RESUMEN DE CRIPTOGRAFÍA
DEL LIBRO DE MANUEL JOSÉ LUCENA LOPES VERSION 1.00

TEMA 2: CONCEPTOS BÁSICOS

TRABAJOS fundamentales:
- Claude Shannon: "Communicattion Theory of Secrery Systems" (1948)
- Whitfield Diffie y Martin Hellman: "New directions in Criptography" (1976)

CRIPTOLOGÍA: engloba a los términos criptografía y criptoanaálisis

CRIPTOGRAFÍA:
- arte de escribir con clave secreta o de un modo enigmático
- sólo hace referencia al uso de códigos

CRIPTOANÁLISIS:
- Definiciones: técnicas para romper los códigos, descifrar un mensaje sin conocer la llave, NO SE CONSIDERA descubrir un algoritmo secreto

- Tipos:
  - texto claro escogido: se utilizan para descubrir la clave (GSM), se estudian las diferencias a pares entre el mensaje y su criptosistema
  - ataques por la fuerza bruta: prueba y error o vuelta-atrás,....
  - diferencial: pares de mensajes con diferencias mínimas (1 bit)
  - lineal: XOR entre algunos bits de texto claro y del cifrado, obteniendo finalmente un único bit

- Seguridad mínima: el tiempo que pueda tardar en descifrar debe de ser superior al tiempo de vida de la información.

CRIPTOSISTEMA:
- Definición: $D_k(E_k(m)) = m$

  M mensaje o texto claro, C criptograma o mensaje cifrado, K klaves, E funciones de cifrado, D descifrado
- Ejemplo:

  Mensaje: casa; *función E de cifrado*: cambiar "a" por ("e" + k) y "c" por ("d"+ k); *función D de descifrado*: al contrario.

  $E_k(m)$ ==> E_2 (casa) = ["d" +2]["e"+2][s]["e"+2] = [f][g][s][g]= fgsg *es el criptograma*

  $D_k(E_k(m))$ ==> $D_2(E_2(\text{casa})) = D_2(\text{fgsg}) = [f-2][g-s][s][g-2]=$ casa *es el mensaje*

- Claves Débiles: son las claves o llaves que no codifican correctamente. En un buen ciptosistema la cantidad de claves débiles es cero o muy pequeña.

- Tipos:
  - simétricos o de llave privada: misma llave para cifrar o descifrar
  - Asimétricos o de llave pública:
    - doble clave: pública y privada; en muchos casos son intercambiables
    - se usan también para cifrar o codificar las claves simétricas

ESTEGANOGRAFÍA (canales subliminales): consiste en ocultar en el interior de una información, otro tipo de información (cifrada o no). Se usan imágenes o cantidades ingentes de *basura.*

SEGURIDAD:

- sistemas que van a albergar información (amenaza):

    - aislados: no están conectados
    - interconectados

- cuestiones de seguridad:

    - **seguridad física**: salvaguarda los soportes físicos contra terremotos, restricción de acceso físico,.... se usan las políticas de **backup**

    - seguridad de la información: criptografía simétrica (sistemas aislados) o asimétrica frente a observadores no autorizados

    - seguridad en el canal de comunicaciones: muy raro

    - problemas de **autentificación**: se soluciona con funciones **resumen**

    - Problemas de **suplantación**: intrusos por usuarios legales, se usa el **password**

    - No repudio: es fundamental impedir que el emisor niegue su autoría.

TEMA 3: TEORÍA DE LA INFORMACIÓN

CANTIDAD DE INFORMACIÓN (I):

- Definición:
  - medida de la disminución de incertidumbre acerca de un suceso
  - la cantidad de información **es** directamente proporcional al número de posibles estados que éste tenía a priori. La disminución de incertidumbre es proporcional al aumento de certeza.

$$I = -\log_2(P(x)); \text{ siendo } I_{min}= 0 \text{ y } I_{max}= +\infty$$

NOTA: $\log_2 x = [\log_{10}(X) / \log_{10} 2]$

- Ejemplos:
  - un único suceso que ocurra siempre hay 1/1 de posibilidades

    $I = -\log_2(1/1) = 0$ *no hay información útil*
  - un dado de con 6 caras y una tirada, hay 1/6 de posibilidad de acertar: esto es

    $I = -\log_2(1/6) = 2.58$ *escasa información útil*
  - una quiniela de $15 = 3^{15}$, hay 1/14348907 de posibilidad de acertar: esto es

    $I = -\log_2(1 / 14348907) = 23.77$

ENTROPÍA(H) de una variable aleatoria **es** el número medio de bits que necesitaremos para codificar cada uno de los estados de la variable. La cantidad de información asociada al suceso más simple (dos posibilidades), será nuestra unidad a la hora de medir esta magnitud que es el bit. Esto es, cuanto más probable sea un valor individual, aportará menos información cuando aparezca y podremos codificarlo empleando un mensaje más corto

- Definición: $H(V) = -\Sigma^{n}_{i=1} P(x_i) \log_2[P(x_i)]$
- Ejemplos anteriores:
  - único suceso: $H = - (1 * \log_2(1)) = 0$ bits; *siempre ocurre, no necesitamos comunicarlo*
  - dado: $H = - (1/6 * \log_2(1/6) + 5/6* \log_2(1/6)) = \log_2(1/6) = 2.584$ bit; menos de un bit
  - Ejemplo de la moneda (dos posibilidades):

    $H = - (1/2 * \log_2(1/2) + 1/2* \log_2(1/2)) =$ 1 bit; coincide con la definición.

