Instituto Politécnico Nacional

Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas

Seguridad de Redes

Profesor. Polanco Montelongo Francisco Antonio

Alumno: Hernández Guzmán Bryan Alexis

Grupo: 3TV4

Práctica: Cifrado y Descifrado con RC4 en C

Introducción:

RC4 es un esquema de cifrado de flujo (no basado en bloques) simétrico.

Fue diseñado por Ron Rivest (la *R* de *RSA*) en 1987. Originalmente era secreto, pero se filtró en 1994 a través de una lista de correo.

Es un esquema de cifrado extremadamente simple y puede implementarse en software de forma muy eficiente. Esto lo ha convertido en uno de los esquemas de cifrado más utilizados del mundo.

Sin embargo, *RC4* hace tiempo que no es considerado un algoritmo seguro. *RC4* es conocido por ser el mismo esquema de cifrado usado por *WEP* (*Wired Equivalent Privacy*), sistema criptográfico totalmente roto hoy en día.

Menos conocido es que *RC4* es usado aún en aproximadamente la mitad de las transmisiones *TLS* que ocurren en el mundo actualmente, desde para consultar tu correo hasta para establecer transferencias bancarias.

Hoy en día el interés por el *RC4* parte de querer conocer hasta qué punto está roto y cómo de vulnerables son los sistemas que lo utilizan.

Elaboración:

Primeramente, creamos un archivo .c en el cual escribiremos el código para cifrar y descifrar

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <openssl/rc4.h>
//cifra40.c 29 de Octubre 2005
// Este archivo cifra un archivo cualquiera con una clave dada


int main(int argc, char *argv[])
{
  //De leerachivo en var
  char nombre_archivo[50];
  char *sArchivote;
  long lTamano;
```

```
  char caracter[500];
  RC4_KEY key; //se define la varialbe key
  unsigned char data[16],buf[1024],*out,*out2;
  int outlen;
  //De leerachivo en var, para abrir el archivo a cifrar
  FILE * dfp; /* archivo a cifrar */
  FILE * Archivo; /* archive cifr en donde se va a depositar */
  if (argc<3) {
     printf ("Use %s archivo llave\n",argv[0]);
     exit(0);
     }
  if ((dfp = fopen(argv[1],"r")) == NULL ) {
  perror (argv[1]);
  exit(-1);
  }

  /*Si no son 2 argumentos (archivo y programa no se ejecuta
  el archivo) y si no se abre el archivo a cifrar no se
  ejecuta el programa */
  {
  // de leerachivo en var
  fseek (dfp,0L,SEEK_END);
  lTamano = ftell (dfp);
  sArchivote = malloc (lTamano);
  fseek (dfp,0L,SEEK_SET);
  fread (sArchivote,1,lTamano,dfp);
  printf("CONTENIDO DEL ARCHIVO ORIGINAL:\n----------------------------------------\n");
  printf("%s\n", sArchivote);
```

```
  fclose(dfp);
  // de leerarchivo en var
  RC4_set_key(&key,40,buf); //se inicai la llave con clave de buf
  RC4(&key, lTamano ,sArchivote,sArchivote);
  printf("CONTENIDO DEL ARCHIVO CIFRADO:\n----------------------------------------
\n%s\n\n",sArchivote);
  Archivo = fopen("cifr", "w");
  fwrite (sArchivote,1,lTamano,Archivo);
  //Se escribe la variable en el archive cifr
  RC4_set_key(&key,40,buf);
  RC4(&key, lTamano ,sArchivote, sArchivote);
  printf("CONTENIDO DEL ARCHIVO DESCIFRADO:\n----------------------------------------
\n%s\n\n",sArchivote);
  // de leer archivo en var
  free (sArchivote);
  // de leer archivo en var
  exit(1);
  }
}
```

Se aplicará los comandos como si se compilar se tratara para proceder a ejecutar, un problema que encontré fue con las librerías tanto las de c como las de openssl. También se tendrá un archivo de teto con un texto de prueba para visualizar

```
┌──(bryan㉿kali)-[~]
└─$ ls
cifr     cifrado.c  Documentos  Imágenes  Plantillas  Público
cifrado  Descargas  Escritorio  Música    prueba.txt  Vídeos

┌──(bryan㉿kali)-[~]
└─$ cat prueba.txt
Este es un texto de prueba
```

Ejecutaremos nuestro archivo .c , al mismo tiempo que el .txt

```
┌──(bryan㉿kali)-[~]
└─$ sudo gcc cifrado.c -o cifrado -lcrypto
cifrado.c: In function ‘main’:
cifrado.c:45:4: warning: ‘RC4_set_key’ is deprecated: Since OpenSSL 3
precated-declarations]
   45 |    RC4_set_key(&key,40,buf); //se inicai la llave con clave d
      |    ^~~~~~~~~~~
In file included from cifrado.c:4:
/usr/include/openssl/rc4.h:35:28: note: declared here
   35 | OSSL_DEPRECATEDIN_3_0 void RC4_set_key(RC4_KEY *key, int len,
      |                            ^~~~~~~~~~~
cifrado.c:46:4: warning: ‘RC4’ is deprecated: Since OpenSSL 3.0 [-Wde
-declarations]
   46 |    RC4(&key, lTamano ,sArchivote,sArchivote);
      |    ^~
/usr/include/openssl/rc4.h:37:28: note: declared here
   37 | OSSL_DEPRECATEDIN_3_0 void RC4(RC4_KEY *key, size_t len,
      |                            ^~
cifrado.c:51:4: warning: ‘RC4_set_key’ is deprecated: Since OpenSSL 3
precated-declarations]
   51 |    RC4_set_key(&key,40,buf);
      |    ^~~~~~~~~~~
```

Al final se puede visualizar el texto del archivo, tanto el original, cifrado y descifrado

```
┌──(bryan㉿kali)-[~]
└─$ ./cifrado prueba.txt private_key.key
CONTENIDO DEL ARCHIVO ORIGINAL:
──────────────────────────────────────────────────

Este es un texto de prueba

CONTENIDO DEL ARCHIVO CIFRADO:
──────────────────────────────────────────────────

dʌd)4z◆J◆w2◆.◆'◆◆

CONTENIDO DEL ARCHIVO DESCIFRADO:
──────────────────────────────────────────────────

Este es un texto de prueba
```

Conclusión:

Aquí lo que yo pude observar mejor fue la variedad de formas que existen para cifrar, y como se trabaja en conjunto con otras herramientas, esto es lo que se busca en lo que respecta a tener opciones o variedad a la hora de hacer las cosas, una divergencia de pensamiento para no solo quedarte con una sola solución, si no que podemos aplicarla a muchas más.