PRÓLOGO DE LA TERCERA EDICIÓN v 3.1 EN INTERNET
Madrid, marzo de 2003

Estimado amigo o amiga:

Transcurridos ya doce meses desde la publicación en Internet de la 2ª versión de este curso gratuito en diapositivas, en los cuales he tenido el grato placer de observar cómo día a día el número de descargas del archivo desde el servidor Web de la Red Temática Iberoamericana de Criptografía y Seguridad de la Información CriptoRed, iba logrando unas cifras para mí verdaderamente astronómicas (más de 40.000 sólo desde ese servidor, y me imagino que unas cuantas más por medio de copia de discos y descargas desde otros sitios Web), ha llegado el momento de hacer una revisión y actualización de dicho curso.

La razón de escribir un libro en formato diapositivas nació prácticamente después de la publicación del libro Aplicaciones Criptográficas editado por el Departamento de Publicaciones de la EUI-UPM en junio de 1999, al ver que en una materia tan cambiante como es la seguridad informática y la criptografía era casi una utopía tener una publicación relativamente al día sobre técnicas, algoritmos, esquemas, etc. Por tanto, comencé a elaborar unos apuntes de clases para mis alumnos con un formato algo más explícito que unas simples diapositivas de apoyo a una clase magistral pero que, a la vez, pudiera ser útil como material de aprendizaje para cualquiera que quisiera adentrarse en este fascinante mundo de la seguridad informática. Nació así una primera versión del curso en el mes de septiembre de 2001, una segunda versión en febrero de 2002 y, por último, ésta que además de su publicación en Internet como material de libre distribución, se publica en formato libro con ISBN y depósito legal.

Por lo tanto, además de su publicación en Internet para su descarga gratuita sólo para fines personales y docentes, existe una edición del mismo en papel –orientada básicamente para los alumnos de la asignatura de Seguridad Informática que imparto y para quien desee tener ese documento en formato libro- por intermedio del Departamento de Publicaciones de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, bajo el patrocinio de la Fundación General de la UPM, con ISBN y depósito legal. Por ello, queda prohibida la venta del mismo en los términos habituales de cualquier propiedad intelectual protegida por copyright como el indicado, excepto a través de dicho departamento de publicaciones.

He corregido algunas erratas, modificado párrafos para su mejor comprensión, incluido más ejemplos y actualizado todos los capítulos. En el capítulo de presentación del curso se incluye el contenido por temas para una búsqueda más rápida y efectiva. Se ha incluido un archivo dedicado a la bibliografía, enlaces, software y tablas de interés criptográfico. Además, al final de cada capítulo o lección se presenta un conjunto de cuestiones y ejercicios (en total 268) para el lector. que le servirán a modo de evaluación personal del avance y comprensión de cada tema. Espero que las soluciones a los mismos estén disponible en los próximos meses en la página Web de la asignatura.

Esta tercera versión del libro electrónico cuenta con 905 diapositivas, 218 más que en la versión v2 del año 2002 y más del doble comparado con la versión v1 del año 2001. Siempre van quedando -y me temo que por mucho tiempo- temas pendientes que por uno u otro motivo voy dejando para más adelante; entre ellos el de técnicas criptográficas avanzadas como por ejemplo la esteganografía, la criptografía visual y la criptografía cuántica, el tema de la seguridad física, de los modelos, normativas y planes de seguridad, la legislación vigente, las Autoridades de Certificación, sistemas de cifra con curvas elípticas, etc.

Pido entonces mis disculpas si precisamente aquel tema sobre seguridad informática que usted estaba buscando no lo encuentra en estos apuntes. Además, cada vez se vuelve más difícil lograr un documento más o menos completo con una cantidad adecuada de páginas; en este caso estas 900 páginas están ya en el tamaño límite para poder publicarlo en formato libro. En este orden de cosas, le informo que el curso que tiene en sus manos está fuertemente orientado hacia los algoritmos, sistemas de cifra, de firma digital y protocolos criptográficos, enfocado eso sí hacia el mundo de los ordenadores personales e Internet.

La seguridad informática como una materia integral que engloba a diversas disciplinas como las matemáticas, algoritmos, lenguajes, sistemas operativos, sistemas de gestión de bases de datos, protocolos y administración de redes, modelos y políticas de seguridad, la planificación estratégica, hardware, software, auditoría, forensia, etc., es cada vez más amplia e imposible de abordar en un solo texto. Más de alguno entre los que me incluyo, hemos propuesto en algún congreso o foro la necesidad de una nueva carrera: la Ingeniería en Seguridad Informática.

Para asociar un nombre de archivo con un tema en particular y al mismo tiempo mantener los nombres genéricos de archivos ya utilizados en la versión anterior, excepto el primer archivo que corresponde a la presentación del curso, todos los archivos siguen denominándose SItema??.ppt, si bien de forma interna cada uno de ellos tiene un título diferente en las propiedades de archivo, por lo que puede saber el nombre de la lección correspondiente con sólo seleccionarlo o pasar el ratón por sobre el icono, sin tener que abrirlo a priori.

Para un mejor seguimiento de su aprendizaje, le recomiendo que descargue desde el servidor de CriptoRed o la página Web de la asignatura los exámenes de la asignatura de Seguridad Informática que imparto desde 1994 (en estos momentos 17 exámenes con 122 páginas) con sus correspondientes soluciones y, cómo no, el software de prácticas de dicha asignatura que encontrará en el mismo servidor y el cuaderno de prácticas de la misma.

Jorge Ramió Aguirre

PRÓLOGO DE LA SEGUNDA EDICIÓN
Madrid, febrero de 2002

Estimado amigo o amiga:

En estas primeras páginas del Libro Electrónico de Seguridad Informática en Diapositivas y de Libre Distribución por Internet que ha descargado del servidor de la Red Temática CriptoRed o de mi página personal en la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, España, he incluido una versión actualizada y breve del prólogo del libro de la asignatura "Aplicaciones Criptográficas" que dio paso a este curso.

Como posiblemente ya lo sepa, el curso de Seguridad Informática en diapositivas con formato Power Point es un proyecto en constante renovación. En esta fase de comienzos del año 2002 se ha ampliado contenidos y adaptado la versión que existía desde 2000, para hacerla compatible con su impresión en papel. Más adelante, en una siguiente fase, se incluirá información de interés, ejercicios, enlaces, etc.

Por lo tanto, le recomiendo que para una correcta visualización de su contenido, imprima ahora los capítulos ppt en formato documento, en lo posible con dos diapositivas por página hoja y con la opción escala de grises. Recuerde que aunque el fondo de las diapositivas sea azul, el fondo de impresión deberá ser blanco y las letras negras. Debe tener en cuenta además que el formato de impresión está adecuado para papel tamaño DIN A4 que es el usado en Europa y particularmente en España, si bien en América es más estándar el tamaño carta.

Sin más, y con la esperanza de que el libro electrónico de este curso que en su primera versión superó las dos mil descargas desde el servidor pueda serle de provecho, reciba un cordial saludo

Jorge Ramió Aguirre

APLICACIONES CRIPTOGRÁFICAS
PRÓLOGO DE LA 2ª EDICIÓN 1999 DEL LIBRO DE LA ASIGNATURA
(reducida y actualizada)

Uno de los retos más fascinantes de la informática del futuro, inmersa en sistemas de interconexión de redes y autopistas de la información, con un espacio virtual para el intercambio de mensajes y datos a través de canales por definición vulnerables será, sin lugar a dudas, la protección de la información. Representada por archivos confidenciales o mensajes que se intercambian dos o más interlocutores autenticados y cuyo contenido en muchos casos debe mantenerse en secreto por razones personales, empresariales, políticas o de otra índole, la información es el bien más preciado en estos días. Por poner sólo un ejemplo sencillo y común, un problema de gran actualidad es el asociado con el correo electrónico que se transmite a través de redes y cuyo nivel seguridad deja mucho que desear. Internet es un claro ejemplo de estas amenazas en tanto es un entorno abierto en su sentido más amplio. Por lo visto en estos pocos años de existencia de la llamada red de redes, sobran los comentarios acerca de pérdida de privacidad, accesos no autorizados, ataques y otros delitos informáticos a nivel nacional e internacional.

Ante tales amenazas, la única solución consiste en proteger nuestros datos mediante el uso de técnicas criptográficas. Esto nos permitirá asegurar al menos dos elementos básicos de la *Seguridad Informática*, a saber la *confidencialidad* o *secreto* de la información y la *integridad* del mensaje, además de la *autenticidad* del emisor.

A la luz de lo anterior, a mediados de esta década hacen su aparición en los nuevos planes de estudios de Ingeniería Informática que comienzan a implantarse en las universidades españolas, diversas asignaturas sobre seguridad informática y protección de la información, con el objeto de formar a los futuros ingenieros en estos temas. Lo que a comienzos de los 90 aparecía como una tímida apuesta por esta enseñanza en carreras relacionadas principalmente con la Informática y las Telecomunicaciones, hoy después de 10 años permite ver un mapa universitario en España en el que ninguna universidad tecnológica se queda fuera en cuanto a la oferta de este tipo de asignaturas, superándose las 40. Es más, comienzan ya a plantearse perfiles en los nuevos planes de estudio con una clara orientación hacia la seguridad informática de forma integral. Personalmente creo que esto es sólo el comienzo; incluso me he atrevido a proponer ya no sólo un perfil, sino una carrera de ingeniería en la que cerca de un 30% de los créditos estén relacionados con esta materia, en una ponencia presentada en un

congreso el año 2001 y que podrá encontrar en el mismo servidor del que ha descargado este curso. Puede parecer una utopía, para otros tal vez una locura, pero por lo que he visto en estos últimos años, tengo el presentimiento de que quizás el tiempo me dé la razón y veamos en algunos años más como Responsable de Seguridad Informática precisamente a un Ingeniero en Seguridad Informática.

Estoy plenamente consciente que quedan muchos temas interesantes fuera del contexto del libro, como son las nuevas técnicas de cifra con curvas elípticas, estudio y profundización en sistemas actuales de clave secreta que tendrán un importante protagonismo en el futuro como son Skipjack y Rijndael; técnicas y protocolos de autenticación; la teoría de cifra y factorización de polinomios; técnicas, algoritmos y esquemas de cifra en flujo; gestión de claves, protección en entornos de red; esquemas de firma; certificados digitales y Autoridades de Certificación; criptografía visual, criptografía cuántica, esteganografía, autenticación biométrica, etc. Resulta imposible abordar tantos temas y plasmarlos en un libro que tenga un tamaño relativamente normal. Por ello, he decido que las actualizaciones del mismo desde el año 2000 sean en formato electrónico como el que acaba de instalar en su computador; ello permite una más rápida y ágil actualización y, por otra parte, como es de libre distribución, el alcance es mucho mayor. Recuerde que estos apuntes se actualizan de forma periódica; como consejo le recomiendo que visite la página Web indicada dos o tres veces al año, seguro encontrará una versión nueva.

Tras estas palabras, sólo me queda animarle a adentrarse al fascinante mundo de la criptografía y escritura secreta que, por muy *lejano* y *misterioso* que pueda hoy parecerle, en este año 2002 la encontrará en muchos sistemas de acceso, consulta, navegación, intercambio de datos, etc., la mayoría de ellos relacionados con Internet. El cifrado de la información, la firma digital y toda la teoría relacionada con este tema será en esta primera década del siglo XXI algo tan común como lo fue en la década anterior escribir con un procesador de textos, establecer una llamada telefónica mediante un teléfono móvil o incluso ver la televisión vía satélite. En una sociedad informatizada al máximo, nos guste o no la *protección* de esta información y más aún la *autenticidad* de la misma, nos llevará de forma irremediable a un único punto: *el uso de técnicas criptográficas*. No sé qué opina usted, en mi caso estoy seguro de ello.