- El método **Huffman** permite transmitir datos en condiciones binarias que se **aproximan bastante** al teórico.

LA ENTROPÍA CONDICIONADA entre dos variables se usa en el análisis de las dos claves (pública y privada) de un algoritmo asimétrico y también en el criptosistema de Shannon.

CRIPTOSISTEMA SEGURO DE SHANNON o criptosistemas ideales:

$I (C_{\text{mesaje cifrado}}, M_{\text{texto claro}}) = 0$

Si la cantidad de información (I) que nos aporta el hecho de conocer el mensaje cifrado (C) sobre la entropía del texto claro (M) vale 0.
La cardinalidad del espacio de claves ha de ser al menos igual que la del espacio de mensajes, esto es, la clave ha de ser tan larga como el mensaje.

- El algoritmo de **Mauborgne y Verman** (1917) utilizaba ésto con **XOR** y **mod 26.**
Inconveniente: la clave era tan larga como el mensaje y había que enviarla por un canal seguro

La **REDUNDANCIA** es una medida de exceso de información. Cuando faltan algunas letras, se puede leer el mensaje por la alta redundancia.

Índice de un lenguaje:

$r_k = H_k(M) / k$; k= longitud de todos los mensajes.
el español tiene 1.4 bits/letra.

Índice ABSOLUTO de un lenguaje:

$R = \log_2 (m)$; R es independiente de la longitud de los mensajes
el español (27 símbolos) es de 4.7 bits/letra

La REDUNDANCIA:

D = R- r en español: D = 4.7 – 1.4 = 3.3 bits/letra

El índice de REDUNDANCIA:

I = D / R en español: I = 3.3 / 4.7 = 0.7021

Aplicaciones relacionadas con la redundancia:

- compresión de datos: trata de eliminar la redundancia dentro de un archivo, considera cada byte como un mensaje

- Códigos de Redundancia Cíclica (**CRC**): permite introducir un campo de longitud mínima en el mensaje, talque éste proporcione la mayor redundancia posible.

Distancia de unicidad: longitud mínima de mensaje de cifrado que aproxima el valor H(K/C) a cero, esto es, la cantidad de mensajes que necesitamos para descubrir la clave. Los criptosistemans de Shanon tienen un unicidad infinita.

CONFUSIÓN (hash) es la sustitución que consiste en cambiar cada ocurrencia de un símbolo en el texto claro por otro. Trata de ocultar la relación entre texto claro y texto cifrado.

DIFUSIÓN (permutaciones) es la transposición que consiste en cambiar de sitio los elementos individuales del texto claro.

TEMA 5: FUNDAMENTOS DE ARITMÉTICA MODULAR

"a" es congruente con "b" módulo "n"
$a \equiv b \pmod n$ si se cumple $a = b + \mathbf{k} * n$ para $K \in \mathbf{Z}$ **(y no a los fraccionarios)**

ALGORITMO DE EUCLIDES (No sirve para cálculo de inversas) es para calcular el m.c.d.

CÁCULO DE INVERSA:

Si mcd(a,b)= 1, entonces primos entre sí.
Si mcd(a,n)= 1, entonces "a" tiene inversa módulo n ; además será el **campo de Galois** GF(n)
Ejemplo: si mcd(3,11) = 1, entonces tiene inversa modula 11;
3 mod 11 ==> 4*3 mod 11 ==> 12 mod 11==> **1** $_{\textbf{elem. simétrico}}$ mod 11
produce el campo de Galois (11)

Función **EULER** :

- si "p" es primo, el nº de residuos modulo "p" o primos relativos con "p" será:

$$\phi(p) = p - 1$$

Ejemplo:
$\phi(11) = 10$ ==> {1,2,3,4,5,6,7,9,10}

- si "n" es NO primo:

$$\phi(m) = \sum 1_{\,0<r<=m \text{ si } mcd(r,m)\,=1}$$

Ejemplo:
$\phi(4) = 2$ y NO ~~$\phi(4) = 3$~~ porque mcd(*2*,4) = 2 esto es, mcd(*2*,4) ≠ 1
==> {1,3}

Teorema **de Femat**: si "a" y "p" son primos entre sí, entonces:
$\mathbf{a^{p-1} = 1 \pmod p}$; si "p" es muy grande, elegir el método de Exponenciación Rápida
Ejemplos:

* $3^{7-1} \equiv 1 \bmod 7$
* $23^{2587} \equiv x \bmod 7$, el exponente: 2587 = (7-1)*431+1 => $23^{6*431+1} \equiv x \bmod 7$ => => $23^{6*431} * 23^{1} \equiv x \bmod 7$ => $23 \equiv x \bmod 7$ ==> x = 2

Algoritmo **EXTENDIDO de Euclides**

TEOREMA CHINO DEL RESTO: sea $p_1, ... p_r$ una serie de primos entre sí, y $n = p_1 * .... * p_r$, entonces el sistemas de ecuaciones en congruencias: $x \equiv x_i \pmod{p_i}$ I = 1...r

Ejemplo: encuentra el menor nº natural que divido por 3 da resto 2. dividido por 5 da 3 y divido por 7 da resto 2.
$x \equiv 2 \bmod 3$; $x \equiv 3 \bmod 5$; $x \equiv 2 \bmod 7$

1) 3,5,7 son primos entre sí
2) mcm (3,5,7) = 105
3) $t_1 = 105/3 = 35$; $t_2 = 105/5 = 21$; $t_3 = 105/7 = 15$
4) según definición:
$35y_1 \equiv 1 \bmod 3$; $21y_2 \equiv 1 \bmod 5$; $15y_3 \equiv 1 \bmod 7$;

5) por otra parte, hallamos el múltiplo INMEDIATAMENTE menor a la base:
$33y_1 \equiv 0 \bmod 3$; $20y_2 \equiv 0 \bmod 5$; $14y_3 \equiv 0 \bmod 7$;
6) restamos las ecuaciones de 4) y 5), resultando:
$2y_1 \equiv 1 \bmod 3$; $y_2 \equiv 1 \bmod 5$; $y_3 \equiv 1 \bmod 7$;
7) luego:
$y_1 = 2$; $y_2 = 1$; $y_3 = 1$
8) por definición:
$x = x_1t_1y_1 + x_2t_2y_2 + x_3t_3y_3 \Longrightarrow x = 2*35*2 + 2*21*1 + 2*15*1 = 233$
9) simplificando:
$233 \equiv 23 \bmod 105$; luego 23 es el menor natural

EXPONENCIACIÓN Y ALGORITMOS DISCRETOS, empleador por los algoritmos de llave pública

Algoritmo de exponenciación rápida (muy similar a al multiplicación "a la russe")
$b = 2^0b_0 + ...+2^nb_n$

$$a^{\,b} = a^{\,2^0b_0 + ...+2^nb_n}$$

Ejemplo: $2^{10368} \bmod 187$
* por Fermat: $2^{10368} = 2^{55*166+138} \equiv 2^{138} \Longrightarrow x \bmod 187 \equiv 2^{138}$ ¡¡¡ muy alto!!!

* por Exponenciación Rápida:

resultado	exponente(cada paso / 2)	base (cada paso * 2)
1	10368	2
1	5184	4
1	2592	8
1	1296	16
1	648	32
1	324	64
1	162	128
1	**81 (81mod 2 ≡ 1)**	**256 mod 187 ≡ 69**
1*69	40	**69*69= 4761 mod 187 ≡ 86**
69	20	86*86=7396 mod 187 ≡ 103
69	10	103*103=10609 mod 187 ≡ 137
69	**5 (5 mod 2 ≡ 1)**	137*137=18769 mod 187 ≡ 69
69*69	2	69*69= 4761 mod 187≡ 86
86 ≡ 4761mod 187	**1(5 mod 2 ≡ 1)**	86*86=7396 mod 187 ≡ 103
86*103 = 8858 mod 187 ≡ ***69 FIN***		

Logaritmos discretos es la inversa a la exponenciación y no existen algoritmos eficientes
$c = \log_b (a) \pmod n \Longleftrightarrow a \equiv b^c \pmod n$

El problema de **Diffie-Hellman** es un ejemplo de su utilidad (logaritmos discretos)

IMPORTANCIA DE LOS NÚMERO PRIMOS: se debe al uso en los algoritmos de llave pública, se debe de obtener un número "n" muy alto y producto de dos primos muy grandes.

ALGORITMOS DE FACTORIZACIÓN

Se puede usar la fuerza bruta: $[2..n^{1/2}]$ pero hay otros métodos más interesantes.

Método Fermat

$n = x^2 - y^2 = a * b$

$n = (x + y)*(x - y) = a * b$

Su coste es exponencial

Método p – 1 de Pollard

Métodos cuadráticos de Factorización: se basan en $x2 \equiv y^2 \pmod n$

- criba cuadrática
- criba del Cuerpo de Números.

TEST DE PRIMALIDAD son métodos probabilísticos que dicen con un alto grado de fiabilidad si un número es primo o compuesto.

- método Lehmann
- método de Rabin-Miller: un número compuesto tiene un 25% de probabilidad para pasar.
- consideraciones prácticas: poner a 1 el bit más y menos significativo, intentar dividirlo con una tabla de primos hasta el número 2000, test de Ravin-Miller 5 veces.

