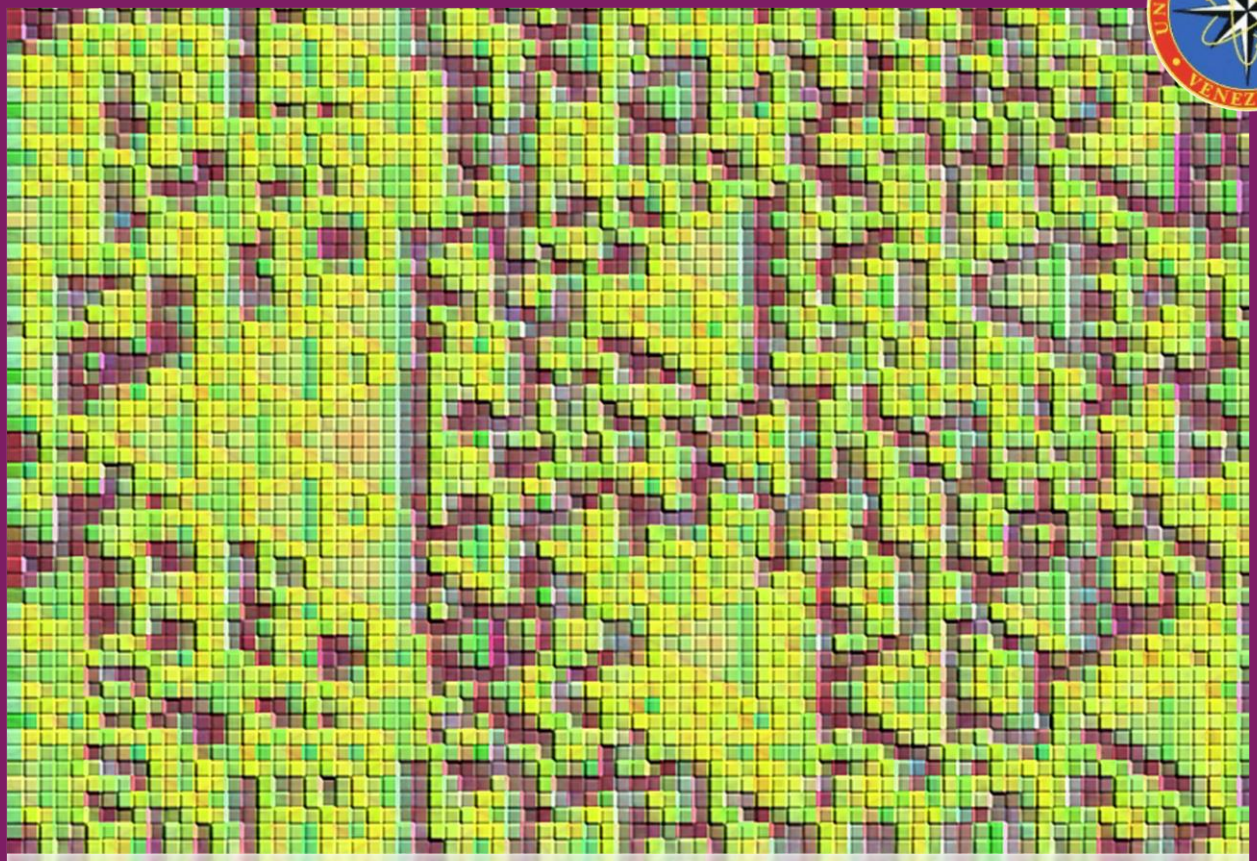


SEGURIDAD CENTRADA EN REDES

ENCRIPTADO

TECNICA DE ENCRIPTACION DE TEXTO PLANO

INFORME Y ANÁLISIS DEL ALGORITMO

POR: CESAR BLASCO, DAVID GÓMEZ Y GABRIEL MOBILIO

EL PROBLEMA

SE PROPONE EL DESARROLLO DE UN MÉTODO DE ENCRIPTAMIENTO ROBUSTO PARA ARCHIVOS DE TEXTO.

