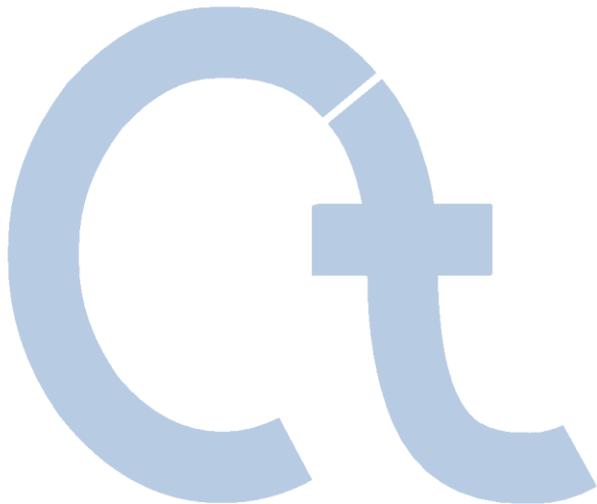


CRIPTOLOGÍA CON CRYPTOOl v1.4.30



Introducción a la
Criptografía y al Criptoanálisis

Alcance, Tecnología y Futuro de CrypTool

Prof. Bernhard Esslinger y el equipo de CrypTool, Julio 2010

www.cryptool.org
www.cryptool.com
www.cryptool.de
www.cryptool.es
www.cryptool.pl

Contenido (I)

I. CrypTool y Criptología – Visión General

1. Definición y relevancia de la Criptología
2. El Proyecto CrypTool
3. Ejemplos de métodos clásicos de cifrado
4. Conocimientos sobre el desarrollo de la criptografía

II. Características de CrypTool

1. Visión General
2. Ejemplos de Interacción
3. Desafíos para los desarrolladores

III. Ejemplos

1. Cifrado con RSA / Test de primalidad / Cifrado Híbrido y certificados digitales
2. Visualización de firma digital
3. Ataque al cifrado RSA (modulo N demasiado pequeño)
4. Ánalisis del cifrado de la PSION 5
5. Claves DES débiles
6. Localizar información de la clave (“clave NSA”)
7. Ataque a la firma digital por localización de colisiones hash
8. Autentificación en un entorno cliente-servidor
9. Demonstración de un ataque de canal lateral(en un protocolo de cifrado híbrido) (...)



Contenido (II)

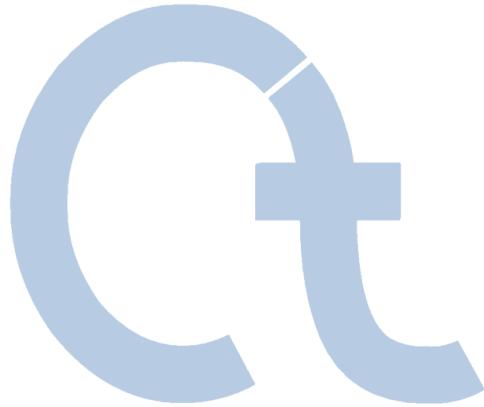
III. Ejemplos

- 10.** [Ataque RSA utilizando reducción de retículos \(Lattice Reduction\)](#)
- 11.** [Análisis de aleatoriedad con visualización 3-D](#)
- 12.** [Secreto Compartido \(Teorema Chino de los Restos \(CRT\) / Shamir\)](#)
- 13.** [Implementación del CRT en Astronomía](#)
- 14.** [Visualización del cifrado utilizando ANIMAL](#)
- 15.** [Visualización del AES](#)
- 16.** [Visualización del cifrado Enigma](#)
- 17.** [Visualización de E-mail seguro con S/MIME](#)
- 18.** [Generación de un código de autentificación de un mensaje \(HMAC\)](#)
- 19.** [Demo Hash](#)
- 20.** [Herramienta de aprendizaje de teoría de números y cifrado asimétrico](#)
- 21.** [Suma de puntos en curvas elípticas](#)
- 22.** [Medidor de calidad de contraseñas](#)
- 23.** [Ánalisis por Fuerza Bruta](#)
- 24.** [Escítala/Rail Fence](#)
- 25.** [Cifrado Hill/Ánalisis Hill](#)
- 26.** [Ayuda online de CrypTool](#)

IV. Proyecto / Perspectiva / Contacto



Contenido



- I. **CrypTool y Criptología – Visión General**
- II. Características de CrypTool
- III. Ejemplos
- IV. Proyecto/Perspectiva/Contacto

Relevancia de la Criptografía

Ejemplos de Uso de la Criptografía

- Cajeros automáticos, transferencias entre bancos
- TV por Satélite, TV de pago
- Sistemas inmovilizadores en coches
- Gestión de Derechos Digitales (DRM)
- Tarjetas telefónicas, teléfonos móviles, controles remotos
- Dinero electrónico, banca electrónica, correo electrónico seguro
- La Criptografía no está limitada a las empresas, diplomacia o a los militares. La Criptografía es una caracterizada ciencia matemática.
- Un gran cambio en la criptografía empezó con la generalización del uso de Internet
- Para las empresas y los gobiernos es importante que los sistemas sean seguros y

... ¡que los usuarios (clientes, empleados) tengan un cierto entendimiento y conciencia sobre la seguridad en TI!



Definición: Criptología y Criptografía

Criptología (*del Griego kryptós, “escondido”, y lógos, “palabra”*) es la ciencia de las comunicaciones seguras (generalmente secretas). Esta seguridad se obtiene con usuarios legítimos, el transmisor y el receptor, siendo capaz de transformar la información en un código utilizando una clave – por ejemplo, una parte de la información solamente conocida por ellos. Aunque el código es inescrutable y muy a menudo inolvidable para cualquiera con su clave secreta, el receptor autorizado podrá descifrar el código para recuperar la información escondida o verificar que fue enviado probablemente por alguien que posee la clave.

Criptografía al principio se preocupaba de proporcionar confidencialidad para los mensajes escritos. Sin embargo, sus leyes se aplican igualmente bien para asegurar un flujo de datos entre ordenadores o para cifrar señales televisivas. ... Hoy, las ciencias (matemáticas) modernas de criptología no sólo contienen mecanismos para cifrar sino también para la integridad, firmas electrónicas, números aleatorios, intercambio seguro de claves, recipientes seguros, voto electrónico y dinero electrónico, y también ha conseguido convertirse en una gran variedad de aplicaciones en la vida moderna.

Fuente: Britannica (www.britannica.com)

Una definición similar se puede encontrar en Wikipedia: <http://es.wikipedia.org/wiki/Criptología>

Criptografía – Objectivos

■ Confidencialidad

- La información prácticamente no puede ser accesible o revelada a individuos, entidades o procesos desautorizados.

■ Autentificación

- La autentificación asegura que los usuarios se han identificado y que sus identidades se han verificado apropiadamente.

■ Integridad

- La integridad asegura que los datos no se han alterado o destruido de una forma no autorizada.

■ No-Repudio

- El principio de que, después de todo, se puede probar que los participantes de una transacción realmente la autorizan y que no pueden negar de ninguna forma su participación.

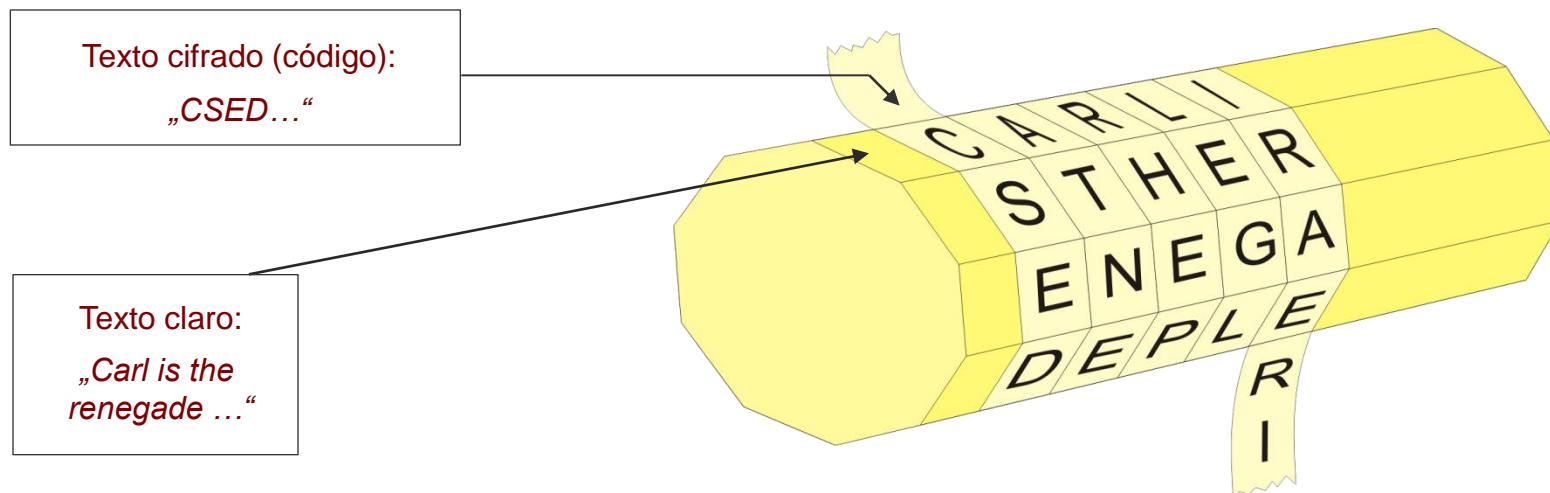
El Proyecto CrypTool

- Origen en un programa de concienciación de un banco (Capacitación Empresarial)
→ **Concienciación para empleados**
- Desarrollado en cooperación con universidades (mejorando la educación)
→ **Enfoque didáctico y orientado a estándares**
 - 1998 **Inicio del proyecto** – el esfuerzo de más de 40 años-hombre desde entonces
 - 2000 CrypTool disponible como **software libre**
 - 2002 CrypTool en el **CD-ROM-Ciudadano de la BSI** (Agencia Alemana de Seguridad de la información)
 - 2003 CrypTool se convierte en **Código-Abierto** – Soporte por Universidad de Darmstadt (Prof. Eckert)
 - 2007 CrypTool disponible en alemán, inglés, español y polaco
 - 2008 Inicio de versiones .NET y Java – Mantenidas por la Univ. de Duisburg (Prof. Weis) y SourceForge
 - 2010 CT1 disponible en su quinto idioma, serbio. Preparando versiones .NET y Java para ser lanzadas
- **Galardones**
 - 2004 TeleTrusT (TTT Förderpreis) 
 - 2004 NRW (IT Security Award NRW)  Ministerium für Innovation, Wissenschaft, Forschung und Technologie des Landes Nordrhein-Westfalen 
 - 2004 RSA Europe (Finalista del European Information Security Award 2004) 
 - 2008 "Selected Landmark" en la iniciativa "Germany – Land of Ideas" 
- **Desarrolladores**
 - Desarrollado por gente de empresas y universidades en distintos países
 - Miembros adicionales del proyecto o códigos útiles siempre se aprecian (actualmente existen alrededor de 50 personas trabajando sobre el universo CrypTool).

Ejemplos de la primera Criptografía (1)

Métodos de cifrado antiguos.

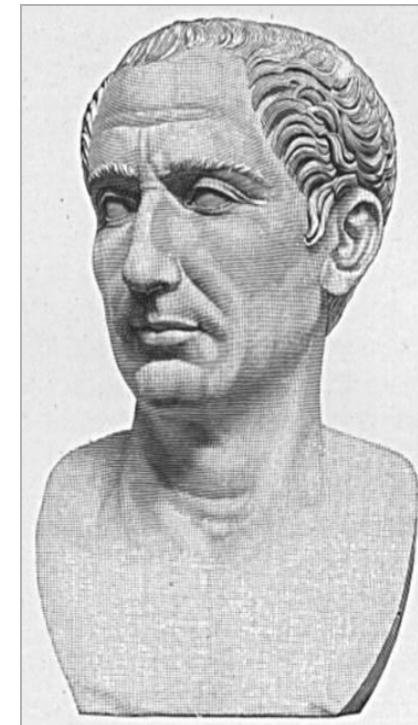
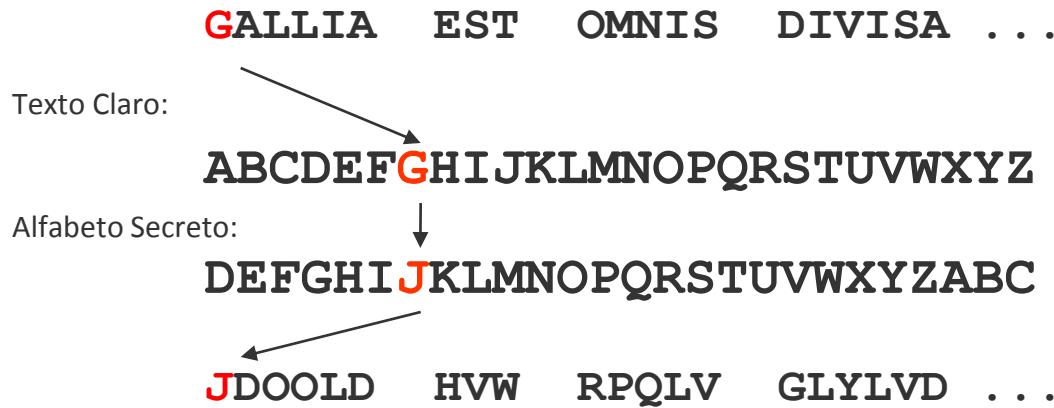
- **Tatuajes en la cabeza de un esclavo cubierto por el cabello**
- **Atbash** (sobre 600 A.C.)
 - Lenguaje secreto Hebreo, alfabeto invertido
- **Scytale de Sparta** (500 A.C.)
 - Descrito por el historiador/autor Griego Plutarco (45 - 125 A.C.)
 - Dos cilindros (varas de madera) con igual diámetro
 - Transposición (los caracteres del texto claro se reordenan)



Ejemplos de la primera Criptografía (2)

Cifrado Simétrico del César

- **Cifrado César** (Julius Caesar, 100 - 44 A.C.)
- Código de sustitución simple



- **Ataque:** Análisis de frecuencias (distribución típica de caracteres)

Presentación con CrypTool mediante los siguientes menus:

- Animación: „Procedimientos Idiv.“ \ „Visualización de algoritmos“ \ „Cesar“
- Implementación: „Cifrar/Descifrar“ \ „Simétrico (clásico)“ \ „Cesar / Rot-13“

Ejemplos de la primera Criptografía (3)

Cifrado Simétrico de Vigenère (Cifrado de sustitución polialfabética)

- **Cifrado Vigenère** (Blaise de Vigenère, 1523-1596)
- Cifrado con una palabra clave utilizando una tabla clave
- Ejemplo:
Palabra clave: **CHIFFRE**
Cifrado: **VIGENERE** resulta **XPOJSVVG**
- El carácter (V) del texto claro se reemplaza por el carácter en la fila correspondiente y en la columna de la primera palabra de la palabra clave (c). El siguiente carácter del texto claro (I) se reemplaza por el carácter en la fila correspondiente y en la columna de la siguiente letra de la palabra clave (h), y así sucesivamente.
- Si se han utilizado todos los caracteres de la palabra clave, entonces el siguiente carácter de la palabra clave es la primera letra de la palabra clave.
- **Ataque** (por el test Kasiski): Pueden darse combinaciones de textos claros con idénticos textos cifrados. La distancia de éstos patrones se pueden utilizar para determinar la longitud de la clave.

Un análisis de frecuencia tradicional se puede utilizar para determinar la clave.

Carácter de la clave

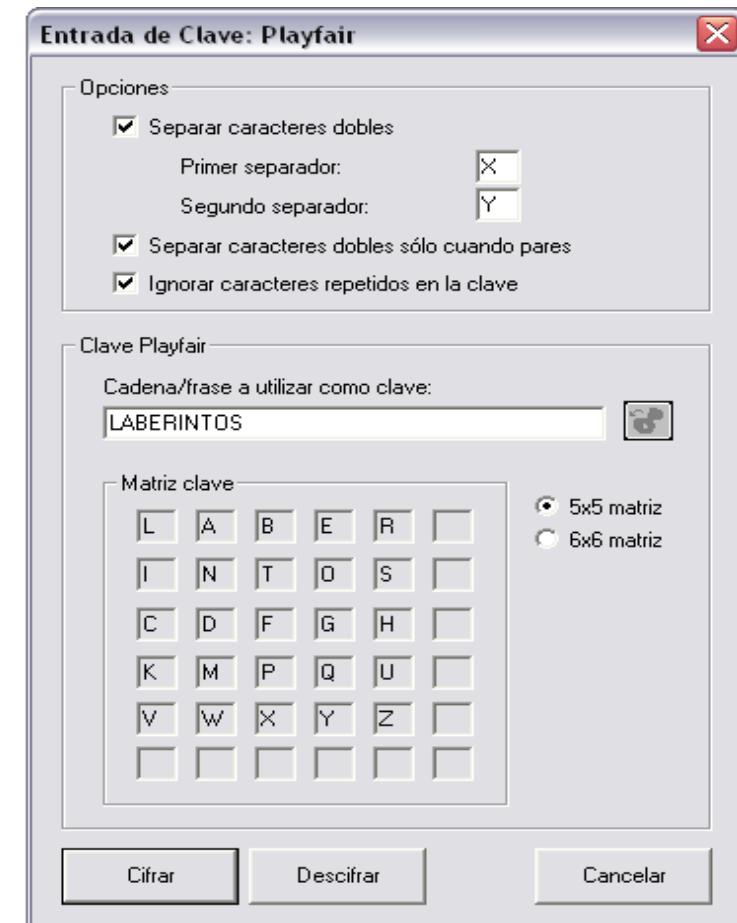
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
N	N	O	E	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	W	X	Y

Tableau carré, dit « Carré de Vigenère »

Ejemplos de la primera Criptografía (4)

Otros métodos de cifrado simétricos

- **Sustitución Homofónica**
- **Playfair** (inventado en 1854 por Sir Charles Wheatstone, 1802-1875)
 - Publicado por el Baron Lyon Playfair
 - Sustitución de un par de caracteres por otro basado en una matriz cuadrada de letras
- **Transferencia de páginas de libro**
 - Adaptación de la Libreta de un sólo uso (OTP)
- **Rejilla giratoria** (Fleissner)
- **Cifrado por permutación**
 - „Doble Dado“ (trasposición de columna doble)
(Trasposición / muy efectiva)



La Criptografía en Tiempos Modernos

Desarrollo de la Criptografía en los últimos 100 años hasta 1970

Métodos Clásicos

- Todavía se utilizan actualmente .
(ya que no todo lo puede hacer un ordenador...)
- Y sus principios de **transposición** y **sustitución** son un gran apoyo para el diseño de algoritmos modernos: la combinación de operaciones simples (un tipo de cifrado múltiple, también llamado cifrado en cascada), a nivel de bit, cifrado en bloque, ciclos.

El cifrado se vuelve

- más **sofisticado**,
- **Mecanizado o computarizado** y
- Permanece **simétrico**.

Ejemplos de la Primera Mitad del S. XX

Máquinas de cifrado mecánico (máquinas de rotores)

Cifrado Enigma (Arthur Scherbius, 1878-1929)

- Se han utilizado más de 200000 máquinas en la II Guerra Mundial.
- El cilindro giratorio elige las causas por las que cada carácter del texto se cifra con una nueva permutación.
- La oficina de cifrado polaca descifró el sistema Enigma prebélico ya en 1932.
- Código roto por un esfuerzo masivo por parte de expertos en criptografía (unas 7000 personas en Reino Unido) con máquinas de descifrado, Enigmas originales capturadas o interceptando comunicados de estado diarios (p.ej. comunicados meteorológicos).
- **Consecuencias de este exitoso criptoanálisis:**

"En general, el exitoso criptoanálisis del cifrado enigma tuvo una ventaja estratégica, que jugó un papel significante para ganar la guerra. Algunos historiadores afirman que el descifrado del código enigma acortó la guerra varios meses o incluso un año."

(traducido de http://de.wikipedia.org/wiki/Enigma_Maschine - Marzo 6, 2006)



Criptografía – Conceptos Importantes (1)

■ **Principio de Kerckhoffs** (establecido en 1883)

- Separación del algoritmo (método) y la clave

p.ej. Cifrado César:

Algoritmo: “Alfabeto desplazado un cierto número de posiciones a la izquierda”

Clave: El “cierto número de posiciones” (César por ejemplo)

- Principio de Kerckhoffs :

El secreto permanece en la clave y no en el algoritmo, es decir, “No hay seguridad por oscuridad”

■ **Libreta de un sólo uso – Shannon / Vernam**

- Demostrado teóricamente seguro, pero no es útil en la realidad (sólo el teléfono rojo)

■ **Conceptos de Shannon : Confusión y Difusión**

- Relación entre M, C y K tiene que ser tan compleja como sea posible (M=mensaje, C=código, K=clave)
- Cada carácter del texto cifrado debe depender de tantos caracteres del texto claro como de la clave de cifrado.
- „Efecto Avalanche“ (una pequeña modificación tiene un gran impacto)

■ **Función de puerta trasera (función en una dirección)**

- Rápido en una dirección pero no en la dirección contraria (sin información secreta)
- La dirección contraria funciona teniendo el secreto (acceso a la puerta trasera)



Ejemplos de una Fisura en el Principio de Kerckhoffs

El secreto está relacionado con la clave y no con el algoritmo

- **Penetración en el cifrado de teléfonos móviles (Diciembre 1999)**

„Científicos Israelíes descubrieron un defecto de diseño que permitía descodificar las conversaciones privadas de cientos de millones de teléfonos móviles. Alex Biryukov y Adi Shamir describen en un artículo publicado esta semana cómo un PC con 128 MB de RAM y unos grandes discos duros puede saltarse la seguridad de una llamada telefónica o de una transmisión de datos en menos de un segundo. El algoritmo erróneo apareció en los teléfonos digitales GSM hechos por empresas como Motorola, Ericsson, y Siemens, y que son utilizados por unos 100 millones de clientes en Europa y Estados Unidos.” [...]”

“Los algoritmos de cifrado GSM habían estado bajo prueba de ataques al estar siendo desarrollados en secreto apartados del escrutinio público –pero muchos expertos dicen que una alta seguridad sólo puede venir de un código publicado. Moran dijo “no fue la actitud a la hora de publicar los algoritmos” cuando los códigos A5 se desarrollaron en 1989, pero los actuales que se están creando se publicarán para una revisión por pares.”

[<http://www.wired.com/politics/law/news/1999/12/32900>]

- **Otro Ejemplo:** En 1999, el navegador Netscape almacenó contraseñas para acceder al servidor de correos utilizando un método de cifrado débil.

Muestra de Adaptación de una Libreta de Uso Único



Percha de un agente Stasi con una *libreta de un sólo uso*
(extraído de: *Spiegel Spezial 1/1990*)

Menú:
"Cifrar/Descifrar" \
"Simétrico (clásico)" \
"Vernam"

Problema de Distribución de Claves

Distribución de claves para métodos de cifrado simétrico

Si 2 personas se comunican utilizando un cifrado simétrico, necesitan una clave secreta común.

Si n personas se comunican entre ellas, entonces necesitan $S_n = n * (n-1) / 2$ claves.

Esto significa que

$n = 100$ personas requieren

$S_{100} = 4.950$ claves; y

$n = 1.000$ personas requieren

$S_{1000} = 499.500$ claves.

⇒ Un factor 10 de más personas, resulta un factor 100 de más claves



Criptografía – Conceptos Importantes (2)

Resolver el problema de distribución de clave mediante criptografía asimétrica

Criptografía Asimétrica

- Durante siglos se creía que: el emisor y el receptor necesitaban el mismo secreto.
- Ahora: Cada miembro necesita un par de claves (solución al problema de distribución de claves)

Cifrado Asimétrico

- „Todo el mundo puede cerrar un candado o puede dejar caer una carta en un buzón.“
- MIT, 1977: Leonard Adleman, Ron Rivest, Adi Shamir (más conocido como RSA)
- GCHQ Cheltenham, 1973: James Ellis, Clifford Cocks (aceptado públicamente en Diciembre de 1997)

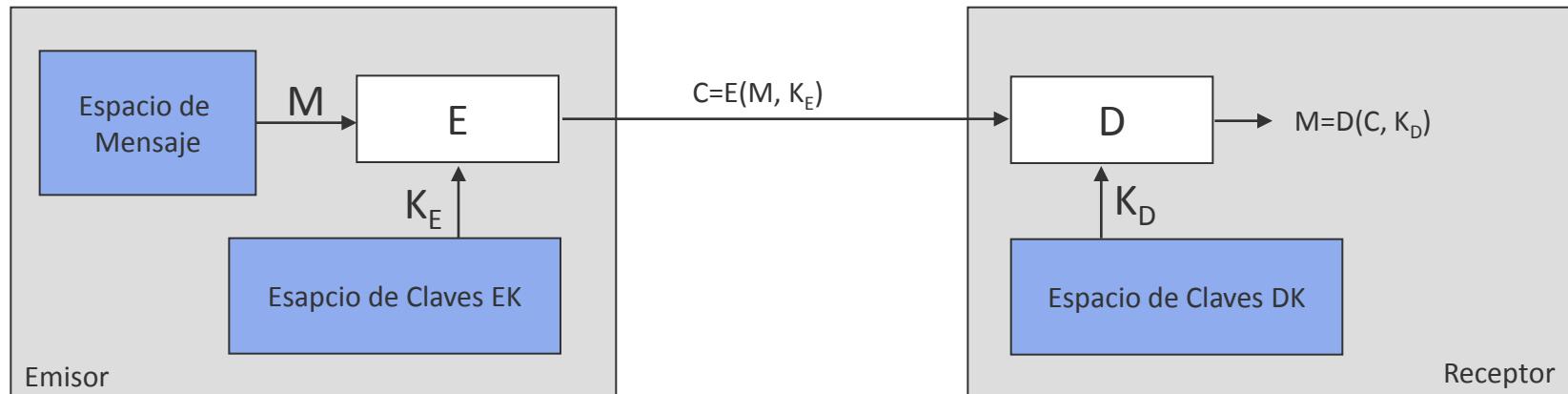
Distribución de claves

- Stanford, 1976: Whitfield Diffie, Martin Hellman, Ralph Merkle (Intercambio de clave Diffie-Hellman)
- GCHQ Cheltenham, 1975: Malcolm Williamson

¡La seguridad en redes abiertas (como Internet) sería extremadamente cara y compleja sin una criptografía asimétrica!

Cifrado y Descifrado

Cifrado Simétrico y asimétrico



a) Cifrado Simétrico:

$$K_E = K_D \quad (\text{p.ej. AES})$$

b) Cifrado Asimétrico:

$$\begin{array}{ccc} K_E & \neq & K_D \\ \text{público} & & \text{privado/segreto} \end{array} \quad (\text{p.ej. RSA})$$

Criptografía – Conceptos Importantes (3)

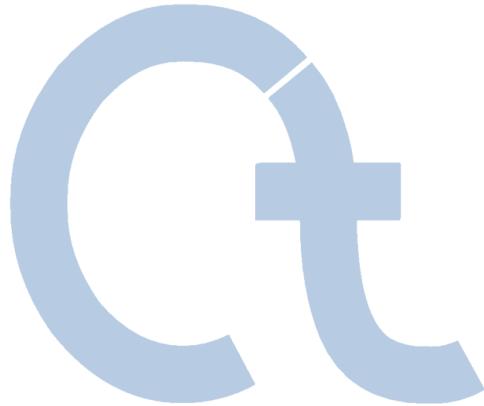
La creciente relevancia de las matemáticas y las tecnologías de la información

- La criptografía moderna se basa en las **matemáticas**
 - A pesar de los nuevos métodos de cifrado simétrico como el AES (mejor funcionamiento y una clave más corta comparados con los métodos asimétricos basados puramente en problemas matemáticos).
- La seguridad de los métodos de cifrado dependen fuertemente del estado en el que se encuentran las **matemáticas** y las **tecnologías de la información** (TI)
 - Complejidad computacional (el principal esfuerzo de procesado está relacionado con la longitud de la clave, demanda de dispositivos y complejidad de los datos)
-> ver RSA: Bernstein, dispositivo TWIRL, RSA-160, RSA-200
 - Actividad muy alta en la investigación actual en:
Factorización, algoritmos no paralelizables (a causa de los ordenadores cuánticos), mejor comprensión de la debilidad de los protocolos y los generadores aleatorios, ...).
- Grave Error: “Las matemáticas reales no tienen efecto sobre la guerra.”
(G.H. Hardy, 1940)
- Los vendedores han descubierto la **seguridad** como un criterio esencial de **compra**.

Demostración con CrypTool

- Análisis Estadístico
- Cifrar dos veces no siempre es mejor:
 - César: $C + D = G$ ($3 + 4 = 7$)
 - Vigenère: - $CAT + DOG = FOZ$ $[(2,0,19)+(3,14,6)=(5,14,25)]$
 - No hay mejoras, sin embargo usando:
 - $GATO + PERRO = VEKFUPXFZOISXRHDKRKC$
 - Se produce una clave mucho más fuerte.
- Vernam (OTP)
- AES (clave de salida, análisis por fuerza bruta)

Contenido



- I. CrypTool y Criptología – Visión General
- II. **CrypTool Características**
- III. Ejemplos
- IV. Proyecto/Perspectiva/Contacto

1. ¿Qué es CrypTool?

- Programa libre con interfaz gráfica
- Se pueden aplicar métodos criptográficos y analizarlos
- Completa ayuda en línea (comprendible sin un conocimiento profundo sobre criptografía)
- Contiene casi todas las funciones criptográficas actuales
- Introducción fácil tanto a la criptografía clásica como a la moderna
- No es una “*herramienta de hackers*”

2. ¿Por qué CrypTool?

- Origen en una iniciativa de concienciación de un instituto financiero
- Desarrollado en una cercana cooperación con universidades.
- Mejora en la educación universitaria y capacitación empresarial

3. Público Objetivo

- Grupo Principal: Estudiantes de informática, negocio informático y matemáticas
- Pero también para: usuarios de ordenador, desarrolladores de aplicaciones, empleados
- Prerrequisitos: conocimiento sobre PC
- Preferiblemente: Interesados en matemáticas y/o programación

Contenido del Paquete del Programa



Programa CrypTool

- Todas las funciones integradas en un *único* programa con una interfaz gráfica consistente
- Funciona sobre Win32
- Librerías Criptográficas de Secude y OpenSSL
- Aritmética de enteros grandes a través de Miracl, APFLOAT y GMP/MPIR, Reducción de base de retículos por NTL (V. Shoup)

Herramienta AES

- Programa independiente para cifrado AES (y creación de archivos autoextraíbles)

Juego Educativo

- „Number Shark“ estimula la comprensión de los factores y los números primos.

Completa Ayuda online (Ayuda HTML)

- Ayuda sensible al contexto disponible con F1 para todas las funciones del programa (incluidos los menús)
- Casos detallados de uso para muchas funciones del programa (tutorial)

Script (archivo .pdf) con información básica

- Métodos de cifrado • Factorización en Primos • Firma Digital
- Curvas Elípticas • certificado de clave pública • Teoría de Números Básica • Crypto 2020

Dos historias cortas relacionadas con la criptografía de Dr. C. Elsner

- „The Dialogue of the Sisters“ (una variante de RSA como elemento clave)
- „The Chinese Labyrinth“ (Tareas de teoría de números para Marco Polo)

Herramienta para el aprendizaje de Teoría de Números

Características (1)

Criptografía

Criptografía Clásica

- César (y ROT-13)
- Sustitución Monoalfabética (y Atbash)
- Vigenère
- Hill
- Sustitución Homofónica
- Playfair
- ADFGVX
- Suma de Bytes
- XOR
- Vernam
- Permutación / Trasposición (Rail Fence, Escítala, ...)
- Solitario

Varias opciones para entender fácilmente los métodos criptográficos

- Alfabeto seleccionable
- Opciones: manejo de espacios, etc.