- Primos fuertes:
  - antes los algoritmos asimétricos basabas su potencia en los primos de entero muy grandes
  - Ronald Rivest y Robert Silverman demostraron que no era necesario al existir las cribas cuadráticas o las curvas elípticas y técnicas como la Lenstra.

ANILLOS DE POLINOMIOS

suma $f(x) + g(x) = \sum C_r X^r$ donde $c_i = a_i + b_i$ ó $c_i = a_i$ XOR b_i es Z_2

producto $f(x) * g(x) = \sum C_r X^r$ donde $c_i = \sum a_j * b_k$, tal que $j + k = i$

Un polinomio es **irreducible** si NO puede ser descompuesto como producto de otros dos polinomios de grado positivo

Polinomios en Z_n , en Z_2 los coeficientes pueden ser 0 ó 1 (binarios) y se opera como tales. Además genera un cuerpo finito o campo de Galois $GF(2^n)$ donde n= grado del polinomio.

Ejemplo: polinomios 100101 y 1011 $GF(2^6)$ polinomio irreducible x^6+x+1

* suma= XOR ==>100101 XOR 001011 =101110 ==> $x^6 + x^4 + x^3 + x^2 + x$
* producto ==>100101 * 1011 => 110010111 => $\mathbf{x^8 + x^7} + x^4 + x^2 + x+1$ excede de $\mathbf{GF(2^6)}$
  se aplica x^6+x+1 ==> $x^6 \equiv x+1 \bmod f(x)$; resultando: $[\mathbf{x^2\ x^6}] + [\mathbf{x\ x^6}] + x^4 + x^2 + x+1$
  ==>$[x^2\ \mathbf{(x+1)}] + [x\ \mathbf{(x+1)}] + x^4 + x^2 + x+1 = [x^3 + x^2] + [x^2 + x] + x^4 + x^2 + x+1$ =>
  => $x^4 + x^3 + \sout{x^2 + x^2} + x^2 + \sout{x + x} + 1 = \mathbf{x^4 + x^3 + x^2 + 1}$ **==> 11101**

TEMA 6 – CURVAS ELÍPTICAS

Neal Koblitz y **Vicotr Miller** en 1985

Se aplican en el algoritmos asimétricos pues con claves más cortas se puede alcanzar el mismo nivel de seguridad

CURVAS ELÍPTICAS EN GF(n)
Dados "a", "b" y "n", deben cumplir: $y^2 = [x^3 + ax + b \ \textbf{(mod n)}\ ]$

CURVAS ELÍPTICAS EN $GF(2^n)$
Dados "a", "b" y "n", deben cumplir: $y^2 + xy = x^3 + ax^2 + b$; con $b \neq 0$

TEMA 8 CRIPTOGRAFÍA Y NÚMEROS ALEATORIOS

Los algoritmos de llave pública (Asimétricos) suelen ser empleados en conjunción con algoritmos con llave privada (simétricos); usan la **clave de sesión**, que será diferente cada vez, y la codifican. La única forma de que estas claves de sesión sean seguras es que no exista dependencias entre ellas, esto es, que sean aleatorias.

TIPOS DE SECUENCIAS ALEATORIAS

Secuencias **pseudo**aletorias son secuencias más o menos largas (finitas y periódicas) empleando operaciones aritméticas y/o lógicas.

- secuencias estadísticamente aleatorias: superan los test estadísticos de aleatoriedad.
- un **generador congruencial lineal** será del todo inútil: se ha demostrado que conociendo un bit de cada valor de la secuencia, ésta puede ser recuperada completamente

Secuencias **criptográficamente** aleatorias, se usan en el cifrado de flujos (tema11); ha de cumplir que sea impredecible y que sea computacionalmente intratable el problema de averiguar el siguiente número de la secuencia. El generador necesita una **semilla** totalmente aleatoria para iniciar. Ésto se convierte en ventaja pues se podrán sincronizar cliente-servidor.

Secuencias **totalmente** aleatorias o aleatorias: si no puede ser reproducida de manera fiable. Será suficiente un generador criptográficamente aleatorio alimentado por una **semilla totalmente aleatoria**

GENERADO DE SECUENCIAS ALEATORIAS CRIPTOGRÁFICAMENTE VÁLIDAS

- no se puede usar el reloj del sistema ni nºde serie de los componentes ni los bits de un cd audio.
- fuentes adecuadas:
  - tarjetas digitalizadoras de sonido o video: capta esencialmente ruido térmico
  - unidades de disco: método para medir el tiempo de acceso a la unidad con suficiente precisión
  - **mezcla fuerte** (produce una salida en la que cada bit es una función compleja y no lineal de todos los bits de entrada, modificar un bit en la entrada debería alterar aproximadamente la mitad de los bits de salida; además, la cantidad de bits de salida serán menores que los de entrada) de dos o más fuentes de información:
    - uso de **algoritmos simétricos** como función de mezcla. Des convierte 120 bits a 64
    - uso de **funciones resumen**

ELIMINACIÓN DEL SESGO

Una fuente aleatoria está sesgada cuando hay más unos que ceros, o viceversa.

