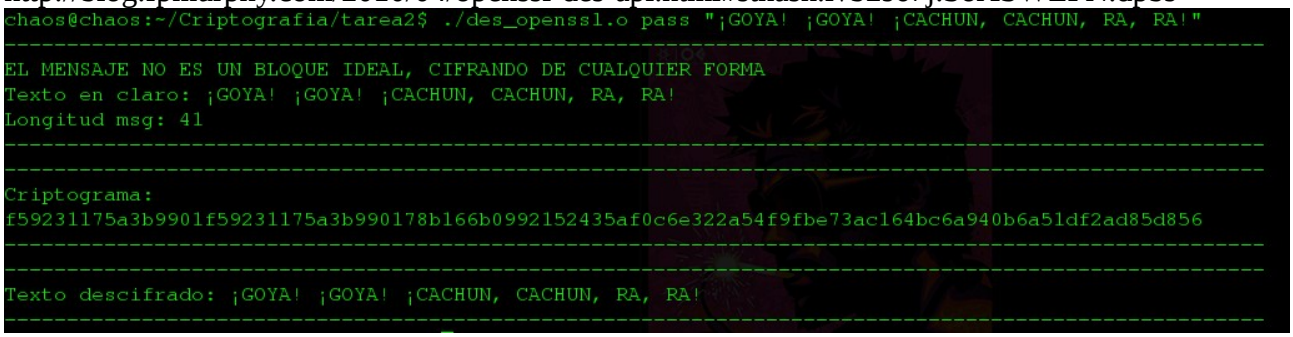
DES
github/truerandom

Modificamos el programa para que pudiera cifrar todo el mensaje basandonos en:
http://blog.fpmurphy.com/2010/04/openssl-des-api.html#sthash.lvb2s67j.SoASWZFN.dpbs



Codigo:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <openssl/des.h>
// Instalar en Debian libssl-dev
// gcc des_openssl.c -o des_openssl.o -lcrypto
// uso: program pass text
//tamanip del buffer
#define BUFSIZE 256

//imprime una linea para separar las secciones
void format(){
	printf("-----------------------------------------------------------------------------------------\n");
}

void main(int argc, unsigned char* argv[]){
	unsigned char in[BUFSIZE], out[BUFSIZE], back[BUFSIZE];
	unsigned char *e = out;
	// Crea una variable llamada key, que es de tipo DES_cblock
	DES_cblock key;
	// Copia 8 bytes de la contraseña dada por el usuario en key
	memcpy( key, argv[1],8);
	// establece la paridad de la llave en impar
	DES_set_odd_parity( &key );
	// Crea una variable llamada keysched, que es de tipo DES_key_schedule
	DES_key_schedule keysched;
	// Copia tantos 0 en las variables como sea su tamanio
	memset(in, 0, sizeof(in));
	memset(out, 0, sizeof(out));
	memset(back, 0, sizeof(back));
```

DES
github/truerandom

```
        // Revisa la paridad de la llave key, y hace la planificación de la llave
        DES_set_key_checked( &key, &keysched );
        /* 8 bytes de texto en claro */
        strcpy(in, argv[2]);
        format();
    //Calculamos la longitud del mensaje
    int longitud = strlen(argv[2]);
        if(longitud%64!=0){
                printf("MSG NO ES UN BLOQUE IDEAL, CIFRANDO DE CUALQUIER
                FORMA\n");
        }
        printf("Texto en claro: %s\n", in);
        printf("Longitud msg: %u\n",longitud);
        format();
        // Cifrado
        int i;
        //Ciframos por bloque
        for(i=0;i<longitud; i+=8){
                DES_ecb_encrypt(&(in[i]),&(out[i]),&keysched,DES_ENCRYPT);
        }
        format();
        printf("Criptograma:\n");
        while (*e) printf("%02x", *e++);
        printf("\n");
        format();
        // Descifrado
        //Desciframos por bloque
        for(i=0;i<longitud; i+=8){
                DES_ecb_encrypt(&(out[i]),&(back[i]),&keysched,DES_DECRYPT);
        }
        format();
        printf("Texto descifrado: %s\n", back);
        format();
}
```

```
chaos@chaos:~/Criptografia/tarea2$ ./des_openssl.o pass "baby, air and light and time and space have noth
ing to do with it and don't create anything except maybe a longer life to find new excuses for "
--------------------------------------------------------------------------------------------------------
EL MENSAJE NO ES UN BLOQUE IDEAL, CIFRANDO DE CUALQUIER FORMA
Texto en claro: baby, air and light and time and space have nothing to do with it and don't create anythi
ng except maybe a longer life to find new excuses for
Longitud msg: 145
--------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------
Criptograma:
4932618b9d1a89111fc348050a19464cdc7d840bf58c5c8af2ab0703a81a467ca56b9c28f3ead39f50760f1dbd6acdd605185760f
347e0ab8cf117510b724bb289d140f3f3b0b8a3fb526905cf48c6758a1db29454b032a32796da5b0fdee567
--------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------
Texto descifrado: baby, air and light and time and space have nothing to do with it and don't create anyt
hing except maybe a longer life to find new excuses for
--------------------------------------------------------------------------------------------------------
```

DES
github/truerandom

Para manejar bloques no ideales el cifrado por bloques realiza lo siguiente:
Muchos de estos algoritmos de cifrado reordenan el texto en patrones particulares, si este no se ajusta exactamente al patron, es necesario agregarle caracteres adicionales para completar los patrones.
Ademas, esto dificulta el criptoanalisis debido a que no es posible conocer la longitud real del mensaje original.
Como desventaja , esto hace al mensaje vulnerable a ataques de oraculo de padding:
Ejemplos:

BitPadding:

```
... | 1011 1001 1101 0100 0010 0111 0000 0000 |
```

Se agrega un unico 1 y despues tantos ceros como sea necesario.

ANSI X.923:

```
... | DD DD DD DD DD DD DD DD | DD DD DD DD 00 00 00 04 |
```

Se agregan bytes cero y el ultimo byte de relleno indica cuantos de estos se ocuparon.

Referencias:

http://blog.fpmurphy.com/2010/04/openssl-des-api.html#sthash.lvb2s67j.SoASWZFN.dpbs
https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation
http://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption_perf/