Madrid, junio de 1999 (y febrero de 2002)
El autor

Dr. Ingeniero de Telecomunicación por la Universidad Politécnica de Madrid. Profesor titular del Departamento de Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid. Es profesor y coordinador de la asignatura de Seguridad Informática que se imparte desde el año 1994 como asignatura optativa de tercer curso en la titulación de Ingeniero Técnico en Informática de Gestión del Plan de Estudios 1992. Es Coordinador General de CriptoRed, la Red Temática Iberoamericana de Criptografía y Seguridad de la Información, en la que participan destacados profesionales e investigadores de más de un centenar de universidades y centros de investigación, así como empresas de sector de las NTIs. Es partner de la Red ECET, European Computer Education and Training, teniendo entre otros objetivos la introducción de las asignaturas de seguridad informática en los planes de estudios superiores de las universidades europeas, y miembro del Subcomité de Seguridad de T.I. (SC 27) del Comité Técnico de Normalización de Tecnología de la Información (CTN 71) de AENOR.
✉ jramio@eui.upm.es
Web personal: http://www.lpsi.eui.upm.es/~jramio
Web asignatura: http://www.lpsi.eui.upm.es/SInformatica/SInformatica.htm
Web CriptoRed: http://www.criptored.upm.es/

CURSO DE SEGURIDAD INFORMÁTICA Y CRIPTOGRAFÍA

LIBRO ELECTRÓNICO DE LIBRE DISTRIBUCIÓN EN INTERNET PARA LA ENSEÑANZA Y EL APRENDIZAJE DE LA SEGURIDAD INFORMÁTICA Y LA PROTECCIÓN DE LA INFORMACIÓN MEDIANTE TÉCNICAS CRIPTOGRÁFICAS

DIAPOSITIVAS ANIMADAS QUE PERMITEN SU IMPRESIÓN EN PAPEL

TERCERA EDICIÓN
Versión v 3.1

Autor: Jorge Ramió Aguirre

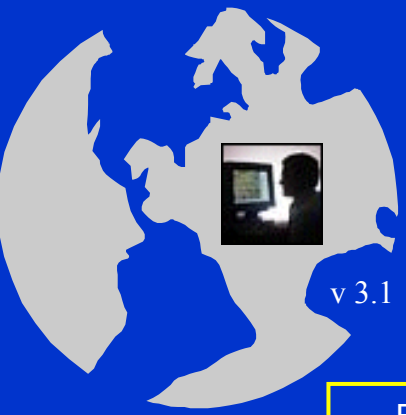
UNIVERSIDAD POLITÉCNICA DE MADRID - ESPAÑA

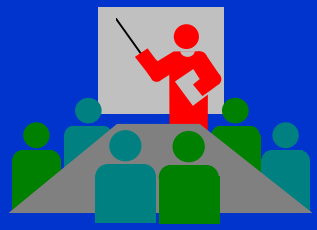
Marzo de 2003

ISBN: 84-86451-69-8
Depósito Legal: M-10039-2003

Presentación del Curso

Seguridad Informática y Criptografía
Tercera edición v 3.1 – 3 de marzo de 2003



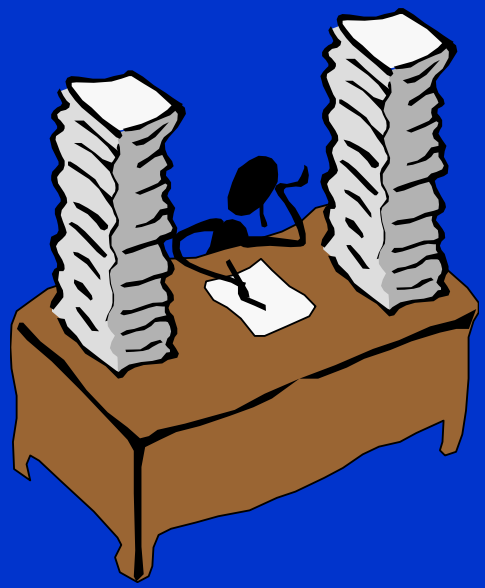


Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 24 diapositivas
Curso completo: 905 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Los derechos de autor



ISBN: 84-86451-69-8
Depósito Legal: M-10039-2003

Recuerde que este curso en diapositivas es el resultado de miles de horas de trabajo.

Temas del curso (1)

Número de diapositivas

Temas del curso (2)

	Número de diapositivas
Tema 10: Cifrado Simétrico en Bloque	87
Tema 11: Cifrado Asimétrico con Mochilas	28
Tema 12: Cifrado Asimétrico Exponencial	56
Tema 13: Funciones Hash en Criptografía	30
Tema 14: Autenticación y Firma Digital	50
Tema 15: Certificados Digitales	11
Tema 16: Aplicaciones Criptográficas	93
Tema 17: Protocolos y Esquemas Criptográficos	72
Tema 18: Bibliografía, Enlaces, SW y Tablas	50
Total diapositivas del curso en la versión v3.1	905

Resumen del Contenido (1)

Resumen del Contenido (2)

Resumen del Contenido (3)

Resumen del Contenido (4)

Resumen del Contenido (5)

Resumen del Contenido (6)

Resumen del Contenido (8)

Resumen del Contenido (9)

Resumen del Contenido (10)

Resumen del Contenido (11)

Resumen del Contenido (12)

Resumen del Contenido (13)

Resumen del Contenido (14)

El curso, la asignatura y este libro

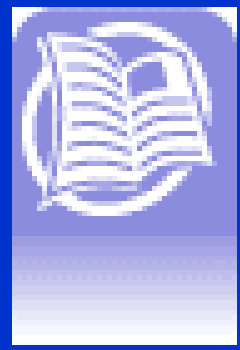
- Este libro electrónico en su calidad de curso en diapositivas de libre distribución en Internet, es parte del material docente que se usa en la asignatura de Seguridad Informática y que el autor imparte desde el año 1994 en la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, España.
- Existe también una versión libro editada por el Departamento de Publicaciones de la Escuela Universitaria de Informática con ISBN: 84-86451-69-8 y Depósito Legal: M-10039-2003.
- Es posible que le resulte más económico adquirir el libro en vez imprimirlo o sacar fotocopias. Para cualquier consulta en este sentido sobre condiciones de compra y envío del mismo, dentro o fuera de España, póngase en contacto con el Departamento de Publicaciones, EUI-UPM, Carretera de Valencia Km 7, 28031, Madrid, España. O bien por email a publicaciones@eui.upm.es o telefónicamente al número +34 91 3367905. Precio € 16.

Sobre el autor

- Jorge Ramió Aguirre es profesor titular en el Departamento de Lenguajes, Proyectos y Sistemas Informáticos de la Universidad Politécnica de Madrid.
  http://www.lpsi.eui.upm.es
- Desde el curso 1994/1995 imparte la asignatura de Seguridad Informática en la titulación de Ingeniero Técnico en Informática de Gestión en la Escuela Universitaria de Informática de dicha universidad.
  http://www.lpsi.eui.upm.es/SInformatica/SInformatica.htm
- Es el creador y Coordinador General de CriptoRed, la Red Temática Iberoamericana de Criptografía y Seguridad de la Información, desde diciembre de 1999 y que en febrero de 2003 cuenta con más de 110 universidades representadas y una centena de empresas.
  http://www.criptored.upm.es
- Ha impartido diversas conferencias y cursos sobre criptografía y seguridad informática en Argentina, Bolivia, Chile, Colombia, Cuba, España, México, Uruguay y Venezuela.
  http://www.lpsi.eui.upm.es/~jramio

El primer libro de la asignatura

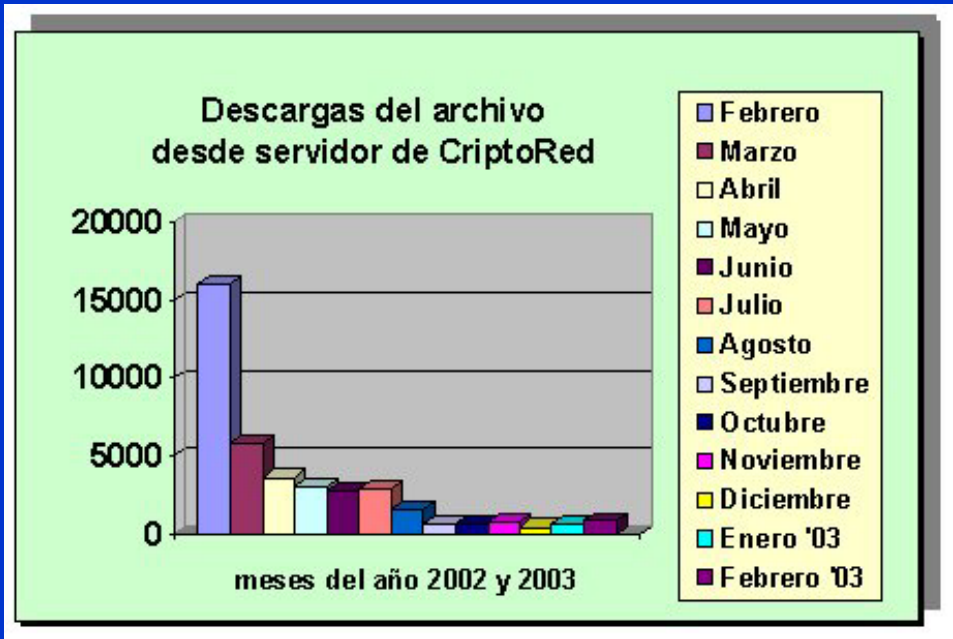
"Aplicaciones Criptográficas", 2ª edición, Junio de 1999
Departamento de Publicaciones - Escuela Universitaria de Informática - Universidad Politécnica de Madrid
Carretera de Valencia km. 7 - 28031 Madrid (España)
I.S.B.N.: 84-87238-57-2. Depósito Legal: M-24709-1999



"Si dotamos a Internet de las medidas de protección y seguridad que nos ofrecen las técnicas criptográficas, dejará de ser ese peligroso y caótico tablón de anuncios para convertirse en el supermercado electrónico del futuro y la herramienta de trabajo de la generación del próximo siglo."

A Anita y Jordi

Descargas de la versión v2 del libro



Febrero 2002	16.020
Marzo 2002	5.892
Abril 2002	3.655
Mayo 2002	3.077
Junio 2002	2.717
Julio 2002	2.940
Agosto 2002	1.602
Septiembre 2002	719
Octubre 2002	674
Noviembre 2002	765
Diciembre 2002	462
Enero 2003	690
Febrero 2003	852

Número total de descargas de la versión v2 desde el 11/02/02 al 28/02/03: 40.065.

- La versión v1 del libro electrónico alcanzó unas descargas ligeramente superiores a 2.000 durante el año 2001 desde el mismo servidor.

Nuevo formato imprimible

- Esta edición ha sido adaptada para que, además de su animación como material de apoyo docente y de autoaprendizaje, pueda también imprimirse en papel como un documento de estudio y consulta. Esto en caso de que opte por no adquirir el libro en el Departamento de Publicaciones según ya se ha comentado.
- Estos archivos forman parte de un Proyecto Docente del autor y cuyo objetivo es la elaboración de una documentación seria sobre seguridad informática y criptografía, en formato electrónico de libre distribución en Internet.
- IMPORTANTE: Para una lectura más cómoda, se recomienda imprimir dos diapositivas por página en formato documentos DIN A4 con impresión en segundo plano, el fondo debe ser blanco. Si imprime más de dos diapositivas por página, algunas de ellas se verán con un tamaño demasiado pequeño.

Diferencias con la versión v2

El libro en su versión v3.1 cuenta con 19 temas, uno más que en la versión v2 de febrero de 2002, al incluir uno dedicado a la bibliografía, enlaces, software y tablas. Se han revisado y actualizado todos los capítulos (en especial los temas de cifra simétrica y cifra asimétrica, de autenticación, protocolos y correo seguro con PGP). Además, al final de cada capítulo se ha incluido un conjunto de cuestiones y ejercicios, en total 268 preguntas. En la medida de lo posible, las soluciones a esta cuestiones y ejercicios se irán incluyendo de forma gradual en la página Web de la asignatura.

En esta versión se siguen usando los mismos nombres para los archivos, es decir SItema??.ppt. No obstante, puede saber el contenido de cada capítulo y su nombre sencillamente viendo las propiedades de éste, sin necesidad de abrirlo. En algunas versiones de Windows, esto se obtiene sólo con pasar el ratón sobre el icono del archivo correspondiente. Si desea ir a una diapositiva específica dentro de un archivo, use la Vista Normal de Power Point, preferiblemente con la opción contraer activada.