Para solucionarlo se emplean:

-bits de paridad

-método **Von Neuman** : elimina los pares 00 y 11 de la secuencia. El problema es que no sabemos a priori cuantos bis de información sesgada necesitamos para obtener cada bit de información NO sesgada.

- Uso funciones **Resumen**: se calcula la entropía (H) de una secuencia sesgada, se obtiene los bits reales y nos quedamos con los "n" bits menos significativos

- Generadores aleatorios **criptográficamente seguros (semilla aleatoria y generador pseudoaleatorio)**

- generador **X9.17**:

  - semilla **inicial** de **64** bits
  - genera semillas de 64 bits, utilizando DES y XOR
  - propuesto por **INEN**

- generador **Blum Blum Shub:**

  - usa la propiedad de los primos grandes
    * p = 3(mod 4); q = 3(mod 4)
    * n = p*q
    * escogemos "x" como coprimo(semilla inicial) de "n"
    * x = clave secreta; n = clave pública
    * calculamos:
      * $s_0 = x^2 \bmod n$
      * $s_{i+1} = s^2_i \bmod n$
    * se emplea como salida los bits menos significativos

TEMA 9 CRIPTOGRAFÍA CLÁSICA

Todos los sistemas de cifrado son anteriores a la II Guerra Mundial.
Todos los algoritmos de cifrado clásico son simétricos.

ALGORITMOS CLÁSICOS DE CIFRADO: correspondencia única.

Cifrado MONOalfabético

César por Julio César; 26 letras C = (M+3) mod 26

Ejemplo: casa ==> C = fdvd

Sustitución Afín la clave son dos números secretos a,b menores que N; k = (a,b);

$E_{a,b}(M) = (aM + b) \bmod N$

Ejemplo: casa , k = (2,3), N=26 ==> C = hdnd

Nota: el algoritmo de Cesar es un caso particular de éste K = (1,3)

Tenemos N**!** posibles claves.

Criptoanálisis: las propiedades estadísticas se conservan en el criptograma

Cifrado POLIalfabéticos: la sustitución del carácter varía en función de la posición que ocupe éste en el texto claro.

Cifrado de Vigère

$K = \{k_0 ... k_{d-1}\}$; $E_k = m_i + k_{(i \bmod d)} \pmod n$

Ejemplo:

{	_	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	}
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	

K = { g r a c i a s] N = 27 ;
7 18 1 3 9 1 19 d = 7

C I F R A R

M =	**{**	**a**	**_**	**f**	**e**	**r**	**n**	**a**	**n**	**d**	**o**	**_**	**y**	**_**	**a**	**_**	**m**	**a**	**r**	**}**								
		1	0	6	5	18	14	1	14	4	15	0	25	0	1	0	13	1	18								
		g	r	a	c	i	a	s	g	r	a	c	i	a	s	g	r	a	c	~~i~~	~~a~~	~~s~~						
	+	7	18	1	3	9	1	19	7	18	1	3	9	1	19	7	18	1	3	~~9~~	~~1~~	~~19~~						
		8	18	7	8	**27**	15	20	21	22	16	3	**34**	1	20	7	**31**	2	21									

m o d 27

		8	18	7	8	**0**	15	20	21	22	16	3	**7**	1	20	7	**4**	2	21								
C ==	**{**	**h**	**r**	**g**	**h**	**_**	**o**	**t**	**u**	**v**	**p**	**c**	**g**	**a**	**t**	**g**	**d**	**b**	**u**	**}**								

D E S C I F R A R

C ==	**{**	**h**	**r**	**g**	**h**	**_**	**o**	**t**	**u**	**v**	**p**	**c**	**g**	**a**	**t**	**g**	**d**	**b**	**u**	**}**								
		8	18	7	8	0	15	20	21	22	16	3	7	1	20	7	4	2	21									
		g	r	a	c	i	a	s	g	r	a	c	i	a	s	g	r	a	c	~~i~~	~~a~~	~~s~~						
	-	7	18	1	3	9	1	19	7	18	1	3	9	1	19	7	18	1	3	~~9~~	~~1~~	~~19~~						
		1	0	6	5	**-9**	14	1	14	4	15	0	**-2**	0	1	0	**-14**	1	18									

m o d 27

		1	0	6	5	**18**	14	1	14	4	15	0	**25**	0	1	0	**13**	1	18									
M =	**{**	**a**	**_**	**f**	**e**	**r**	**n**	**a**	**n**	**d**	**o**	**_**	**y**	**_**	**a**	**_**	**m**	**a**	**r**	**}**								

Cifrados por Sustitución HOMOFÓNICA: cada carácter del texto claro, es sustituido en el criptograma por uno o más caracteres, dependiendo de la frecuencia de su aparición. En la práctica, NO se usa pues presenta demasiados inconvenientes.

Cifrados de TRANSPOSICIÓN: escitalo, bastón cilíndrico. NO se sustituyen unos por otros, sino que cambia su orden dentro del texto.