Criptoanálisis

Ataque a métodos clásicos

- Sólo texto cifrado
 - César
 - Vigenère (según Friedman + Schroedel)
 - Suma
 - XOR
 - Sustitución
 - Playfair
- Texto Claro conocido
 - Hill
 - Transposición de Columna Simple
- Manual (soportado)
 - Sustitución mono-alfabética
 - Playfair, ADFGVX, Solitario

Métodos de Análisis soportados

- Entropía, frecuencia real
- Histograma, análisis de n-grama
- Autocorrelación
- Periodicidad
- Análisis de aleatoriedad
- Base64 / UU-Encode

Características (2)

Criptografía

Cifrado simétrico moderno

- IDEA, RC2, RC4, RC6, DES, 3DES, DESX
- Candidatos AES de la última ronda de selección (Serpent, Twofish, ...)
- AES (=Rijndael)
- DESL, DESXL

Cifrado Asimétrico

- RSA con certificados X.509
- Demostración RSA
 - Comprensión de ejemplos
 - Alfabeto y longitud de bloque seleccionable

Cifrado Híbrido (RSA + AES)

- Diagrama de flujo de datos interactivo

Criptoanálisis

Ataque por fuerza bruta para algoritmos simétricos

- Para todos los algoritmos
- Suposiciones:
 - La entropía de un texto claro es pequeña o la clave se conoce parcialmente o se conoce el alfabeto del texto claro

Ataque al cifrado RSA

- Factorización del módulo RSA
- Ataques de bases de Retículos

Ataque al cifrado híbrido

- Ataque a RSA o
- Ataque a AES (ataque del canal lateral)

Características (3)

Criptografía

Firma Digital

- RSA con certificados X.509
 - Firma como un diagrama de flujo de datos
- DSA con certificados X.509
- Curva Elíptica DSA, Nyberg-Rueppel

Funciones Hash

- MD2, MD4, MD5
- SHA, SHA-1, SHA-2, RIPEMD-160

Generadores Aleatorios

- Secude
- $x^2 \bmod n$
- Generador de congruencias Lineal (LCG)
- Generador de congruencias Inverso (ICG)

Criptoanálisis

Ataque a la firma RSA

- Factorización del módulo RSA
- Factible hasta los 250 bits o 75 decimales (en un PC estándar)

Ataque a las funciones hash / firma digital

- Generar colisiones hash para un texto en ASCII (paradoja del cumpleaños) (hasta 40 bit en unos 5 min)

Análisis de datos aleatorios

- Batería de pruebas FIPS-PUB-140-1
- Periodicidad, Vitany, entropía
- Frecuencia real, histograma
- Análisis de n-gramas, autocorrelación
- Test de compresión ZIP

Características (4)

Animaciones / Demostraciones

- César, Vigenère, Nihilist, DES (todo con ANIMAL)
- Enigma (Flash)
- Rijdael/AES (Flash)
- Cifrado y descifrado Híbrido (AES-RSA y AES-ECC)
- Generación y verificación de firmas digitales
- Intercambio de claves Diffie-Hellman
- Secreto compartido (con CRT o Shamir)
- Método Desafío-Respuesta (autentificación)
- Ataque del canal lateral
- E-mail seguro con protocolo S/MIME (con Java y Flash)
- Presentación gráfica en 3D de chorros de datos (aleatorios)
- Sensibilidad de funciones hash con respecto a cambios en el texto claro
- Teoría de Números y criptosistema RSA (con Authorware)



Características (5)

Funciones Adicionales

- Diferentes funciones para RSA y números primos
- Cifrado Homofónico y por permutación (Transposición Doble Columna)
- PKCS #12 importado y exportado para PSEs (Entorno Personal de Seguridad)
- Generar archivos has de archivos grandes sin cargarlos
- Ataques por fuerza bruta flexibles sobre cualquier algoritmo simétrico moderno.
- Demostración de ECC (como aplicación Java)
- Medidor de Calidad de Contraseñas (PQM)
- Y mucho más ...

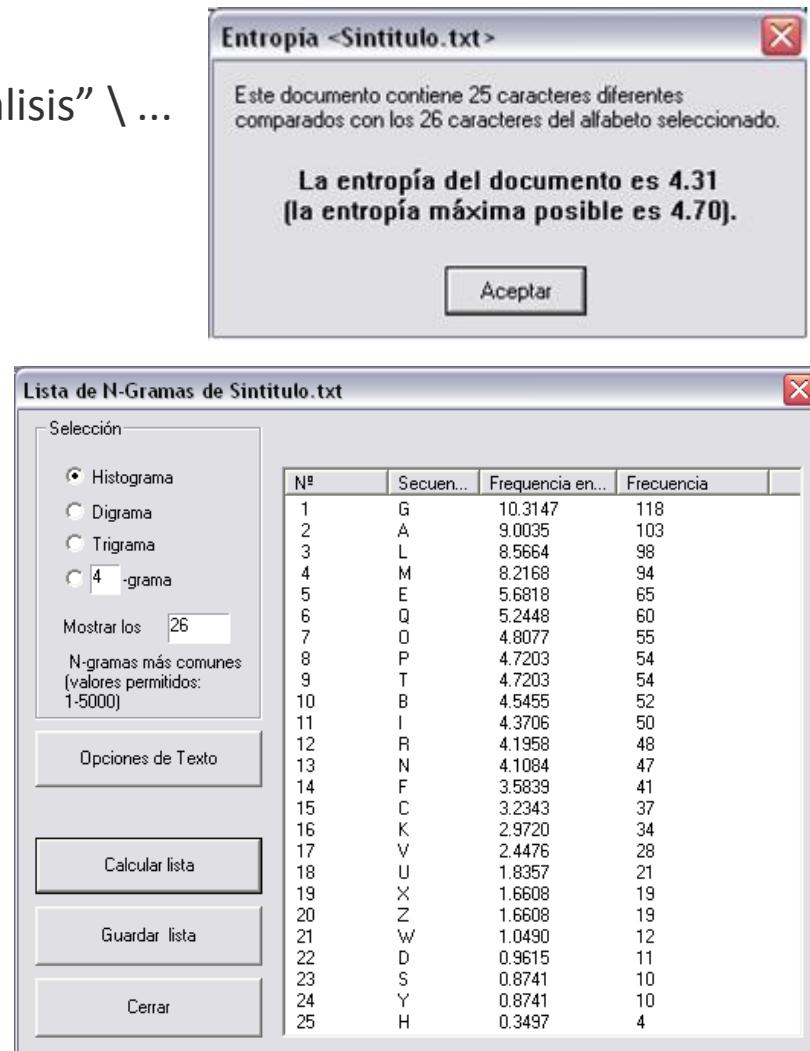
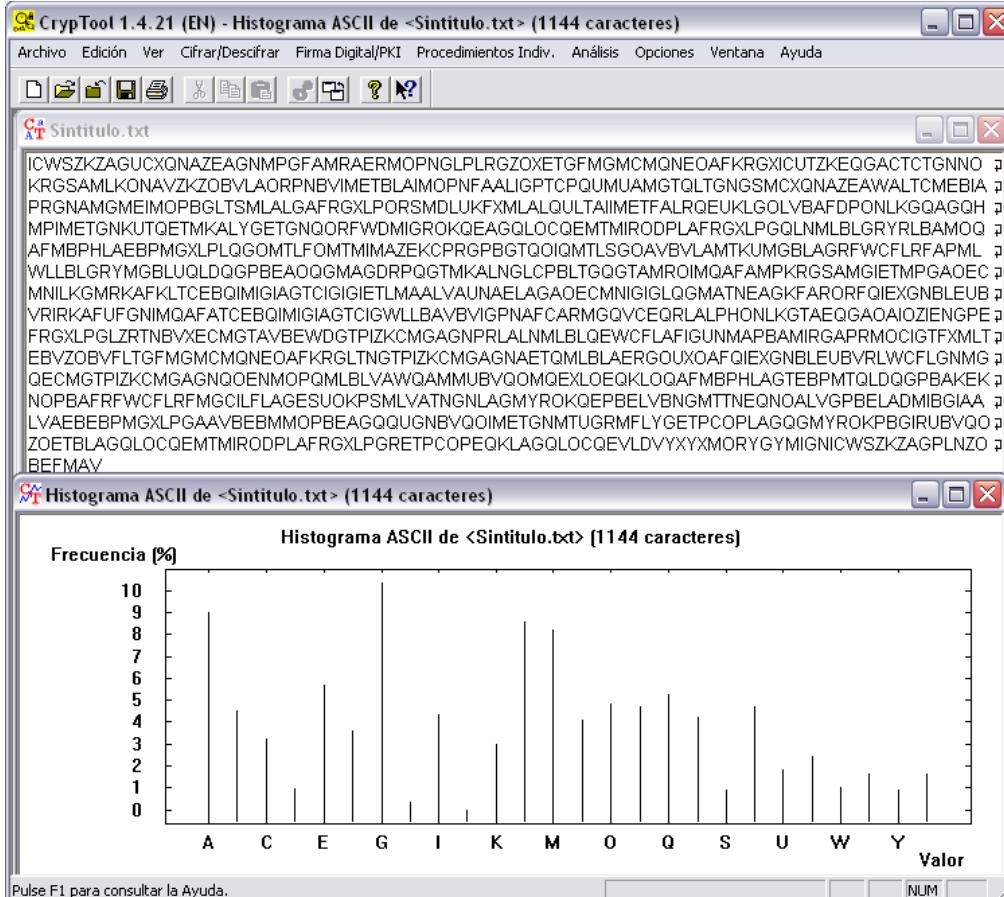


Análisis de la Estructura de un Idioma

Opciones de análisis disponibles en CrypTool

Número de caracteres, n-grama, entropía

- ver menú “Análisis” \ “Herramientas para el Análisis” \ ...



Demonstración de Interactividad (1)

Análisis Vigenère

Demostración en
CrypTool

El resultado del análisis de Vigenère puede rehacerse manualmente (cambiando la longitud de la clave):

1. Cifrar el ejemplo inicial con: TESTETE

- “Cifrar/Descifrar” \ “Simétrico (clásico)” \ “Vigenère”
- Introducir TESTETE \Rightarrow “Cifrar”

Análisis del resultado del cifrado:

- “Análisis” \ “Cifrado Simétrico (clásico)” \ “Sólo texto cifrado” \ “Vigenère”
- Longitud de clave deducida: 7, Clave deducida: TESTETE

2. Cifrar el ejemplo inicial con: TEST

- “Cifrar/Descifrar” \ “Simétrico (clásico)” \ “Vigenère”
- Introducir TEST \Rightarrow “Cifrar”

Análisis del resultado del cifrado:

- “Análisis” \ “Cifrado Simétrico (clásico)” \ “Sólo texto cifrado” \ “Vigenère”
- Longitud de clave deducida: 8 – Falso
- Longitud de clave seleccionada automáticamente a 4 (puede ajustarse manualmente)
- Clave deducida: TEST

Demonstración de Interactividad (2)

Factorización automatizada

Demostración en
CrypTool

Factorización de un número compuesto con algoritmos de factorización

- Algunos métodos se ejecutan en paralelo (multihilo)
- Los métodos tienen ventajas e inconvenientes específicos (p.ej. Algunos métodos sólo pueden determinar factores pequeños)

Ejemplo de Factorización 1:

316775895367314538931177095642205088158145887517

Número decimal de 48-dígitos

=

3 * 1129 * 6353 * 1159777 * 22383173213963 * 567102977853788110597

Ejemplo de Factorización 2:

$2^{250} - 1$

Número decimal de 75-dígitos

=

3 * 11 * 31 * 251 * 601 * 1801 * 4051 * 229668251 * 269089806001 *
4710883168879506001 * 5519485418336288303251

Menú: “Procedimientos Indiv.” \ “Criptosistema RSA” \ “Factorización de un Número”

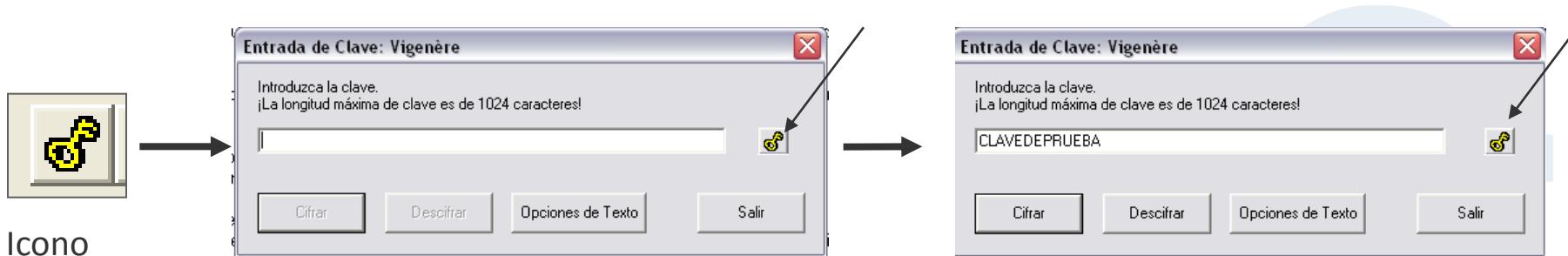
Conceptos para una Interfaz de fácil manejo

1. Ayuda sensible al contexto (F1)

- F1 sobre una entrada de menú seleccionada nos muestra información sobre el algoritmo/método.
- F1 en una ventana de diálogo explica la utilidad de la ventana.
- Estas asistencias y los contenidos de los menús principales están vinculados de forma cruzada en la ayuda en línea.

2. Pegar claves en una ventana de entrada de claves

- Se puede utilizar CTRL-V para pegar contenidos desde el porta papeles.
- Las claves utilizadas se pueden obtener de una ventana de texto cifrado por medio de un ícono de la barra de herramientas. Su correspondiente ícono en la ventana de entrada de clave se puede utilizar para pegar la clave en el campo de entrada. Se utiliza una memoria interna de CrypTool que está disponible para cada método (útil para claves largas y/o “específicas” - p.ej. en el cifrado homofónico).



Desafíos para los Desarrolladores (Ejemplos)

1. Muchas funciones trabajan en paralelo

- La factorización trabaja con algoritmos multihilo

2. Alto Rendimiento

- Localizar colisiones hash (paradoja del cumpleaños) o ejecutar análisis por fuerza bruta

3. Considera límites de memoria

- Algoritmo de Floyd (mapeados para localizar colisiones hash) o con una factorización con criba cuadrática

4. Medida del tiempo y estimaciones

- Muestra el tiempo restante durante la fuerza bruta

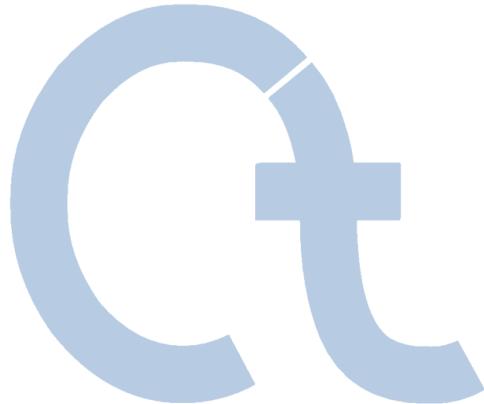
5. Reusabilidad / Integración

- Aplicaciones para la generación de números primos
- Criptosistema RSA (cambia la vista después de un ataque exitoso de un usuario de clave pública al propietario de clave privada)

6. Automatizar parcialmente la consistencia de funciones, GUI y ayuda en línea (incluyendo varios idiomas y los SOs de Windows: XP, Vista y 7)



Contenido



- I. CrypTool y Criptología – Vista General
- II. Características CrypTool
- III. Ejemplos**
- IV. Proyecto/Perspectiva/Contacto

Ejemplos de CrypTool

Visión general de los ejemplos

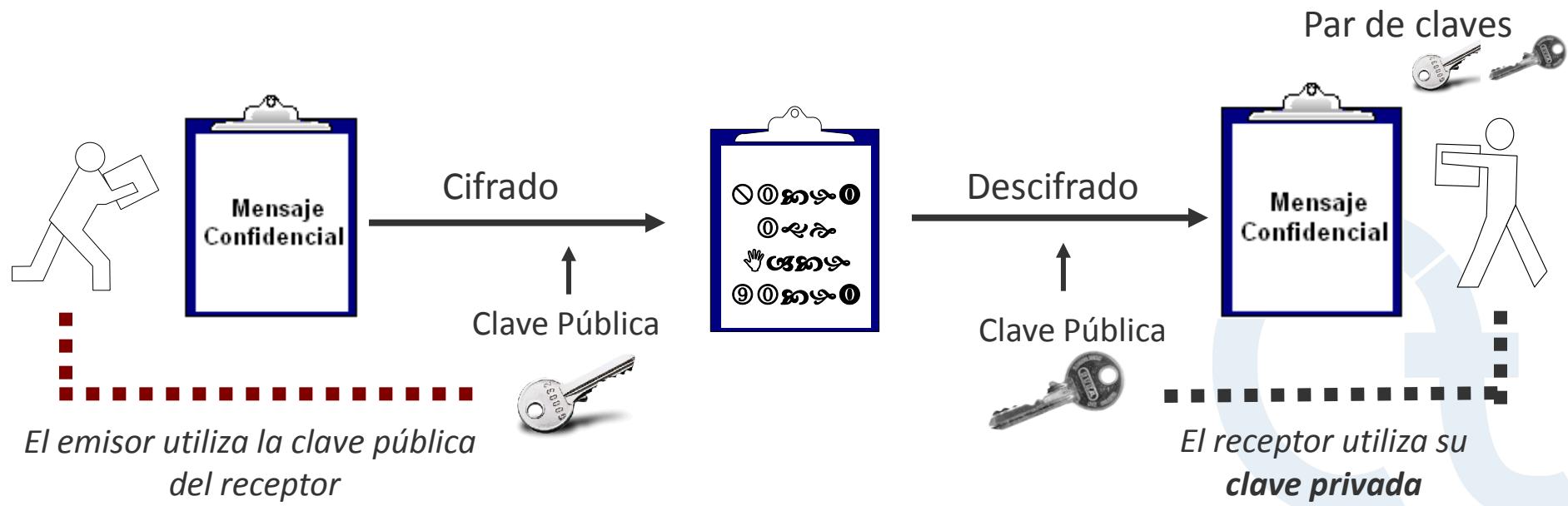
1. [Cifrado con RSA / Test de primalidad / Cifrado híbrido y certificados digitales / SSL](#)
2. [Visualización de firma digital](#)
3. [Ataque al cifrado RSA \(módulo N demasiado pequeño\)](#)
4. [Ánalisis del cifrado en la PSION 5](#)
5. [Claves DES débiles](#)
6. [Localizando información de la clave \("clave NSA"\)](#)
7. [Ataque a la firma digital por búsqueda de colisiones hash](#)
8. [Autentificación en un entorno cliente-servidor](#)
9. [Demostración de un ataque de canal lateral \(en un protocolo de cifrado híbrido\)](#)
10. [Ataque RSA utilizando reducción de retículos \(lattice reduction\)](#)
11. [Ánalisis de aleatoriedad con visualización 3-D](#)
12. [Secreto Compartido \(Teorema Chino de los Restos \(CRT\) / Shamir\)](#)
13. [Implementación del CRT en Astronomía](#)
14. [Visualización del cifrado utilizando ANIMAL](#)
15. [Visualización del AES](#)
16. [Visualización del cifrado Enigma](#)
17. [Visualización de E-mail seguro con S/MIME](#)
18. [Generación de un código de autentificación de un mensaje \(HMAC\)](#)
19. [Demo Hash](#)
20. [Herramienta de aprendizaje de teoría de números y cifrado asimétrico](#)
21. [Suma de puntos en curvas elípticas](#)
22. [Medidor de calidad de contraseñas](#)
23. [Ánalisis por Fuerza Bruta](#)
24. [Escítala / Rail Fence](#)
25. [Cifrado Hill / Análisis Hill](#)
26. [Ayuda online de CrypTool](#)



Ejemplos de CrypTool

Cifrado con RSA (en realidad, la mayoría de cifrados híbridos)

- Bases para, por ejemplo, el protocolo SSL (acceso a sitios web protegidos)
- Cifrado asimétrico utilizando RSA
 - Cada usuario tiene un par de claves— una pública y otra privada
 - El emisor cifra con la clave pública del receptor
 - El receptor descifra con su clave privada
- Normalmente se implementa combinándolo con métodos simétricos (transferencia de la clave simétrica por codificación/descodificación RSA)



Ejemplos (1)

Cifrado utilizando RSA – trasfondo Matemático / algoritmo

- Clave Pública: (n , e)
- Clave Privada: (d)

donde:

p , q grandes, números primos elegidos aleatoriamente con $n = p * q$;

d se calcula bajo las constantes $\text{mcd}[\varphi(n), e] = 1$; $e^d \equiv 1 \pmod{\varphi(n)}$.

Operación de cifrado y descifrado: $(m^e)^d \equiv m \pmod{n}$

- n es el módulo, cuya longitud en bit se refiere a la longitud de la clave RSA.
- mcd = máximo común divisor.
- $\varphi(n)$ es la función fi de Euler.

Procedimiento :

- Transformación del mensaje en representación binaria
- Mensaje Cifrado $m = m_1, \dots, m_k$ en el sentido de los bloques, con para todo m_j : $0 \leq m_j < n$; tamaño de bloque máximo r , por eso: $2^r \leq n$ ($2^r - 1 < n$)

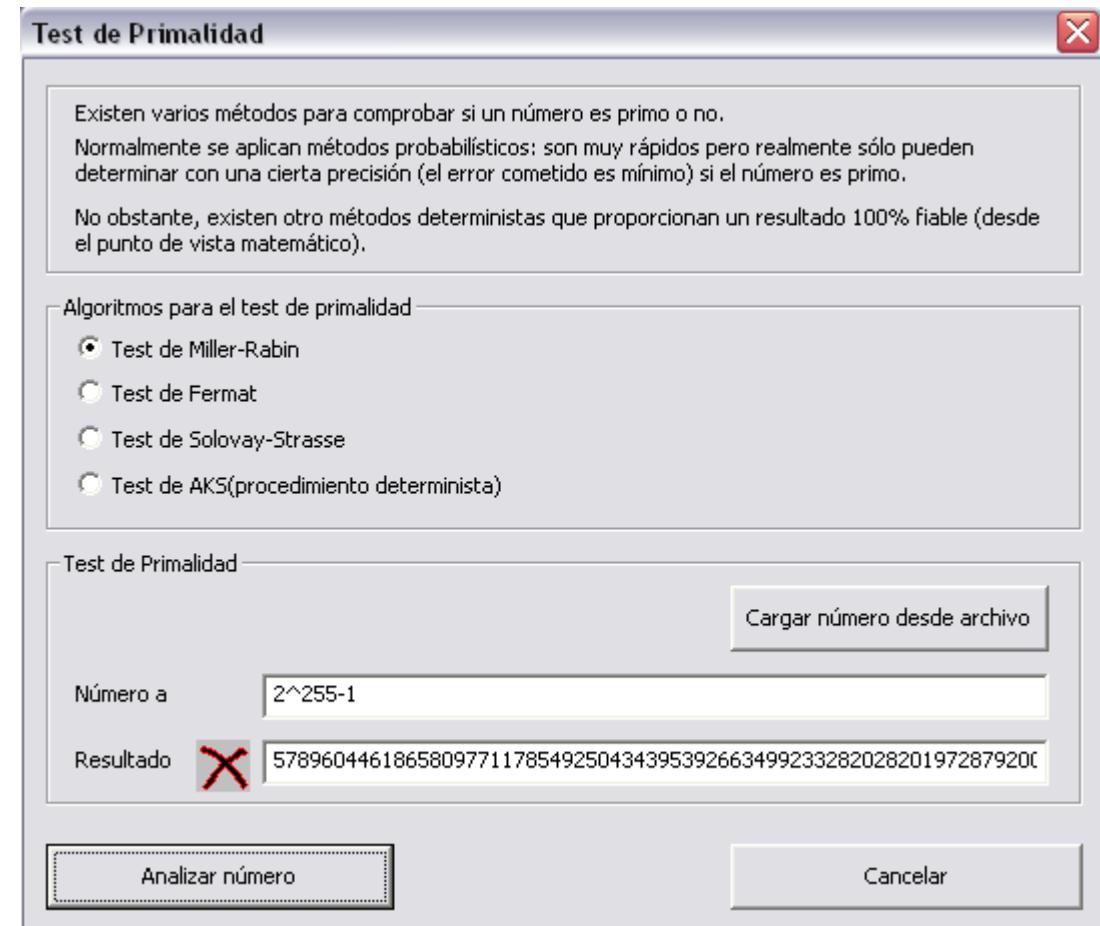
Ejemplos (1)

Test de Primalidad – Se necesita para los enormes primos de RSA.

- Pruebas probabilísticas rápidas
- Pruebas Deterministas

Los métodos de prueba de números primos se realizan mucho más rápido si un número grande es primo, entonces los métodos de factorización conocidos pueden separar un número de tamaño similar en sus factores primos.

Para los test AKS se integraron a CrypTool las bibliotecas GMP (**GNU Multiple Precision Arithmetic Library**) y MPIR (**Multiple Precision Integers and Rationals**).



Menú: "Procedimientos. Indiv" \ "RSA Criptosistema" \ "Test de Primalidad"

Ejemplos (1)

Cifrado Híbrido y certificados digitales

- Cifrado Híbrido – **Combinación de cifrado simétrico y asimétrico**
 1. Generación de una clave simétrica aleatoria (clave de sesión)
 2. Se transfiere la clave de sesión – protegida por una clave asimétrica
 3. Se transfiere el mensaje – protegida por la clave de sesión
- Problema: **Ataques del hombre en el medio – ¿La clave pública del receptor pertenece realmente al receptor?**
- Solución: Certificados Digitales – **Una central (p. ej. Telesec, VeriSign, Deutsche Bank PKI), en la que confían los usuarios, asegura la autenticidad del certificado y la clave pública contenida (parecido al pasaporte expedido por el estado).**
- El cifrado híbrido basado en certificados digitales **es la base para todas las comunicaciones electrónicas seguras:**
 - Compra por Internet y Banca Online
 - Correo electrónico seguro



Ejemplos (1)

Conexión online segura utilizando SSL y certificados

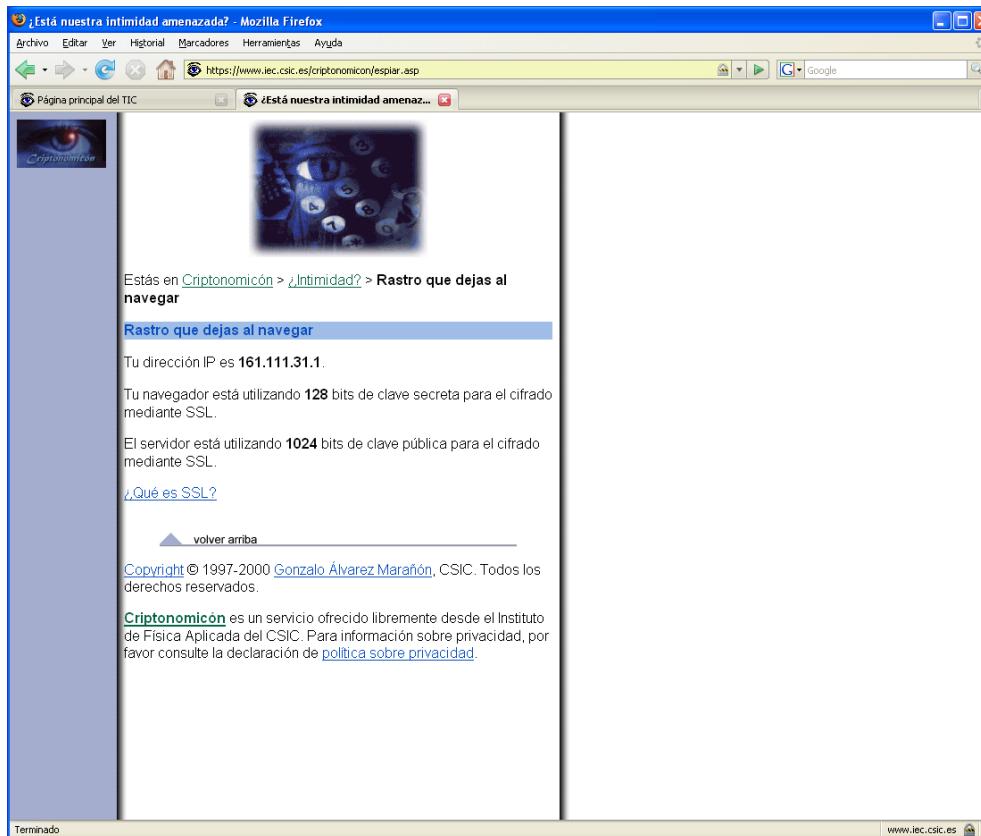
Está nuestra intimidad amenazada? - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

https://www.iec.csic.es/cryptonomicón/espiar.asp

Página principal del TIC

¿Está nuestra intimidad amenazada...



Estás en Criptonomicón > ¿Intimidad? > Rastro que dejas al navegar

Rastro que dejas al navegar

Tu dirección IP es 161.111.31.1.

Tu navegador está utilizando 128 bits de clave secreta para el cifrado mediante SSL.

El servidor está utilizando 1024 bits de clave pública para el cifrado mediante SSL.

¿Qué es SSL?

volver arriba

Copyright © 1997-2000 Gonzalo Álvarez Marañón, CSIC. Todos los derechos reservados.

Criptonomicón es un servicio ofrecido libremente desde el Instituto de Física Aplicada del CSIC. Para información sobre privacidad, por favor consulte la declaración de política sobre privacidad.

Terminado

Visor de certificados:"www.iec.csic.es"

General | Detalles

No se pudo verificar este certificado por razones desconocidas.

Emitido para

Nombre común (CN)	www.iec.csic.es
Organización (O)	CSIC
Unidad organizativa (OU)	TIC
Número de serie	61:0A:AD:94:00:08:00:00:43

Emitido por

Nombre común (CN)	Dep. Tratamiento de la Información y Codificación
Organización (O)	CSIC
Unidad organizativa (OU)	TIC

Validez

Emitido el	30/11/2007
Expira el	30/11/2008

Huellas digitales

Huella digital SHA1	36:53:07:D4:70:B6:79:F5:59:77:48:CC:A0:C5:A2:1E:64:F0:C0:EA
Huella digital MD5	DF:9B:4B:2B:86:A3:C6:29:89:B0:B5:45:70:D3:9B:31

Cerrar

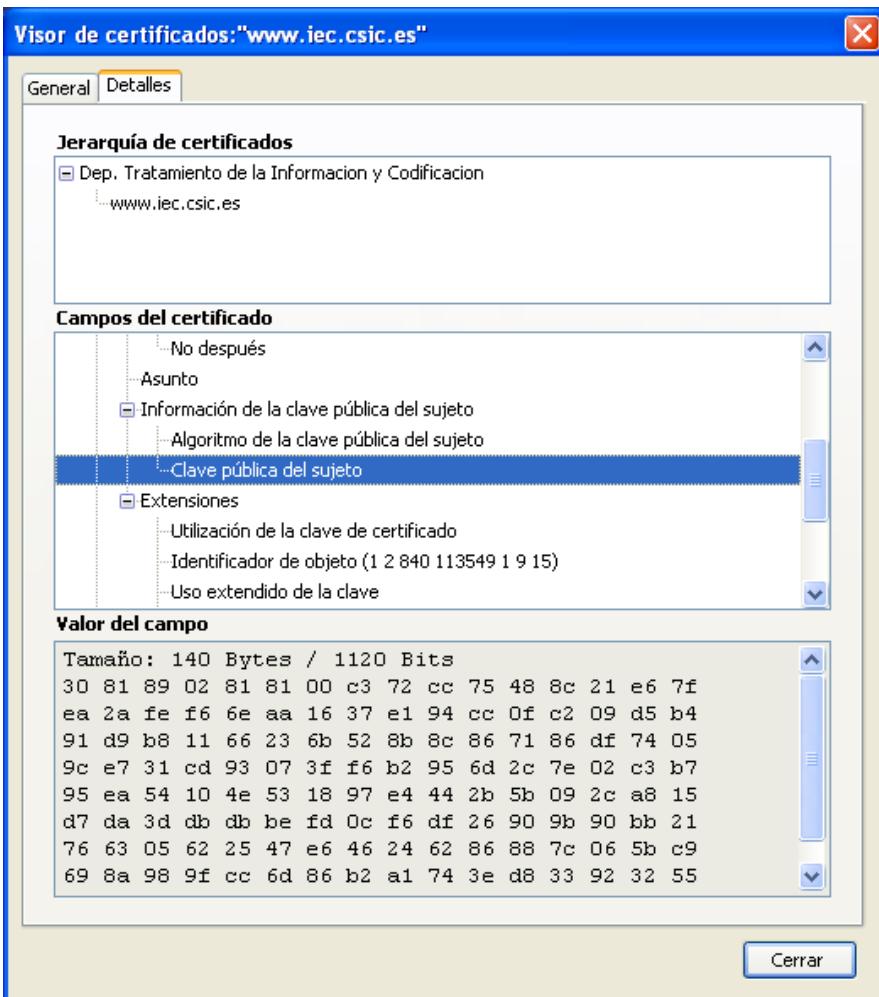
Esto significa que la conexión es autenticada y (al menos en un sentido) la transferencia de datos está fuertemente cifrada.



