 x 8 conformado por 64 casilleros



Fila	Col
2	1
2	-1
-2	1
-2	-1
-1	2
-1	-2
1	2
1	-2

El caballo se mueve en "L" o sea en desplazamientos 2 x 1 o 1 x 2 en todos los sentidos, esto genera que existan 8 movimientos legales

El juego inicia de una posición que puede ser cualquier vértice o sea una posición (x,y) y desde ahí debe jugar a otra posición legal pero además posible, o sea, que caiga dentro del tablero y que no este ocupada.

El objetivo del juego es completar todo el tablero con números del 1 al 64 de forma tal que requiere moverse o jugar 64 veces.

EL SALTO DEL CABALLO

- Se debe considerar que existen ***8 movimientos legales*** y dentro de ellos se van a encontrar ***x movimientos posibles***.
- El algoritmo persigue el objetivo de jugar desde una posición preestablecida a una posición posible a ser utilizada.
- El algoritmo debería terminar después de ejecutarse 64 veces, de forma tal de completar el tablero con números de 1 a 64.
- En función de ello el algoritmo se resume moverse a una posición posible desde una posición en dada.
- Esto se debería repetir hasta llegar a los 64 movimientos.

EL SALTO DEL CABALLO

```
int caballo(int *tablero, int x, int y, int *f, int *c, int jugada)
tablero[x][y] = ++jugada;
if jugada == 64
    return 1;
for (i=1;i<8;i++)
{
    if ((x+f[i])<8 && (x+f[i])>=0 && (y+c[i])<8 && (y+c[i])>=0
         && !(tablero[x+f[i]][y+c[i]))
        if caballo(tablero, x+f[i], y+c[i], f, c, jugada)
            return 1;
        tablero[x+f[i]][y+c[i]] = 0;
};
return 0;
```