El texto es el tipo de dato de transito más común en la Internet y dicha data debiera transmitirse de forma confiable, asegurando que la información transmitida sea accesible sólo por las partes autorizadas. Para satisfacer el requerimiento de confidencialidad se han propuesto diversos tipos de mecanismos entre los cuales el más importante, con diferencia, es el cifrado o encriptamiento de la información.

En el proceso de encriptamiento el mensaje original es convertido en un mensaje aparentemente aleatorio y sin sentido denominado. El proceso de encriptamiento consta de un algoritmo y una clave. La clave es un valor independiente del mensaje original, de la que tiene conocimiento únicamente aquellos autorizados a acceder a la información real del mensaje; en cambio, el algoritmo y el mensaje encriptado son por lo general de conocimiento público.

Para un mismo mensaje original el algoritmo producirá salidas diferentes dependiendo de la clave específica que se haya utilizado. Al cambiar la clave cambia la salida del algoritmo. De ahí que sea la clave el mecanismo de acceso al mensaje. Una vez que el mensaje ha sido producido, este puede ser transmitido. Luego de la recepción, el texto cifrado puede ser transformado en el texto plano original usando un algoritmo de desencriptamiento y la misma clave que fue usada para el encriptamiento.

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

DESARROLLAR E IMPLEMENTAR UN MÉTODO DE ENCRIPTAMIENTO SIMÉTRICO BASADO EN LA APLICACIÓN DE MÁSCARAS XOR Y REORDENAMIENTO ITERATIVO.

El primer concepto a tratar es el de la clave y sus propiedades. Una clave es una cadena de por lo menos 5 caracteres alfanuméricos, que son transformados en valores enteros durante la operación del algoritmo.

Una condición indispensable para todo algoritmo de encriptamiento es que el mensaje encriptado resultante debe ser único para cada clave. Es absolutamente inadmisible que, por ejemplo, la clave 50000 produzca el mismo mensaje que la clave 00005. Aun cuando dos claves posean los mismos caracteres, la posición de estos debería generar un mensaje distinto y de igual forma debería suceder si dos claves tienen valores idénticos por la sumatorias de sus dígitos, por su producto o cualquier otro procedimiento.

C#

```
public int[] keyfix(String key)
{
    char[] res = key.ToCharArray();
    int[] i = new int[res.Length];
    for(int c=0;c<i.Length;c++)
    {
        i[c] = res[c];
        i[c]++;
    }
    return i;
}

public int getbyte(int[] key)
{
    int c = 0;
    foreach(int k in key)
    {
        c+=k+((13*k)/11)+((7*k)/3)+c/2;
    }
    Return c;
}
```

Para atender esta necesidad se crearon de 2 métodos: **getbyte** y **keyfix** Keyfix se encarga de traducir la String que se recibe a un arreglo de enteros. Dado que el valor **0** debilita algunos aspectos de la fortaleza del encriptado, todos los valores de la clave se incrementan.

getbyte es un método muy importante, este se ocupa de devolver un valor deformado único a partir de la clave suministrada por keyfix, dado que 13, 11, 7 y 3 son números primos, ningún producto de 13 que sea múltiplo de 11 será a su vez un producto de 7 y múltiplo de 3; lo que provee un aumento suficientemente irregular para cada valor de k. Además, cada valor es luego aumentado por la mitad del valor de la sumatoria, a manera de que sea todavía más difícil intuir un patrón para revertir el resultado. La razón por la que se consideró necesario deformar la clave viene explicado en el próximo concepto importante: la máscara.