SSL-secured (128 Bit)

Ejemplos (1)

Atributos o campos de un certificado



Atributos generales / campos

- Emisor (p.ej. VeriSign)
- Solicitante
- Período de validez
- Número de Serie
- Tipo de Certificado/ Versión (X.509v3)
- Algoritmo de firma
- Clave Pública (y método)

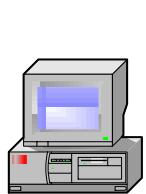
Clave Pública



Ejemplos (1)

Establecer una conexión segura SSL (Autenticación del Servidor)

Client



1. Inicio SSL

Server



2.

Enviar certificado del servidor

3. Validar certificado del servidor (utilizado raíces de certificados instalados localmente)

4. Recuperar la clave pública del servidor (desde el certificado del servidor)

5. Generar una clave simétrica aleatoria (clave de sesión)

6. Enviar clave de sesión

(cifrada con la clave pública del servidor)

Recuperar clave de sesión

(descifrada por la clave privada del servidor)

7.



SSL Secured (128 Bit)

*Comunicación cifrada basada en el
intercambio de la clave de sesión*

Ejemplos (1)

Establecer una conexión segura SSL (Autenticación del Servidor)

General

- El ejemplo muestra el establecimiento de una conexión SSL típica para transferir datos delicados a través de internet (p.ej. compra online).
- Al establecer la conexión SSL solamente se autentifica el servidor utilizando el certificado digital (la autenticación del usuario se da normalmente a través del nombre de usuario y su contraseña después de que se haya establecido la conexión SSL).
- SSL también ofrece la opción de la autenticación del cliente basada en certificados digitales.

Comentarios al establecimiento de la conexión SSL (ver diapositiva anterior)

- Paso 1: Inicialización SSL – durante esta fase, se negocian las características tanto de la clave de sesión (p.ej. Tamaño en bits) como del algoritmo de cifrado simétrico (p.ej. 3DES, AES).
- Paso 2: En el caso de una jerarquía de certificado multinivel, también se pasan los certificados intermedios al cliente.
- Paso 3: En esta fase los certificados raíz instalados en la memoria de certificados del navegador se utilizan para validar el certificado del servidor.
- Paso 5: La clave de sesión se basan en las características negociadas (ver 1).

Ejemplos (2)

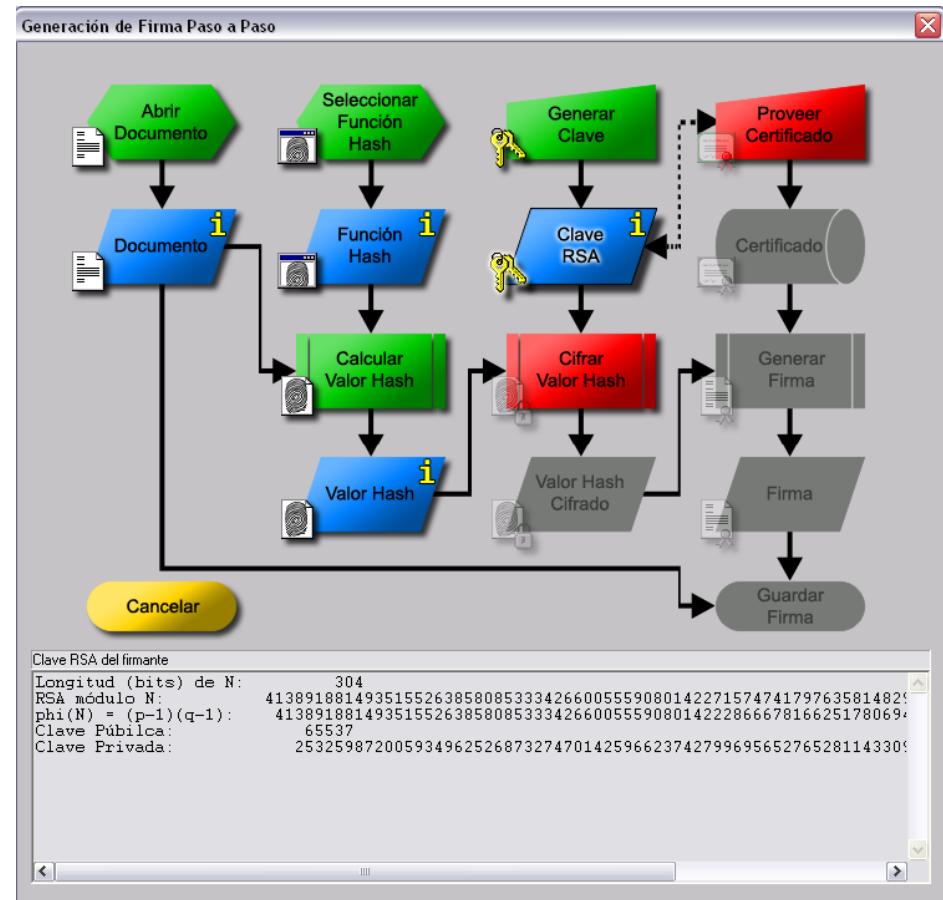
Visualización de una firma digital

Firma Digital

- Importancia creciente:
 - Equivalencia con la firma manual (ley de la firma digital)
 - Cada vez más utilizada en la industria,
 - Gobierno y usuarios
- Poca gente sabe cómo funciona exactamente

Visualización en CrypTool

- Diagrama de flujo de datos interactivo
- Parecida a la visualización del cifrado híbrido



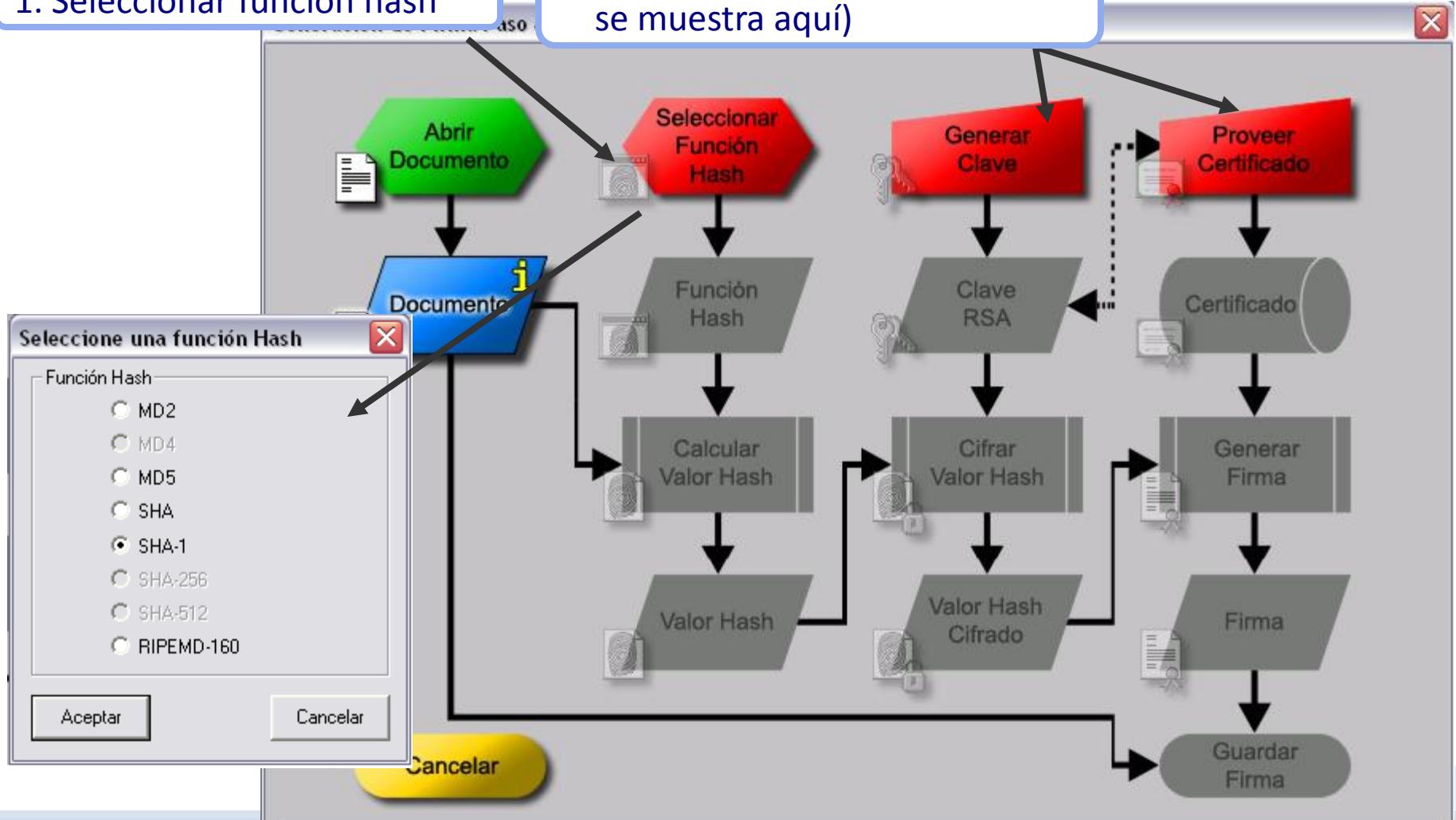
Menú: „Firma Digital/PKI“ \
„Demostración de firma(Generación de Firma)“

Ejemplos (2)

Visualización de una firma digital : a) Preparación

1. Seleccionar función hash

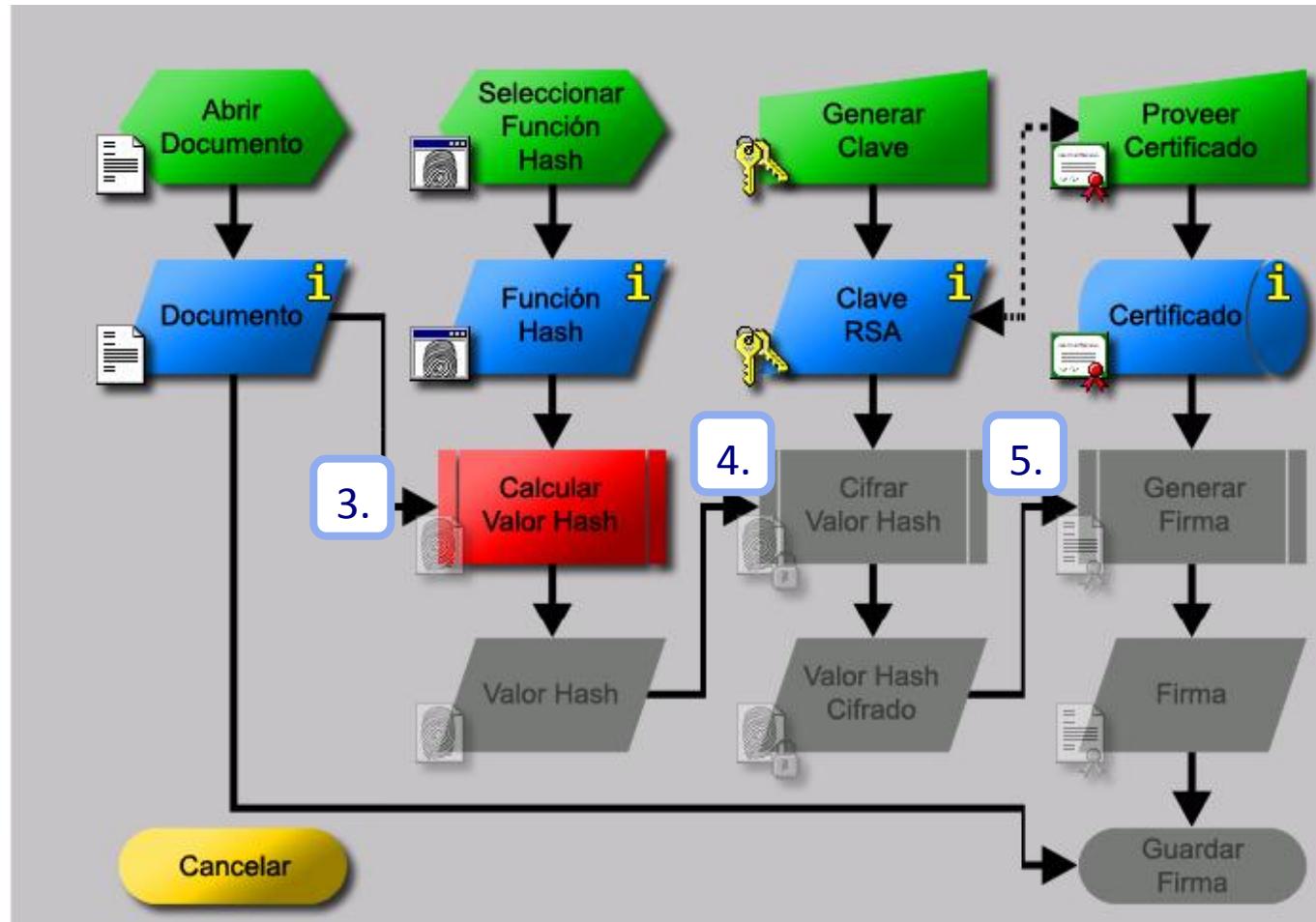
2. Facilitar clave y certificado (no se muestra aquí)



Ejemplos (2)

Visualización de una firma digital : b) Criptografía

- 3. Calcular valor hash
- 4. Cifrar el valor hash con la clave privada (firmar)
- 5. Generar firma

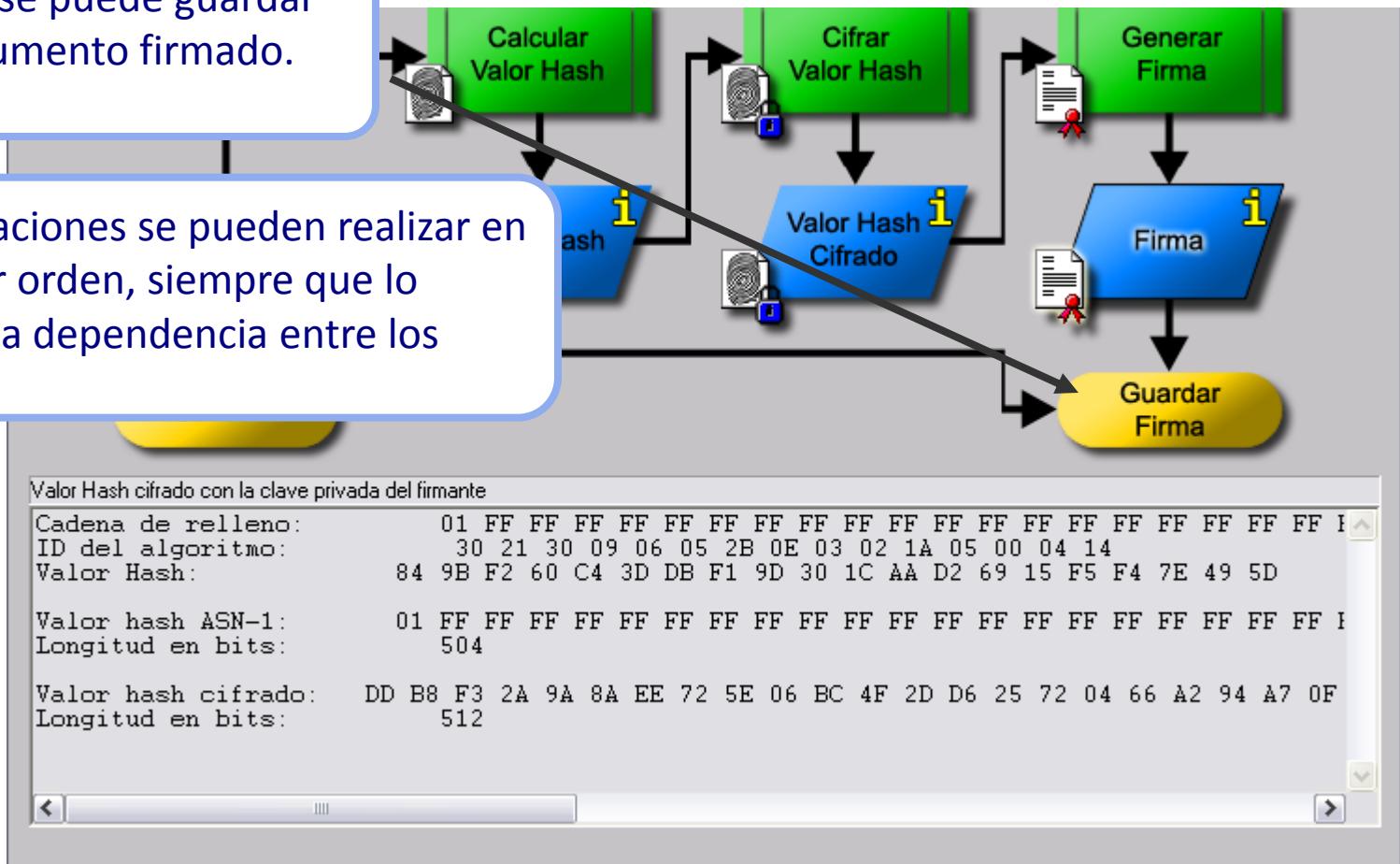


Ejemplos (2)

Visualización de una firma digital : c) Resultado

6. Ahora se puede guardar el documento firmado.

Las operaciones se pueden realizar en cualquier orden, siempre que lo permita la dependencia entre los datos.



Ejemplos (3)

Ataque al cifrado RSA con un RSA de módulo pequeño

Ejemplo de *Song Y. Yan, Number Theory for Computing, Springer, 2000*

Clave pública

- Módulo RSA **N = 63978486879527143858831415041** (95 bit, 29 dígitos decimales)
- exponente publico **e = 17579**

Texto cifrado(longitud de bloque = 8):

$C_1 = 45411667895024938209259253423,$
 $C_2 = 16597091621432020076311552201,$
 $C_3 = 46468979279750354732637631044,$
 $C_4 = 32870167545903741339819671379$

¡El texto cifrado no se necesita para el criptoanálisis actual (localizar la clave privada)!

¡el texto debe ser descifrado!

Solución utilizando CrypTool (de forma más detallada en la sección de la ayuda online)

- Introducir parámetros públicos en “RSA Criptosistema” (menú: “Procedimientos Indiv.”)
- Botón “Factorizar módulo RSA” produciendo los dos factores primos $pq = N$
- Basado en la información del exponente privado se determina $d = e^{-1} \text{ mod } (p-1)(q-1)$
- Descifrar el texto cifrado con d : $M_i = C_i^d \text{ mod } N$

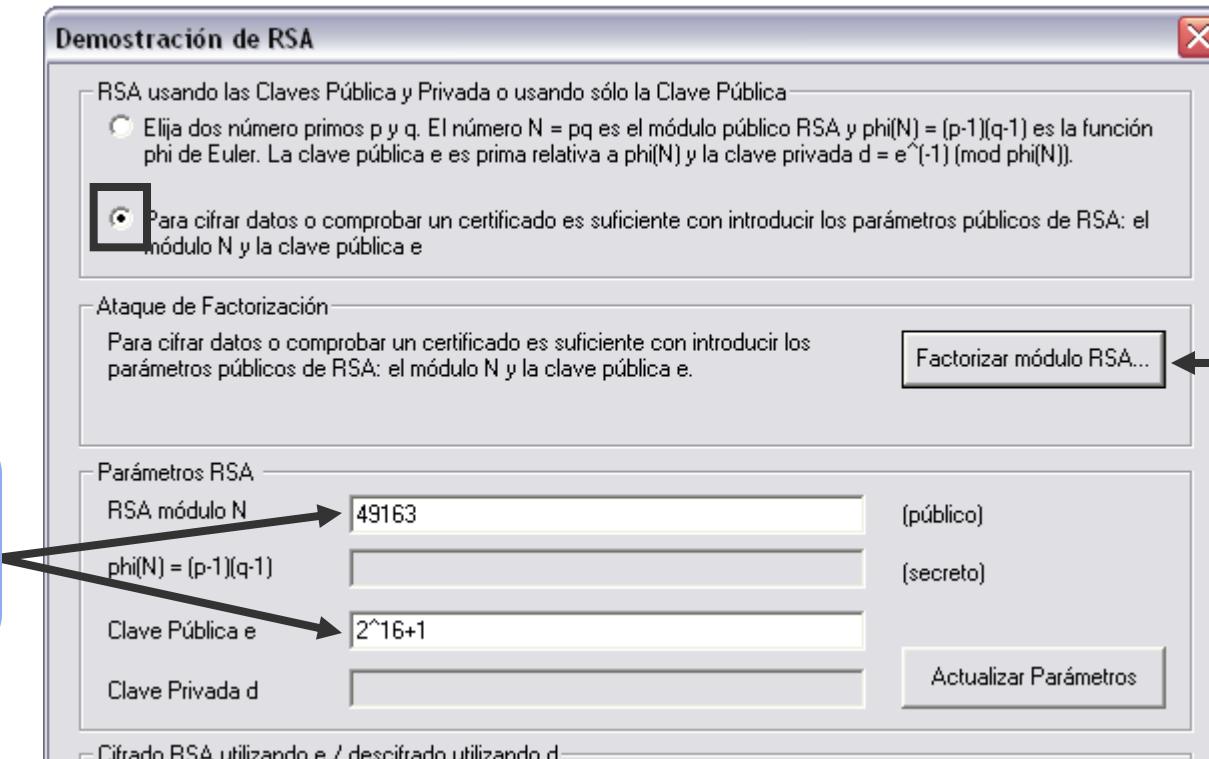
El ataque con CrypTool funciona para el módulo RSA hasta 250 bits.

¡Entonces podría firmar digitalmente por otra persona!

Ejemplos (3)

RSA de módulo pequeño: Introducir parámetros públicos del RSA

Menú: "Procedimientos. Indiv." \ "RSA Criptosistema" \ "RSA Demonstración ..."

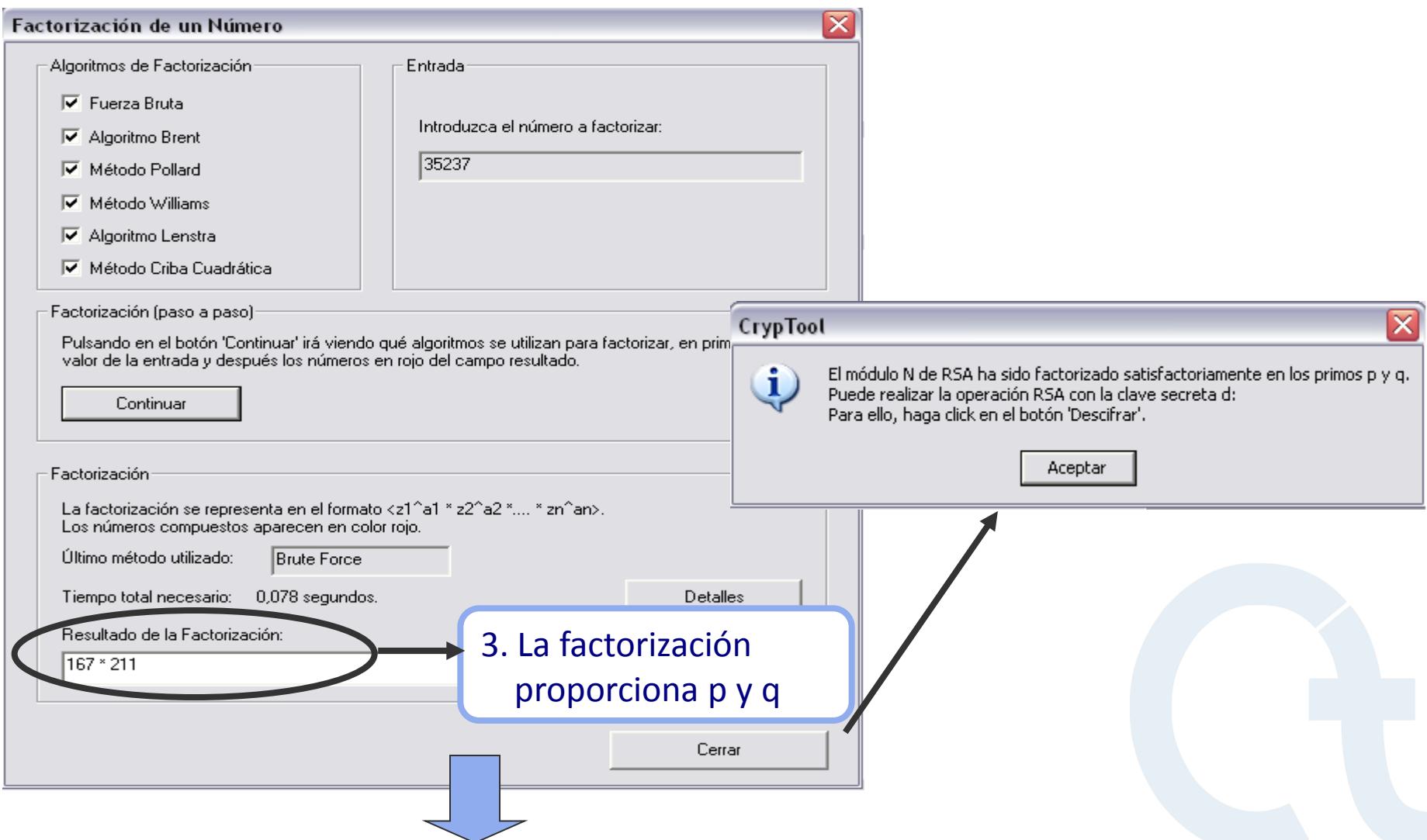


1. Introducir
parámetros:
"N" y "e"

2. Factorizar

Ejemplos (3)

RSA de módulo pequeño: Factorizar módulo RSA



Ejemplos (3)

RSA de módulo pequeño: determinar la clave privada d

Demostración de RSA

RSA usando las Claves Pública y Privada o usando sólo la Clave Pública

Elija dos números primos p y q. El número N = pq es el módulo público RSA y $\phi(N) = (p-1)(q-1)$ es la función phi de Euler. La clave pública e es prima relativa a $\phi(N)$ y la clave privada d = $e^{-1} \pmod{\phi(N)}$.

Para cifrar datos o comprobar un certificado es suficiente con introducir los parámetros públicos de RSA: el módulo N y la clave pública e

Entrada de número primo

Número primo p: 7997393 Generar números primos...

Número primo q: 2258651

Parámetros RSA

RSA módulo N: 18063319696843 (público)

$\phi(N) = (p-1)(q-1)$: 18063309440800 (secreto)

Clave Pública e: $2^{16}+1$

Clave Privada d: 5351989604673

Actualizar Parámetros

Cifrado RSA utilizando e / descifrado utilizando d

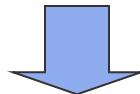
Entrada: texto números

Opciones del alfabeto y sistema numérico...

Cambia la visión del propietario de la clave secreta.

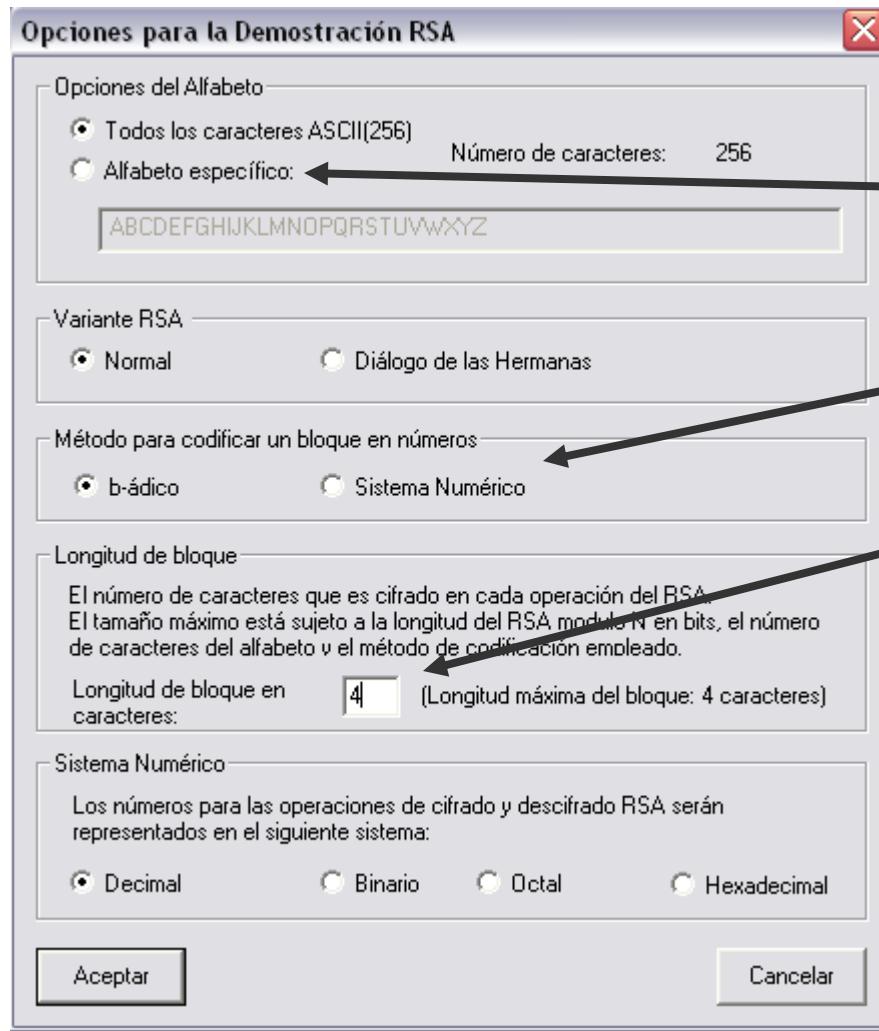
4. p y q se han introducido automáticamente y se ha calculado la clave privada

5. Ajustar Opciones



Ejemplos (3)

RSA de módulo pequeño: Ajustar Opciones



6. Seleccionar alfabeto

7. Seleccionar método de codificación

8. Seleccionar longitud de bloque

Ejemplos (3)

RSA de módulo pequeño: Descifrar texto cifrado

Parámetros RSA

RSA módulo N	345947477033	(público)
phi(N) = (p-1)(q-1)	345946300500	(secreto)
Clave Pública e	2^16+1	
Clave Privada d	333504543473	Actualizar Parámetros

Cifrado RSA utilizando e / descifrado utilizando d

Entrada texto números Opciones del alfabeto y sistema numérico...