Ejemplo: "la casa de la pradera" con n= 6 y llave {5,4,3,1,2,6}

1	2	3	4	5	6
l	a		c	a	s
a		d	e		l
a		p	r	a	d
e	r	a	∅	∅	∅

"ac las eda larpa daer"

El criptoanálisis es un estudio de pares y tripletas de símbolos en el lenguaje.

MÁQUINA ENIGMA:

Arthur Scherbius

Ciclómero: descifrar mensajes

Otras máquinas similares: **PURPLE, RED, SIGABA**

TEMA 10 CIFRADOS POR BLOQUES

CIFRADO DEL PRODUCTO consiste en trocear el mensaje en **bloques de tamaño fijo** y aplicar la función de cifrado a cada uno de ellos. Se usa combinado a la **confusión** y **difusión.**

Redes de FEISTEL dividen un bloque en dos mitades, se produce un cifrado iterativo: la salida de cada ronda es la entrada de la siguiente.
Lo usan: **DES, Lucifer, FEAL, CAST, Blowfish,**

Cifrados con Estructura de Grupo: el algoritmo criptográfico debe de **carecer** de esta propiedad.
$\forall k_1, k_2 \ \exists k_3 \ / E_{k2}(E_{k1}(M)) = E_{k3}(M)$

S-Cajas (Hash): generalmente, cuanto más grandes sean las S-cajas, más resistente ser el algortimo

DES ==> **8** S-Cajas de **6 * 4** bits

CAST => - **6** S-Cajas de **8 * 32** bits,
- codifica bloques de 64 bits con claves de 64 bits
- sólo la fuerza bruta puede atacarlo

ALGORITMO DES:

- es **seguro** a partir de claves de **256** bits
- se basa en **Lucifer**
- codifica **bloques de 64 bits empleando claves de 56 bits** (en principio 64 bits)
- utiliza **16** rondas **Feistel**: 16 subclaves de 48 bits, desplaza 2 bits a la izquierda excepto 1, 2 9 y 16

claves débiles: 16
claves semidébiles: generan 2 valores diferentes K_i cada uno de los cuales aparece 8 veces.
Variantes DES:

- DES **Múltiple:** aplica DES varias veces con diferentes claves. El más común es el **TripleDES** con **DOS** claves $C = E_{k1}(E^{-1}_{k2} (E_{k1}(M)))$

- DES con SUBCLAVES independientes: emplea subclaves diferentes en cada una de las 16 rondas. La clave inicial es de 768 bits (48*16) . NO presenta un avance sustancial.

- DES generalizado emplea "n" trozos en cada ronda en vez de dos. Pierde seguridad.
- DES con S-Cajas Alternativas: NO se han encontrado S-Cajas mejores que las propias.

Algoritmo IDEA:

- trabaja con **bloques de 64 bits**, emplea **claves de 128 bits**
- genera **52 subclaves de 16 bits** , **4 bloques de16** bits en **8 rondas**
- No presenta claves débiles.
- la longitud de clave hace imposible el ataque de fuerza bruta
- operaciones: XOR, suma módulo 2^{16}, producto módulo $2^{16} + 1$

Algoritmo Rijndael – AES:

- NO posee estructura de red Feistel.
- es el primer algoritmo hecho por la comunidad mundial criptológica.
- longitud de **clave y bloque variables**, comprendidas **entre 128 y 256 bits**
- tiene 4 funciones invertibles, formando 3 capas:
    - capa mezcla lineal con funciones Desplazar Fila y Mezclar Columnas
    - capa NO lineal: ByteSub (**1 S-Caja 8*8** se aplica a cada byte)
    - capa de adición de clave: XOR
- no posee claves débiles y actualmente es el más seguro

MODOS DE OPERACIÓN PARA ALGORITMOS DE CIFRADO POR BLOQUES

ECB:

- subdivide la cadena que se quiere codificar en bloques del tamaño adecuado y se cifran todos con la misma clave
- a favor: adecuado para codificar BBDD o ficheros aleatorios.
- en contra:
    - información redundante
    - se pueden cambiar los bloques y alterar mensajes son conocer la clave.

CBC:

- incorpora un mecanismo de retroalimentación, mediante XOR
- se emplea un bloque aleatorio como **vector de inicialización**
- NO empieza a codificar hasta que no empieza a transmitir

CFB:

- muy similar a CBC
- permite codificar la información en unidades inferiores del bloque, manteniendo un nivel adecuado de seguridad

Otros:

- algoritmo **OFB** que incorpora el sincronismo entre origen y destino

CRIPTOANÁLISIS DIFERENCIAL Biham y Shamir. Pares

CRIPTOANÁLISIS LINEAL Mitsuru Matsui. XOR

TEMA 11 CIFRADOS DE FLUJO

TIPOS DE GENERADORES DE SECUENCIA

Generadores SÍNCRONOS

- el receptor y emisor están sincronizados para que el texto pueda descifrarse.
- se pierde o añade algún bit, el resto de mensaje será imposible de descifrar
- **función resumen**: evita que un atacante introduzca cambios

Generadores Asíncronos o AUTO-sincronizados:

- son **más resistentes** a ataques que los síncronos
- la secuencia generada es función de una semilla, más una cantidad fija de los bits de la propia secuencia.
- resiste a la pérdida, alteración o inserción de información
- mediante la **dispersión** resisten a ataques basados en la redundancia del texto claro.

