# Práctica 2: Seguridad Perfecta y Criptografía Simétrica

20 de octubre de 2015

## Índice



**Resumen**

El objetivo de esta práctica es la familiarización con los conceptos de seguridad perfecta, entropía y con los método de cifrado simétrico tomando como referencia el *Data Encryption Standard (DES) y el Advanced Encryption Standard (AES)*. Para ello se procederá a su implementación y al estudio de diversos elementos básicos de su estructura.

## Introducción

Se deberá elaborar una memoria sobre la práctica (en un fichero en formato pdf) que explique detalladamente la realización de todos los apartados con todos sus resultados correspondientes. En la memoria se podrá integrar el código necesario para entender la práctica correctamente. La elaboración de esta memoria es indispensable para la superación de la práctica.

# Problemas

## 1. Seguridad Perfecta

### a. Comprobación empírica de la Seguridad Perfecta del cifrado afín

Comprobar mediante un programa que el método de cifrado afín implementado en la práctica anterior:

1. *Consigue Seguridad Perfecta si se eligen las claves con igual probabilidad.*
2. *No consigue Seguridad Perfecta si las claves no son equiprobables.*

Recordar que la definición de Seguridad Perfecta asume que cada elemento de texto plano se cifra con una clave distinta. Por ello, se deberán utilizar dos métodos distintos de cambio de clave entre un elemento de texto plano y el siguiente: uno elegirá claves de manera equiprobable y el otro no.

El programa tendrá la siguiente interfaz:

`seg-perf {-P | -I} [-i` $file_{in}$`] [-o` $file_{out}$`]`

Explicación de los argumentos:

**-P** se utiliza el método equiprobable.

**-I** se utiliza el método no equiprobable.

En todo caso, la salida consistirá en las probabilidades $P_p(x)$ de los elementos de texto plano, y las probabilidades condicionadas $P_p(x|y)$ para cada elemento de texto plano y de texto cifrado, con el siguiente formato:

$P_p(A) = \%lf$

$P_p(B) = \%lf$

...

$P_p(Z) = \%lf$

$P_p(A|A) = \%lf \;\; P_p(A|B) = \%lf \;\; ... \;\; P_p(A|Z) = \%lf$

$P_p(B|A) = \%lf \;\; P_p(B|B) = \%lf \;\; ... \;\; P_p(B|Z) = \%lf$

...

$P_p(Z|A) = \%lf \;\; P_p(Z|B) = \%lf \;\; ... \;\; P_p(Z|Z) = \%lf$

Donde $\%lf$ representa un double de C impreso de la forma estándar, y las distintas probabilidades condicionadas van separadas por espacios.

En la memoria se comentarán entre otros los siguientes aspectos:

1. *Esquemas de cambio de clave implementados para los modos -P y -I.*
2. *Resultados obtenidos para los dos modos con distintos casos de prueba, en función del tamaño del mensaje cifrado.*

### b. Entropía del idioma inglés (OPCIONAL)

Realizar un estudio de la entropía del inglés, a partir de la definición vista en teoría. Se valorará especialmente el cálculo eficiente de los valores de la variable $P^n$, que representa la distribución de los n-gramas de un texto plano dado. El programa implementado tendrá la interfaz:

`entropia {-n tope} [-i` $file_{in}$`] [-o` $file_{out}$`]`

Donde:

**-n** longitud máxima de los n-gramas.

En la salida se tabularán en líneas diferentes los valores de entropía para distintos tamaños de los n-gramas, desde n=1 hasta lo que indique el parámetro -n, con el formato siguiente:

$H(1) = \%lf$

$H(2) = \%lf$

...

$H(tope) = \%lf$

Comentar en la memoria los resultados obtenidos, así como el método de cálculo de $P^n$. Nota: Es posible que te ayude para los calcular de la entropía las siguientes referencias [1, 2]

## 2. Implementación del DES

### a. Programación del DES

Programa el método de DES de 16 rondas en el modo ECB [3, 4, 5]. Se creará el programa llamado `desECB` que recibirá argumentos de acuerdo con el siguiente esquema:

`desECB {-C | -D -k clave} {-S s} [-i` $file_{in}$`] [-o` $file_{out}$`]`

Explicación de los argumentos:

**-C** el programa cifra

**-D** el programa descifra

**-k** clave de 64 bits: 56 bits + 8 bits de paridad

**-i** fichero de entrada

**-o** fichero de salida

Si la longitud en bits del fichero de entrada no es múltiplo de 64, se añadirán los caracteres necesarios para que lo sea.