Texto cifrado codificado en número de base 16

```
06307C1243 # 4794CB1007 # 15FC2E59E6 # 1DB9F279FC # 2768F3C4F7 # 27C208A3CA # 219E0C88E # 004C4F5320 # 004E554D45 # 00524F5320 # 004E415455 # 0052414C45 # 005320534F # 004E204449 # 0
```

Descifrar mensaje $m[i] = c[i]^d \pmod{N}$

El texto de salida del proceso de descifrado (en segmentos de tamaño 4; el símbolo '#' es el usado como separador)

```
LOS # NUME # ROS # NATU # RALE # S SO # N DI # VINO # S
```

Texto claro

```
LOS NUMEROS NATURALES SON DIVINOS
```

Cifrar **Descifrar** **Cerrar**

9. Introducir texto cifrado

10. Descifrar

Ejemplos (4)

Análisis del cifrado utilizado en la PSION 5

Aplicación práctica del criptoanálisis:

Ataque a la opción de cifrado de la aplicación de Procesador de texto de la PDA PSION 5

Punto de partida: un archivo cifrado con PSION

Requisitos

- Texto en alemán o en inglés cifrado
- Dependiendo del método y la longitud de la clave, texto desde 100 Bytes hasta varios kB

Procedimiento

- preanálisis
 - entropía
 - Entropía real
 - Test de compresión
- autocorrelación
- Intentar análisis automático con métodos clásicos

Probablemente un algoritmo de cifrado clásico



Ejemplos (4)

PDA PSION 5 – determinar entropía, test de compresión

The screenshot shows the CryptTool interface with several windows open:

- Main Window:** Shows the hex dump of the file `psion-enc.hex`. A blue callout box with an arrow points to the right pane, containing partially visible text: "w.....Vf", "...)%uZ", "...x.", "L...[t...t", "cA...C.v...sx.", "...w...{.....", "...w...s.xUM[", "...N...ZY...3...".
- Compression Dialog:** A modal window titled "CrypTool" displays the message "Tasa de compresión: 21 %". An arrow from a blue callout box points to this dialog.
- Entropy Dialog:** A modal window titled "Entropía <psion-enc.hex>" displays the message "Este documento contiene los 256 valores posibles." and "La entropía del documento es 7.56 (la entropía máxima posible es 8.00)". An arrow from another blue callout box points to this dialog.
- Frequency Analysis Plot:** A graph titled "Frecuencia real de <psion-enc.hex>" showing "Caracteres diferentes por bloque de 64 bytes". The Y-axis ranges from 40 to 60, and the X-axis has markers at 1, 5000, and 10000. The plot shows a highly noisy, flat line around a value of 50.
- Status Bar:** At the bottom left, it says "Pulse F1 para consultar la Ayuda." At the bottom right, there are buttons for "OVR", "CAP", and "NUM".

Annotations:

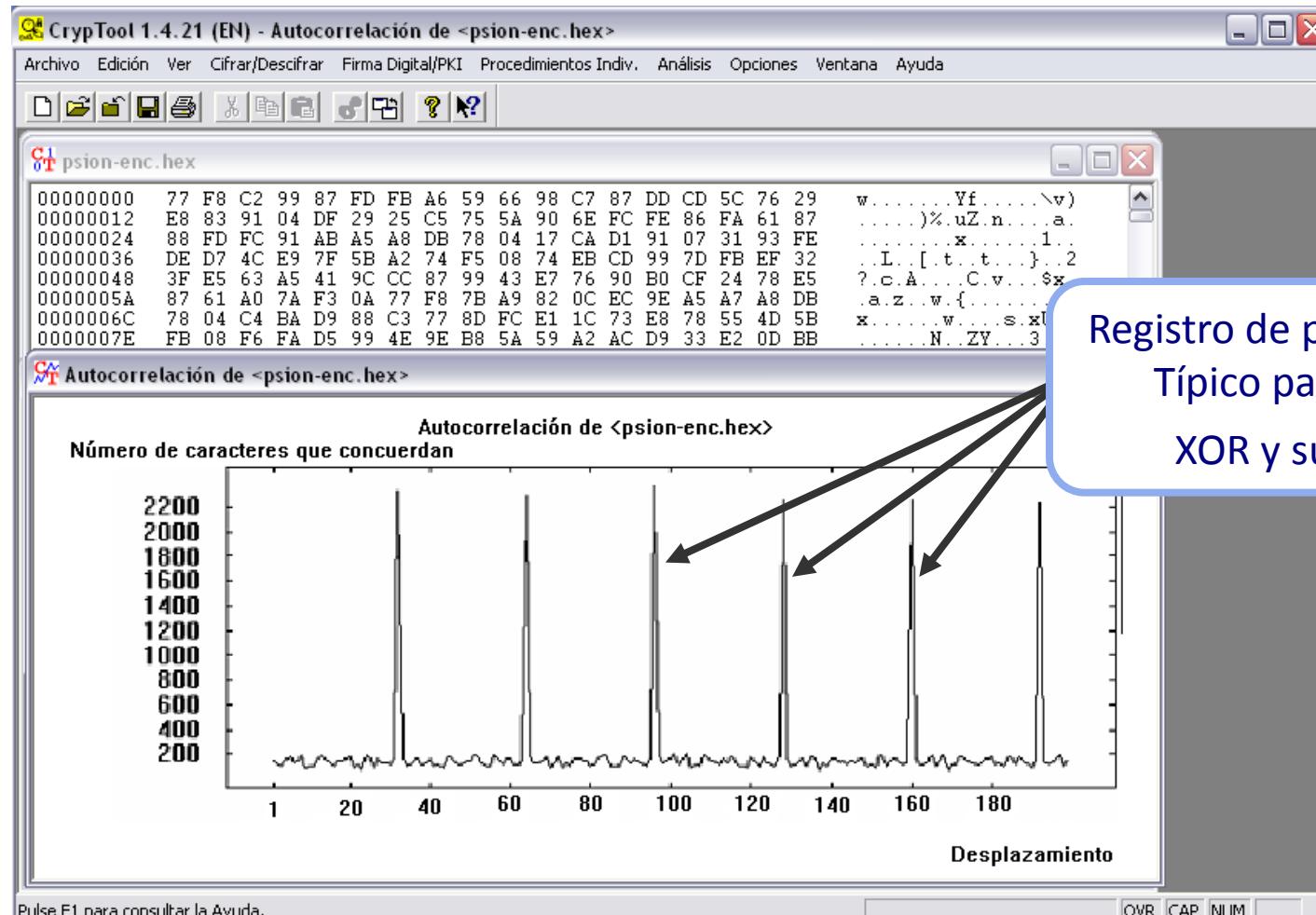
- A blue callout box with an arrow points to the compressed text in the main window, stating: "Comprendibilidad: indicador claro para la criptografía débil (se redujo el tamaño un 21%)".
- A blue callout box with an arrow points to the entropy dialog, stating: "La entropía no proporciona información sobre el método de cifrado específico."

CryptTool 1.4.30

Página 57

Ejemplos (4)

PDA PSION 5 – Determinar auto-correlación



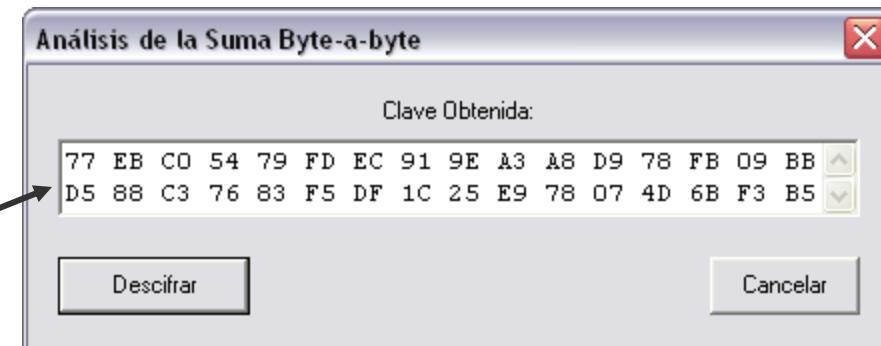
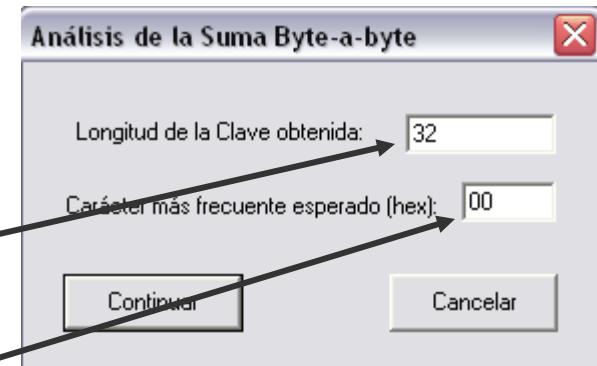
* El archivo cifrado está disponible en CrypTool (ver: CrypTool\examples\psion-en-enc.hex)

Ejemplos (4)

PDA PSION 5 – análisis automático

Análisis automático utilizando

- Vigenère: sin éxito
- XOR: sin éxito
- Suma binaria
 - CrypTool calcula la longitud de la clave utilizando la auto-correlación: 32 bytes
 - El usuario puede elegir el carácter que se espera que aparezca con mayor frecuencia:
“e” = 0x65 (código ASCII)
 - El análisis calcula la clave más probable (basada en las suposiciones sobre la distribución)
 - Resultado: bueno, pero no perfecto



Ejemplos (4)

PDA PSION 5 – resultados del análisis automático

Resultados del análisis automático asumiendo “suma binaria”:

- El resultado es bueno, pero no perfecto: 24 de los 32 bytes de la clave son correctos.
- Se determina correctamente la longitud de la clave:32 .

The screenshot shows the CryptTool interface with two panes. The left pane displays a hex dump of a file named 'psion-enc.hex'. The first 16 bytes of the dump are: 000000 65 72 67 AA 73 65 74 7A 20 28 55 53 74 47 29 06. The right pane shows the corresponding ASCII text: erg@setz (UStG). ...rstereAb, hn® tt...teuergegenst an® und el¹ng, ber²ich...S 1..(1) er Um,ati,te® er ²nterliegen d ieeffolge³de³eUm, ätz³:1. die Lie fe·ungeneun@eso³ sti·en Leistunge n,edie e@n .³te· neh²er im Inland g³gen E³tg³tt ® m R|hmen seines Un¹erneh²en,eau,. A red arrow points from the text "Se determina correctamente la longitud de la clave:32 ." to the key length value "32" in the hex dump pane.

- la contraseña introducida no tenía 32 bytes de longitud.
⇒ PSION Word deduce la clave actual de la contraseña.
- El post-procesado manual produce el texto cifrado (no se muestra)

Ejemplos (4)

PDA PSION 5 – determinar los bytes de clave restantes

Copiar la clave en el porta papeles durante el análisis automático

En análisis automático en hex dump,

- Determinar las posiciones de bytes incorrectos, p.ej. 0xAA en la posición 3
- Adivinar y escribir los bytes correctos: “e” = 0x65

En un archivo inicial hex dump cifrado

- Determinar los bytes iniciales de las posiciones de bytes calculadas: 0x99
- Calcular los bytes correctos de la clave con CALC.EXE: $0x99 - 0x65 = 0x34$

Clave desde el portapapeles

- Corrección 12865B~~34~~1498872C393E43741396A45670235E111E907AB7C0841...
- Descifrar el documento inicial cifrado utilizando la suma binaria
- bytes en la posición 3, 3+32, 3+2*32, ... Ahora son correctos

Address	Hex	Dec	Text
00000	65 72 67 65 73 65 74 7A 20 28 55 53 74 47 29 06	97 72 67 65 73 65 74 7A 20 28 55 53 74 47 29 06	ergesetz (UStG)
00010	06 06 8A 72 73 74 65 72 65 41 62 B8 A8 68 6E AE	10 06 8A 72 73 74 65 72 65 41 62 B8 A8 68 6E AE	rstereAb, "hn®
00020	74 74 06 53 74 65 75 65 72 67 65 67 65 6E 73 74	74 74 06 53 74 65 75 65 72 67 65 67 65 6E 73 74	tt. Steuiergegenst

Ejemplos (5)

clave DES débil

Cifrando 2 veces con esta clave volvemos a obtener el texto claro

CryptTool 1.4.30 - Ej_DES_Debil.txt

Archivo Edición Ver Cifrar/Descifrar Firma Digital/PKI Procedimientos Indiv. Análisis Opciones Ventana Ayuda

Simétrico (clásico) ► IDEA...
Simétrico (moderno) ► RC2...
Asimétrico ► RC4...
Híbrido ► DES (ECB)...
DES (CBC)...
Triple DES (ECB)...
Triple DES (CBC)...
Rijndael (AES)... Shift + Strg + R
Otros Algoritmos...
AES (autoextracción)...

Ej_DES_Debil.txt
DEMONSTRACION CLAVE DES DEBIL

Longitud de la clave: 56 bit
01 01 01 01 01 01 01 01

Cifrar Descifrar Cancelar

L:1 C:30 P:30 NUM

CryptTool 1.4.30 - DES (ECB) cifrado de <DES (ECB) cifrado de <Ej_DES_Debil.txt>, clave <01 01 01 01 01 01 01 01>, clave <01 01 01 01 01 01 01 01>

Archivo Edición Ver Cifrar/Descifrar Firma Digital/PKI Procedimientos Indiv. Análisis Opciones Ventana Ayuda

DEMONSTRACION CLAVE DES DEBIL

DES (ECB) cifrado de <Ej_DES_Debil.txt>, clave <01 01 01 01 01 01 01 01>
00000000 9E AD 94 FB 2E 6F 42 7D 22 5E C8 5A 91 4B 78 54 F9 3B 04 C2 3D 95 6B 5A ...oB}^Z.KxT;=.Z.R[...

DES (ECB) cifrado de <DES (ECB) cifrado de <Ej_DES_Debil.txt>, clave <01 01 01 01 01 01 01 01>, clave <01 01 01 01 01 01 01 01>
00000000 44 45 4D 4F 4E 53 54 52 41 43 49 4F 4E 20 43 4C 41 56 45 20 44 45 53 20 DEMONSTRACION CLAVE DES DEBIL...|

Pulse F1 para consultar la Ayuda.

L:2 C:9 P:33 NUM

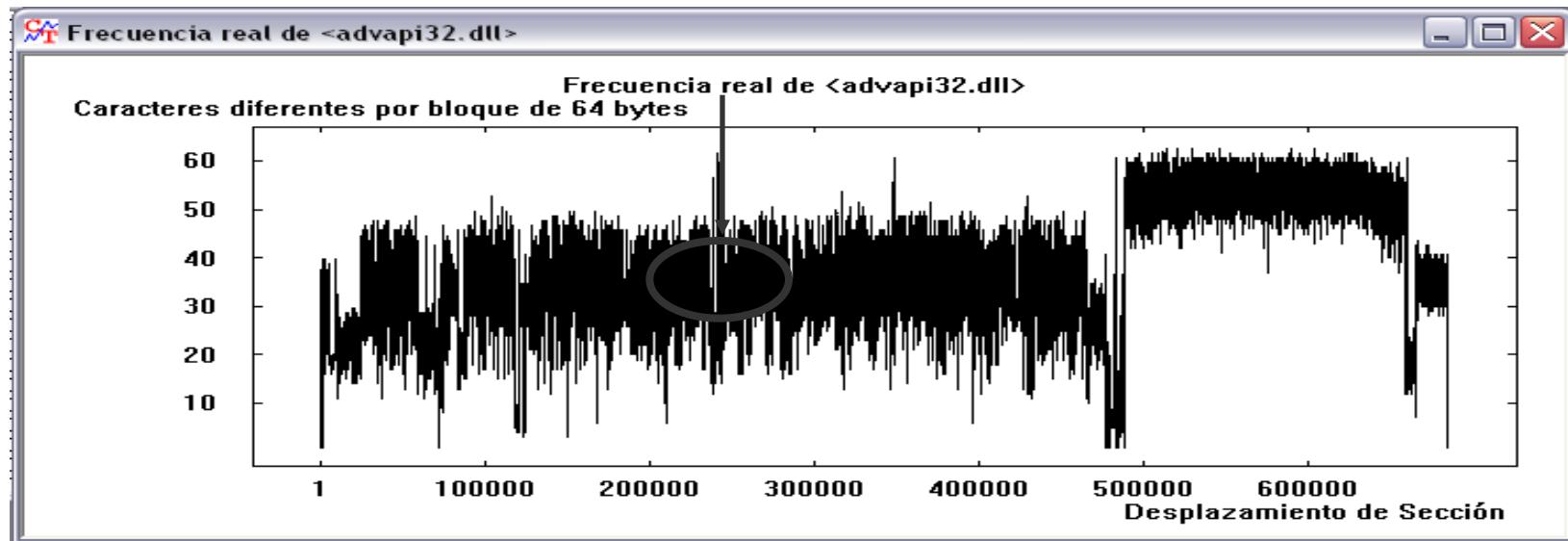
Ejemplos (6)

Localizar información de la clave

La función “Frecuencia real” es adecuada para localizar la información sobre la clave y áreas cifradas en archivos.

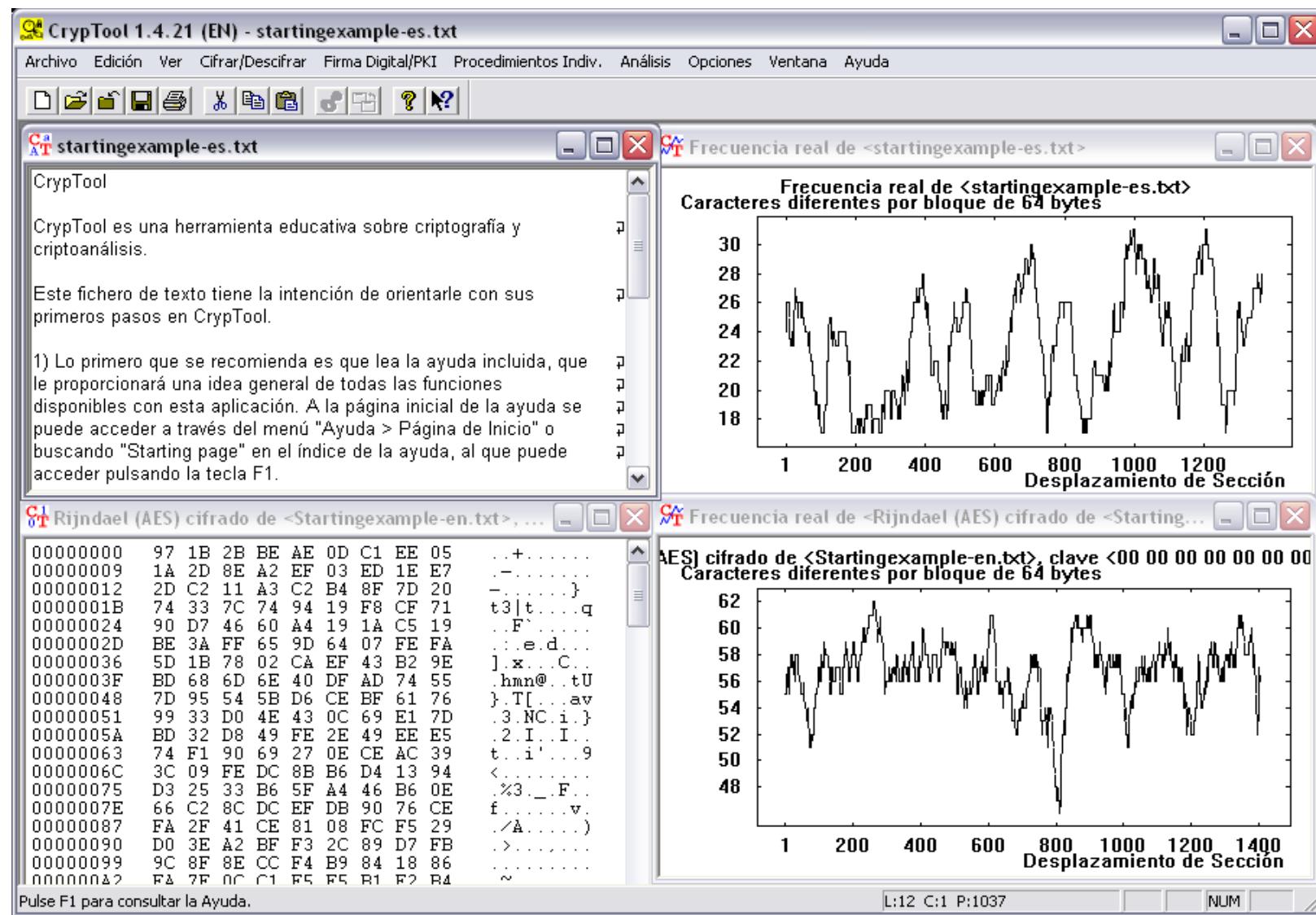
Trasfondo:

- Los datos de la clave son “más aleatorios” que el texto o el código programado
- Se puede reconocer como los picos en la “frecuencia real”
- Ejemplo: la “clave NSA” de advapi32.dll (Windows NT)



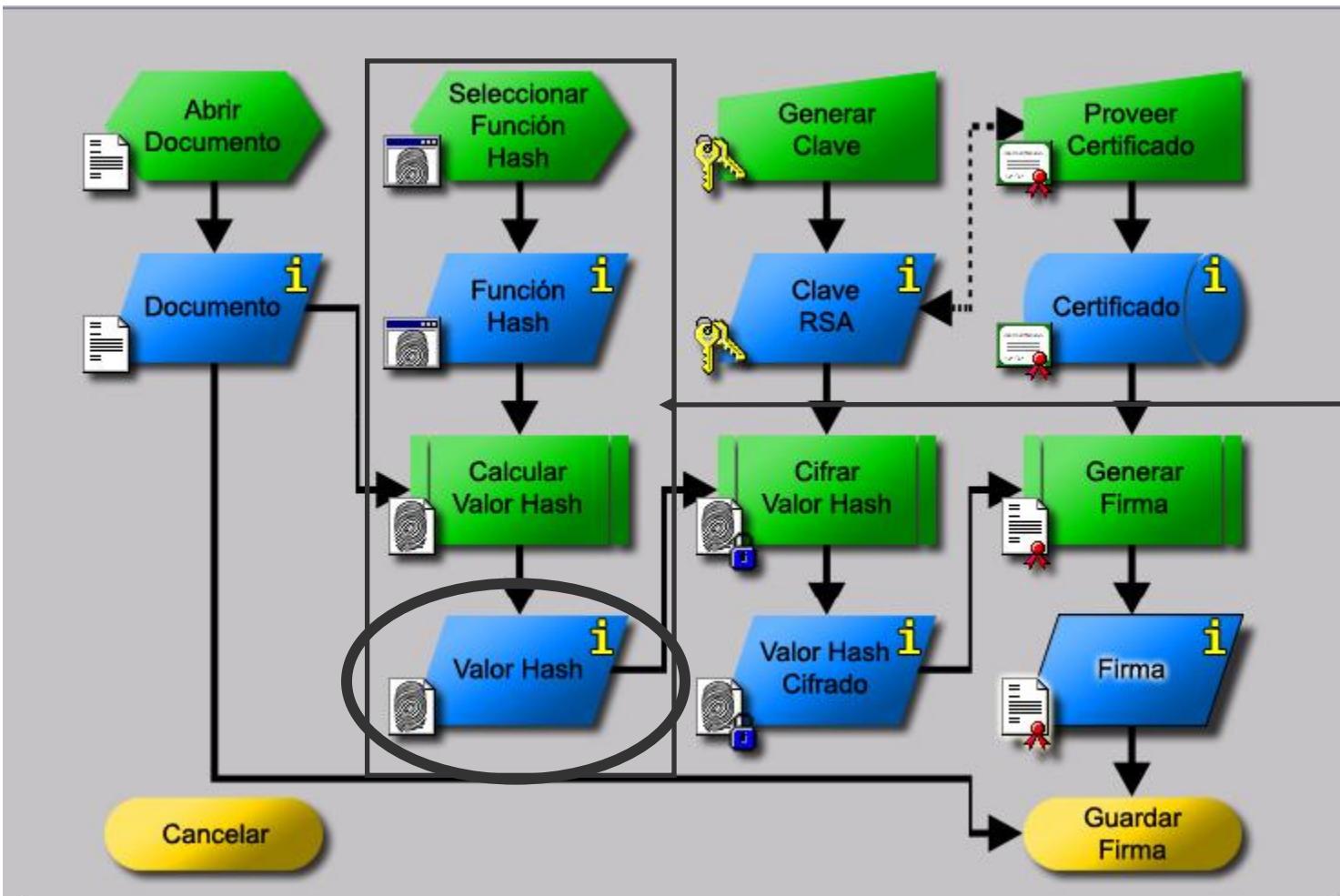
Ejemplos (6)

Comparación de frecuencia real con otros archivos



Ejemplos (7)

Ataque a la firma digital



Ataque:

¡Encontrar dos mensajes con el mismo valor hash!

Menú: "Análisis" \ "Hash" \ "Ataque al valor Hash de una firma digital"

Ejemplos (7)

Ataque a la firma digital– idea (I)

Ataque a la firma digital de un texto en ASCII basado en la búsqueda de colisiones.

Idea:

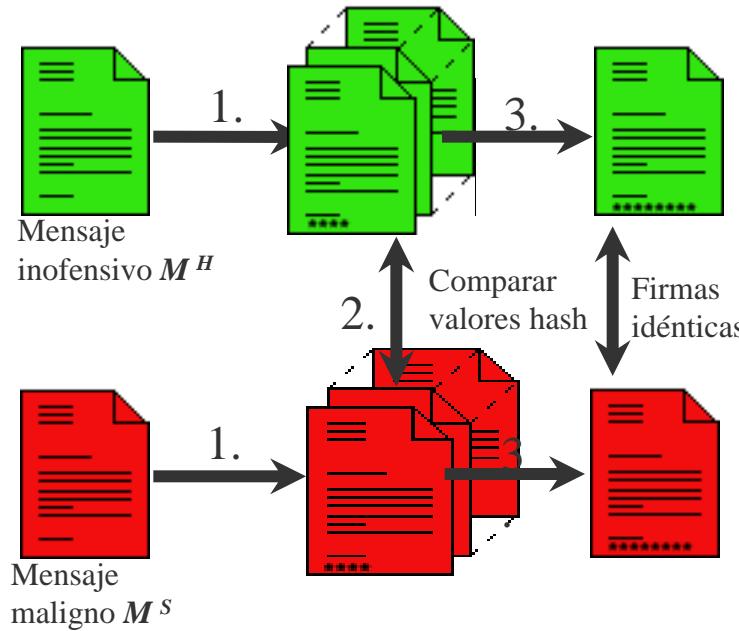
- Los textos ASCII se pueden modificar cambiando/insertando caracteres **no imprimibles**, sin cambiar el contenido visible
- Modificar dos textos en paralelo hasta encontrar una colisión hash
- Aprovechar la paradoja del cumpleaños (ataque del cumpleaños)
- Ataque genérico aplicable también a las funciones hash
- Se puede ejecutar en paralelo en varias máquinas (no está implementado)
- Se ha implementado en CrypTool como parte de la tesis de un licenciado “*Métodos y herramientas para ataques a firmas digitales*” (alemán), 2003.

Conceptos:

- Mapeados
- Algoritmo de Floyd modificado (consumo de memoria constante)

Ejemplos (7)

Ataque a la firma digital– idea (II)



- Modificación:** empezando desde un mensaje M se crean N mensajes distintos M_1, \dots, M_N con el mismo “contenido” como M .
- Búsqueda:** encontrar mensajes modificados M_i^H y M_j^S con el mismo valor hash.
- Ataque:** las firmas de ambos documentos M_i^H y M_j^S son iguales.

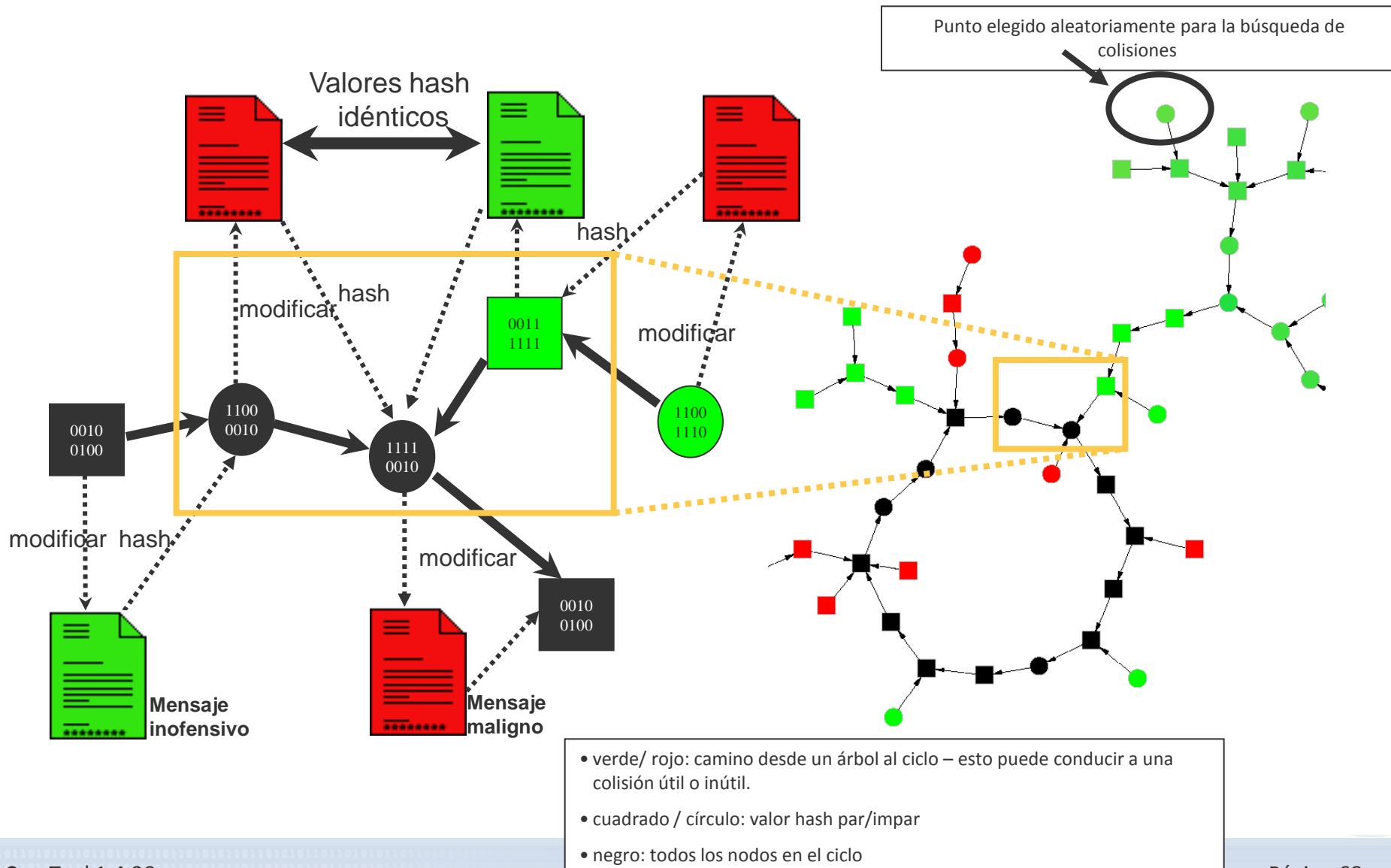
Conocemos por la paradoja del cumpleaños que para valores hash de longitud n en bits:

- buscar colisiones entre M^H y M_1^S, \dots, M_N^S : $N \approx 2^n$
- Buscar colisiones entre M_1^H, \dots, M_N^H y M_1^S, \dots, M_N^S : $N \approx 2^{n/2}$

Número estimado de mensajes generados para encontrar una colisión hash.