La máscara es el proceso de transformación de un valor binario a uno derivado de la aplicación de una operación de deformación. El operador lógico XOR es comúnmente usado para esto, En criptografía, el cifrado XOR es, como su nombre indica, un algoritmo de cifrado basado en el operador binario XOR:

$$A \oplus 0 = A,$$

$$A \oplus A = 0,$$

$$(B \oplus A) \oplus A = B \oplus 0 = B,$$

El operador XOR es muy común como parte de cifrados más complejos. Sin embargo, por sí solo el cifrado XOR es muy vulnerable y es muy fácil obtener la clave a través del análisis de varios mensajes cifrados con la misma clave. Por esta razón, dos consideraciones fueron tomadas: como se mencionó anteriormente, getbyte se ocupa de deformar las claves para que en caso de revertir la máscara esta no sea revelada propiamente; en segundo lugar, la llamada a aplicación de la máscara nunca ocurre usando la transformación de la clave como único operador de cambio. Se usa en cambio una combinación del valor de la clave, su longitud y el mensaje. Mask, el método encargado de aplicar la máscara, viene definido como:

C#

```
public void mask(int key,ref char[] subject)
        {
            for(int c=0;c<subject.Length;c++)
            {
                subject[c] = (char)(subject[c] ^ key);
            }
        }
```

Con las bases explicadas, la implementación de los mecanismos de encriptado es como sigue:

C#

```
public string encripting(int[] key, string subject)
        {
            char[] res = subject.ToCharArray();
            int s = res.Length - 1;
            for(int x=0;x<key.Length;x++)
            {
                int k=key[x];
                for (int c = 0; c < k; c++)
                {
                    if (c > 0 && k % c != 0)
                    {
                        for (int i = 0; i < s; i++)
                        {
                            if (i % 2 == 0 && i < s - 1)
                            {
                                char aux = res[i];
                                res[i] = res[i + 2];
                                res[i + 2] = aux;
                            }
                            else
                            {
                                char aux = res[i];
                                res[i] = res[i + 1];
                                res[i + 1] = aux;
                            }
                        }
                    }
                }
                mask(k+getbyte(key)*
                ((key.Length+(int)(s/4))*
                (key.Length+(int)(s/4))),ref res);
            }
            mask(getbyte(key) + ( s / key.Length), ref res);
            String result = new string(res);
            return result;
        }
```

Obsérvese que existe un mecanismo adicional al enmascarado: el reordenamiento. El reordenamiento es una técnica simple que se ocupa de desordenar el mensaje original para romper con cualquier patrón reconocible que pudiera servir para intuir su contenido; en este caso se ha usado un intercambio entre posiciones pares e impares y un estado de ruptura en el patrón

que ocurre por cada múltiplo de k para hacer el desorden menos predecible. Es una técnica que, aunque débil, complementa adecuadamente el enmascarado ya que, como se mencionó antes, el resultado de la máscara depende del mensaje y el valor usado para transformarlo; si se altera el mensaje se obtiene un valor transformado distinto del que se obtendría al aplicar la máscara sobe el mensaje sin desordenar.

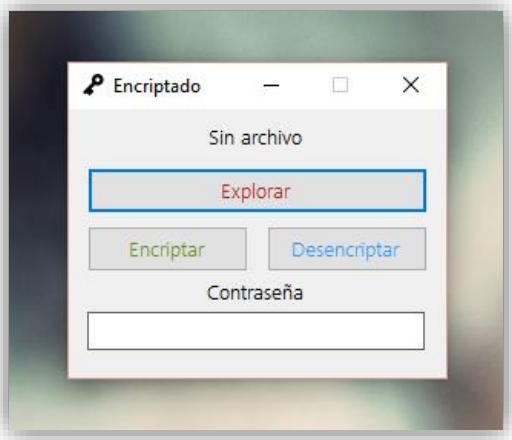
Sigue entonces el método de desencriptado:

C#

```
public string decripting(int[] key, string subject)
        {
            char[] res = subject.ToCharArray();
            int s = res.Length - 1;
            mask(getbyte(key) + (s / key.Length), ref res);
            for (int x = key.Length - 1; x >= 0; x--)
            {
                int k = key[x];
                mask(k+getbyte(key)*
                ((key.Length+(int)(s/4))*
                (key.Length+(int)(s/4))),ref res);
                for (int c = k - 1; c >= 0; c--)
                {
                    if (c > 0 && k % c != 0)
                    {
                        for (int i = s; i >= 1; i--)
                        {
                            if (i % 2 == 0)
                            {
                                char aux = res[i];
                                res[i] = res[i - 2];
                                res[i - 2] = aux;
                            }
                            else
                            {
                                char aux = res[i];
                                res[i] = res[i - 1];
                                res[i - 1] = aux;
                            }
                        }
                    }
                }
            }
            String result = new string(res);
            return result;
        }
```

El método se construye como la aplicación inversa del encriptado debido a que tanto la máscara como el reordenamiento son dinámicas reversibles.