Cuando se cifre un texto, el programa generará automáticamente la clave de cifrado y la mostrará en la salida estándar para poder utilizarla en el modo de descifrado. La clave tendrá 56 bits de datos más 8 bits de paridad impar. Los bits de paridad ocuparán las posiciones 8, 16, 24, 32, 40, 48, 56 y 64.

Para facilitar la codificación de al algoritmo se proporciona el fichero [7]. Este fichero contiene los valores numéricos de las 8 *S-boxes* del algoritmo DES y diversas permutaciones que necesitas. Comprueba que el modo ECB no es bueno para mensajes largos, como por ejemplo imágenes:
`http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation`.

### b. Programación del Triple DES (OPCIONAL)

Utilizando los programas implementados en los apartados anteriores, programar el Triple DES en el modo de operación CBC (consultar [6] para los modos de operación en Triple DES). Para ello, se creará un programa con la siguiente interfaz:

`tripleDES {-C | -D} {-k clave -v IV } {-S s} [-i` $file_{in}$`] [-o` $file_{out}$`]`

Se deberá tener en cuenta que la clave especificada se dividirá en las tres claves de cifrado de cada DES, por lo que será una clave de 168 bits con 24 bits de paridad. De nuevo, para los modos -C y -D deberá comprobarse el cumplimiento de dicha paridad cuando corresponda.

## 3. Principios de diseño del DES

### a. El Criterio de Avalancha Estricto (SAC) y el Criterio de Independencia de Bits (BIC)

Estudia los criterios *SAC* y *BIC* en función del número de iteraciones para las cajas $S$ del DES. Explica en qué consiste cada una de estas características del DES y analiza, utilizando la implementación realizada para el apartado anterior, cómo se manifiestan a través del proceso de cifrado DES.

### b. Construcción de *S-boxes* propias (OPCIONAL)

Describe los criterios de diseño seguidos para el diseño de las *S-boxes* del algoritmo DES [8]. Basándote en ellos, construye nuevas *S-boxes* con 6 bits de entrada y 4 bits de salida. Explica los criterios de diseño seguidos en la construcción de estas *S-boxes* y el método seguido para su aplicación.

### c. Estudio de la no linealidad de las S-boxes del DES

Estudiar la no linealidad de las S-boxes del DES y para ello construir un programa que haga las medidas adecuadas pra comprobar tal hecho.

### d. Estudio del Efecto de Avalancha (OPCIONAL)

Estudia experimentalmente el efecto avalancha del algoritmo DES en el bloque y la clave, para cada una de las rondas del algoritmo. Para ello diseña los programas que creas conveniente para probarlos con diferentes bloques y claves para el algoritmo.

### e. Generación de números pseudoaleatorios con DES (OPCIONAL)

Utiliza el DES triple (E-D-E) para generar números pseudoaleatorios tal como se indica en el estándar ANSIX9.17 (consulta [9] para averiguar el algoritmo). Para ello se creará un programa llamado ansix9.17 que se utilizará de esta manera:

`ansix9.17 -n` $n_a$ `-E` $k_1$ `-D` $k_2$ `[-o` $file_{out}$`]`

Explicación de los parámetros:

**-n** Número de bloques de 64 bits a generar.

**-E** Llave de cifrado del DES.

**-D** Llave de descifrado del DES.

El programa generará como salida $n_a$ números pseudoaleatorios de 64 bits.

### f. Comprobación de aleatoriedad de números pseudoaleatorio (OPCIONAL)

Estudiar la eficiencia del método implementado en el apartado anterior para generar números aleatorios, en función del tamaño de la serie temporal, comparándolo con las funciones *rand*() de *C* y las funciones *mpz_urandom*() y *mpz_random*() de la librería GMP.

NOTA: No utilizar las métodos de validación de aleatoriedad monobit, poker y runs definidos en [10] para el DES, ya que son métodos obsoletos (mirar la documentación adicional en la página de moodle de la asignatura).