Localizar Colisiones Hash (1)

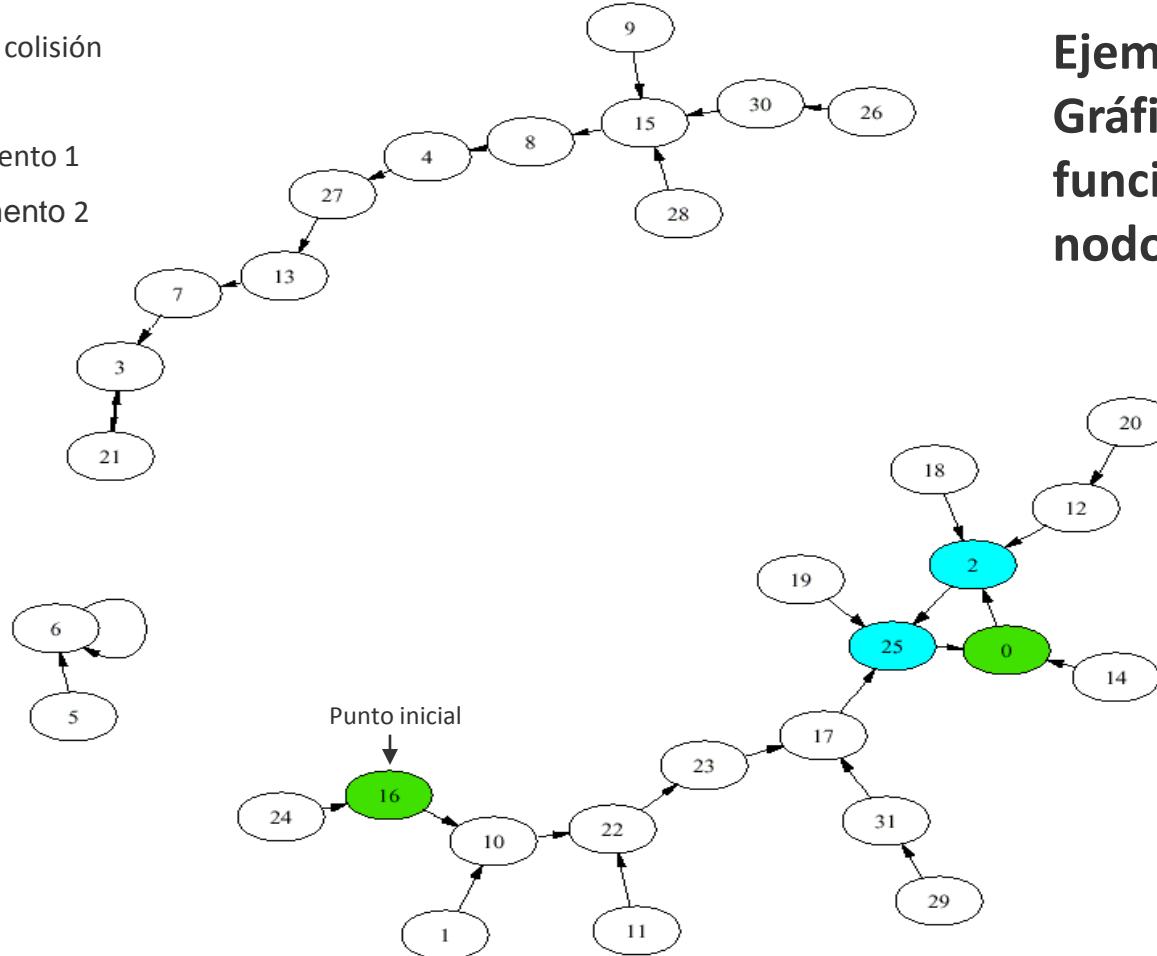
Mapeado por modificaciones del texto



Localizar Colisiones Hash (2)

Algoritmo de Floyd: encontrar el ciclo

-  inicio / colisión
-  ciclo
-  incremento 1
-  incremento 2



Ejemplo:
Gráfico de una
función con 32
nodos

Paso 1: Localizar el punto que concuerde con el ciclo:

- Dos series con idéntico punto de inicio[16]:
una serie con incremento 1, la otra con incremento 2.

Resultado (basado en teoría de grafos):

- Ambas series siempre terminan en un ciclo.
- Ambas series coinciden en un nodo en el ciclo (en este caso 0).

Localizar Colisiones Hash (3)

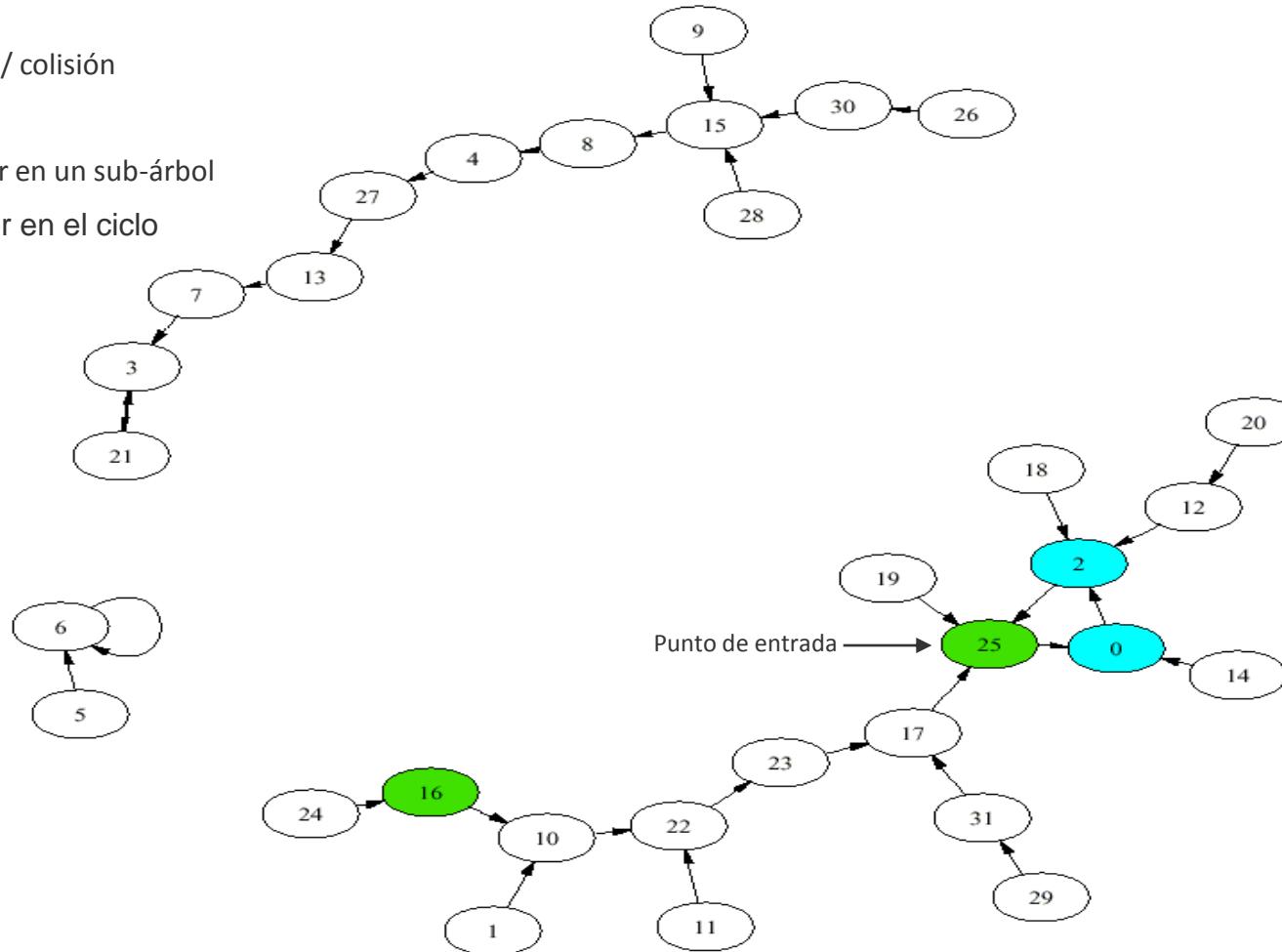
Seguir el ciclo (Extensión de Floyd): encontrar el punto de entrada

 inicio / colisión

 ciclo

 Mover en un sub-árbol

 Mover en el ciclo



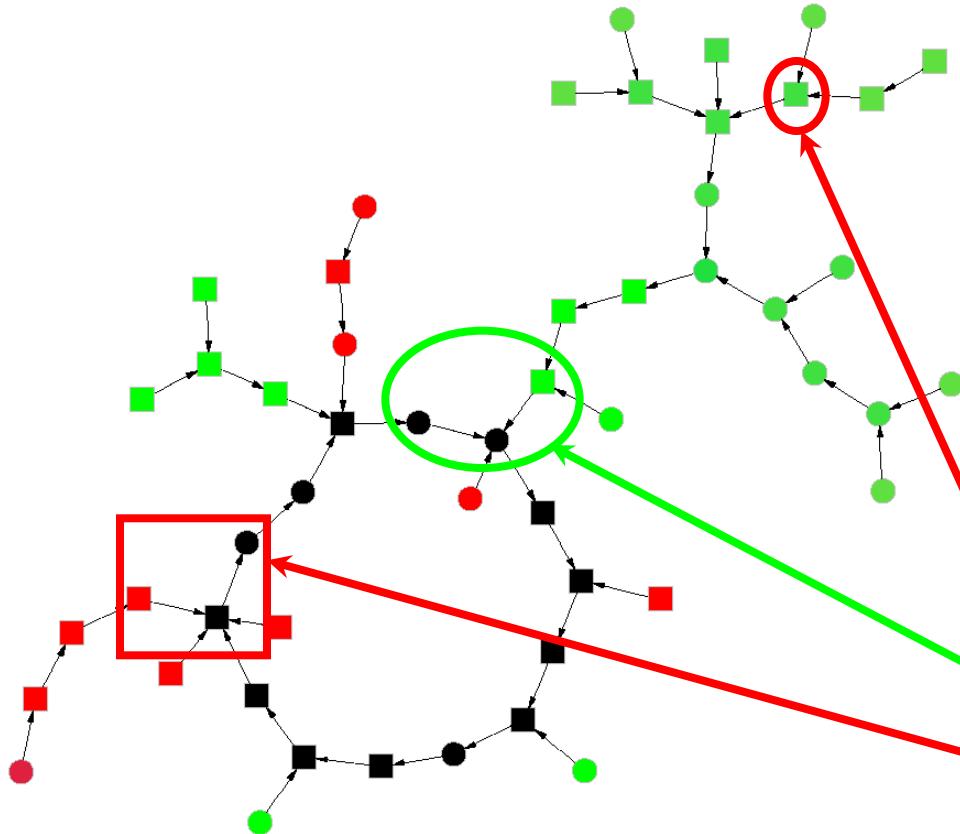
Paso 2: Localizar el punto de entrada de las series 1 en el ciclo [25]:

- La Serie 1 empieza otra vez desde el punto de entrada; la serie 3 con un incremento de 1 empieza en el punto de encuentro con el ciclo (en este caso en 0).

Resultado: Las series (1 y 3) coinciden en el punto de entrada del ciclo de la serie 1 (en este caso 25)

- Los predecesores (en este caso 17 y 2) resultan en una colisión hash.

Ataque de la Paradoja del Cumpleaños a la Firma Digital



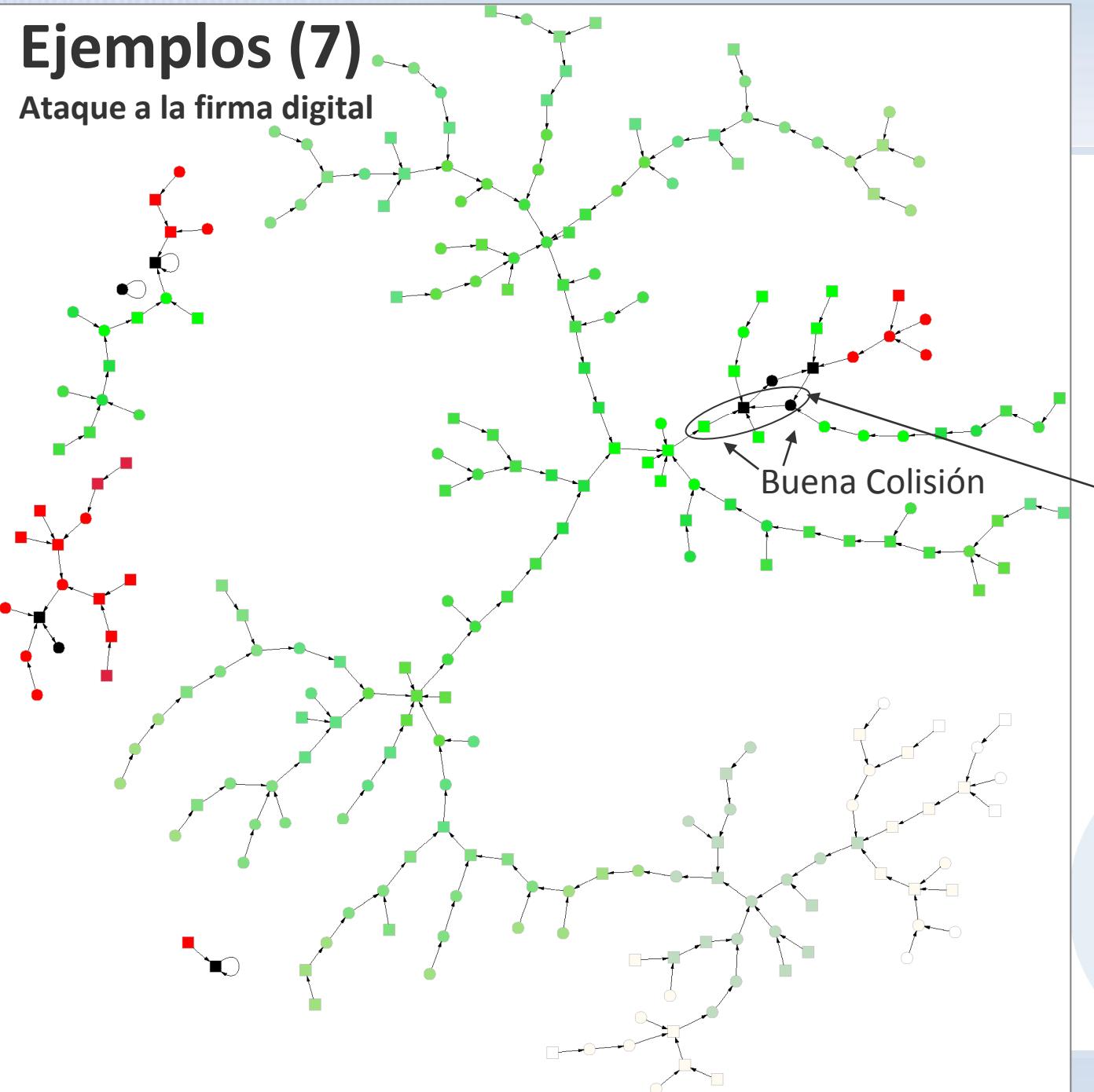
Examinar el algoritmo Floyd

- Presentación visual e interactiva del algoritmo Floyd (“Desplazándose a través del mapeo” en un ciclo).
- Adaptación del algoritmo de Floyd para un ataque de firma digital.

*El algoritmo de Floyd se implementa en CrypTool, pero la visualización del algoritmo aún no está implementada.

Ejemplos (7)

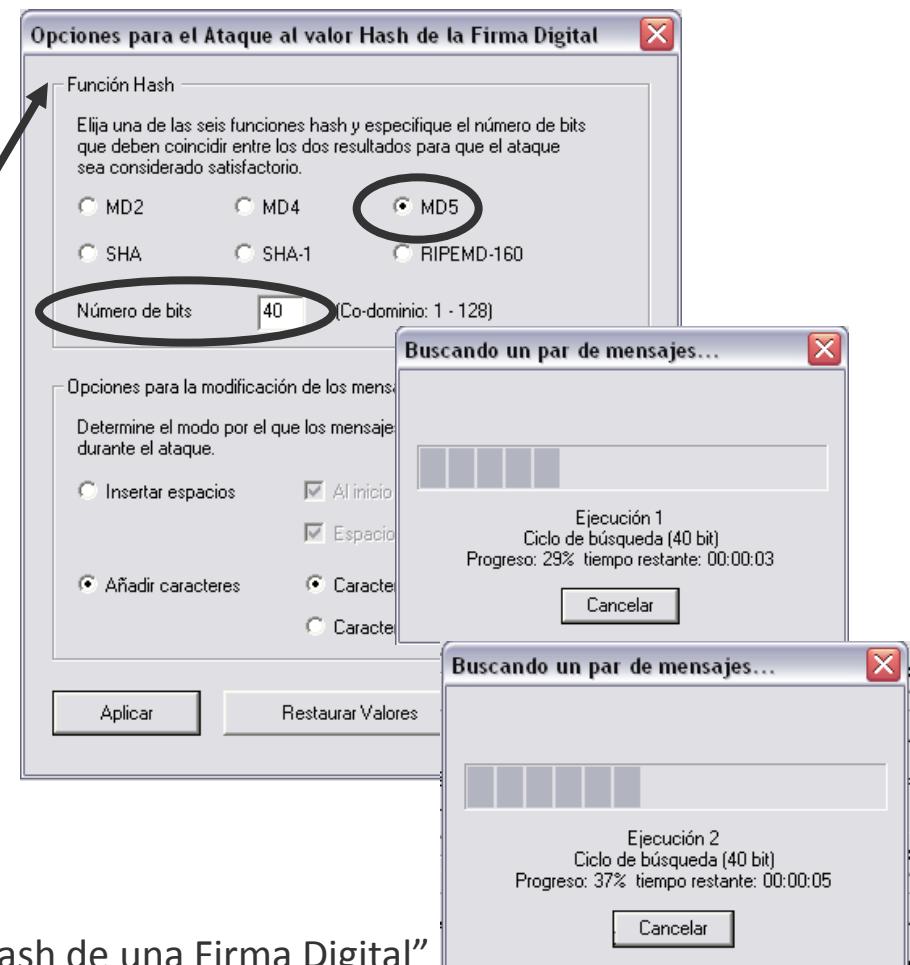
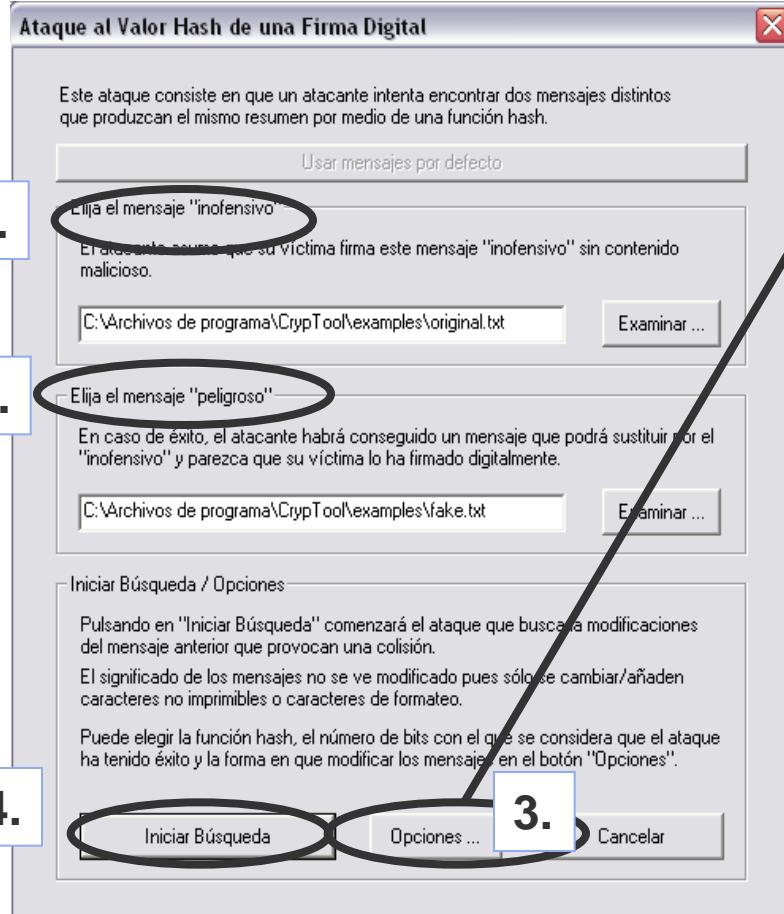
Ataque a la firma digital



Un ejemplo de un “buen” Mapeado (casi todos los nodos son verdes). En este grafo casi todos los nodos pertenecen al árbol grande, el cual se encuentra con el ciclo cuando se igualan los valores hash y donde el predecesor al punto de entrada en el ciclo es impar. Esto significa que el atacante encuentra útil la colisión para casi cualquier punto de inicio.

Ejemplos (7)

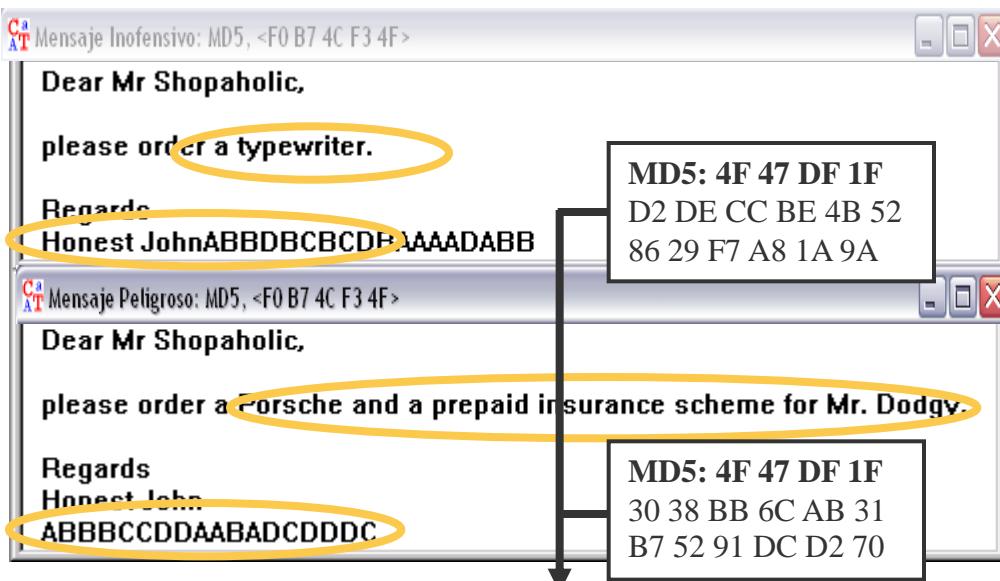
Ataque a la firma digital: Ataque



Menú: "Análisis" \ "Hash" \ "Ataque en el valor Hash de una Firma Digital"

Ejemplos (7)

Ataque a la firma digital: Resultados



Los primeros 32 bits de los valores hash son idénticos.

Adicionalmente al manejo manual:

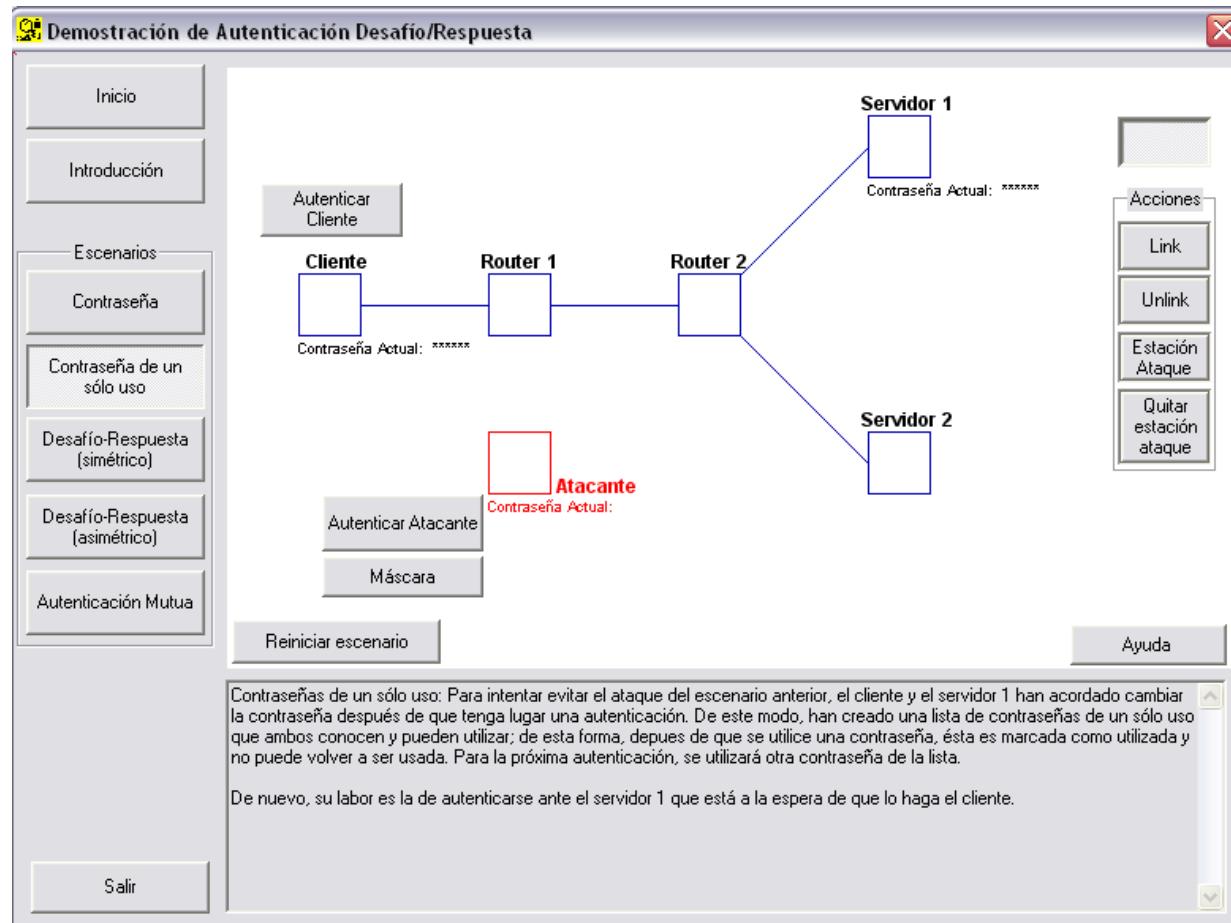
Característica automática desconectada en CrypTool: Ejecuta y registra los resultados para todos los conjuntos de configuraciones de parámetros. Disponible a través de la ejecución de CrypTool por línea de comandos.

Resultados Experimentales

- *Colisión parcial de 72 Bit* (igualdad para los valores has de los primeros 72 bits) se encontró en un par de días en un único PC.
- ¡Las firmas que utilizan valores hash de hasta 128 bits se pueden atacar hoy en día con búsqueda en paralelo!
- Utilizar valores has de como poco 160bits de longitud.

Ejemplos (8)

Autenticación en un entorno cliente-servidor

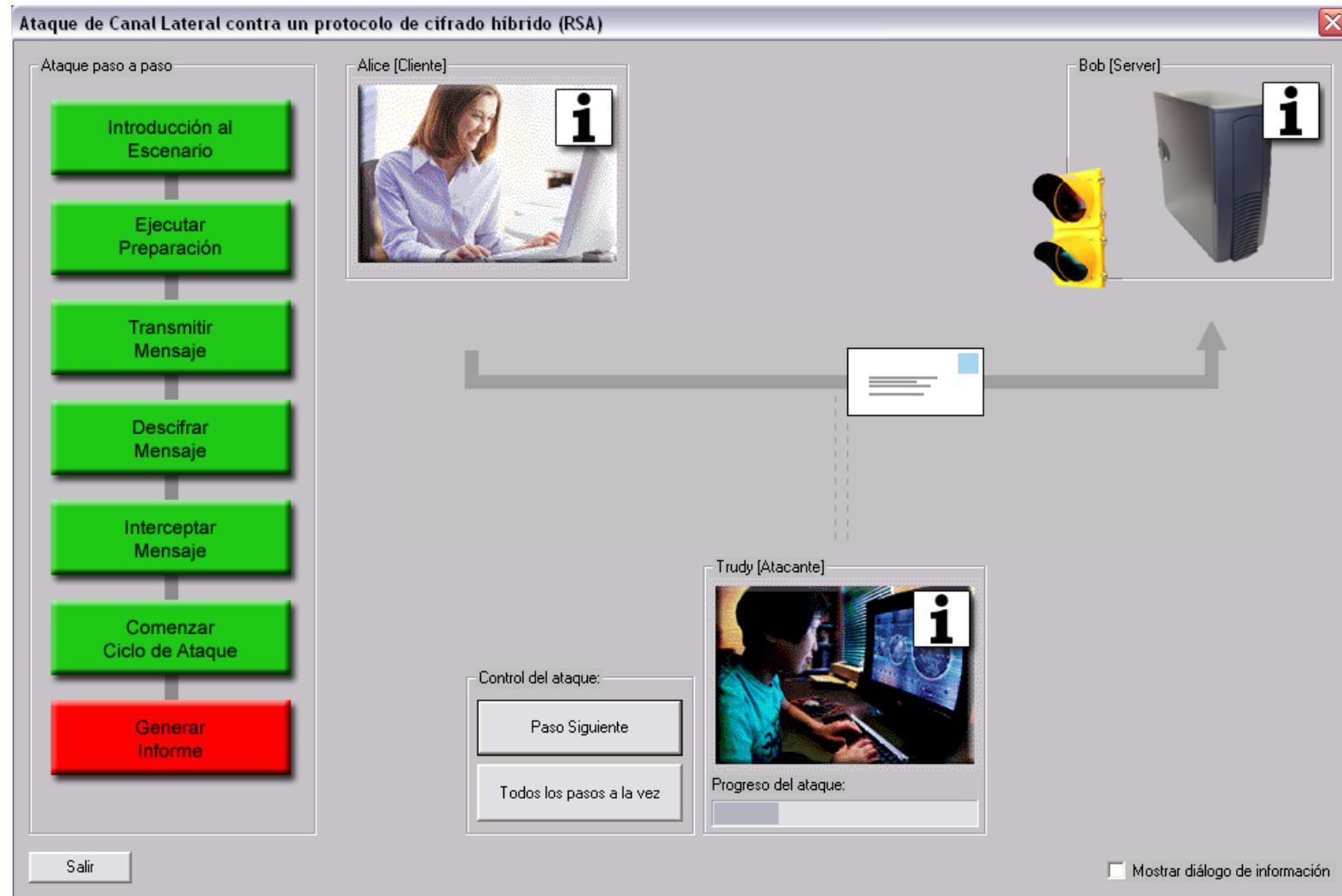


- Demostración interactiva para distintos métodos de autenticación.
- Oportunidades definidas del atacante.
- Puede tomar el papel del atacante.
- **Moraleja:**
Sólo es segura la autenticación mutua.

Menú: "Procedimientos Indiv." \ "Protocolos" \ "Autenticación en Red"

Ejemplos (9)

Demonstración de un ataque de canal lateral (en un protocolo de cifrado híbrido)



Menú: “Análisis” \ “Cifrado Asimétrico” \ “Ataque de canal lateral”

Ejemplos (9)

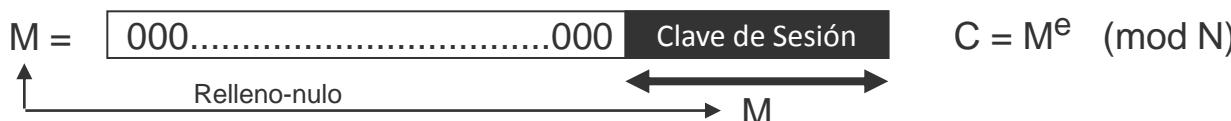
Idea para este ataque de canal lateral

Ulrich Kühn “*Side-channel attacks on textbook RSA and ElGamal encryption*”(2003)

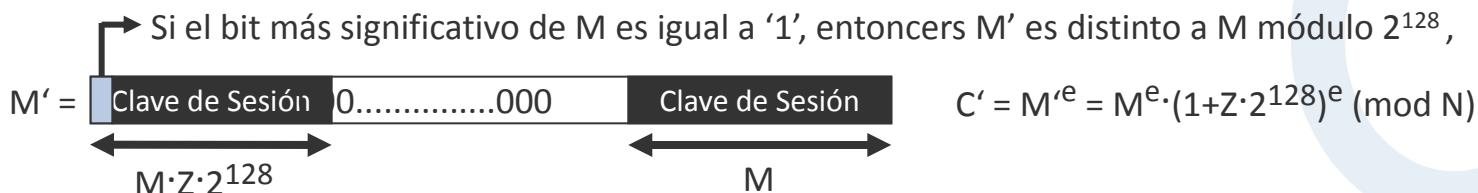
Prerrequisitos:

- Cifrado RSA: $C = M^e \pmod{N}$ y descifrado: $M = C^d \pmod{N}$.
- Las Claves de sesión de 128-Bits (en M) están „cifradas por diccionario“ (relleno nulo).
- El servidor conoce la clave secreta d y
 - La utiliza después de descifrar sólo los 128 bits menos significativos (sin validación de los bits 0 de relleno) (esto significa que el servidor no reconoce si hay algo distinto a cero).
 - Avisos y mensajes de error, si del intento de cifrado resulta una clave de sesión errónea (el texto descifrado no se puede interpretar en el servidor). En el resto de casos no habrá mensajes.

Idea para el ataque: Aproximación para Z en la ecuación $N = M * Z$ para cada $M = \lfloor |N/Z| \rfloor$



Se calculan de forma sucesiva todas las posiciones de bits para Z: En cada paso se toma un bit más. El atacante modifica C a C' (ver más abajo). Si ocurre un desbordamiento de bits mientras se calcula M' en el servidor (receptor), el servidor envía un mensaje de error. Basándose en esta información, el atacante obtiene un bit para Z.