Actualización de los archivos

Tenga en cuenta que los temas y contenidos de estos archivos serán actualizados de forma continua. Luego, cuando observe en las páginas Web indicadas la existencia de una versión más actualizada del archivo SItemas.zip que la que tiene en su ordenador, reemplace el archivo zip anterior y, al descomprimirlo, reemplace todos los archivos ppt.

NOTA: No obstante, la documentación será en esencia la misma que la del libro editado mientras se mantenga la versión 3.1. En la página Web del autor y de la asignatura se irán poniendo en un archivo aparte las nuevas diapositivas que se vayan generando de cara a una nueva edición.

El curso comienza con un primer archivo llamado SItema00 cuyo objetivo es entregar una introducción a la criptografía. Si está comenzando estos estudios, le recomiendo su visualización previa al curso en tanto es un resumen claro y preciso sobre muchos de los temas y algoritmos que serán tratados en más detalle a lo largo de los capítulos.

Fin presentación

Tema 0
Una Introducción a la Criptografía

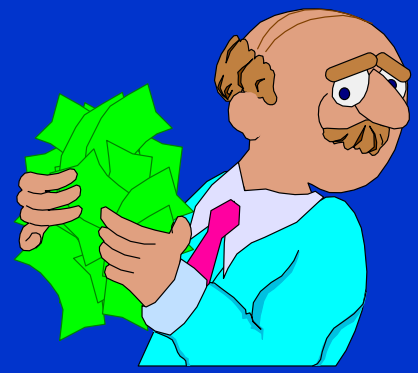
Seguridad Informática y Criptografía

Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 14 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Nota del autor



El contenido de este tema de introducción está orientado a una primera visión de tipo generalista al tema sobre en el que se profundizará a lo largo del libro: la seguridad y protección de la información mediante el uso de técnicas y algoritmos criptográficos. En particular, se trata de visión rápida del concepto de cifra y firma digital asociado a los criptosistemas y que serán analizados en detalle en los próximos capítulos. Si lo desea, puede utilizar estas diapositivas para una charla introductoria al tema de la seguridad y criptografía de unos 30 minutos.

Criptografía según la RAE

Criptografía

una definición ...

La Real Academia Española define criptografía (oculto + escritura) como:

"el arte de escribir mensajes con una clave secreta o de modo enigmático".

Resulta *difícil* dar una definición *tan poco ajustada* a la realidad actual.

Véase la siguiente diapositiva

Imprecisiones de esta definición

Arte: la criptografía ha dejado de ser un arte: es una ciencia.

Escritura de mensajes: ya no sólo se escriben mensajes; se envía o se guarda en un ordenador todo tipo de documentos e información de distintos formatos (txt, doc, exe, gif, jpg, ...).

Una clave: los sistemas actuales usan más de una clave.

Clave secreta: existirán sistemas de clave secreta que usan una sola clave y sistemas de clave pública (muy importantes) que usan dos: una clave privada (secreta) y la otra pública.

Representación enigmática: la representación binaria de la información podría ser enigmática para nosotros los humanos pero jamás para los ordenadores ☺ ... es su lenguaje natural.

Una definición más técnica de criptografía

Criptografía

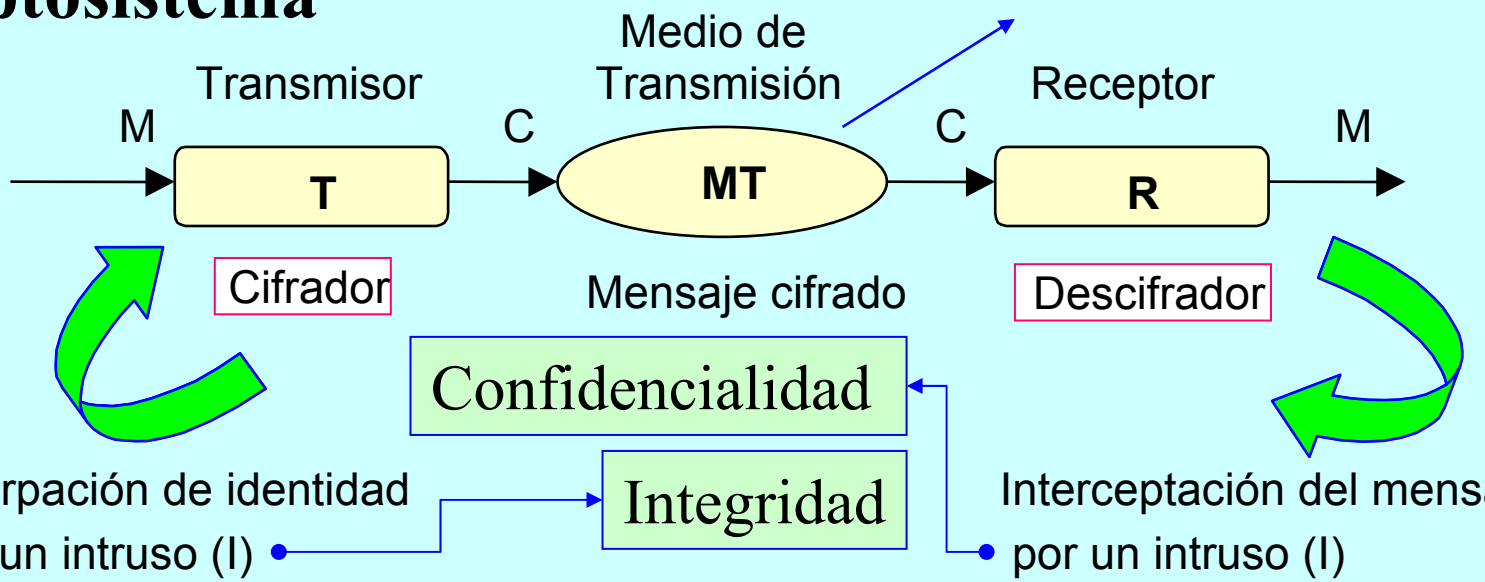
Lo que realmente es

Rama inicial de las Matemáticas y en la actualidad de la Informática y la Telemática, que hace uso de métodos y técnicas con el objeto principal de cifrar un mensaje o archivo por medio de un algoritmo, usando una o más claves. Esto da lugar a diferentes tipos de criptosistemas que permiten asegurar cuatro aspectos fundamentales de la seguridad informática: confidencialidad, integridad, disponibilidad y no repudio de emisor y receptor.

Confidencialidad e integridad

Criptosistema

Cualquier medio de transmisión es inseguro



Estos dos principios de la seguridad informática, el de la confidencialidad y la integridad, (además de la disponibilidad y el no repudio) serán muy importantes en un sistema de intercambio de información segura a través de Internet.

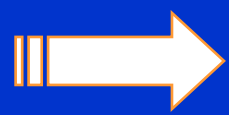
Tipos de criptosistemas

Clasificación de los criptosistemas

Según el tratamiento del mensaje se dividen en:

- Cifrado en bloque (DES, IDEA, RSA) 64-128 bits
- Cifrado en flujo (A5, RC4, SEAL) cifrado bit a bit

Según el tipo de claves se dividen en:



- Cifrado con clave secreta — **Sistemas simétricos**
- Cifrado con clave pública — **Sistemas asimétricos**

Criptosistemas simétricos y asimétricos

Criptosistemas simétricos:

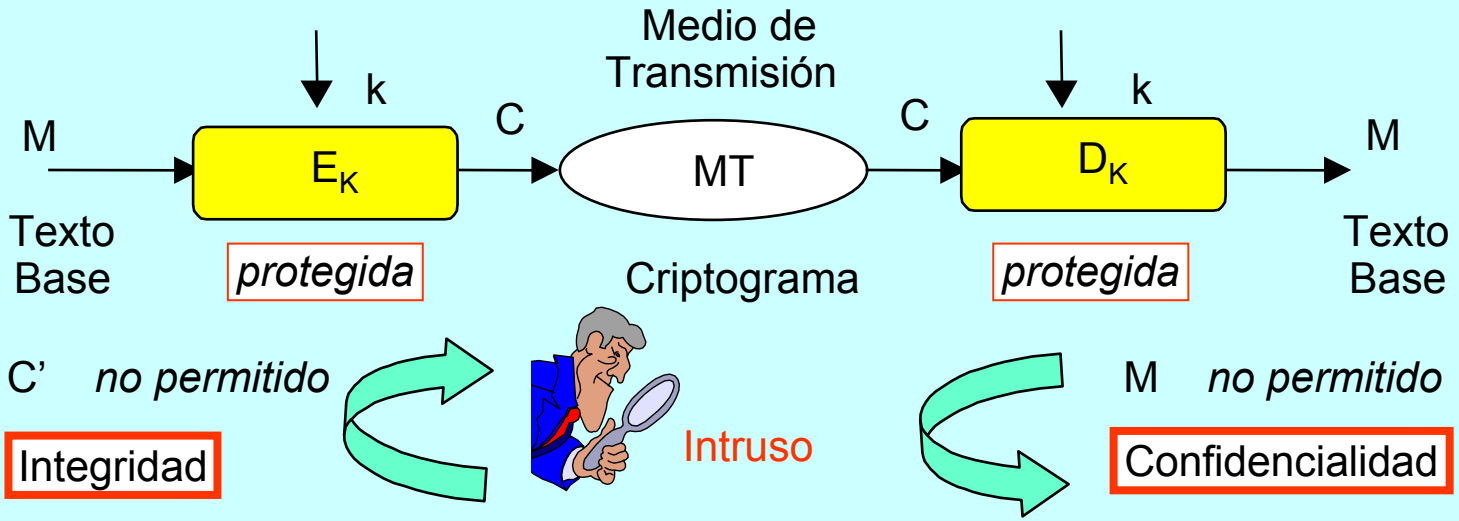
Existirá una única clave (secreta) que deben compartir emisor y receptor. Con la misma clave se cifra y se descifra por lo que la seguridad reside sólo en mantener dicha clave en secreto.

Criptosistemas asimétricos:

Cada usuario crea un par de claves, una privada y otra pública, inversas dentro de un cuerpo finito. Lo que se cifra en emisión con una clave, se descifra en recepción con la clave inversa. La seguridad del sistema reside en la dificultad computacional de descubrir la clave privada a partir de la pública. Para ello usan funciones matemáticas de un solo sentido con trampa.

Criptosistemas simétricos

Cifrado con criptosistemas de clave secreta



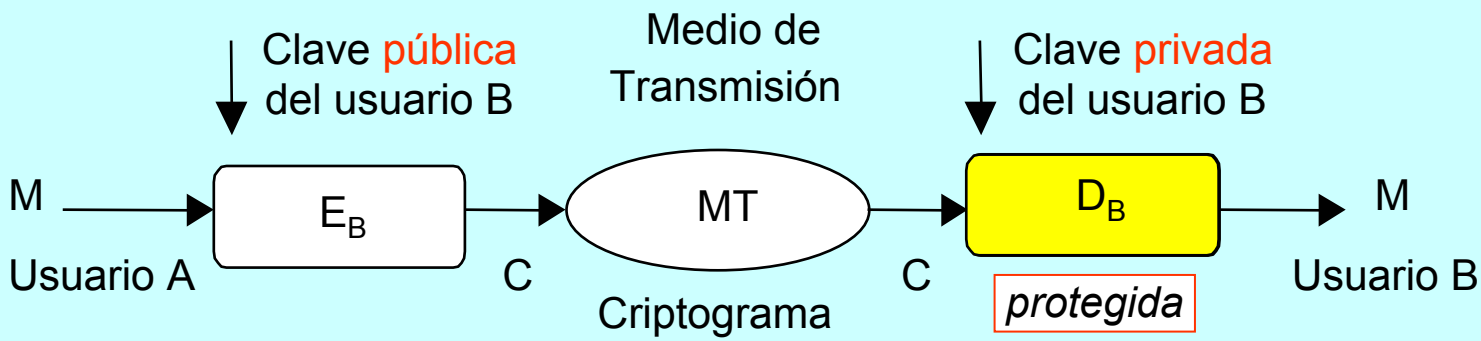
La confidencialidad y la integridad se lograrán si se protegen las claves en el cifrado y en el descifrado. Es decir, se obtienen *simultáneamente* si se protege la clave secreta.

