Práctica 2: Seguridad Perfecta y Criptografía Simétrica

29 de octubre de 2020

Índice

Resumen

El objetivo de esta práctica es la familiarización el concepto de seguridad perfecta, y 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 por despalzamiento:

Comprobar mediante un programa que el método de cifrado por desplazamiento:

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.*

2. Implementación del DES

a. Programación del DES

Programa el método de DES de 16 rondas en el modo CBC (Cipher-block chaining) [1, 2, 3, 4]. Se creará el programa llamado `desCBC` que podrá recibir argumentos de acuerdo con el siguiente esquema:

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

Tener en cuenta que el bloque que se cifra y se encadena es de 64 bits.

Explicación de los argumentos:

-C el programa cifra

-D el programa descifra

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

-iv vector de incialización

-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 [6]. Este fichero contiene los valores numéricos de las 8 *S-boxes* del algoritmo DES y diversas permutaciones que necesitas. Para depurar el DES te puede servir el apartado "Depurar el DES" en la pagina moodle de la asignatura. Por útimo, comprueba que el modo de operación ECB no es bueno para mensajes largos y estructurados, como por ejemplo imágenes:
`http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation`. Comparalo con la función diseñada "desCBC", explicando porque las diferencias con el modo ECB (Electronic Code-Book). Para depurar los posibles errores del DES podéis utilizar The DES Algorithm Illustrated (`http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm`). .

b. Programación del Triple DES

Utilizando los programas implementados en los apartados anteriores, programar el Triple DES en el modo de operación CBC (consultar [2, 5] para los modos de operación en Triple DES). Para ello, se creará un programa con la siguiente interfaz:

`TDEA-CBC {-C | -D} {-k clave -iv vectorincializacion} [-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. 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 para comprobar tal hecho.

b. Estudio del Efecto de Avalancha

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.

4. Principios de diseño del AES

a. 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 para comprobar tal hecho (mirar la documentación adicional en la página de moodle de la asignatura [7]).

b. Generacion de las S-boxes AES

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 correcta implementación comparando las S-boxes obtenidas con las reales del AES, que pueden consultarse en las páginas 16 y 22 de [9] y de [10].

c. Programación del AES (OPCIONAL)

Programa el método de AES en el modo ECB [10, 3, 4], para claves de 128 bits. Se creará el programa llamados `AESECB` que recibirá argumentos de acuerdo con el siguiente esquema:

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

Información complementaria

Plazo de realización y entrega: La realización será los días 29/10/2020, 05/11/2020, 12/11/2022, 19/11/2020, y la entrega el día 25/11/2020 (23:55 horas) .

Bibliografía de referencia

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

[2] NIST 800-67 (`http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-67r1.pdf`).

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

[4] NIST Special Puplication 800-38A (`http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf`).

[5] NIST 800-20 (`http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-20.pdf`).

[6] DES_tables.c (ver moodle)

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

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