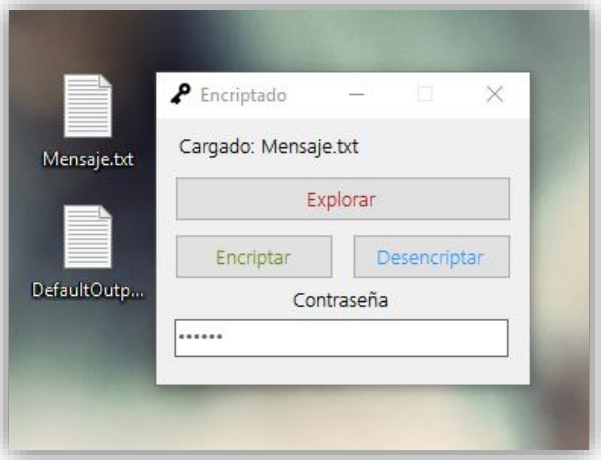
EXPLICACIÓN DE USO, RESULTADOS Y CONCLUSIONES

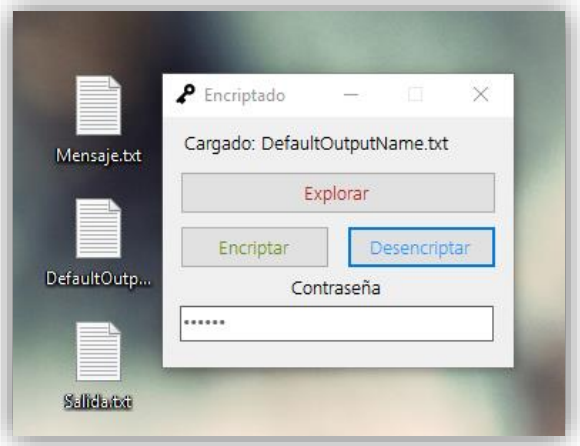


Se creó una interfaz al usuario del método, con los mecanismos necesarios para leer y escribir los archivos a encriptar, así como ingresar la contraseña a usar.

Explorar abre un selector de archivos con el cual cargar el texto, encriptar encripta dicho archivo y dirige a un dialogo de guardado.

Al desencriptar, debería antes seleccionar el archivo. Tenga en cuenta que el programa tiene precargado el último archivo que se cargó, por lo que deberá verificar que el archivo a desencriptar es aquel que se produjo con la clave en cuestión.





RESULTADOS

- Las pruebas realizadas demostraron que el mensaje encriptado era único para cada clave introducida.
- El mensaje que se produjo al desencriptar un mensaje encriptado con una clave diferente era siempre ininteligible sin importar cuan similares eran las claves.
- Todos los mensajes encriptados fueron, sin excepción, imposibles de leer y carentes de sentido. Solo al desencriptarlos se podía leer el mensaje original
- El mensaje producto del desencriptado es siempre exactamente igual al mensaje original

CONCLUSION

- LA NOTACIÓN UTILIZADA Y LOS MECANISMOS CREADOS PARA DESCRIBIR LA DINÁMICA, EVIDENCIAN QUE EL CIFRADOR XOR OFRECE UN MARCO DE REFERENCIA APROPIADO PARA EL DISEÑO DE ALGORITMOS DE ENCRIPTAMIENTO.
- LOS RESULTADOS OBTENIDOS DEMUESTRAN QUE LA DINÁMICA PROPUESTA PRODUCE UN CIFRADO DE ALTA CALIDAD.
- LA INCORPORACIÓN DEL PARÁMETRO DE DESORDENADO PROPICIA UN PROCESO DE DIFUSIÓN DE NATURALEZA NO LINEAL QUE FAVORECE LA CALIDAD DE LA DINÁMICA.