DES, TDES, IDEA, CAST, RIJNDAEL

Criptosistemas asimétricos (1)

Cifrado con clave pública del receptor

Intercambio de claves RSA



Intruso

M *no permitido*

Confidencialidad

Observe que se cifra con la clave pública del destinatario.

Las cifras E_B y D_B (claves) son inversas dentro de un cuerpo

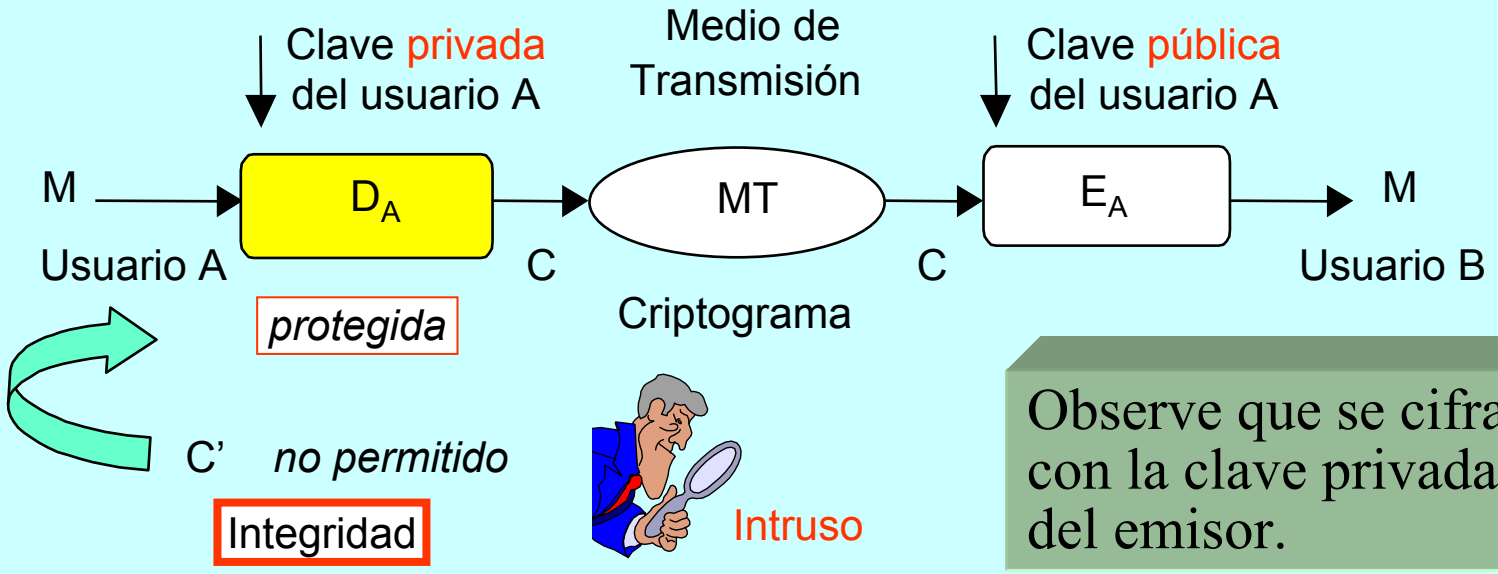
Un sistema similar es el intercambio de clave de Diffie y Hellman (DH)

Criptosistemas asimétricos (2)

Cifrado con clave privada del emisor

Firma digital RSA

Firmas: RSA y DSS



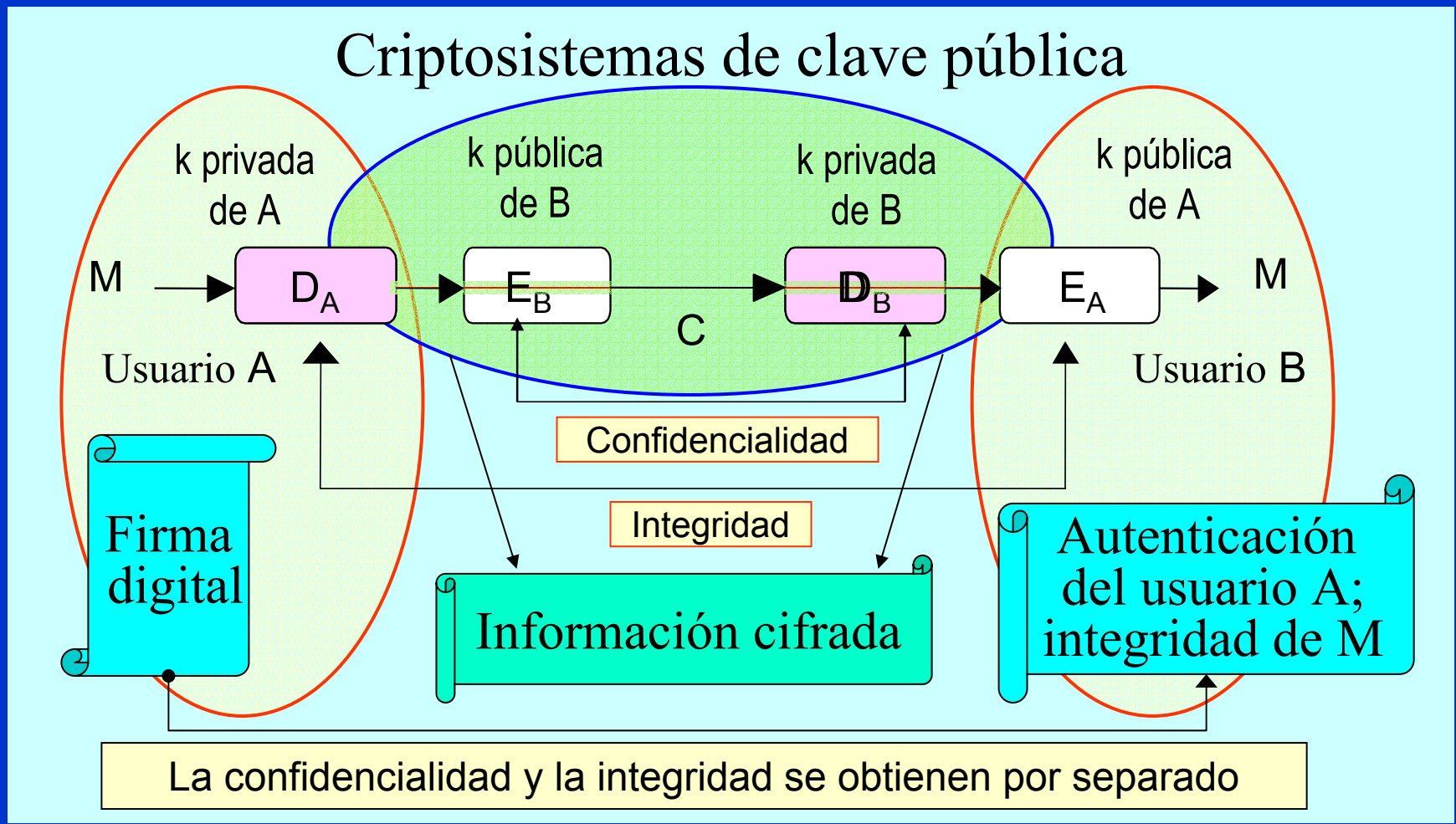
Observe que se cifra con la clave privada del emisor.

Se firma sobre un hash H(M) del mensaje, por ejemplo MD5 o SHA-1

Las cifras D_A y E_A (claves) son inversas dentro de un cuerpo

La firma DSS está basada en el algoritmo de cifra de El Gamal

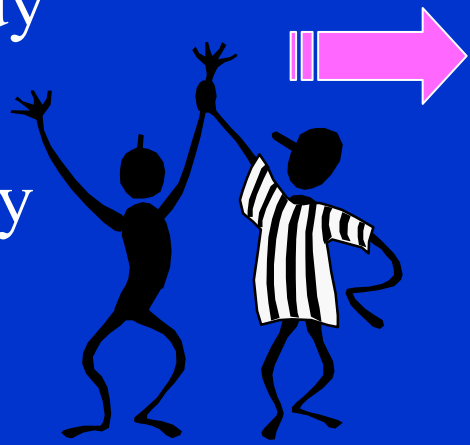
Tipos de cifra con sistemas asimétricos

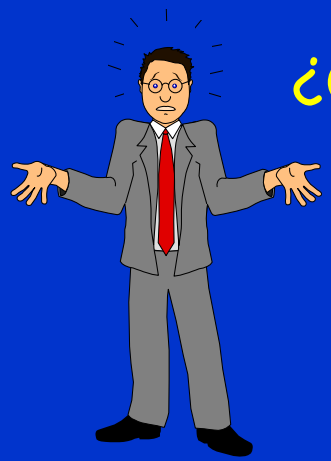


¿Qué usar, simétricos o asimétricos?

Los sistemas de clave pública son muy lentos pero tienen firma digital.

Los sistemas de clave secreta son muy rápidos pero no tienen firma digital.



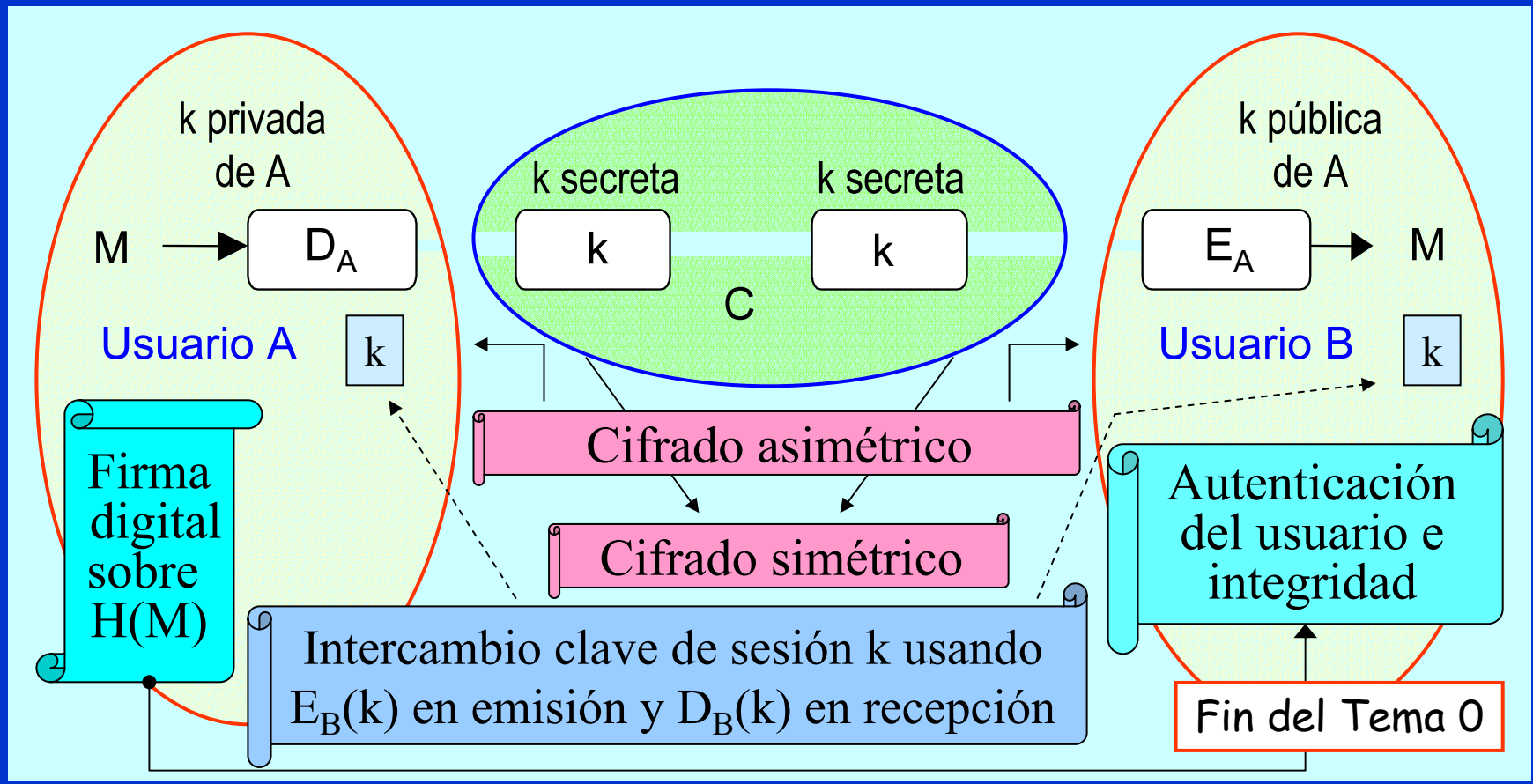


¿Qué hacer?