FSR REGISTROS DE DESPLAZAMIENTO RETROALIMENTADOS

FSRL registros de desplazamiento retroalimentados **LINEALES**: generan secuencias con periodos muy grandes y con buenas propiedades estadísticas; pueden ser implementados por hardware. Operaciones: desplazamiento, suma módulo 1

FSRNL registros de desplazamiento retroalimentados **NO lineales:** tiene mejores condiciones que los lineales. Operaciones: desplazamiento, suma módulo 1 y una función **booleana**

OTROS:

- RC4: Ron Rivest para RSA. Características:
- Genera secuencias de un byte y utiliza:
  - **claves de longitud variable**
  - **1 S-Caja de 8*8** ==> almacena una permutación de {0..255}
- es inmune al criptoanálisis diferencial y lineal
- puede poseer claves débiles

- SEAL: Phil Rogaway y Don Coppersmith para IBM. Características:
- eficiente en computadoras de 32 bits
- NO se basa en un sistema lineal de generación, sino que define una **familia de funciones pseudoaleatorias.**
- **claves de 160 bits**
- se basa en SHA y SHA-1, según revisiones

TEMA 12 ALGORITMOS ASIMÉTICOS DE CIFRADO

Emplean longitudes de claves mayores, al **menos de 1024** bits
Introducidos por **Diffie y Hellman** sobre los 70

APLICACIONES DE LOS ALGORITMOS ASIMÉTRICOS

Protección de la información o Confidencialidad:
- la llave pública está en un servidor y sirve para codificar el mensaje a enviar
- la llave privada está en el receptor y sirve para decodificar

Autentificación
- el emisor cifra con la clave pública del receptor y lo envía
- el emisor cifra el resumen del mensaje con su clave privada (no con la pública)
- el receptor descifra el mensaje.
- el receptor descifra el resumen con la clave pública del emisor, además el receptor regenera el resumen del mensaje recibido y compara su resumen generado con el enviado

ALGORITMO **RSA**:
- **Ronald Rivest, Adi Shamir y Leonard Adleman**
- se utilizó en la primeras versiones de **PGP**
- se basa en la dificultad para **factoriza**r grandes números: n = p*q; si "p" ó "q" fueran compuestos el algoritmo no funcionaría
- claves **débiles: siempre** hay mensajes que quedan por codificar, se cual sea el valor de "n"
- las **claves** deben de ser de 768 bits al menos, se recomienda no inferiores a 1024 bits
- **Twinkle**: máquina capaz de factorizar muy rápidamente.

- ataques intermedio:
  - el emisor solicita al receptor su clave pública, sin embargo, el atacante le envía su clave pública y se queda con la pública del receptor.
  - el emisor transmite el mensaje y lo descifra el atacante; lo cifra con la clave pública del receptor y se lo envía (modificado o no).

- Ataques de texto en claro escogido: cuando el emisor firma (su privada) y codifica (pública) empleando el mismo par de llaves. Se debe usar **siempre** el **resumen** para **firmar**
- Ataques módulo común: calculados "p" y "q", se generan muchas llaves con los mismos primos, puede ser atacada. Se evitará generando diferentes primos para cada par de llaves.
- Ataques de Exponente Bajo
- Firmar y decodificar: nunca se debe codificar el mensaje y luego firmarlo, aunque sea un resumen

OTROS Asimétricos:
- **Diffeie-Hellman** (primos)**:** no necesita llaves públicas en el estricto sentido
- **ElGamal** (primos): se utilizaba en un principio para producir firmas digitales
- **Rabin** (raices): después de descifrar, produce 4 posibles mensajes en claro
- **DSA:** variante de ElGamal, para NIST

PROTOCOLOS SSL Y TLS: liberan a las aplicaciones informáticas de las operaciones criptográficas.
- el TSL es una nueva versión del SSL
- comunicación:
  - saludo o handshacking: identificación con el X.509
  - comunicación: intercambio

TEMA 13 MÉTODOS DE AUTENTIFICACIÓN

Tipos: mensaje ==> **firma digital,** usuario ==> contraseña, dispositivo ==> llave electrónica

FIRMAS DIGITALES O SIGNATURAS. FUNCIONES RESUMEN (MDC)

Autentificar información es confirmar o asegurar que un mensaje proviene de un emisor verdadero. Se realiza a través de las funciones resumen (MDC).Además la función resumen (MDC) debe de cumplir:

- r(m) es de longitud fija, independientemente de m
- dado m, es fácil calcular r(m)
- dado r(m), es computacionalmente intratable recuperar m
- dado m, es computacionalmente intratable obtener m' tal que r(m) = r(m')

Longitud adecuada para la firma o signatura: se recomienda al **menos de 128** bits, siendo **160** bits el más usado.