Ejemplos (10)

Matemáticas: Ataques a RSA utilizando reducción de retículos (Lattice Reduction)

Ataque a claves demasiado cortas (según Bloemer / May)

Descripción
Este ataque permite factorizar un módulo N de RSA siempre que la clave secreta d elegida sea suficientemente pequeña en comparación con N. El número $\delta = \log(d)/\log(N)$ es llamado "tamaño de d". Este ataque es posible para deltas < 0.290.

Para aplicar ejemplos de la bibliografía, puede introducir la clave pública (N,e).
 Después introduzca el valor estimado de delta, o bien, puede introducir directamente el d que es utilizado para calcular delta.
 Para generar un ejemplo aleatorio introduzca los parámetros delta y la longitud (en bits) de N
Pulsando sobre 'Generar Clave Aleatoria' se generarán las claves.

Después pulse sobre

Paso 1: Introduzca la clave y sus parámetros

Longitud de N: delta: Parámetros por defecto

N:
e:
d:

Generar clave RSA aleatoria

Paso 2: Introduzca parámetros para la reducción de la celosía

m: Determina el tamaño de la celosía a reducir y el tamaño máximo de delta. Debe ser al menos 4.
t: Se calcula óptimamente en función de m.
Dimensión de la celosía: Tamaño a reducir de la celosía. Tiene un impacto importante en el tiempo
Maximal delta: Maximal size of delta for big N (N>1000 Bit).

Paso 3: Iniciar Ataque

Construyendo: Reduciendo: Reducciones: Iniciar
Calculando resultado: Resultante: Cancelar
Tiempo Total:

Factorización
p: q:

■ Muestra cómo se tienen que elegir los parámetros del método RSA, por eso el algoritmo resiste al ataque de reducción de retículos descrito en la bibliografía.

■ 3 variantes

1. El exponente secreto d es demasiado pequeño en comparación con N.
2. Se conoce parcialmente a uno de los factores de N.
3. Se conoce una parte del texto claro.

■ Estas suposiciones son realistas.

Menú: "Análisis" \ "Cifrado Asimétrico" \ "Colección de ataques basados en RSA" \ ...

Ejemplos (11)

Análisis de Aleatoriedad con visualización 3-D

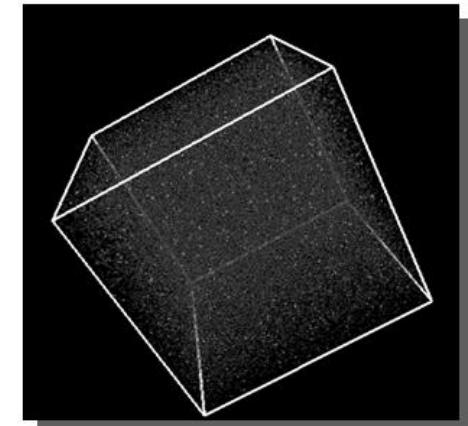
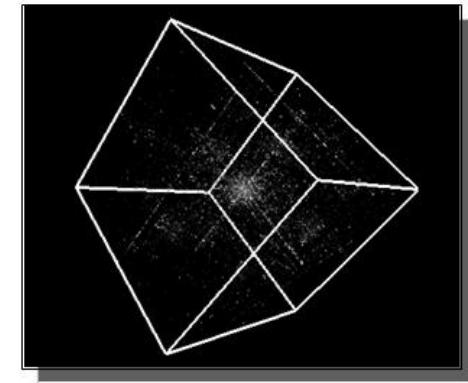
Visualización 3-D para el análisis de aleatoriedad

Ejemplo 1

- Abrir un archivo arbitrario (p.ej. Un informe en Word o una presentación de PowerPoint)
- Se recomienda seleccionar un archivo de al menos 100 Kb
- Análisis 3-D utilizando en menú: „Análisis“ \ „Análisis de Aleatoriedad“ \ „Visualización 3-D“
- Resultado: **se reconocen fácilmente las estructuras**

Ejemplo 2

- Generación de números aleatorios: „Procedimientos Indiv.“ \ „Herramientas“ \ „Generar Números Aleatorios“
- Se recomienda generar al menos 100.000 bytes aleatorios
- Análisis 3-D utilizando en menú: „Análisis“ \ „Análisis de Aleatoriedad“ \ „Visualización 3-D“
- Resultado: **distribución uniforme (no se reconocen las estructuras)**

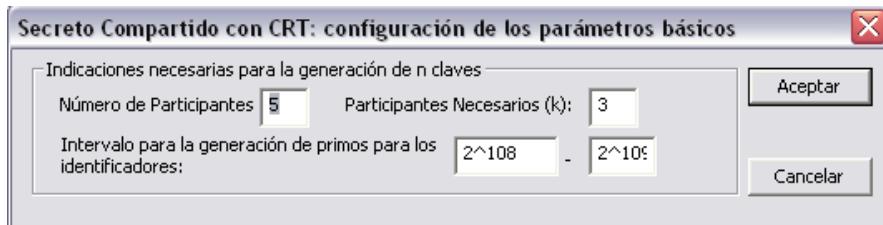


Ejemplos (12)

Secreto Compartido con CRT – Implementación del Teorema Chino de los Restos (CRT)

Ejemplo de Secreto compartido (1):

- **Problema:**
 - 5 personas tienen una clave
 - Para ganar acceso al menos 3 de esas 5 personas tienen que estar presentes
- “**Opciones**” permite configurar los detalles del método.



- “**Pasos de Calc.**” muestra todos los pasos para generar la clave.

Menú: “Procedimientos Indiv.” \
“Aplicaciones del Teorema Chino de los Restos”\
“Secreto Compartido por CRT”

Compartir un secreto con CRT permite distribuirlo en n claves diferentes. Para reconstruir ese secreto es necesario reunir un número $k \leq n$ de claves.

Para generar y distribuir el secreto, pulse sobre el botón "Calcular". Con el botón "Opciones" puede modificar el número de identificadores y cuántas partes serán necesarias para reconstruir el secreto. Para reconstruir el secreto, puede usar los botones '+/-' para añadir/eliminar las claves que se utilizarán. Después, pulse en el botón 'Reconstruir Secreto'.

Construyendo y compartiendo el secreto con $n = 5$ partes

Identificador (= Número Primo)	Contraseña (= Clave)	Generación:
372809584840037959915511708765699	130764953420472647725315864620287	<input checked="" type="radio"/> automática
405804884216408345799507156818917	258906749313015609616340007840535	<input type="radio"/> manual
487583282637495446535122914399147	407073388372430803921691018095695	
538840152093869636808076397599027	238966047870751761470720771347665	
602979550252885254443545493977093	589444215642371412299977068987231	
		Calcular
		Opciones
		Proceso Calc.
		Reiniciar

Reconstruyendo el secreto con un mínimo de $k = 3$ claves
Elija algunos de estos $n = 5$ identificadores, cuyas claves serán usadas para reconstruir el secreto:

1: + / -	130764953420472647725315864620287
2: + / -	258906749313015609616340007840535
3: + / -	407073388372430803921691018095695
4: + / -	
5: + / -	
6: + / -	
7: + / -	

Mostrar Introducción al Iniciar

Reconstruir Secreto

Ayuda Fichero Log Salir

Ejemplos (12)

Secreto Compartido de Shamir

Ejemplo Secreto Compartido (2)

■ Problema

- Un valor secreto se puede separar para n personas.
- t de las n personas se necesitan para recuperar el valor secreto K.
- (t, n) esquema de borde

■ Menú: “Procedimientos Indiv.” \

“Demostración Secreto Compartido (Shamir)”

1. Introducir el secreto K, número de personas n y el umbral t

2. Generar polinomio

3. Utilizar Parámetros

■ Utilizando “Reconstrucción” se puede recuperar el secreto

Secreto Compartido : Inicializando el esquema umbral

Por definición de un esquema Shamir (t, n), un secreto puede ser distribuido en n personas. Despues de esto, se necesitarán al menos t de esas personas ($t \leq n$) para reconstruir el secreto original combinando sus secretos. Para configurar el esquema, se debe generar un polinomio $f(x)$ de grado máximo $t-1$ (con $t-1$ coeficientes elegidos aleatoriamente) y un primo aleatorio p . Cada participante recibe un valor público x elegido aleatoriamente y su secreto se corresponde con el valor $y=f(x)$. Para obtener más información puede consultar la ayuda pulsando la tecla F1.

Elija un secreto y determine los parámetros para configurar un esquema

Secreto S con $S \geq 0$	18	Parámetros por defecto
Número de participantes n con $n > 0$	10	Opciones
Umbral (mínimo) t con $t > 0$	4	

Generar polinomio Editar polinomio

Parámetros concernentes el polinomio $f(x)$ de grado $t-1$

Todos los cálculos tienen lugar en el espacio discreto $GF(p)$

Polinomio $f(x)$ $18+4974x+4035x^2+492x^3$

Primo p 5927

Valor de los participantes, calculado a partir de los parámetros:

Participants	Valor público x	Parte [valor secreto f(x)]
<input checked="" type="checkbox"/> participante 1	2996	1069
<input type="checkbox"/> participante 2	89	3665
<input checked="" type="checkbox"/> participante 3	4828	5009
<input type="checkbox"/> participante 4	5437	3696
<input checked="" type="checkbox"/> participante 5	2757	3378
<input type="checkbox"/> participante 6	2751	3808
<input type="checkbox"/> participante 7	3903	4625
<input type="checkbox"/> participante 8	154	5666

Seleccione entre los participantes aquellos que reconstruiran el secreto.

Mostrar información al inicio.

Ejemplos (13)

Implementación del CRT para resolver sistemas de ecuaciones modulares lineales

Escenario en astronomía

- ¿Cuánto tiempo tiene que pasar hasta que un número dado de planetas (con distintos períodos de rotación) se alineen?
- El resultado es un sistema lineal de ecuaciones modulares, que se puede resolver con el Teorema Chino de los Restos (CRT).
- En esta demostración se pueden introducir hasta 9 ecuaciones y calcular una solución utilizando el CRT.

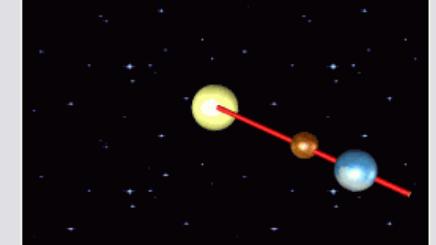
Ejemplo de uso: Visualización del Teorema Chino de los Restos aplicado en la Astronomía - Movimiento Planetario

Utilizando el Teorema Chino de los Restos (TCR) se pueden resolver sistemas de congruencias. En este ejemplo puedes introducir hasta 9 ecuaciones de la forma: $x = a[i] \text{ mod } m[i]$ ($i=1, \dots, 9$); que serán utilizadas para calcular el tiempo que tardan los planetas en alinearse

Sistema de Congruencias

$x \equiv 15$	mod 88
$x \equiv$	mod
$x \equiv 100$	mod 365
$x \equiv$	mod
$x \equiv 0$	mod 4327
$x \equiv$	mod
$x \equiv$	mod
$x \equiv 0$	mod 60149
$x \equiv$	mod

Ejemplo de uso en Astronomía (visualización)



El periodo de los planetas Mercurio y Tierra alrededor del Sol es de 88 y 365 días. Hasta alcanzar el rayo s (en rojo) quedan:

15 y 100 días.

¿Podría ocurrir que Mercurio y la Tierra volvieran a coincidir sobre el rayo s?

Elige un Planeta

<input checked="" type="checkbox"/> Mercurio	<input type="checkbox"/> Marte	<input type="checkbox"/> Urano
<input type="checkbox"/> Venus	<input checked="" type="checkbox"/> Júpiter	<input checked="" type="checkbox"/> Neptuno
<input checked="" type="checkbox"/> La Tierra	<input type="checkbox"/> Saturno	<input type="checkbox"/> Plutón

Intervalo de tiempo (en días) hasta que se repita el incidente

126.228.390.655

Resolver Salir

Borrar Todos los Parámetros Restaurar los valores por defecto

Menú: "Procedimientos Indiv." \ "Aplicación del Teorema chino de los restos" \ "Astronomía y Movimiento planetario"

Ejemplos (14)

Visualización de métodos de cifrado simétrico utilizando ANIMAL (1)

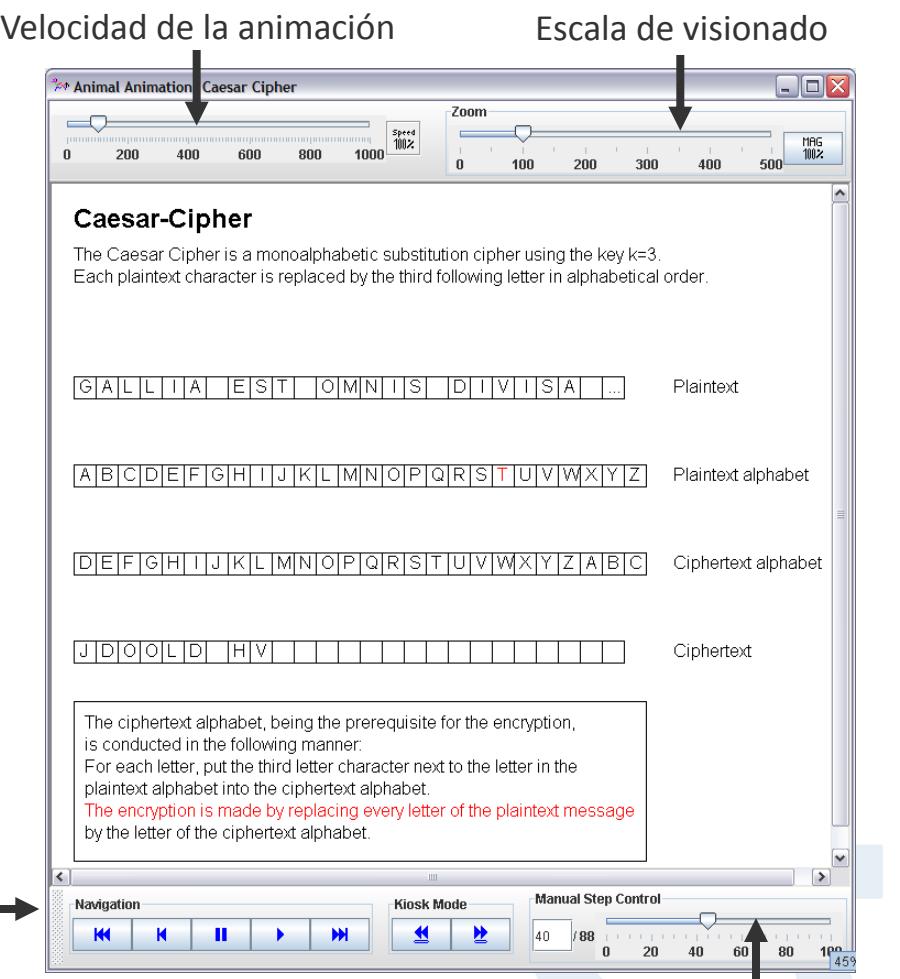
Visualizaciones animadas de varios algoritmos simétricos

- César
- Vigenère
- Nihilist
- DES

CrypTool

- Menú: “Procedimientos Indiv.” \ “Visualización de algoritmos” \ ...
- Control de la animación interactivo utilizando los controles integrados en la ventana.

Controles de la
animación (siguiente,
atrás, pausa, etc.)

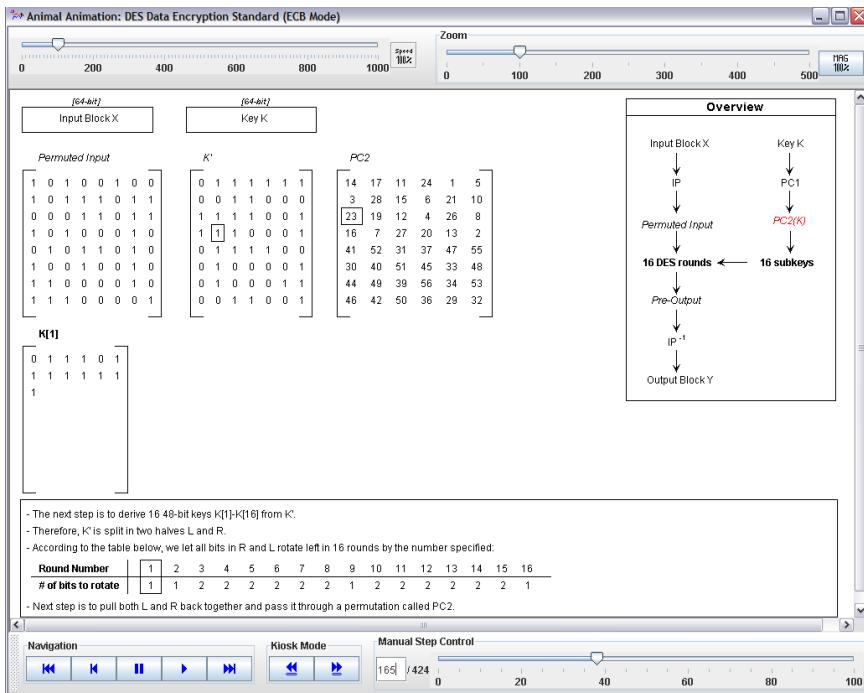


Selección directa de un paso de la animación

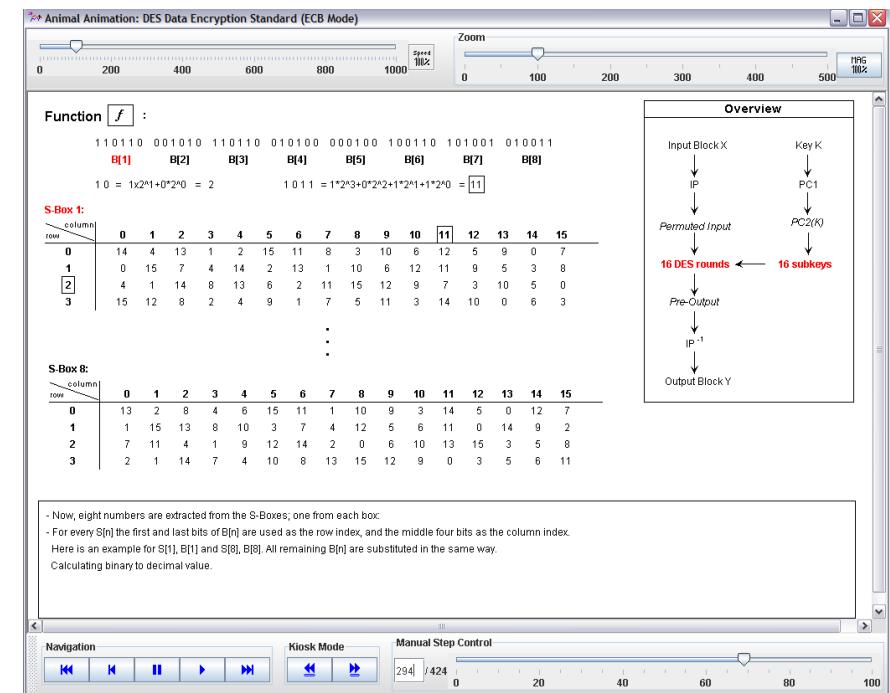
Ejemplos (14)

Visualización de métodos de cifrado simétrico utilizando ANIMAL (2)

Visualización del cifrado DES



Después de la permutación del bloque de entrada utilizando el vector de inicialización IV, la clave K se permuta con PC1 y PC2.



La función de núcleo **f** del DES, que se enlaza la mitad derecha del bloque R_{i-1} con la clave parcial K_i .

Ejemplos (15)

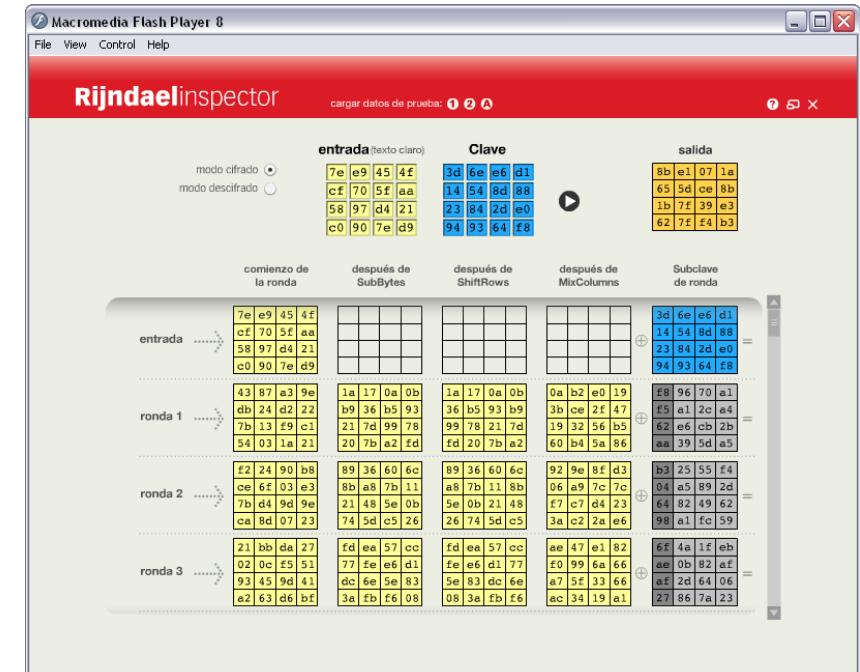
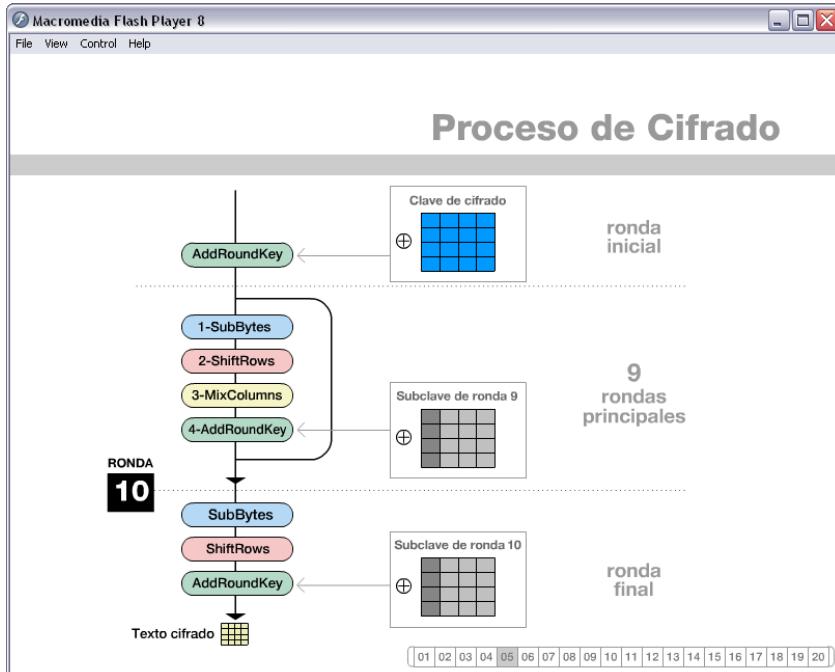
Visualización de AES (cifrado Rijndael)

Animación Rijndael (el cifrado Rijndael fue el ganador de la dependencia del AES)

- La visualización muestra la animación del proceso del cifrado basado en ciclos (utilizando datos fijos)

Inspector Rijndael

- El proceso de cifrado para probar (utilizando tus propios datos)



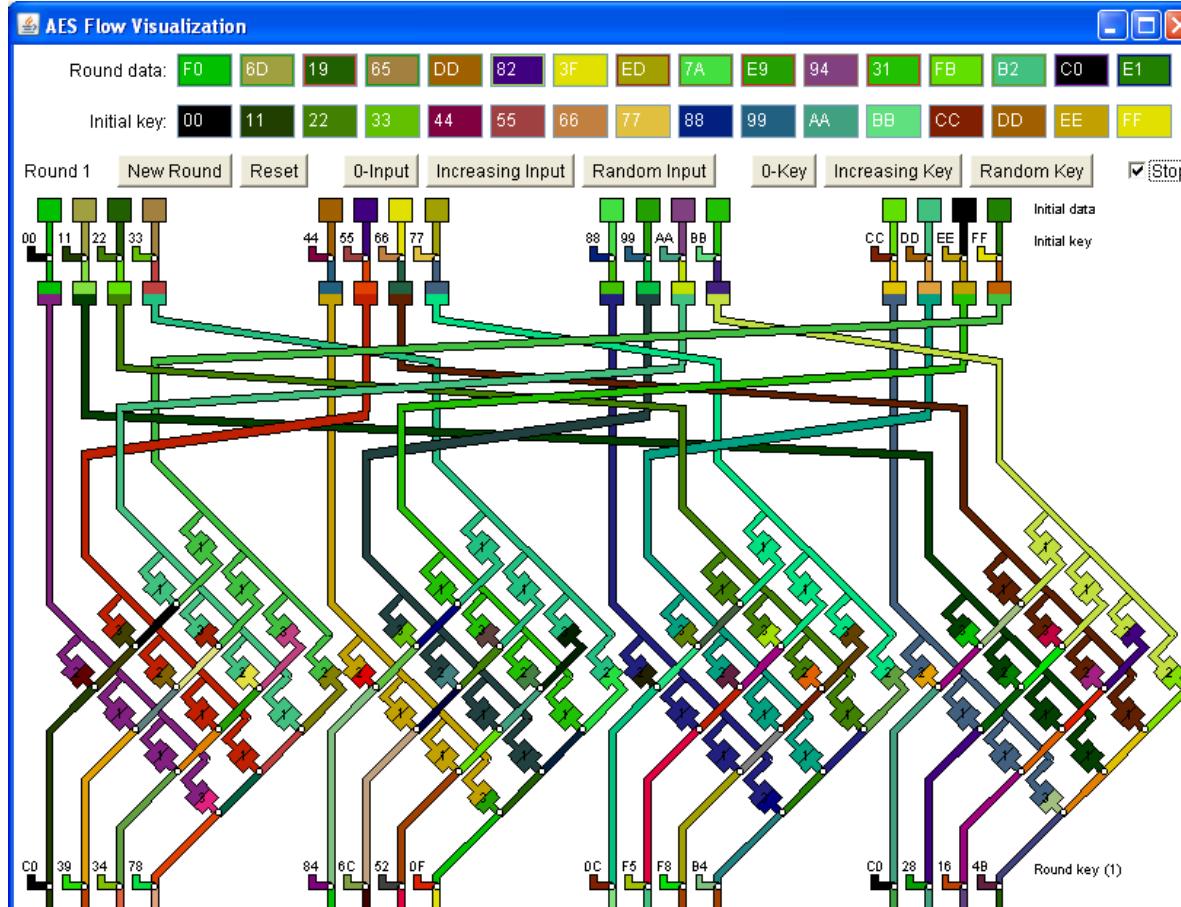
Menú: “Procedimientos Indiv.” \ “Visualización de Algoritmos” \ “AES” \ “Animación Rijndael” o “Inspector Rijndael”

Ejemplos (15)

Visualización de AES (cifrado Rijndael) – usando Java

Visualización de flujo de Rijndael

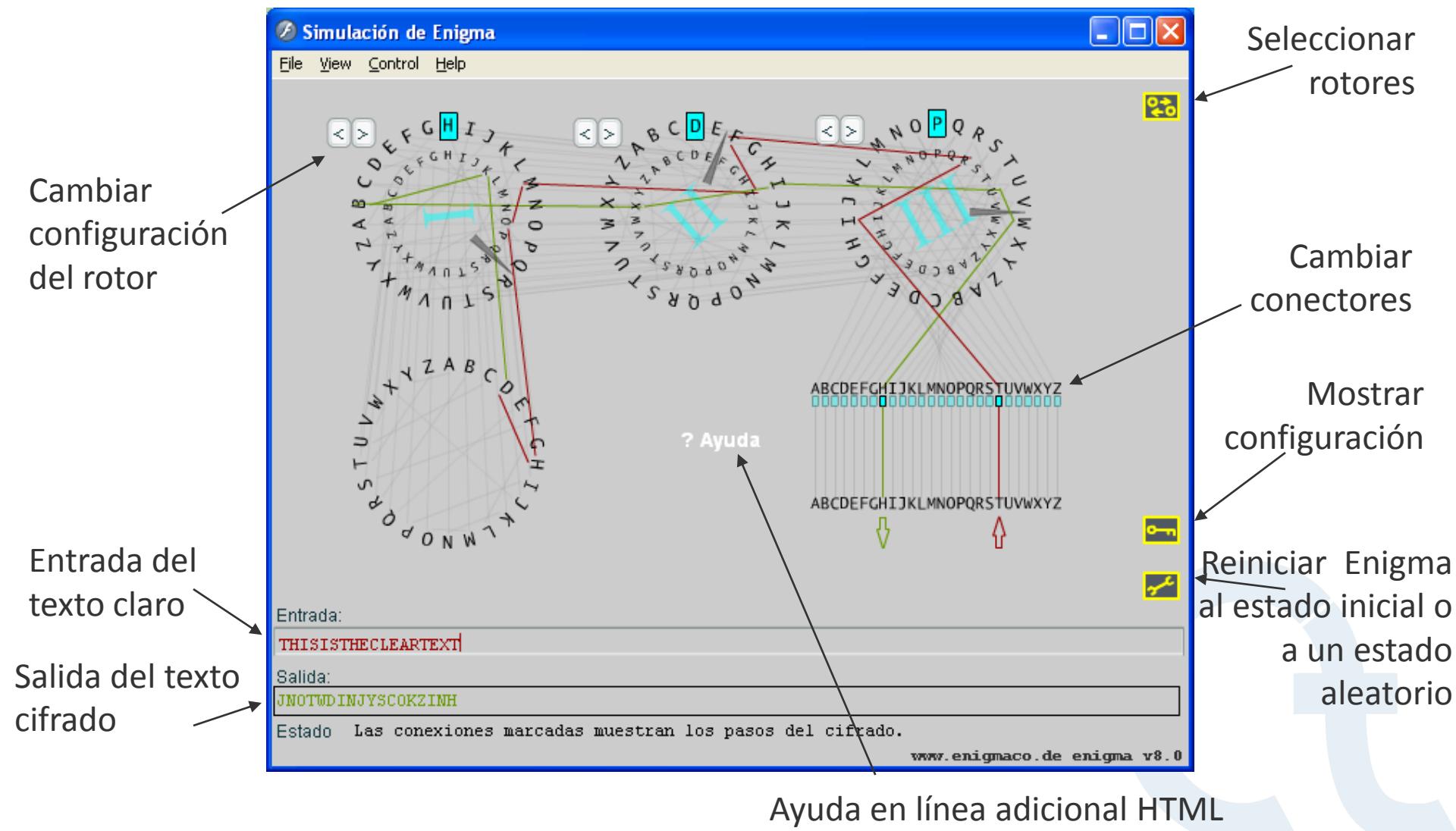
- Visualización de cambios por ronda a través de escala de colores.



Menú: "Procedimientos Indiv." \ "Visualización de Algoritmos" \ "AES" \ "Animación Rijndael ..."