Cifrado de la información:
Sistemas de clave secreta

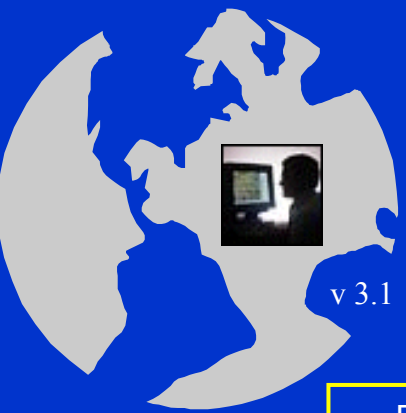
Firma e intercambio de clave de sesión:
Sistemas de clave pública

Sistema híbrido de cifra y firma



Tema 1
Introducción a la Seguridad Informática

Seguridad Informática y Criptografía



Material Docente de
Libre Distribución

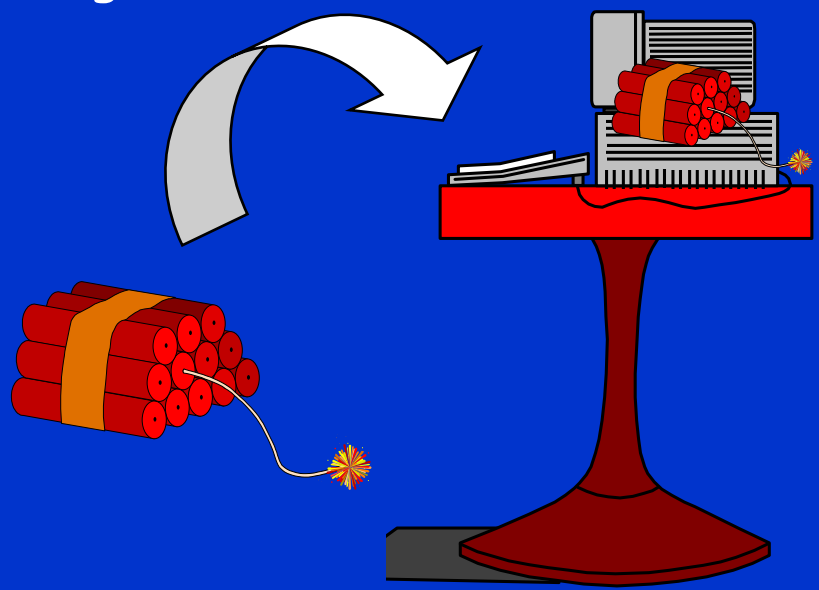
Ultima actualización: 03/03/03
Archivo con 54 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

¿Conectado o desconectado?

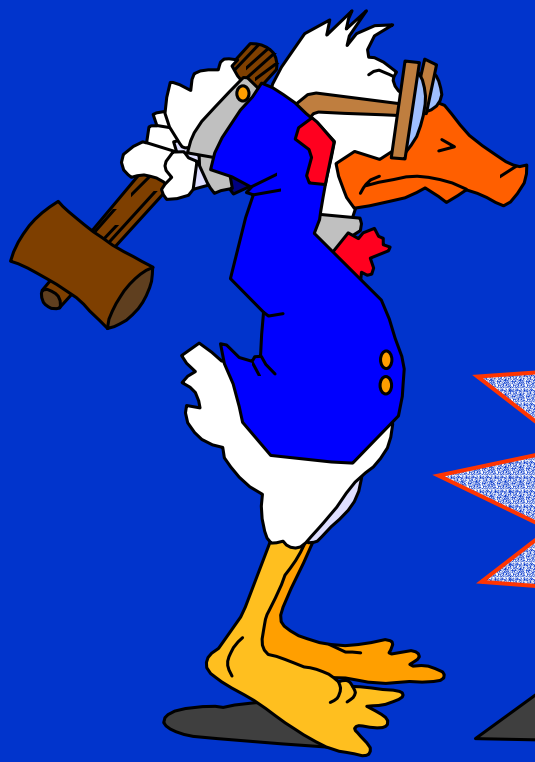
No podemos aceptar esa afirmación popular que dice que el computador más seguro ...

... es aquel que está apagado y, por tanto, desconectado de la red.



A pesar de todas las amenazas del entorno que, como veremos, serán muchas y variadas.

¿Hay conciencia de las debilidades?



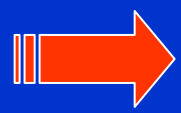
internas o externas

La seguridad informática será un motivo de preocupación.

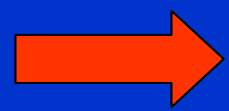
A finales del siglo XX las empresas, organismos y particulares comienzan a tener verdadera conciencia de su importancia.

Acontecimientos en dos últimas décadas

- A partir de los años 80 el uso del ordenador personal comienza a ser común. Asoma ya la preocupación por la integridad de los datos.
- En la década de los años 90 proliferan los ataques a sistemas informáticos, aparecen los virus y se toma conciencia del peligro que nos acecha como usuarios de PCs y equipos conectados a Internet.
- Las amenazas se generalizan a finales de los 90.
- En los 00s los acontecimientos fuerzan a que se tome en serio la seguridad informática.



¿Qué hay de nuevo en los 00s?

- Principalmente por el uso de Internet, el tema de la protección de la información se transforma en una necesidad y con ello se populariza la terminología técnica asociada a la criptología:
  - Cifrado, descifrado, criptoanálisis, firma digital.
  - Autoridades de Certificación, comercio electrónico.
- Ya no sólo se transmiten estas enseñanzas en las universidades. El usuario final desea saber, por ejemplo, qué significa *firmar* un e-mail.
- Productos futuros:  Seguridad añadida

Una definición de criptografía

Criptografía:

Rama de las Matemáticas y en la actualidad de la Informática que hace uso de métodos y herramientas con el objeto principal de cifrar un mensaje o archivo por medio de un algoritmo y una o más claves, dando lugar a distintos criptosistemas que permiten asegurar, al menos, dos aspectos básicos de la seguridad como son la confidencialidad y la integridad de la información.

Aquí tenemos una definición algo menos afortunada de criptografía que vemos en el diccionario de la Real Academia Española ...

Una definición menos afortunada...

Criptografía:

"Arte de escribir mensajes con una clave secreta o de modo enigmático".

Desde el punto de vista de la ingeniería y la informática, es difícil encontrar una definición menos apropiada ☹.

- Hoy ya no es un arte sino una ciencia.
- No sólo se protegen mensajes que se escriben, sino archivos y documentos en general que se generan.
- Muchos sistemas usan dos claves: secreta y pública.
- No hay nada de enigmático ☺ en una cadena de bits.

¿Cifrar o encriptar?

Cifra o cifrado:

Técnica que, en general, protege o autentica a un documento o usuario al aplicar un algoritmo criptográfico. Sin conocer una clave específica, no será posible descifrarlo o recuperarlo.

En algunos países por influencia del inglés se usará la palabra encriptar. Si bien esta palabra todavía no existe, bien podría ser el acto de "meter a alguien dentro de una cripta", ✝☠✝... algo bastante distinto a lo que deseamos expresar ☺.

Situaciones parecidas a ésta encontraremos muchas. Por ejemplo, podemos ver en algunos documentos las palabras autentificación, securizar y otras parecidas, que a la fecha no están recogidas en el diccionario de la Real Academia Española. Peor aún, no podemos encontrar en ese diccionario palabras tan comunes como factorizar, factorización, primalidad, criptólogo, criptógrafo, criptoanalista, ...

Algunas definiciones previas

Criptología: ciencia que estudia e investiga todo aquello relacionado con la criptografía: incluye cifra y criptoanálisis.

Criptógrafo: máquina o artilugio para cifrar.

Criptólogo: persona que trabaja de forma legítima para proteger la información creando algoritmos criptográficos.

Criptoanalista: persona cuya función es romper algoritmos de cifra en busca de debilidades, la clave o del texto en claro.

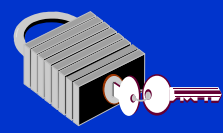
Texto en claro: documento original. Se denotará como M.

Criptograma: documento/texto cifrado. Se denotará como C.

Claves: datos (llaves) privados/públicos que permitirán cifrar.

¿Es el delito informático interesante?

- El delito informático parece ser un *buen negocio*:
  - Objeto Pequeño: la información está almacenada en contenedores pequeños: no es necesario un camión para robar un banco, llevarse las joyas, el dinero, ...
  - Contacto Físico: no existe contacto físico en la mayoría de los casos. Se asegura el anonimato y la integridad física del propio delincuente.
  - Alto Valor: el objeto codiciado tiene un alto valor. El contenido (los datos) puede valer mucho más que el soporte que los almacena: computador, disquete, CD, ...
- ¿Solución? uso de técnicas criptográficas.



Seguridad Física v/s Seguridad Lógica

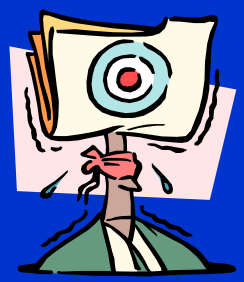
- El estudio de la seguridad informática podemos plantearlo desde dos enfoques:
  - Seguridad Física: protección del sistema ante las amenazas físicas, planes de contingencia, control de acceso físico, políticas de seguridad, normativas, etc. Este tema será tratado brevemente en el capítulo 3.
  - Seguridad Lógica: protección de la información en su propio medio mediante el enmascaramiento de la misma usando técnicas de criptografía. Este enfoque propio de las Aplicaciones Criptográficas es el que será tratado a lo largo de todo el curso.
  - No obstante, tenga en cuenta que esta clasificación en la práctica no es tan rigurosa.

Principios de la seguridad informática

- Veremos a continuación los tres principios básicos de la seguridad informática: el del acceso más fácil, el de la caducidad del secreto, y el de la eficiencia de las medidas tomadas.
- Tras los acontecimientos del 11 de septiembre de 2001, que de alguna forma ha hecho a la gente pensar en las debilidades de los sistemas, vale la pena tenerlos muy en cuenta.

Deberíamos aprender la lección ☹

1er principio de la seguridad informática



PREGUNTA:
¿Cuáles son los puntos débiles de un sistema informático?

- "El intruso al sistema utilizará cualquier artilugio que haga más fácil su acceso y posterior ataque".
- Existirá una diversidad de frentes desde los que puede producirse un ataque. Esto dificulta el análisis de riesgos porque el delincuente aplica la filosofía de ataque hacia el punto más débil.

2º principio de la seguridad informática

PREGUNTA:
¿Cuánto tiempo deberá protegerse un dato?

- "Los datos confidenciales deben protegerse sólo hasta ese secreto pierda su valor".
- Se habla, por tanto, de la caducidad del sistema de protección: tiempo en el que debe mantenerse la confidencialidad o secreto del dato.
- Esto nos llevará a la fortaleza del sistema de cifra.

3er principio de la seguridad informática

- "Las medidas de control se implementan para ser utilizadas de forma efectiva. Deben ser eficientes, fáciles de usar y apropiadas al medio".
  - Que funcionen en el momento oportuno.
  - Que lo hagan optimizando los recursos del sistema.
  - Que pasen desapercibidas para el usuario.

Medidas de control

- Y lo más importante: ningún sistema de control resulta efectivo hasta que es utilizado al surgir la necesidad de aplicarlo. Este es uno de los grandes problemas de la Seguridad Informática.