## 4. Principios de diseño del AES

### a. S-boxes AES (OPCIONAL)

Implementar los algoritmos de Euclides y de Euclides extendido para $GF(2^8)$ con $m(x) = x^8 + x^4 + x^3 + x + 1$ (polinomio irreducible del AES). Utilizando dichos algoritmos, codificar un programa que calcule las S-boxes para el AES, tanto la directa como la inversa. La interfaz será:

`SBOX_AES {-C | -D} [-o` $file_{out}$`]`

Donde:

**-C** calcular la S-box directa.

**-D** calcular la S-box inversa.

Comprobar la corrección de la implementación comparando las S-boxes obtenidas con las reales del AES, que pueden consultarse en las páginas 16 y 22 de [11] y de [12].

### b. Estudio de la no linealidad de las S-boxes del AES

Estudiar la no linealidad de las S-boxes del AES y para ello construir un programa que haga las medidas adecuadas pra comprobar tal hecho.

### c. Estudio de los criterios SAC y BIC

Estudiar los criterios SAC y BIC para las S-boxes del AES mediante un procedimiento análogo al del DES. La interfaz del programa será:

`SAC_BIC_AES {-S | -B} {-C | -D} [-o` $file_{out}$`]`

Donde -C (-D) indica la S-box directa (inversa) y los otros parámetros son iguales a los del caso del DES. En la memoria se compararán los resultados obtenidos para el AES con los del DES, y se explicarán las diferencias en los resultados, así como las causas de dichas diferencias.

### d. Programación del AES (OPCIONAL)

Programa el método de AES de 16 rondas en el modo CBC [12, 4, 5]. Se creará el programa llamados `desCBC` que recibirá argumentos de acuerdo con el siguiente esquema:

`AesCBC {-C | -D -k clave} [-i` $file_{in}$`] [-o` $file_{out}$`]`

### e. Estudio del Efecto de Avalancha (OPCIONAL)

Estudiar el efecto de avalancha en el AES tanto en el cifrado como en la generación de claves, al estilo de como se hizo en la parte anterior para el DES. Por ello, el programa correspondiente tendrá la interfaz:

$$avalancha_{aes}\ \texttt{-K|-B -k clave -b bloque [-o}\ file_{out}\texttt{]}$$

Donde los parámetros funcionan igual que para el DES, sólo que ahora las claves y los bloques son de 128 bits.

En la salida se detallarán los resultados para cada fase de cada ronda. Comentar en la memoria los resultados obtenidos para distintas claves y bloques de entrada.

## Información complementaria

**Plazo de realización y entrega:** La realización será los días 22/10/2015, 29/10/2015, 05/11/2015, 12/11/2015, 19/11/2015 y la entrega el día 25/11/2015 (23:55 horas).

## Bibliografía de referencia

[1] Entropy and Redundancy in English. (`http://people.seas.harvard.edu/~jones/cscie129/papers/stanford_info_paper/entropy_of_english_9.htm`)

[2] Prediction and entropy of printed English C.E. Shannon January 1951. (`http://www.princeton.edu/~wbialek/rome/refs/shannon_51.pdf`).

[3] FIPS 46-3 (`http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf`).

[4] FIPS 81 (`http://csrc.nist.gov/publications/fips/fips81/fips81.htm`).

[5] NIST Special Puplication 800-38A (`http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf`).

[6] NIST 800-20 (`http://csrc.nist.gov/publications/nistpubs/800-20/800-20.pdf`).

[7] DES_tables.c (ver moodle)

[8] Diseño S-Boxes DES: D. Coppersmith. 1994. The Data Encryption Standard (DES) and its strength against attacks. IBM J. Res. Dev. 38, 3, 243-250 y Heys, H.M. and Tavares, S.E. 1995. Avalanche characteristics of substitution-permutation encryption networks, Computers, IEEE Transactions on, vol.44, no.9, pp.1131-1139 (ver moodle).

[9] Menezes, Alfred J. Handbook of applied cryptography Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone. 1997.(`http://www.cacr.math.uwaterloo.ca/hac/`).

[10] FIPS 140-1 (`http://csrc.nist.gov/publications/fips/fips1401.htm`).

[11] FIPS 197: Advanced Encryption Standard (`http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`).

[12] AES según los autores (`http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.36.640`).