Estructura de una función Resumen (**MDC**): se basa en la compresión, se suele incluir en alguno de los bloques del mensaje

Algoritmo **MD5**: **Ron Rivest**. Se incluía en las primeras versiones de PGP. Emplea: **4** registros y **512 bits de entrada** contra **128 bits de salida**

Algoritmo **SHA-1** por **NSA**. Incluido en DSS. Produce **firmas de 160** bits a partir de **bloques de 512** bits. **Emplea 5 registros de 32 bits**.

Se diferencia con SHA (antiguo) por el desplazamiento a la izquierda.

Funciones de Autentificación de Mensaje (**MAC**): emplea una clave secreta conocida por emisor y receptor para calcular la integridad del mensaje, lo más común es cifrar el mensaje con el algoritmo simétrico **CBC**

Autentificación de DISPOSITIVOS por desafío

Autentificación de USUARIO mediante CONTRASEÑAS en un terminal seguro distinguimos:

- el usuario carece de acceso a los archivos del sistema (cajero automáticos)
- el sistema permite entrar.

Ataques mediante DICCIONARIO son elecciones de clave poco afortunadas. Se emplea la **sal:** conjunto de bits aleatorios que se añaden a la palabra clave antes de calcular su firma; en el fichero de claves se almacenará junto a la firma o signatura.
Consejos para no ser atacados: no se deben de escribir, complejas y al menos de 8 letras, fáciles de recordar, deben de ser modificadas con frecuencia

Dinero electrónico. Protocolos:

- Crip. Simétrica: NetBill y NetCheque
- Crip. Asimétrica: CAFE, ECasn, NetCash, CyberCash, iHP, Anonymous Cards

No basados en Cript: ISN, Compuserve, FIRST VIRTUAL

Certificados X.509

- consiste en una llave pública y un identificados, firmados por una **autoridad de certificación.**
- mecanismos:
  - se envía la clave pública a la autoridad de certificación
  - el autorizador envía las dos claves, en este caso, éste podrá descifrar nuestros mensajes.

TEMA 15 SEGURIDAD EN REDES

El protocolo TCP/IP se ha erigido como estándar

Redes internas LAN:

- usualmente a un edificio
- riesgos mínimos:
    - posibles pérdidas de información, se minimizan con **copias de respaldo**
    - uso inadecuado del sistema por parte de los usuarios.
- control: protocolo de autentificación tipo **Kerberos**, deshabilitación dinámica de conexiones de red no utilizadas, verificador del identificador único de la tarjeta de red concreta.
- WLAN:
    - ondas de radio
    - proporciona un nivel deficiente de seguridad

Redes Externas:

- uso de cortafuegos entre la red local y la red interna
- tipos de peligro:
    - ataques indiscriminados
    - ataques a medida: hackers

Intranets: redes externas que se comportan como redes internas mediante protocolos criptográficos y de autentificación.

TEMA 16 HACKERS

Términos:

- **crackers**: copia ilegal de software
- **phreakers**: contra las compañías telefónicas
- **hackers**: se infiltran en sistemas informáticos

PROTOCOLO TCP/IP. DEMONIOS Y PUERTOS

Un computador con TCP/IP puede establecer comunicaciones simultáneamente a través de los **puertos** que se comportan como los canales de un televisor.

Un demonio es un programa que escucha a través de un puerto a la espera de establecer comunicaciones. Los errores y bugs del programa son aprovechados por los harkers

Desbordamientos de Buffer: los datos de entrada pueden ser mayores y ocasiona el abortamiento del demonio, ésto es aprovechado por el hacker para entrar

Suplantación de usuarios

Borrado de huellas: se emplean usuarios más modestos

Ataques Pasivos son fallos en los navegadores de internet

Ataques coordinados o Denegación de Servicio (**DoS**): no consigue robar información pero si paralizar el servidor ya que por cada solicitud se reserva una determinada memoria.

zombi es una computadora que contribuye al ataque de forma inadvertida para sus usuarios

COMO PROTEGERSE:

- sólo información necesaria en los computadores que salen al exterior
- instalación de demonios con la versión actualizada y reciente
- software criptográfico: claves al menos de **128 bits**
- cambiar contraseñas,.....
- archivos de registro: examinando las entradas sospechosas

TEMA 17 VIRUS

Virus: cualquier programa capaz de infiltrarse en un sistema y ejecutarse sin que el usuario tenga noticia

Tipos:

- gusano: capacidad de propagarse y menos destructivo
- bombas lógicas: capaces de destruir y menos de propagación
- troyanos: envoltorio atractivo e interior destructivo

Métodos de contagio:

- sectores de arranque
- red
- multiplataforma: macros
- ...
- falsedades:
  - leer correo electrónico

Fases:

- letargo
- destrucción