Debilidades del sistema informático (1)

HARDWARE - SOFTWARE - DATOS

MEMORIA - USUARIOS

Los tres primeros puntos conforman el llamado Triángulo de Debilidades del Sistema:

- Hardware: pueden producirse errores intermitentes, conexiones suelta, desconexión de tarjetas, etc.
- Software: puede producirse la sustracción de programas, ejecución errónea, modificación, defectos en llamadas al sistema, etc.
- Datos: puede producirse la alteración de contenidos, introducción de datos falsos, manipulación fraudulenta de datos, etc.

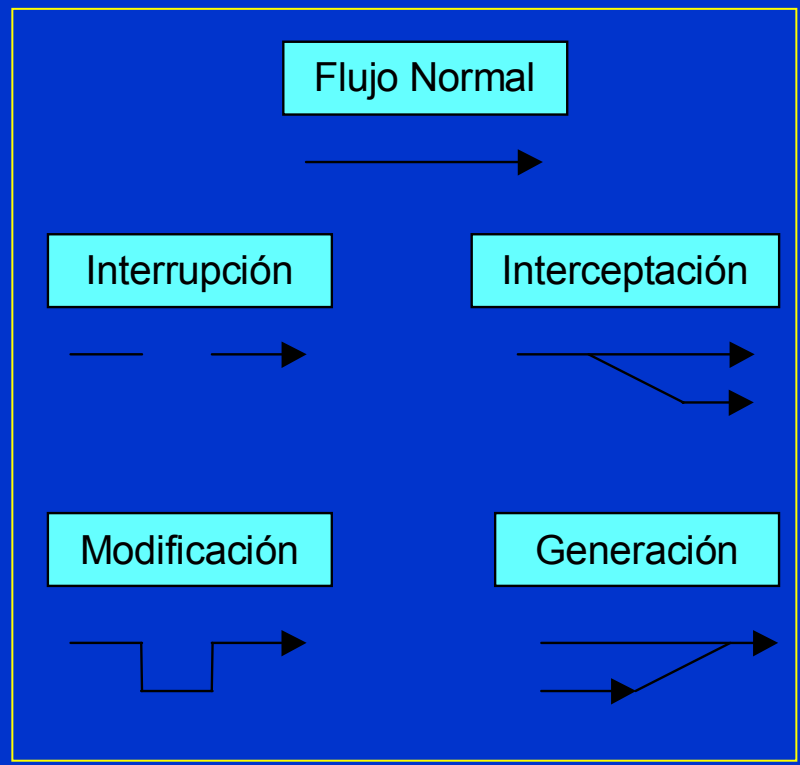
Debilidades del sistema informático (2)

- Memoria: puede producirse la introducción de un virus, mal uso de la gestión de memoria, bloqueo del sistema, etc.
- Usuarios: puede producirse la suplantación de identidad, el acceso no autorizado, visualización de datos confidenciales, etc.

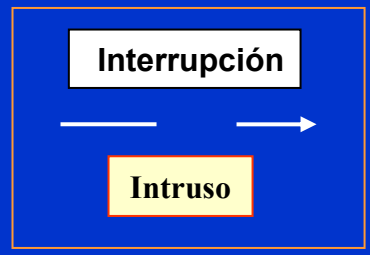
- Es muy difícil diseñar un plan que contemple minimizar de forma eficiente todos estos aspectos negativos.
- Debido al Principio de Acceso más Fácil, no se deberá descuidar ninguno de los cinco elementos susceptibles de ataque del sistema informático.

Amenazas del sistema

- Las amenazas afectan principalmente al Hardware, al Software y a los Datos. Éstas se deben a fenómenos de:
  - Interrupción
  - Interceptación
  - Modificación
  - Generación



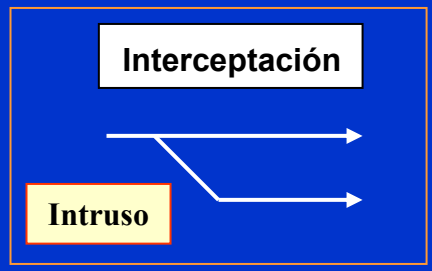
Amenazas de interrupción



- Se daña, pierde o deja de funcionar un punto del sistema.
- Su detección es inmediata.

Ejemplos: Destrucción del hardware.
Borrado de programas, datos.
Fallos en el sistema operativo.

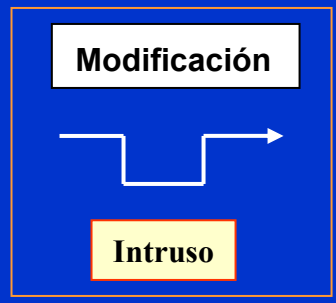
Amenazas de interceptación



- Acceso a la información por parte de personas no autorizadas. Uso de privilegios no adquiridos.
- Su detección es difícil, a veces no deja huellas.

Ejemplos: Copias ilícitas de programas.
Escucha en línea de datos.

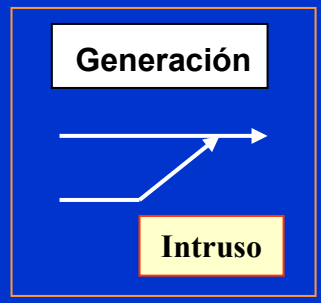
Amenazas de modificación



- Acceso no autorizado que cambia el entorno para su beneficio.
- Su detección es difícil según las circunstancias.

Ejemplos: Modificación de bases de datos.
Modificación de elementos del HW.

Amenazas de generación



- Creación de nuevos objetos dentro del sistema.
- Su detección es difícil: delitos de falsificación.

Ejemplos: Añadir transacciones en red.
Añadir registros en base de datos.

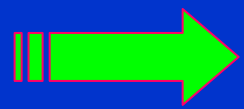
El triángulo de debilidades

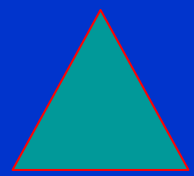
Interrupción (pérdida)	**Interceptación** (acceso)	**Modificación** (cambio)	**Generación** (alteración)

DATOS

Los datos serán la parte más vulnerable del sistema.

Ejemplos de ataques





HW

SW

Interrupción (denegar servicio)
Interceptación (robo)

Modificación (falsificación)
Interrupción (borrado)
Interceptación (copia)

Ataques característicos

- Hardware:
  - Agua, fuego, electricidad, polvo, cigarrillos, comida.
- Software:
  - Además de algunos de hardware, borrados accidentales o intencionados, estática, fallos de líneas de programa, bombas lógicas, robo, copias ilegales.
- Datos:
  - Tiene los mismos puntos débiles que el software. Pero hay dos problemas añadidos: no tienen valor intrínseco pero sí su interpretación y, por otra parte, algunos datos pueden ser de carácter público.

Confidencialidad, integridad, disponibilidad

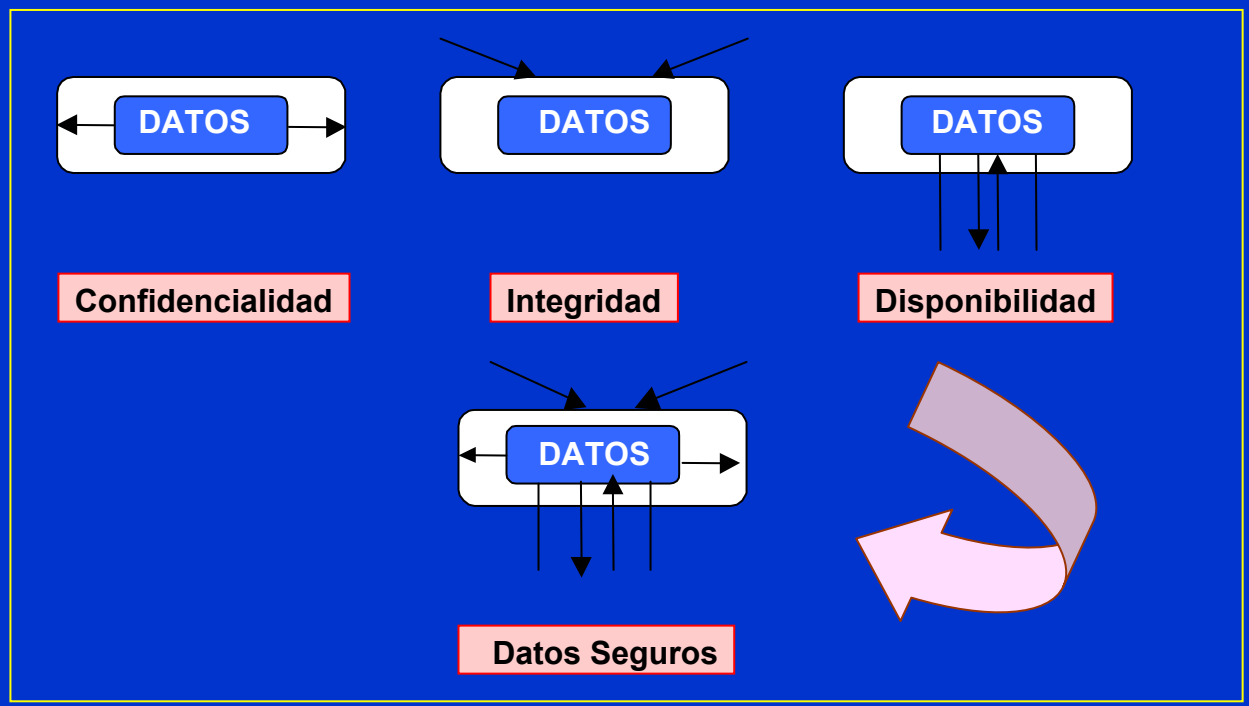
- Confidencialidad
  - Los componentes del sistema serán accesibles sólo por aquellos usuarios autorizados.
- Integridad
  - Los componentes del sistema sólo pueden ser creados y modificados por los usuarios autorizados.
- Disponibilidad
  - Los usuarios deben tener disponibles todos los componentes del sistema cuando así lo deseen.

No repudio de origen y destino

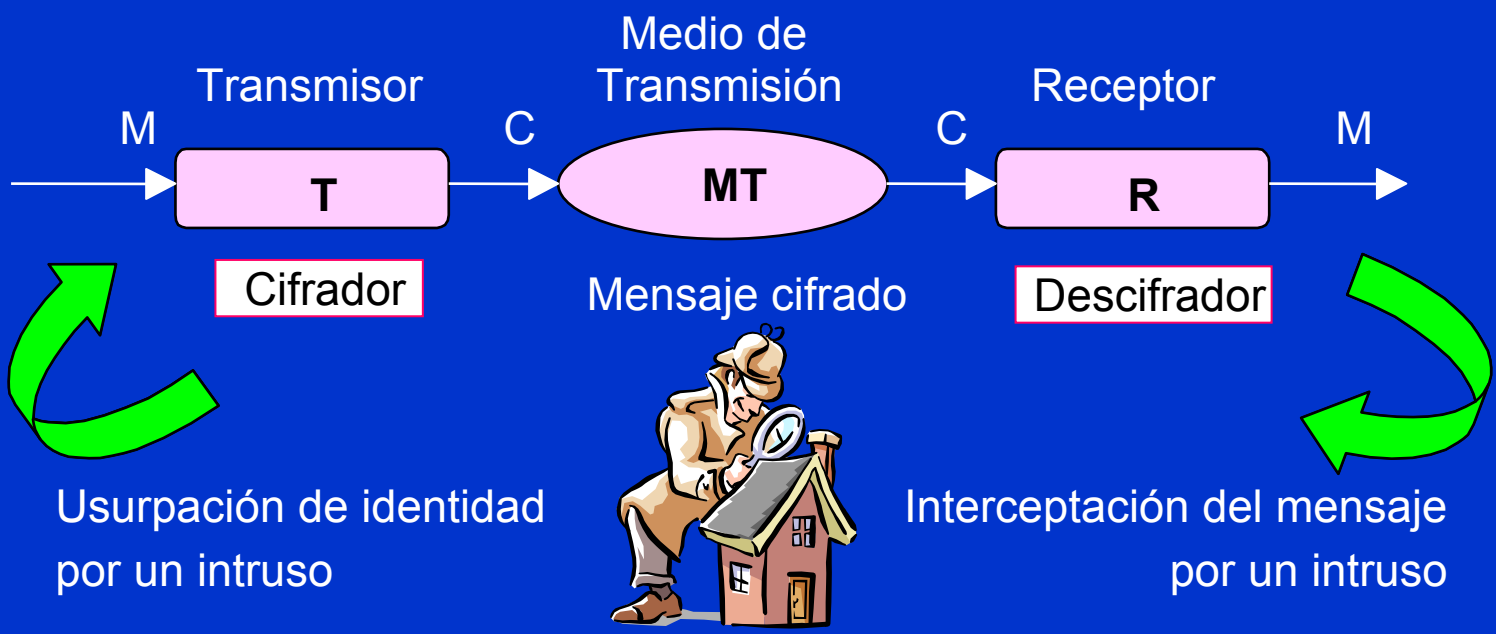
- No Repudio
  - Este término se ha introducido en los últimos años como una característica más de los elementos que conforman la seguridad en un sistema informático.
  - Está asociado a la aceptación de un protocolo de comunicación entre emisor y receptor (cliente y servidor) normalmente a través del intercambio de sendos certificados digitales de autenticación.
  - Se habla entonces de No Repudio de Origen y No Repudio de Destino, forzando a que se cumplan todas las operaciones por ambas partes en una comunicación.