Ejemplos (16)

Visualización del cifrado de Enigma

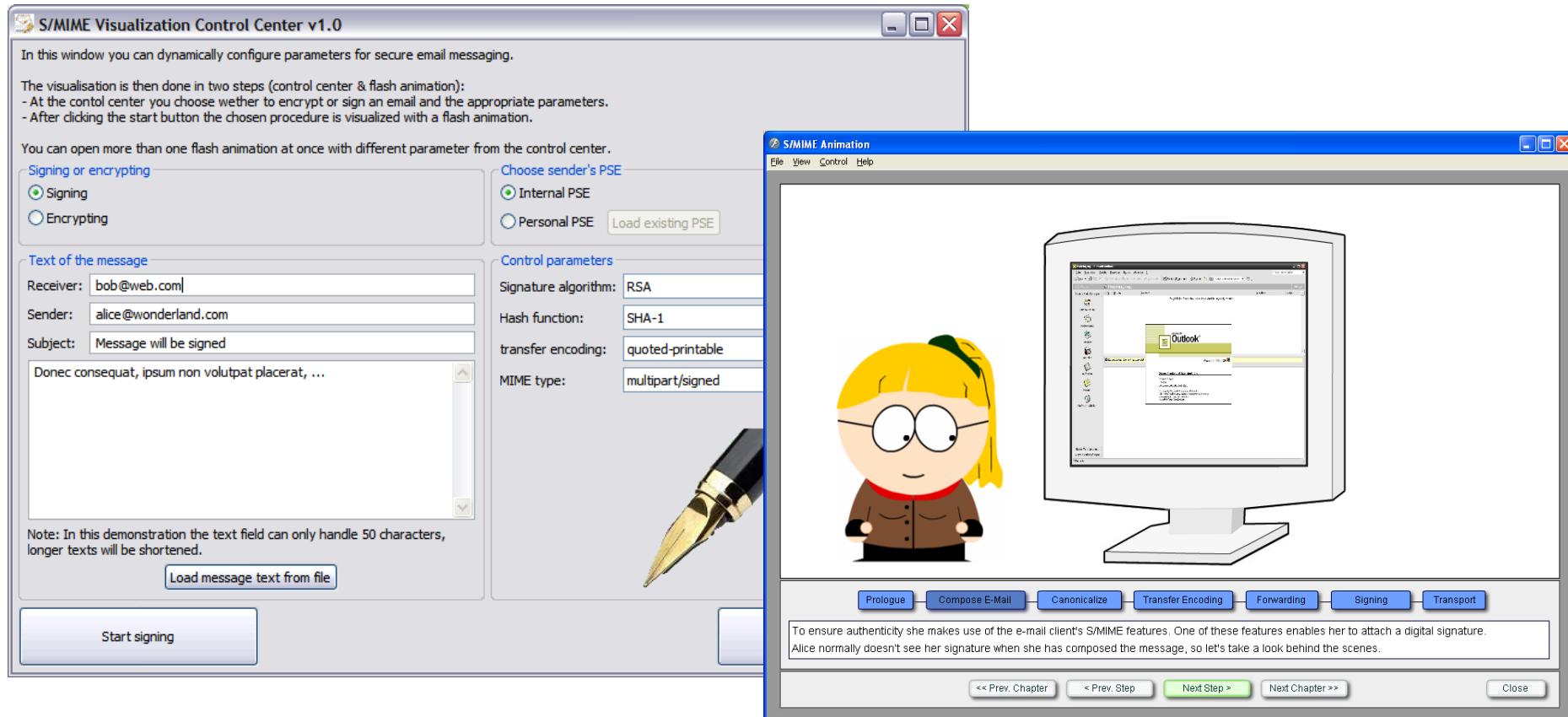


Ejemplos (17)

Visualización de E-Mail seguro usando S/MIME

Visualización S/MIME

- Centro de Control: Firmar/Cifrar mensajes con diferentes parámetros
- Animación: Desde la creación en el emisor hasta la lectura en el receptor



Menú: "Procedimientos Indiv." \ "Protocolos" \ "Seguridad E-Mail con S/MIME..."

Ejemplos (18)

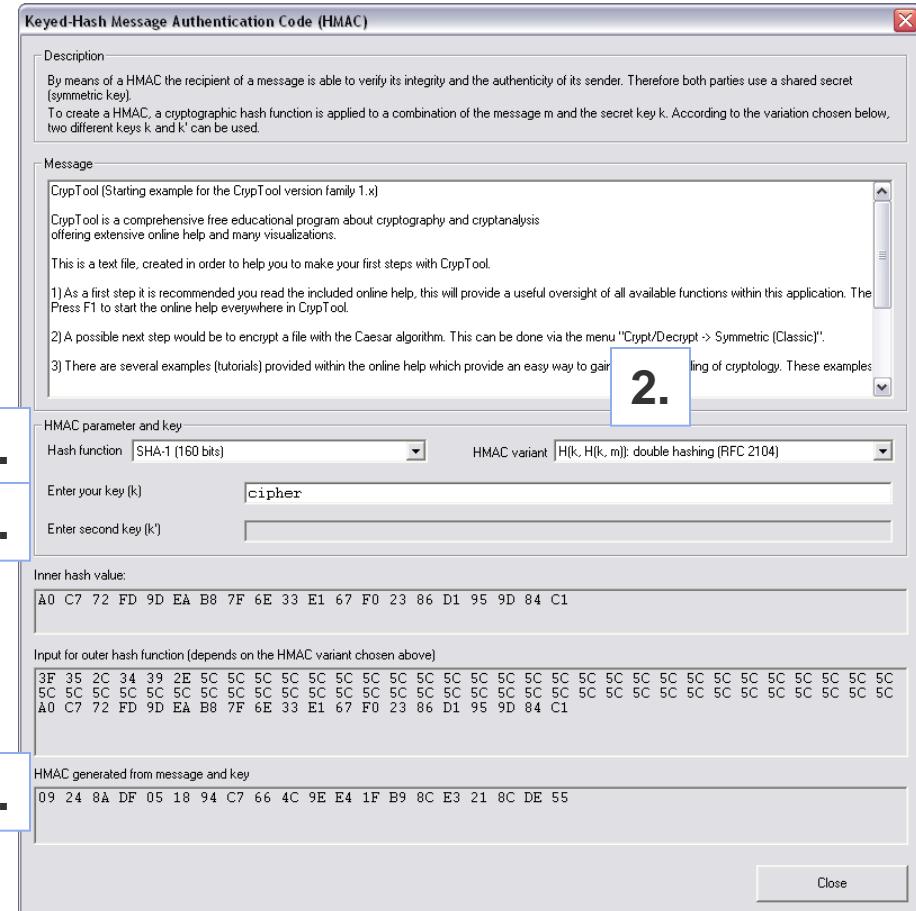
Generación de un código de autentificación de un mensaje (HMAC)

Código de Autentificación de Mensaje (HMAC)

- Asegura:
 - La integridad de un mensaje
 - La autenticación del mensaje
 - Bases: una clave común
 - Alternativa: Firma Digital

Generación de un MAC en CrypTool

1. Elija una función hash
 2. Seleccione una variante de MAC
 3. Introduzca una clave (dependiendo de la variante del MAC pueden ser dos claves)
 4. Generación del MAC (automático)



Menú: “Procedimientos Indiv.” \ “Hash” \ “Generación de MACs”

Ejemplos (19)

Demostración Hash

Sensibilidad de las funciones hash a las modificaciones del texto claro

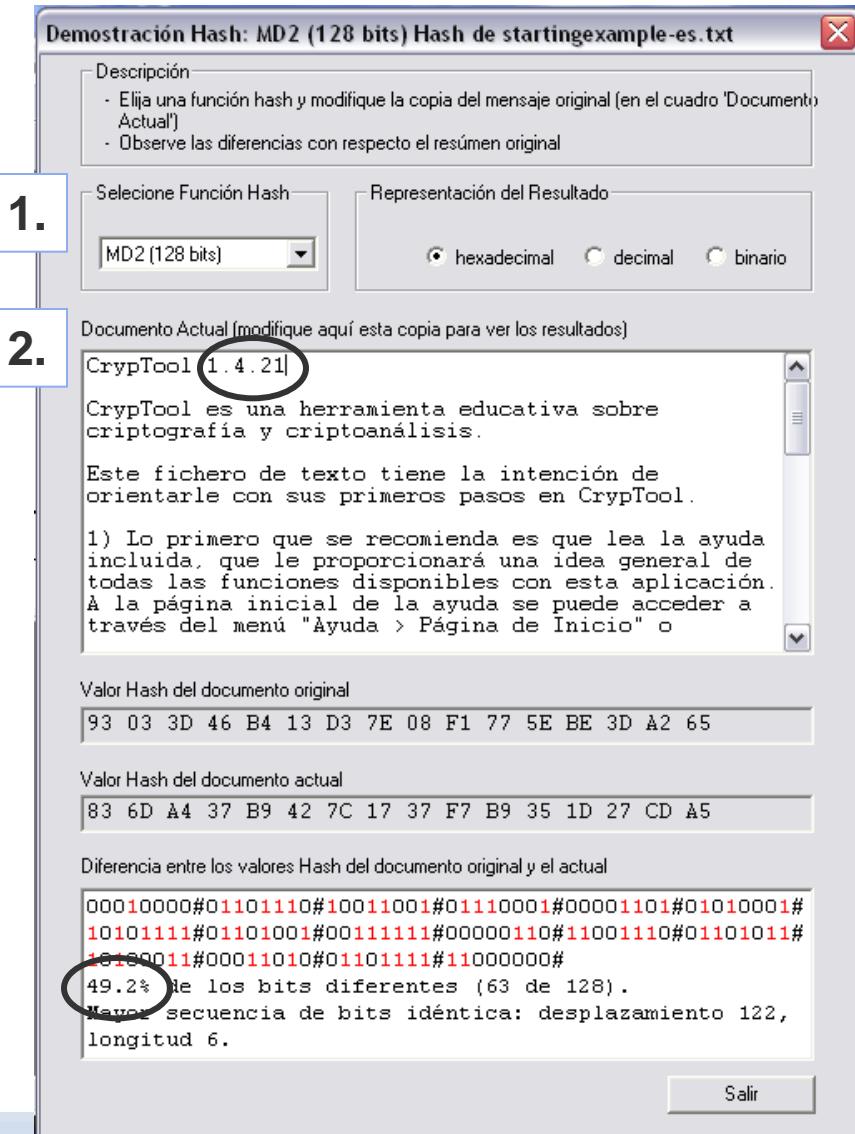
1. Seleccione una función hash
2. Modificar los caracteres del texto claro

Ejemplo:

Introduciendo un espacio después de “CrypTool” en el texto de ejemplo implica un cambio en 49,2% de los bits del valor hash generado.

Una buena función hash debe reaccionar sensiblemente frente a los más pequeños cambios en el texto claro – “efecto avalancha” (cambio pequeño, gran impacto).

Menú: “Procedimientos Indiv.” / “Hash” / “Demostración Hash”



Ejemplos (20)

Herramienta para el aprendizaje de teoría de números

- **Teoría de Números**
soportada por
elementos gráficos y
herramientas para
probar
- **Temas:**
 1. Enteros
 2. Clases de Restos
 3. Generación de primos
 4. Criptografía de clave Pública
 5. Factorización
 6. Logaritmo Discreto

The screenshot shows a software window titled "NT" with a menu bar: Calculators, Navigation, Glossaries, Help. The main content area is titled "3.2 Fermat Test" and displays the following text:

Each prime p passes a test that results from Fermat's Little Theorem:
Try for a $b \in \{2, \dots, p-1\}$, if $b^{p-1} \equiv 1 \pmod{p}$.
This test is called **Fermat Test**. Unfortunately some composite numbers pass it as well.

Example: $341 = 11 \cdot 31$, even so is $2^{340} \equiv 1 \pmod{341}$.

A passed test gives no information, one repeats it with a different base b :

$n = 341$ $2^{n-1} \equiv 1 \pmod{n}$ Test passed
 $\text{GCD}(b, n) = 1$

Definition: Let n be a composite number, b coprime to n .
If $b^{n-1} \equiv 1 \pmod{n}$, then one calls

- n **Pseudo Prime to Base b** ,
- b **Liar for** (the primality of) n ,

otherwise one calls b **Witness against** (the primality of) n .

Theorem: If there are any witnesses against n ,
then they make up at least 50% of all $b \in \{1, \dots, n\}$ coprime to n . Proof

(Go on to the next page.)

Menú: „Procedimientos Indiv.“ \ „Teoría de Números - Interactiva“ \ „Herramienta para el aprendizaje sobre teoría de números“

Ejemplos (21)

Suma de puntos en curvas elípticas

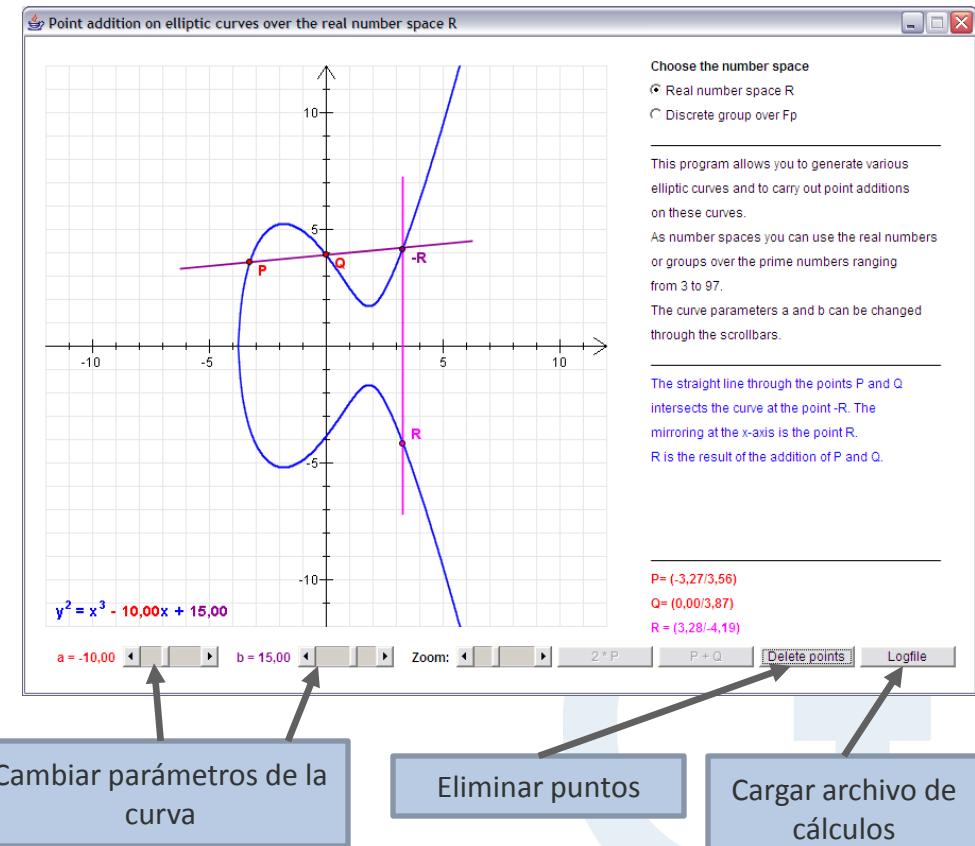
- Visualización de suma de puntos en curvas elípticas
- Bases para la criptografía basada en curvas Elípticas (ECC)

Ejemplo 1

- Marcar un punto P en la curva
- Marcar un punto Q en la curva
- Presionar el botón “P+Q”: La línea recta que une los puntos P y Q e interseca a la curva en el punto -R
- Reflejando en el eje de las X el resultado está en el punto R

Ejemplo 2

- Marcar el punto P en la curva
- Presionar el botón “2*P”: La tangente al punto P que interseca a la curva en el punto -R
- Reflejando en el eje X el resultado está en el punto R



Menú: “Procedimientos Indiv.” \ “Teoría de Números – Interactiva” \ “Suma de Puntos en Curvas Elípticas”

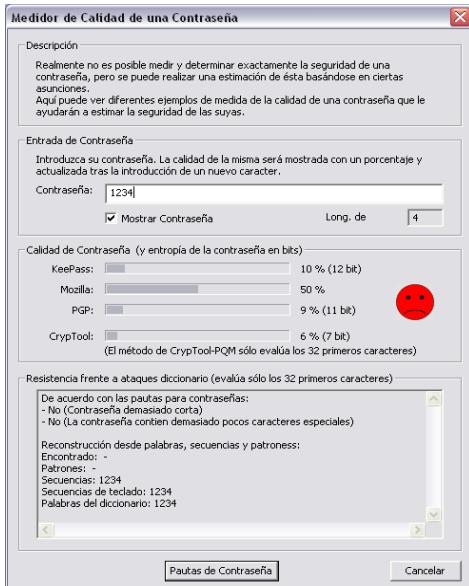
Ejemplos (22)

Medidor de Calidad de Contraseñas (Password Quality Meter “PQM”) 1

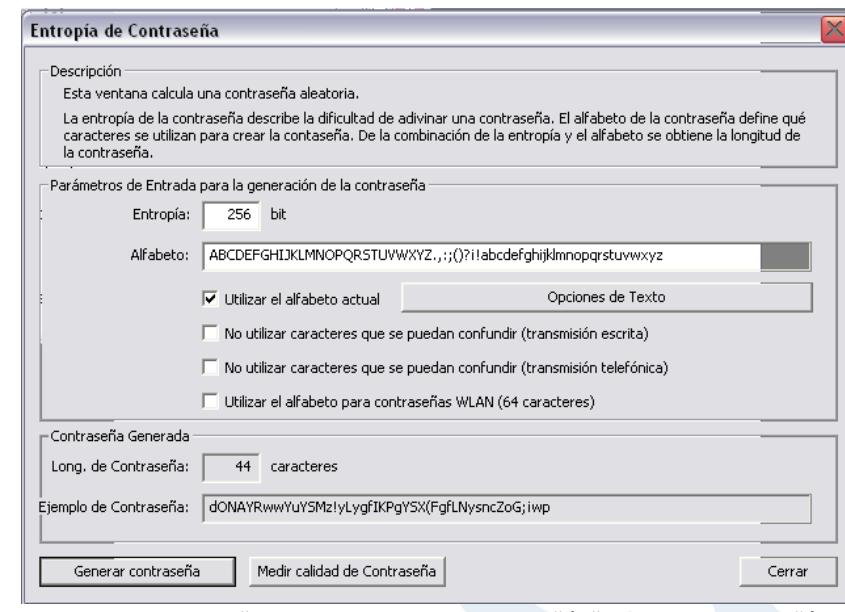
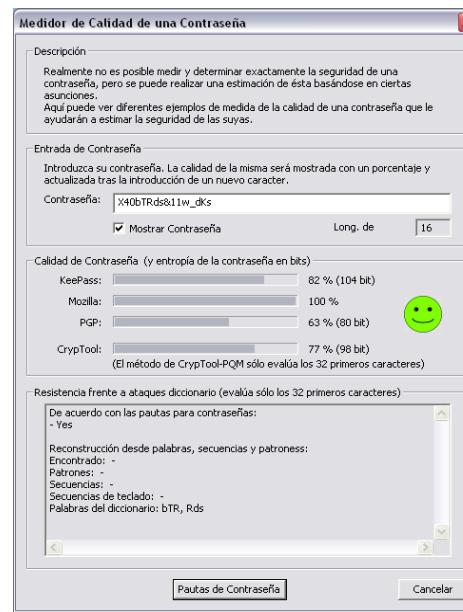
Funciones

- Medida de la calidad de contraseñas
- Comparar con PQMs en otras aplicaciones: KeePass, Mozilla y PGP
- Medida experimental con el algoritmo de CrypTool
- Ejemplo: Entrada de una contraseña (mientras se muestra la contraseña)

Password: **1234**



Password: **X40bTRds&11w_dks**



Menú: “Procedimientos Indiv.” \ “Herramientas” \ “Medidor de Calidad de Contraseñas” Menú: “Procedimientos Indiv.” \ “Herramientas” \ “Entropía de Contraseña”

Ejemplos (22)

Medidor de Calidad de Contraseñas (Password Quality Meter “PQM”) 2

Conclusiones del Medidor de Calidad de Contraseñas

- La calidad de la contraseña depende principalmente de la **longitud de la contraseña**.
- Se puede alcanzar una mayor calidad en la contraseña utilizando **distintos tipos de caracteres**: mayúsculas/minúsculas, números y caracteres especiales (**espacio de contraseña**)
- **Entropía de Contraseña** como indicador de la aleatoriedad de los caracteres de la contraseña o del espacio de contraseña (una mayor entropía aparece en una calidad de contraseña mejorada)
- Las contraseñas **NO deben existir en un diccionario** (nota: una comprobación con diccionario aún no se ha implementado en CrypTool).

Calidad de una contraseña desde la perspectiva de un atacante

- Ataque a una contraseña (con número de intentos ilimitado):
 1. **Ataque diccionario** clásico
 2. Ataque diccionario **con variantes** (p.ej. Combinaciones con números de 4 cifras: Verano2007)
 3. **Ataque por fuerza bruta** probando todas las combinaciones posibles (con parámetros adicionales como limitaciones en los tipos de conjuntos de caracteres)
- ⇒ Una buena contraseña se debe elegir para que los ataques 1. y 2. no la comprometan.
Con respecto a los ataques de fuerza bruta, son importantes la longitud de la contraseña (al menos 8 caracteres) así como los conjuntos de caracteres utilizados.

Ejemplos (23)

Análisis por Fuerza Bruta 1

Análisis por fuerza bruta

Análisis por fuerza bruta optimizado bajo la suposición de que la clave se conoce parcialmente.

Ejemplo – Análisis con DES (ECB)

Intento de encontrar el resto de la clave para descifrar el texto cifrado (Suposición: el texto claro es un bloque de 8 caracteres ASCII)

Clave (Hex)

68ac78dd40bbef*
0123456789ab****
98765432106*****
0000000000*****
000000000000****
abacadaba*****
ddddddddd*****

Texto Cifrado (Hex)

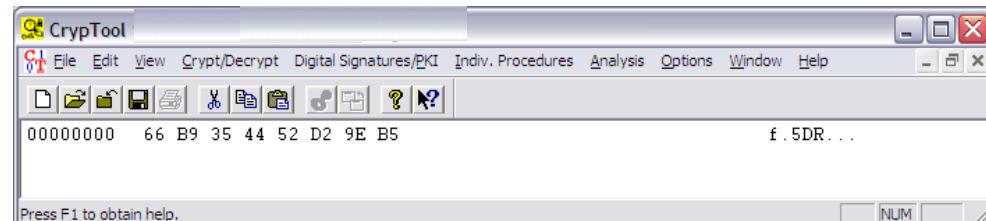
66b9354452d29eb5
1f0dd05d8ed51583
bcf9ebd1979ead6a
8cf42d40e004a1d4
0ed33fed7f46c585
d6d8641bc4fb2478
a2e66d852e175f5c

Ejemplos (23)

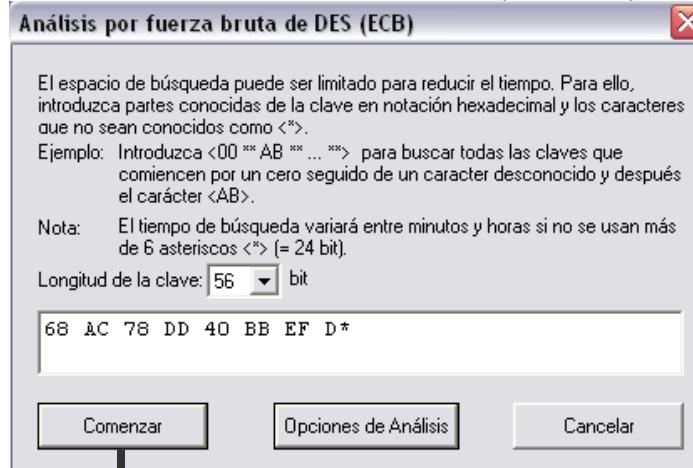
Análisis por Fuerza Bruta 2

1. Entrada de texto cifrado
2. Utilizar análisis por fuerza bruta
3. Introducir parte de la clave conocida
4. Empezar análisis por fuerza bruta
5. Análisis de los resultados: una baja entropía evidencia un posible descifrado. Sin embargo, a causa del texto claro corto utilizado en este ejemplo, el resultado correcto no es el que tiene la entropía más baja.

Utiliza "Ver" \ "Mostrar como código hexadecimal"



Menú: "Análisis" \ "Cifrado Simétrico (moderno)" \ "DES (ECB)"



The 'Análisis por Fuerza Bruta : Resultados' dialog box displays the analysis results. It includes the following text:

El análisis por Fuerza-Bruta descifra el texto cifrado proporcionado con todas las claves posibles fuera del espacio de claves seleccionado y después calcula el valor de la entropía para cada mensaje descifrado. En una lista se almacenan los mensajes descifrados con los menores valores de entropía.
En el caso especial de textos cifrados muy cortos, podría ser que candidatos erróneos a clave obtengan un valor muy bajo para la entropía. Aquí tiene la oportunidad de elegir el candidato con el texto descifrado correcto.

Descifrado: representación hexadecimal

Entropía	Descifrado	
2.4056	2B AA 0C 13 A9 B0 2B 2B	+.....++
2.5000	62 72 6F 77 73 65 72 73	browsers
2.7500	7A 62 95 C9 2D EB 9C 95	zb..~...
3.0000	58 FC 0F B9 F2 D2 6E 2A	X.....n*
3.0000	22 68 92 41 7E 2F 7A BD	"h.A~z.
3.0000	FF 17 43 46 9A 0D E1 88	..CF....
3.0000	3A AA 63 25 C9 CE 7E EF	:c%..~.
3.0000	C1 67 A7 4B 41 BE 13 D8	.g.KA...

At the bottom right are two buttons: 'Aceptar Selección' and 'Cancelar'.

Ejemplos (24)

Escítala / Rail Fence

Escítala y Rail Fence

- Transposiciones mezclan el orden de las letras en el texto claro

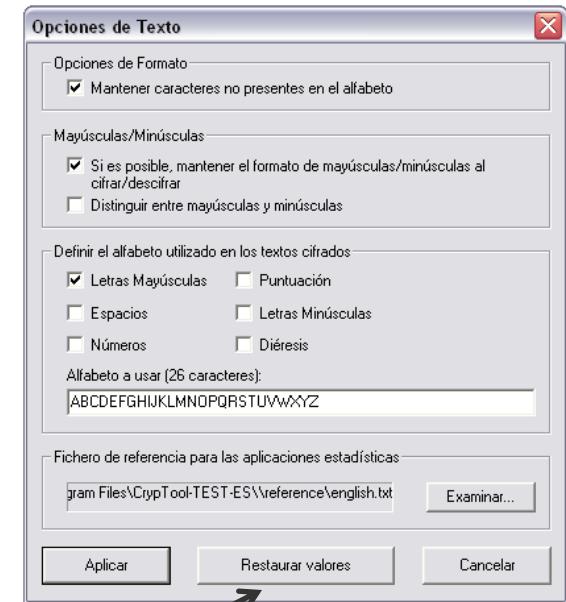
Parámetro de Transposición

- Número de esquinas (Escítala)
- Número de líneas (Rail Fence)
- Offset

Opciones de Texto

- Opciones generales de texto (Menú: “Opciones” \ “Opciones de Texto...”)
- Opciones de formato para texto claro y cifrado
- Distinción entre mayúsculas y minúsculas
- Alfabeto para el procesamiento de texto (muestra los caracteres que deben ser cifrados/descifrados)

Menú: “Cifrado/Descifrado” \ “Simétrico (clásico)” \ “Escítala/Rail Fence ...”



Ejemplos (25)

Cifrado Hill / Análisis Hill (1)

Cifrado Hill

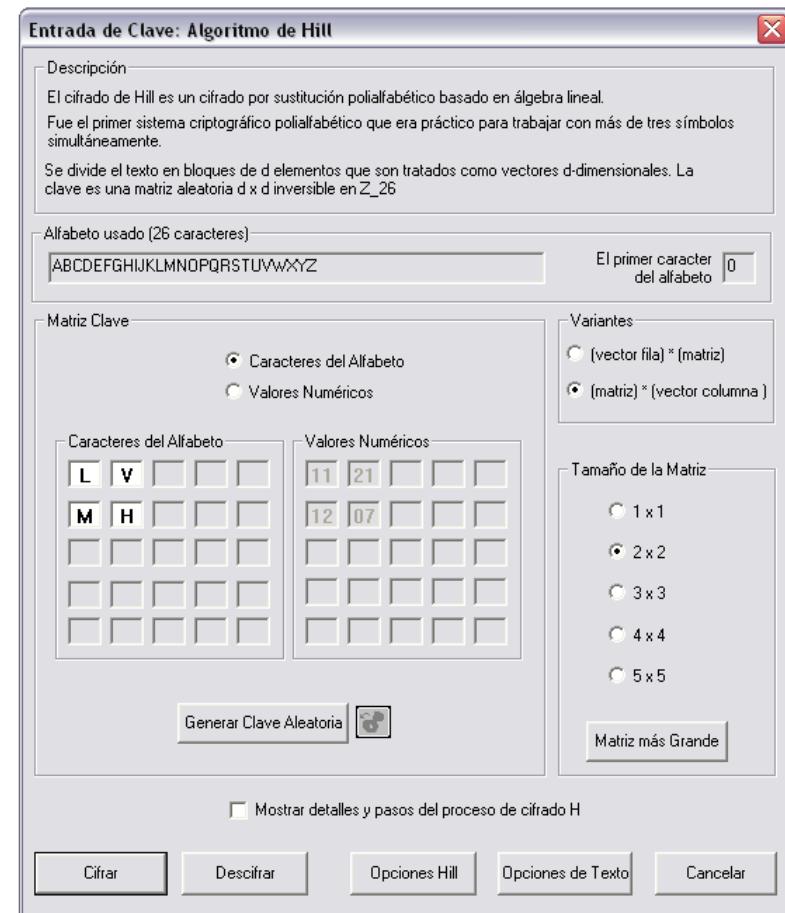
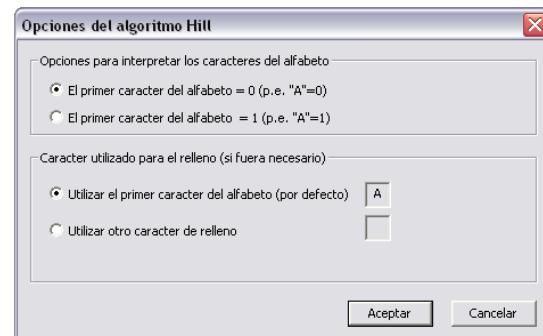
- Cifrado de sustitución poligráfico
- Basado en álgebra lineal

Clave

- Caracteres de alfabeto (Ver opciones de texto) o valores numéricos
- Ingresar clave o generar una aleatoria
- Seleccionar parámetro de multiplicación
- Tamaño de la matriz
- Opciones

Menú:

“Cifrar/Descifrar” \
“Simétrico (clásico)” \
“Hill ...”



Ejemplos (25)

Cifrado Hill / Análisis Hill (2)

Cifrado Hill

- Texto de ejemplo con la clave: LVMH

Análisis Hill (texto claro conocido)

1. Texto claro / texto cifrado - Largo

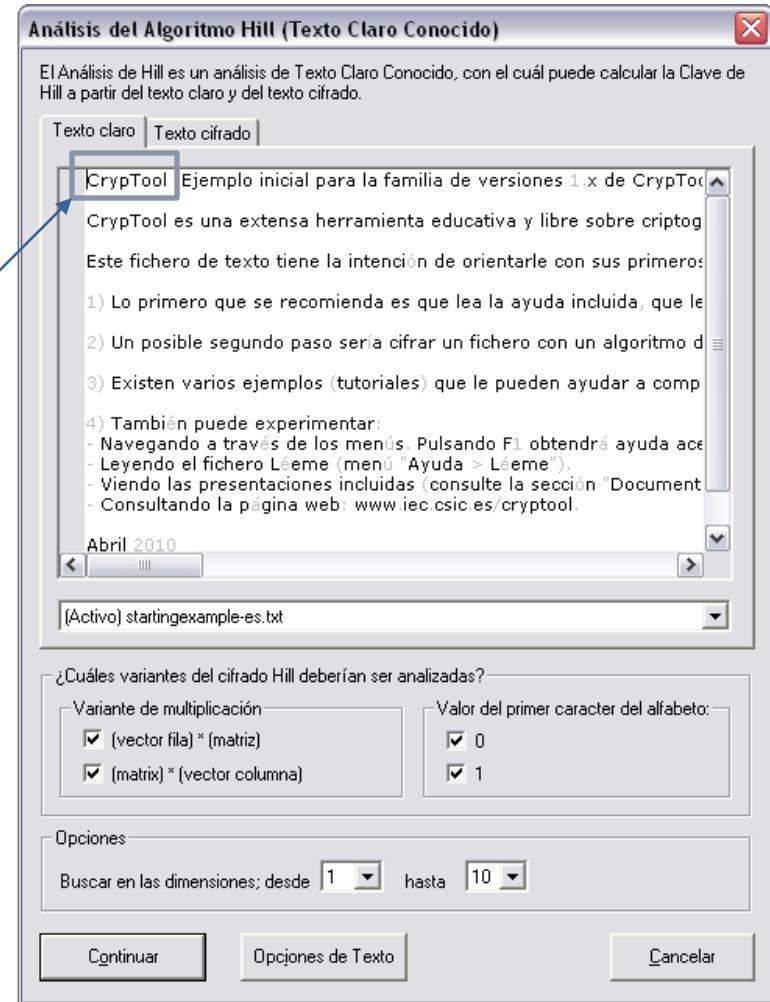
- Seleccionar texto plano (startingexample-es.txt)
- Seleccionar texto cifrado
(Cifrado Hill de <startingexample-es.txt>)
- “Continuar” para buscar la clave

2. Texto claro / texto cifrado - Reducido

- Eliminar todo, excepto el inicio del texto claro (“CrypTool”)
- Reducir texto cifrado a “PnhdJovl”
- “Continuar” encuentra la clave correcta

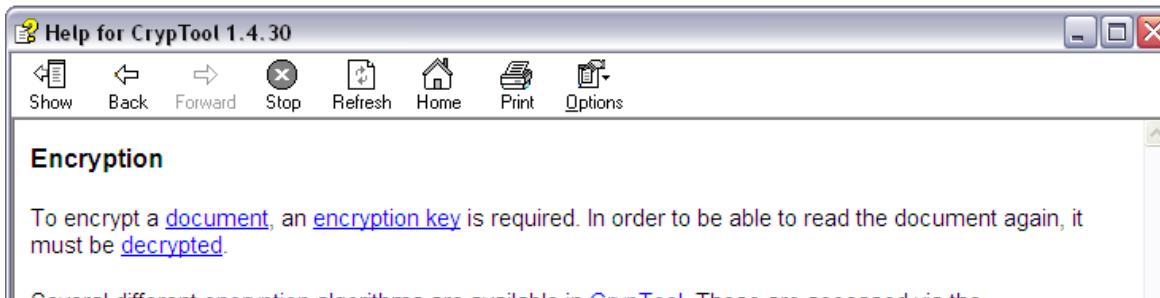
¿Qué cantidad de texto claro/cifrado es necesario para encontrar la clave de cifrado correcta?