Datos seguros

Si se cumplen estos principios, diremos en general que los datos están protegidos y seguros.

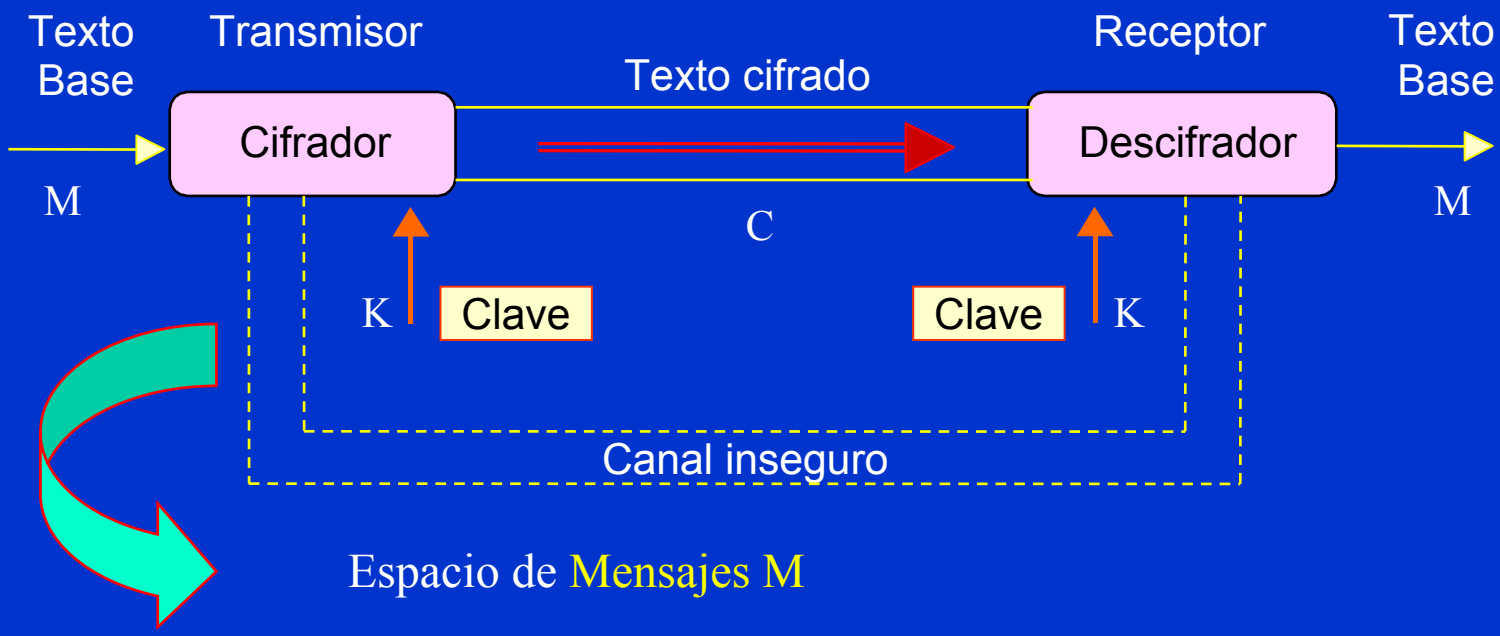


Sistema de cifra



Sea cual sea el medio de transmisión (enlace, red telefónica, red de datos, disco magnético, disco óptico, etc.) éste será siempre y por definición inseguro.

Esquema de un criptosistema



Se habla entonces de:

Espacio de Mensajes M

Espacio de Textos Cifrados C

Espacio de Claves K

Transformaciones de Cifrado y de Descifrado

Funciones y operaciones de cifra

- C = E(M)
- M = D(C)
- M = D(E(M))

Si se usa una clave k:

- C = E(k,M) o $E_k(M)$
- M = D(k, E(k,M))
- M = $D(k_D, E(k_E,M))$

E: Cifrado del mensaje M

D: Descifrado del criptograma C

Las operaciones D y E son inversas o bien lo son las claves que intervienen. Esto último es lo más normal, con los inversos dentro de un cuerpo finito. Por lo tanto, se recupera el mensaje en claro.

En este último caso los algoritmos E y D son iguales

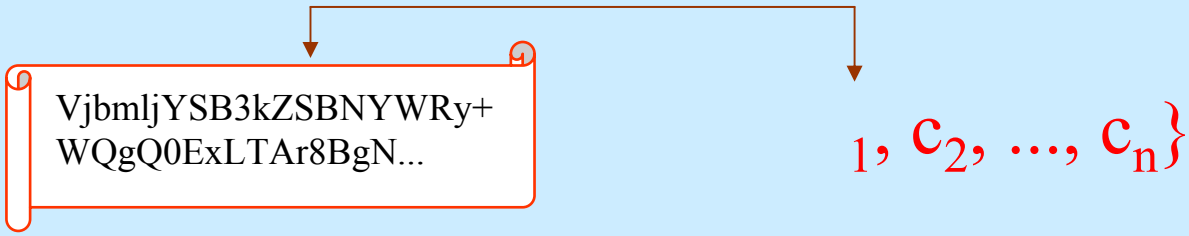
Espacio de mensajes M

Está muy claro que esto es un texto en claro...

$$M = \{m_1, m_2, ..., m_n\}$$

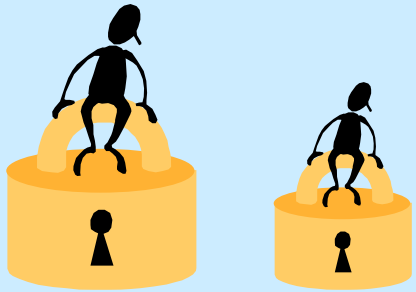
- Componentes de un mensaje inteligible (bits, bytes, pixels, signos, caracteres, etc.) que provienen de un alfabeto previamente establecido como en el ejemplo.
- El lenguaje tiene unas reglas sintácticas y semánticas.
- En algunos casos y para los sistemas de cifra clásicos la longitud del alfabeto indicará el módulo en el cual se trabaja. En los modernos, no guarda relación.
- Habrá mensajes con sentido y mensajes sin sentido.

Espacio de textos cifrados C



- Normalmente el alfabeto es el mismo que el utilizado para crear el mensaje en claro.
- Supondremos que el espacio de los textos cifrados C y el espacio de los mensaje M (con y sin sentido) tienen igual magnitud.
- En este caso, a diferencia del espacio de mensajes M, serán válidos todo tipo de criptogramas.

Espacio de claves K



$K = \{k_1, k_2, ..., k_n\}$

- Si el espacio de claves K es tan grande como el de los mensajes M, se obtendrá un criptosistema con secreto perfecto.
- Se supone que es un conjunto altamente aleatorio de caracteres, palabras, bits, bytes, etc., en función del sistema de cifra. Al menos una de las claves en un criptosistema se guardará en secreto.

Transformaciones de cifrado E_k

$$E_k: M \rightarrow C \qquad k \in K$$

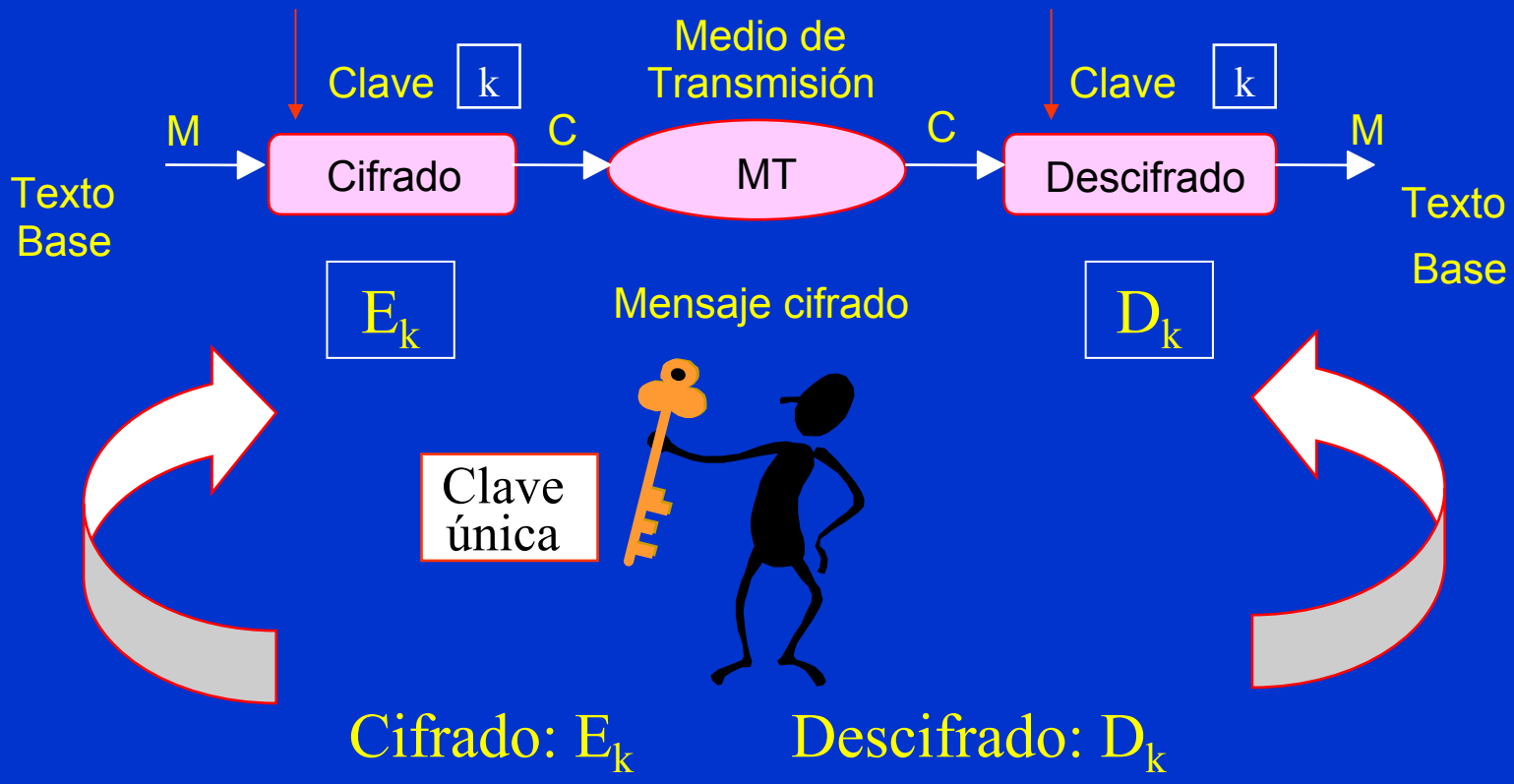
- E_k es una aplicación con una clave k, que está en el espacio de claves K, sobre el mensaje M y que lo transforma en el criptograma C.
- Es el algoritmo de cifra. Sólo en algunos sistemas clásicos el algoritmo es secreto. Por lo general el algoritmo de cifra será de dominio público y su código fuente debería estar disponible en Internet.