Menú: “Análisis” \ “Simétrico (clásico)” \ “Texto claro conocido” \ “Hill...”



Ejemplos (26)

Ayuda Online de CrypTool (1)



The screenshot shows the 'Encryption' section of the CrypTool help documentation. It includes a toolbar with standard navigation buttons (Show, Back, Forward, Stop, Refresh, Home, Print, Options) and a main content area with the following text:

To encrypt a [document](#), an [encryption key](#) is required. In order to be able to read the document again, it must be [decrypted](#).

Several different [encryption algorithms](#) are available in [CrypTool](#). These are accessed via the [Crypt/Decrypt](#) menu.

Encryption algorithms

An encryption algorithm is required in order to transmit confidential information over insecure channels, for example, over a network. The information is [encrypted](#) by the originator prior to transmission and [decrypted](#) by the recipient following transmission.

A symmetric encryption algorithm is one in which the originator's and recipient's [keys](#) are identical. Encryption algorithms in which the originator and recipient have different keys are called asymmetric.

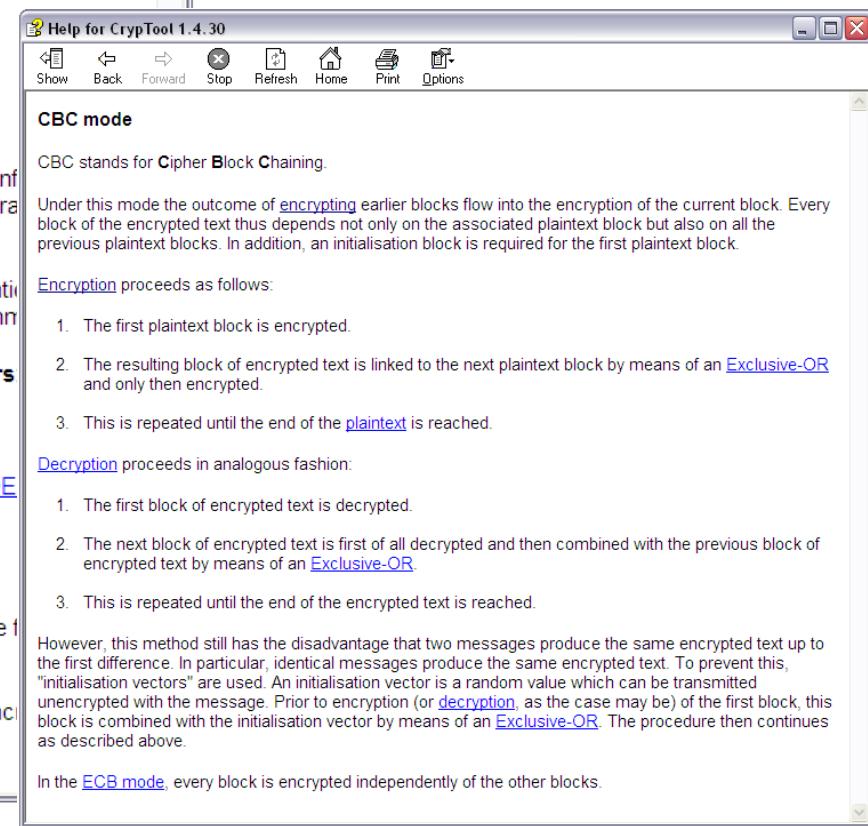
Modern symmetric encryption algorithms can be divided in **block ciphers** and **stream ciphers**.

- [Block ciphers](#) encrypt blocks of fixed length (e.g. 64 or 128 bit). Available in CrypTool are [IDEA](#), [RC2](#), [DES \(ECB\)](#), [DES \(CBC\)](#), [Triple DES \(ECB\)](#), [Triple DES \(CBC\)](#), [Rijndael \(AES\)](#), [MARS](#), [RC6](#), [Serpent](#), [Twofish](#), [DESX](#), [DESL](#) and [DEA](#).
- [Stream ciphers](#) encrypt messages bit by bit. In this category CrypTool provides [RC4](#).

A summary of all the encryption algorithms available in CrypTool is contained on the help page [Encryption](#) in the [Crypt/Decrypt](#) menu.

Further information on encryption algorithms can be found in the [script](#), e.g. in the chapter "Encryption Procedures".

Menú: "Ayuda" \ "Página de Inicio"



The screenshot shows the 'CBC mode' section of the CrypTool help documentation. It includes a toolbar with standard navigation buttons (Show, Back, Forward, Stop, Refresh, Home, Print, Options) and a main content area with the following text:

CBC mode

CBC stands for **Cipher Block Chaining**.

Under this mode the outcome of [encrypting](#) earlier blocks flow into the encryption of the current block. Every block of the encrypted text thus depends not only on the associated plaintext block but also on all the previous plaintext blocks. In addition, an [initialisation block](#) is required for the first plaintext block.

[Encryption](#) proceeds as follows:

1. The first plaintext block is encrypted.
2. The resulting block of encrypted text is linked to the next plaintext block by means of an [Exclusive-OR](#) and only then encrypted.
3. This is repeated until the end of the [plaintext](#) is reached.

[Decryption](#) proceeds in analogous fashion:

1. The first block of encrypted text is decrypted.
2. The next block of encrypted text is first of all decrypted and then combined with the previous block of encrypted text by means of an [Exclusive-OR](#).
3. This is repeated until the end of the encrypted text is reached.

However, this method still has the disadvantage that two messages produce the same encrypted text up to the first difference. In particular, identical messages produce the same encrypted text. To prevent this, "initialisation vectors" are used. An initialisation vector is a random value which can be transmitted unencrypted with the message. Prior to encryption (or [decryption](#), as the case may be) of the first block, this block is combined with the initialisation vector by means of an [Exclusive-OR](#). The procedure then continues as described above.

In the [ECB mode](#), every block is encrypted independently of the other blocks.

Ejemplos (26)

Ayuda Online de CrypTool (2)

Help for CrypTool 1.4.30

Hide Back Forward Stop Refresh Home Print Options

Contents Index Search

Type in the keyword to find:

lattice reduction

Lattice reduction
Liability (exclusion)
License terms
Line wrap
Links
Literature
MARS encryption algorithm
MD2 hash value
MD4 hash value
MD5 hash value
Menu (overview of all menus)
Miracl
Modular transformation
Modulo operator
Monoalphabetic substitution encrypt
Network authentication
N-gram
Nihilist encryption algorithm
NIST
Normal distribution
NSA
NTL
Number Shark
Number system
Number theory
Offset
One-time pad
OpenGL
OpenPGP
OpenSSL
Options
Overview / Subsumption / Broader C
Padding
Parent window
Password
Pattern search

Display

Menu Lattice Based Attacks on RSA (Menu Individual Procedures \ RSA Cryptosystem)

The menu **Lattice Based Attacks on RSA** contains the following commands:

[Factoring with a Hint](#) Attacks RSA with lattice reduction algorithms, if a part of one of the primes of N is known.

[Attack on Stereotyped Messages](#) Attacks RSA with lattice reduction algorithms, if a part of the original cleartext of an intercepted ciphertext is known and if e is small.

[Attack on Small Secret Keys](#) Attacks RSA with lattice reduction algorithms, if d is too small compared to N.

All attacks presented here are based on a common approach: first the task of breaking RSA is transformed into finding the root of a polynomial modulo an integer (mostly N) but to find such a root is a difficult problem.

To solve this problem further polynomials are generated which are known to have the same root. From the coefficients of these polynomials a latticebase is built. This is then reduced with, i.e. the LLL-algorithm to find a small vector.

From this newly found short vector a new polynomial is built. It can be proven that if the vector is short enough, the polynomial has the desired root not only modulo N, but also over the integers.

Example:

The polynomial $q_1(x) = 3x+1$ has a root x_0 modulo 7. It is supposed, that the polynomial $q_2(x) = 4x-1$ has the same root x_0 modulo 7. From these polynomials the vectors $b_1=[3 1]$ and $b_2=[4 -1]$ are built. All integer linear combinations of these vectors form points in a lattice. The Figure on the left shows a part of this lattice. Each point of the lattice now can again be interpreted as a polynomial having the desired root. A short vector of the lattice is $b_3=[1 -2]$ from which the polynomial $h(x) = x-2$ is built. This polynomial has a root in $x_0=2$ over the integers as well as modulo 7. That $x_0=2$ is also a root of the polynomials $q_1(x)$ and $q_2(x)$ modulo 7 can be easily established.
 $(3x_0+1=7, 7 \text{ modulo } 7 = 0)$

Ejemplos (26)

Ayuda Online de CrypTool (3)

Help for CrypTool 1.4.30

Hide Back Forward Stop Refresh Home Print Options

Contents Index Search

Type in the keyword to find:

base

Base64 coding

- BC
- Binary exclusive-OR
- Birthday attack / birthday paradox
- Bit length
- Block cipher
- Blocks
- Books
- Bounding box
- Brute-force attack
- Byte addition
- Caesar encryption algorithm
- Card game
- Cascade
- Cascading cipher
- CBC mode
- Certificate
- Challenge
- Challenge-response demonstration
- Chi² distribution
- Chinese remainder theorem
- Chosen-plaintext attack
- Ciphertext
- Ciphertext-only attack
- Clipboard
- Codings
- Coin toss
- Column transposition
- Compress
- Congruence generator
- Contact
- Context / Subsumption / Ov
- Copyright
- Correlation
- Cryptanalysis
- Crypto competitions / Crypt

Display

Comparison of Base64 and UU coding

The encoding procedures of [Base64](#) and [UUencode](#) are quite similar, which is shown by the following figure:

Base64 UUencode

Step 1: Splitting the data stream -- same procedure in both encodings.

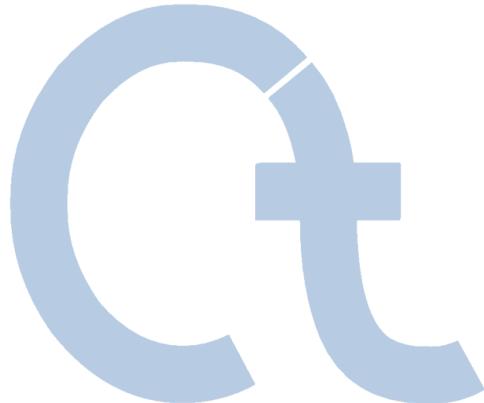
Step 2: Representation of the 6 bit values -- different procedures.

Dividing of 3 x 8 bit to 4 x 6 bit.

Because of the similar encoding procedure, there are also shared advantages and drawbacks:

Advantages	Drawbacks
<ul style="list-style-type: none">Arbitrary binary data can be represented with a 6-bit	

Contenido



- I. CrypTool y Criptología – Vista General
- II. CrypTool Características
- III. Ejemplos
- IV. Proyecto / Perspectiva / Contacto**



Desarrollo de CrypTool en el Futuro (1)

Plan después de la publicación de 1.4.30 (ver archivo Léeme)

CT1	Test FIPS para investigar partes de tamaño mayor a 2500 bytes
JCT	Acuerdos de clave tri-partita
JCT	Visualización de la interoperabilidad de los formatos S/MIME y OpenPGP
JCT	Análisis de entropía
JCT	Grille, Vigenère Autoclave, Criptoanálisis interactivo de Cifrados clásicos
JCT	Cifrados de análisis de transposición usando el algoritmo ACO
JCT	Visualización de las pruebas de cero conocimiento
JCT	Visualización del acuerdo de clave Quantum, Protocolo BB84
JCT	Visualización del ataque SETUP contra la generación de claves RSA (Kleptografía)
JCT	Action-History con característica adicional para crear y reproducir cualquier cifrado (cascada)
CT2	Visualización comprensible del tema de los números primos
CT2	Cifrado y criptoanálisis automatizado de la máquina Enigma y tal vez de Sigaba
CT2	Ataque del Cubo (I. Dinur y A. Shamir, "Cube Attacks on Tweakable Black Box Polynomials", 2008)
CT2	Demonstración de la falsificación de la firma RSA de Bleichenbacher
CT2	Demonstración virtual de números de tarjetas de crédito (enfoque contra el abuso en tarjetas de crédito)
CT2	Cifrado WEP y análisis WEP
CT2	Búsqueda masiva de patrones
CT2	Criptoanálisis distribuido
CT2	Demonstración de seguridad de SOA (mensajes SOAP a través de seguridad WS entre los participantes)
CT2	Framework para crear y analizar cifrados de flujo LFSR
CT2	Diseño gráfico orientado tanto a los principiantes como a los expertos
CT2/JCT	Creación de una versión en línea de comandos para un procesado por lotes
CT2/JCT	Moderna arquitectura <i>pure plugin</i> con plugins cargados
Todo	Parametrización adicional / Incrementando la flexibilidad de los algoritmos presentes
Ideas	Visualización del protocolo SSL // Demonstración de criptografía visual

CT1 = CrypTool 1.x

Nuevas versiones:

CT2 = CrypTool 2.0

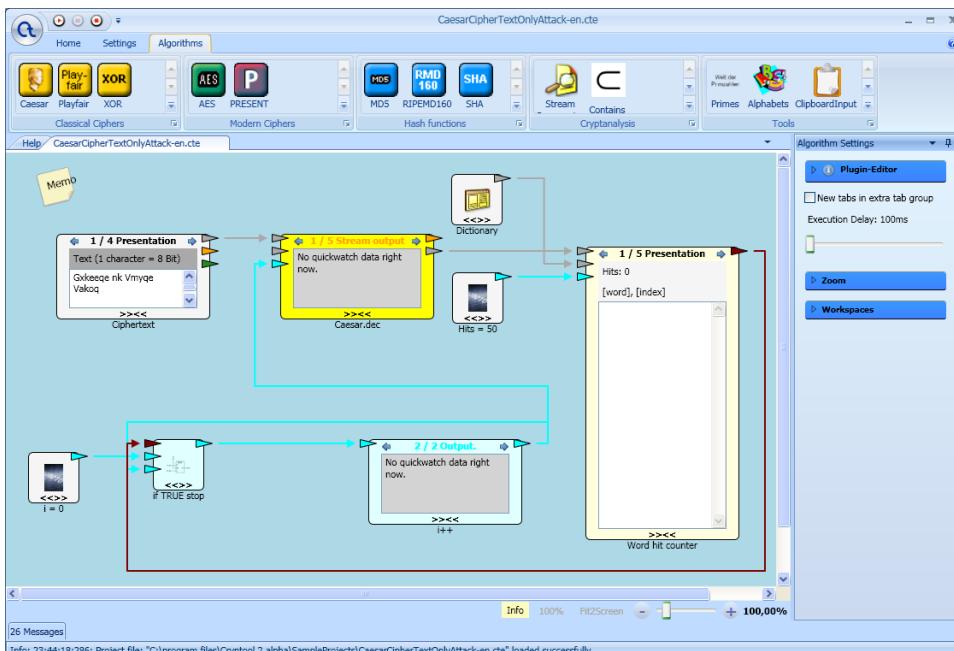
JCT = JCrypTool
(Ambos presentados a continuación)



Desarrollo de CrypTool en el Futuro (2)

En desarrollo: Las dos versiones sucesoras de CT v1 (ver archivo Léeme)

1. JCT: Portabilidad y rediseño de CrypTool en Java / SWT / Eclipse 3.6 / RPC
 - ver: <http://jcryptool.sourceforge.net>
 - Release Candidate RC2 está disponible para desarrolladores y usuarios (Mayo 2010)
2. CT2: Portabilidad y rediseño de la versión en C++ con C# / WPF / VS2010 / .NET 4.0
 - Sucesor directo de las versiones actuales: permite programación visual, etc.
 - Descargar de: <http://cryptool2.vs.uni-due.de/index.php?page=14&lm=1&ql=4>
 - La versión Beta3 está disponible desde Julio 2010 (Actualizada continuamente desde Junio de 2008)

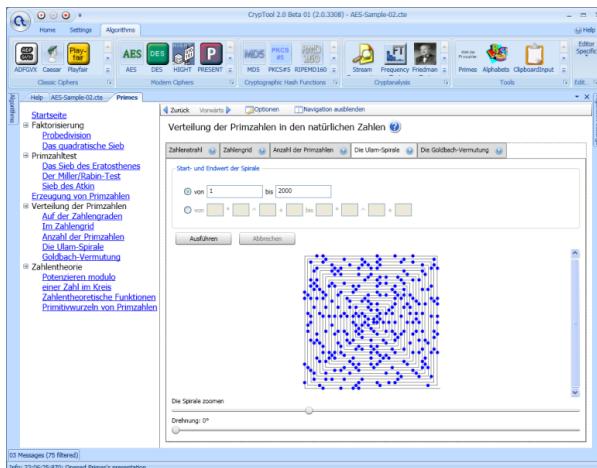
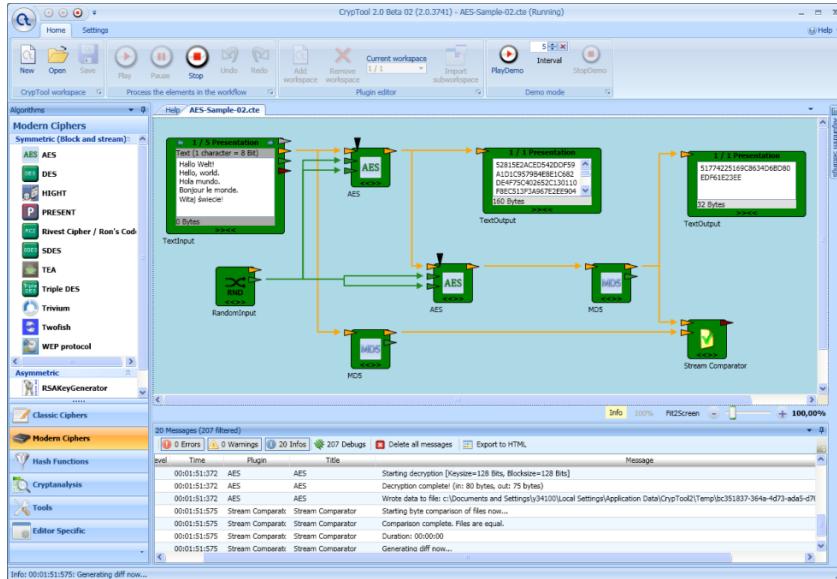


CrypTool 2 (CT2)

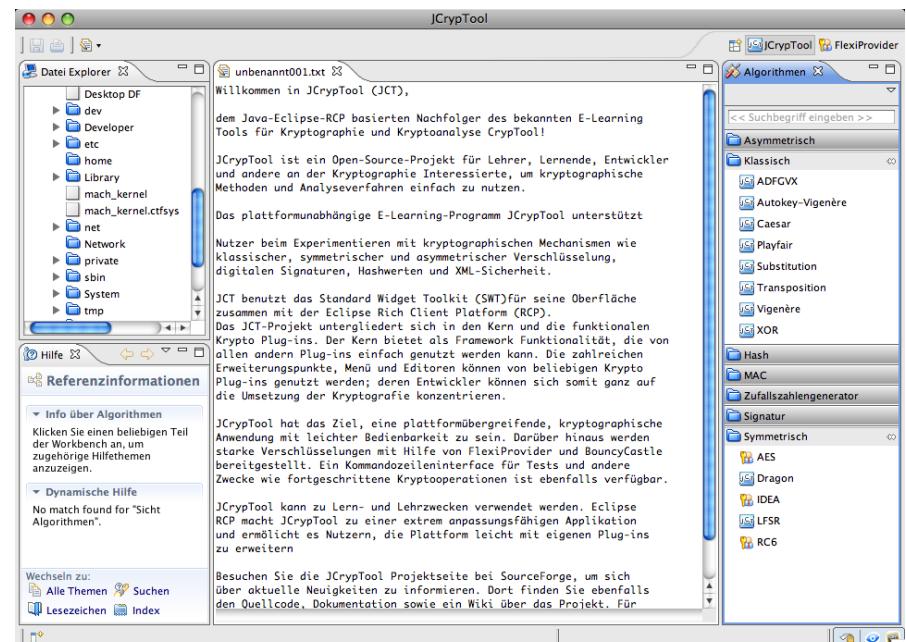


JCrypTool (JCT)

Desarrollo de CrypTool en el Futuro (3)



CrypTool 2 (CT2)



JCrypTool (JCT)

CrypTool como un “Framework”

Propuesta

- Reutilizar el amplio conjunto de algoritmos, incluyendo las librerías y los elementos de la interfaz como base
- Entrenamiento gratuito en Frankfurt, cómo empezar con el desarrollo de CrypTool
- Ventaja: Tu propio código no “desaparece”, se mantendrá

Entorno de desarrollo actual para CT1: Microsoft Visual Studio C++ , Perl,
Subversion Source-Code Management

- Hasta CrypTool 1.4.30: Visual C++ .NET (= VC++ 9.0)(= Visual Studio 2008 Standard)
- Descripción para desarrolladores: ver readme-source.txt
- Descarga: de fuentes y binarios de las publicaciones. Para obtener los archivos fuente de las betas actuales, por favor vea el repositorio de subversiones.

Entornos de desarrollo para CT2 y JCT

- CT2 – versión C# : .NET con Visual Studio 2010 Express Edition (gratis) y WPF
- Java – versión Java: Eclipse 3.6, RCP, SWT (gratis)



CrypTool – Petición de Colaboración

Toda colaboración con el proyecto se agradece enormemente

- Realimentación de información, críticas, sugerencias e ideas
- Integración de algoritmos adicionales, protocolos, análisis (consistencia y completitud)
- Desarrollo de asistencia (programación, diseño, traducción, prueba)
 - Para el proyecto C/C++ actual
 - Para los nuevos proyectos (preferencialmente)
 - Proyecto C# : “CrypTool 2.0” = CT2
 - Proyecto Java : “JCrypTool” = JCT
 - Especialmente se invita al desarrollo adicional a las Universidades que utilizan CrypTool para propósitos educativos.
- Ejemplos de tareas abiertas se encuentran en las páginas de desarrollo respectivas:
 - CT2: Ver la lista: <http://cryptool2.vs.uni-due.de>, voluntarios, tareas actuales
 - JCT: Ver: wiki <http://sourceforge.net/apps/mediawiki/jcryptool/index.php?title=CurrentDevelopment>
- Las colaboraciones significativas se pueden referenciar por nombre (en la ayuda, Léeme, ventana Acerca de, o en la página web de CrypTool).
- Actualmente CrypTool posee más de 6000 descargas al mes (de las cuales un poco más del 50% se realiza sobre la versión en inglés).
- Las versiones Betas de las dos herramientas sucesoras (JCT y CT2) registran ya más de 1000 descargas mensuales.

CrypTool – Resumen

- EL programa de aprendizaje electrónico para criptología
- Un proyecto de Código Abierto con más de 10 años de éxito
- Más de 300.000 descargas
- Uso internacional en escuelas, universidades, así como empresas y agencias del gobierno
- Amplia ayuda online y documentación
- Disponible gratuitamente y con soporte multi-idioma

Contacto

Prof. Bernhard Esslinger

Universidad de Siegen
Facultad 5, Computación y Sistemas de Información

Deutsche Bank S.A.
Director, IT Security Manager

esslinger@fb5.uni-siegen.de

www.cryptool.org
www.cryptool.com
www.cryptool.de
www.cryptool.es
www.cryptool.pl

Contactos adicionales: ver Léeme en la carpeta CrypTool



Bibliografía Adicional

Para introducirse en la Criptología

- Simon Singh, “*Los códigos secretos*”, 2000, Doubleday
- J. Ortega y Miguel Ángel López Guerrero, “*Introducción a la Criptografía*”, 2006
- Jorge Ramió, “Libro Electrónico de Seguridad Informática y Criptografía”
- Johannes Buchmann, “*Introduction to Cryptography*”, 2nd edition, 2004, Springer [inglés]
- Paar / Pelzl: „*Understanding Cryptography – A Textbook for Students and Practitioners*”, 2009, Springer [inglés]
- Klaus Schmeh, “*Codeknacker gegen Codemacher. Die faszinierende Geschichte der Verschlüsselung*”, 2nd edition, 2007, W3L [alemán]
- [HAC] Menezes / van Oorschot / Vanstone, “*Handbook of Applied Cryptography*”, 1996, CRC Press
- van Oorschot / Wiener, “*Parallel Collision Search with Application to Hash Functions and Discrete Logarithms*”, 1994, ACM [inglés]
- Bibliografía adicional sobre criptografía – ver también la página web de CrypTool y la bibliografía de la ayuda online de CrypTool (p.ej. por Wätjen, Salomaa, Brands, Schneier, Shoup, Stamp/Low, ...)
- La importancia de la criptografía en el amplio contexto de la seguridad en TI y la gestión de riesgos
 - Ver p.ej. Kenneth C. Laudon / Jane P. Laudon / Detlef Schoder, “*Wirtschaftsinformatik*”, 2005, Pearson, chapter 14 [alemán]
 - Ver Wikipedia (http://en.wikipedia.org/wiki/Risk_management) [inglés]
 - Página de CrypTool: <http://cryptool.com/index.php/en/cryptool-for-awareness-aboutmenu-74.html>



Cryptool

Acerca de Características Capturas de pantalla Documentación Descargar    

Latest stable version: 1.4.21 [Download](#)

Acerca de

- Introducción a Cryptool
- Cryptool en la Educación
- Cryptool para el conocimiento
- Cobertura en los medios
- Premios
- Colaboradores
- Proyectos Relacionados
- Contacto

Selected Landmark 2008 in:
Germany
Land of Ideas


"In the field of educating students or IT-professionals, this tool gives you much support and wonderful help to increase their knowledge. Beside of theoretical background, it gives the"

Introducción a Cryptool

Cryptool es una aplicación de aprendizaje electrónico gratuita para Windows. Puede utilizarla para aplicar y analizar algoritmos criptográficos. La versión actual de Cryptool se utiliza en todo el mundo. Soporta tanto los métodos actuales de enseñanza en escuelas y universidades como la concienciación de los empleados.

La versión actual ofrece, [entre otras cosas](#), lo siguiente:

- Numerosos algoritmos criptográficos, clásicos y modernos (cifrado y descifrado, generación de clave, contraseñas seguras, autenticación, protocolos seguros, ...)
- Visualización de varios métodos (p.ej. César, Enigma, RSA, Diffie-Hellman, firmas digitales, AES)
- Criptoanálisis de ciertos algoritmos (p.ej. Vigenère, RSA, AES)
- Métodos de medida criptoanalítica (p.ej. entropía, n-grams, autocorrelación)
- Métodos auxiliares (p.ej. tests de primalidad, factorización, codificación en base64)
- Tutorial sobre teoría de números.
- Ayuda detallada on-line.
- Script con más información sobre criptografía.

Desde su uso original para la formación en seguridad de una compañía, Cryptool ha evolucionado en un destacado proyecto de código abierto para temas relacionados en criptografía.

Desde la primavera de 2008, está funcionando en el proyecto Cryptool el [Cripto Portal para profesores](#). Por ahora el portal sólo está disponible en alemán y se espera que actúe como una plataforma para que los profesores puedan compartir material para la enseñanza de la criptografía y temas relacionados.

Actualmente el equipo de Cryptool está trabajando en dos proyectos futuros que se espera que sean los sucesores de la actual versión CrypTool 1.4.x que está escrita en C++. Ambos proyectos de continuación utilizan el último modelo de estándares de programación, pero aún están en un estado alfa/beta:

www.cryptool-online.org

CrypTool-Online es el más nuevo miembro de la familia de CrypTool
Sitios relacionados:

- **CrypTool** (CT1)
- **CT2** – para desarrolladores
- **JCT** – para desarrolladores
- **CryptoPortal** para profesores (actualmente sólo en alemán)
- **CrypTool-Online** (pruebe métodos criptográficos en su navegador).



¡Descargue el software y el CrypTool Script!

The screenshot shows two versions of the CrypTool website. The top version has a red circle around the 'Documentación' menu item, and the bottom version has a red circle around the 'Descargas' menu item. Both versions feature a prominent 'Beta estable 1.4.30' banner.

Script
Versión Actual: 10^a edición, Enero 2010
[Descarga el script](#)

En este script proporcionado con la aplicación CrypTool encontrará pormatícamente para el uso en procedimientos criptográficos. Los autores y son, por lo tanto, independientes los unos de los otros. Al bibliográficas y enlaces web.

A screenshot of the 'script-en.pdf' document viewed in Adobe Reader. The table of contents includes:

- Overview
- Contents Overview
- Contents
- Preface to the 10th Edition of the CrypTool Script
- Introduction -- How do the Script and the Program Play together?
- 1 Encryption Procedures
- 2 Paper and Pencil Encryption Methods
- 3 Prime Numbers
- 4 Introduction to Elementary Number Theory with Examples
- 5 The Mathematical Ideas behind Modern Cryptography
- 6 Hash Functions and Digital Signatures
 - 6.1 Hash functions
 - 6.2 RSA signatures
 - 6.3 DSA signatures
 - 6.4 Public key certification
 - Bibliography
- 7 Elliptic Curves
- 8 Crypto 2020 --- Perspectives for Long-Term Cryptographic Security
- A Appendix
 - A.1 CrypTool Menus
 - A.2 Authors of the CrypTool Script
 - A.3 Movies and Fictional Literature with Relation to Cryptography, Books for Kids with Simple Ciphers
 - A.4 A Learning Tool for Elementary Number Theory
 - A.5 Using Sage with this Script
 - GNU Free Documentation License
 - List of Figures
 - List of Tables
 - List of Crypto Procedures
 - List of Sage Code Examples
 - Index

The CrypTool Script landing page features the title 'The CrypTool Script' and subtitle 'Cryptography, Mathematics, and More'. It credits Prof. Bernhard Esslinger and the CrypTool Development Team. The page also mentions the '10th Edition'.

Descarga

[Download CrypTool 1.4.x](#) [Download CrypTool 2.0 Beta](#) [Download JCrypTool Beta](#)

CrypTool 1.4.21
(Para esta versión de CrypTool en español, favor descargar adicionalmente los archivos de ayuda [aqui](#))

La versión actual publicada para usuarios es CrypTool 1.4.21 (publicada el 11 de Julio de 2008).

Esta versión necesita un entorno Win32. El programa contiene algunas funciones que llaman a aplicaciones Java. Para poder ejecutar estas aplicaciones, deberá tener instalada una máquina virtual Java (JRE 1.5 ó superior).

El código de la versión publicada (etiqueta "CrypTool_1_4_21") y los códigos actuales de desarrollo están disponibles en el repositorio. Todo el mundo tiene acceso de lectura a este [repositorio](#) (Usuario y contraseña: *anonymous*).

CrypTool 1.4.x está disponible en Inglés, alemán, español y polaco:

- [CrypTool 1.4.21 - Inglés](#)
- [CrypTool 1.4.21 - Alemán](#)
- [CrypTool 1.4.21 - Español](#)
- [CrypTool 1.4.10 - Polaco](#)