Transformaciones de descifrado D_k

$$D_k: C \rightarrow M \qquad k \in K$$

- D_k es una aplicación con una clave k, que está en el espacio de claves K, sobre el criptograma C y que lo transforma en el texto en claro M.
- Se usa el concepto de inverso. D_k será la operación inversa de E_k o bien -que es lo más común- se usa la misma transformación E_k para descifrar pero con una clave k' que es la inversa de k dentro de un cuerpo.

Criptosistemas de clave secreta



Requisitos de un criptosistema

- Algoritmo de cifrado/descifrado rápido y fiable.
- Posibilidad de transmitir ficheros por una línea de datos, almacenarlos o transferirlos.
- No debe existir retardo debido al cifrado o descifrado.
- La seguridad del sistema deberá residir solamente en el secreto de una clave y no de las funciones de cifra.
- La fortaleza del sistema se entenderá como la imposibilidad computacional (tiempo de cálculo en años que excede cualquier valor razonable) de romper la cifra o encontrar la clave secreta a partir de otros datos de carácter público.

Recomendaciones de Bacon

- Filósofo y estadista inglés del siglo XVI
  - Dado un texto en claro M y un algoritmo de cifra E_k, el cálculo de $E_k(M)$ y su inversa debe ser sencillo.
  - Será imposible encontrar el texto en claro M a partir del criptograma C si se desconoce la función de descifrado D_k.
  - El criptograma deberá contener caracteres distribuidos para que su apariencia sea inocente y no dé pistas a un intruso.
  - Teniendo en cuenta los siglos trascurridos desde estas afirmaciones, éstas siguen siendo válidas hoy en día.

Recomendaciones de Kerckhoffs

- Profesor holandés en París del siglo XIX
  - K_1: El sistema debe ser en la práctica imposible de criptoanalizar.
  - K_2: Las limitaciones del sistema no deben plantear dificultades a sus usuarios.
  - K_3: Método de elección de claves fácil de recordar.
  - K_4: Transmisión del texto cifrado por telégrafo.
  - K_5: El criptógrafo debe ser portable.
  - K_6: No debe existir una larga lista de reglas de uso.

Al igual que en el caso anterior, siguen siendo "válidas".

Fortaleza: tipos de ataques

Conociendo el algoritmo de cifra, el criptoanalista intentará romper la cifra:

1. Contando únicamente con el criptograma.
2. Contando con texto en claro conocido.
3. Eligiendo un texto en claro.
4. A partir de texto cifrado elegido.

ATAQUE POR FUERZA BRUTA

5. Buscando combinaciones de claves.

Mayor trabajo

Clasificación de los criptosistemas

- Sistemas de cifra: clásicos v/s modernos
  - Clasificación histórica y cultural (no científica).
- Sistemas de cifra: en bloque v/s en flujo
  - Clasificación de acuerdo a cómo se produce la cifra.

- Sistemas de clave: secreta v/s pública
  - Clasificación de acuerdo a la cifra usando una única clave secreta o bien sistemas con dos claves, una de ellas pública y la otra privada.

Cifrado en bloque y en flujo

- CIFRADO EN BLOQUE:
  - El mismo algoritmo de cifra se aplica a un bloque de información (grupo de caracteres, número de bytes, etc.) repetidas veces, usando la misma clave. El bloque será normalmente de 64 ó 128 bits.

- CIFRADO EN FLUJO:
  - El algoritmo de cifra se aplica a un elemento de información (carácter, bit) mediante un flujo de clave en teoría aleatoria y mayor que el mensaje. La cifra se hace bit a bit.

Comparativa cifrado en bloque v/s flujo

CIFRADO EN BLOQUE

Ventajas:

* Alta difusión de los elementos en el criptograma.
* Inmune: imposible introducir bloques extraños sin detectarlo.

Desventajas:

* Baja velocidad de cifrado al tener que leer antes el bloque completo.
* Propenso a errores de cifra. Un error se propagará a todo el bloque.

CIFRADO EN FLUJO

Ventajas:

* Alta velocidad de cifra al no tener en cuenta otros elementos.
* Resistente a errores. La cifra es independiente en cada elemento.

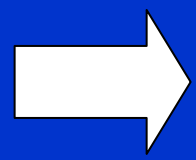
Desventajas:

* Baja difusión de elementos en el criptograma.
* Vulnerable. Pueden alterarse los elementos por separado.

Confidencialidad v/s integridad

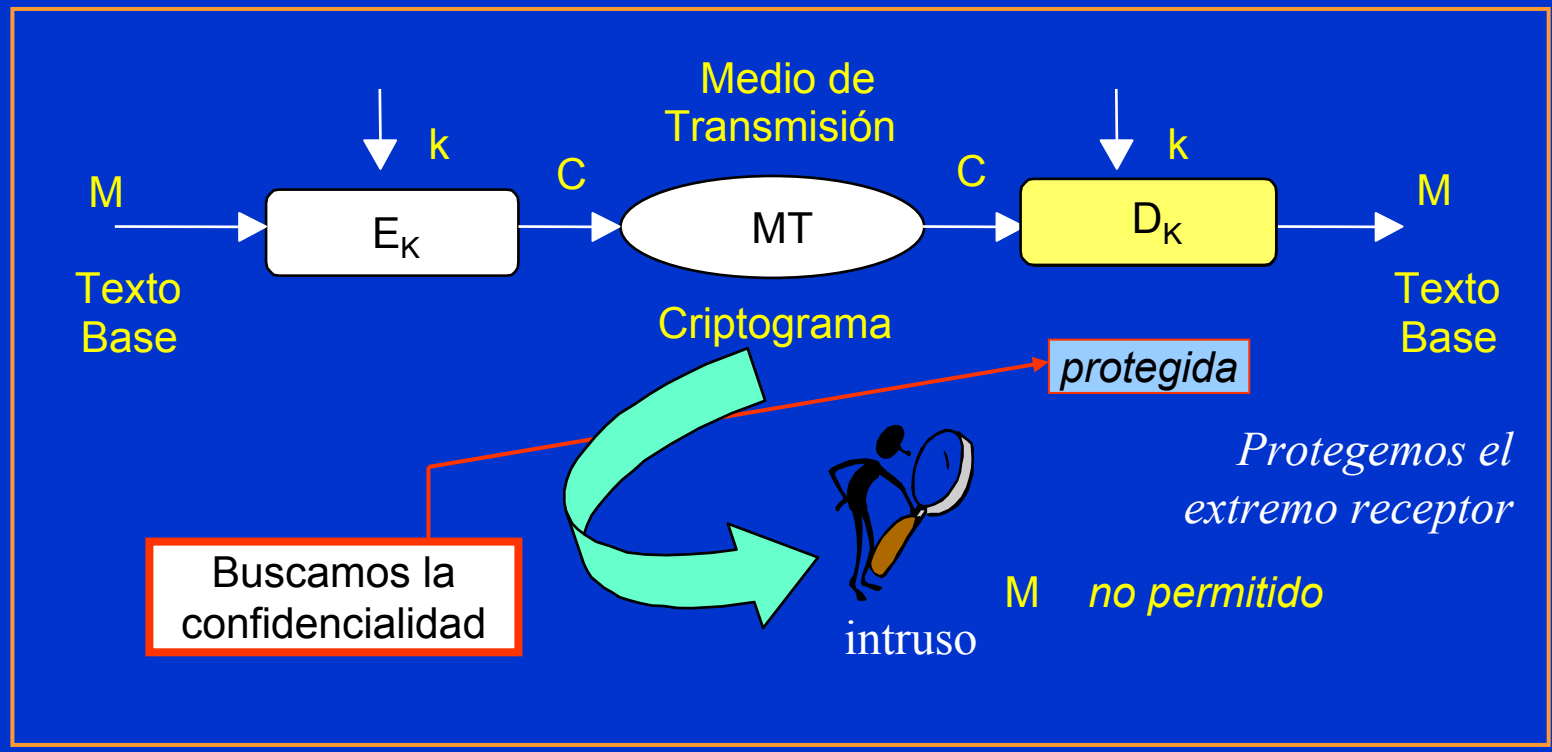
- Vamos a ver cómo se obtienen en cada uno de estos sistemas de cifra (cifrado con clave secreta y cifrado con clave pública) los dos aspectos más relevantes de la seguridad informática:

La confidencialidad
y la integridad



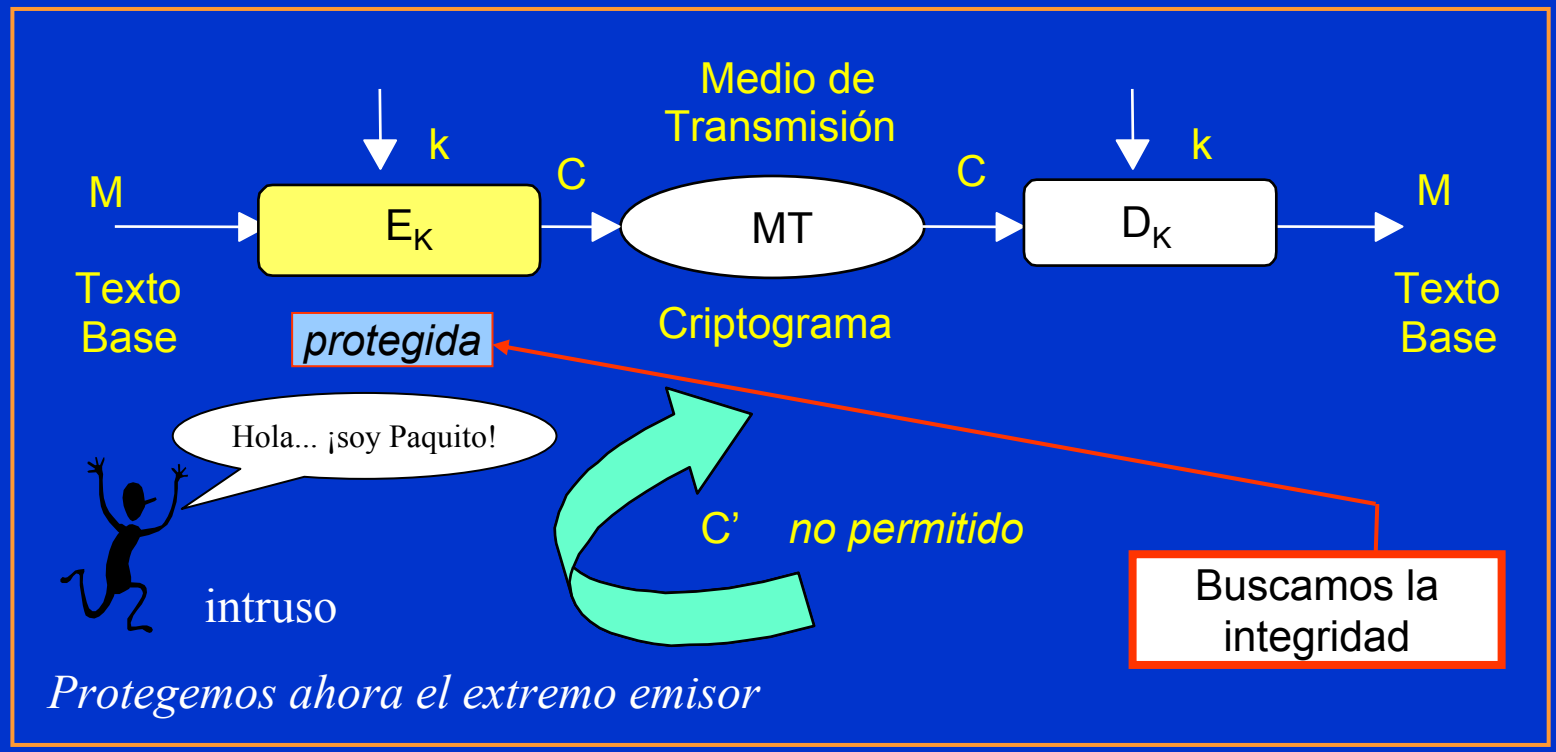
Llegaremos a un concepto de mucha utilidad en criptografía al analizar el sistema con clave pública...

Confidencialidad con clave secreta



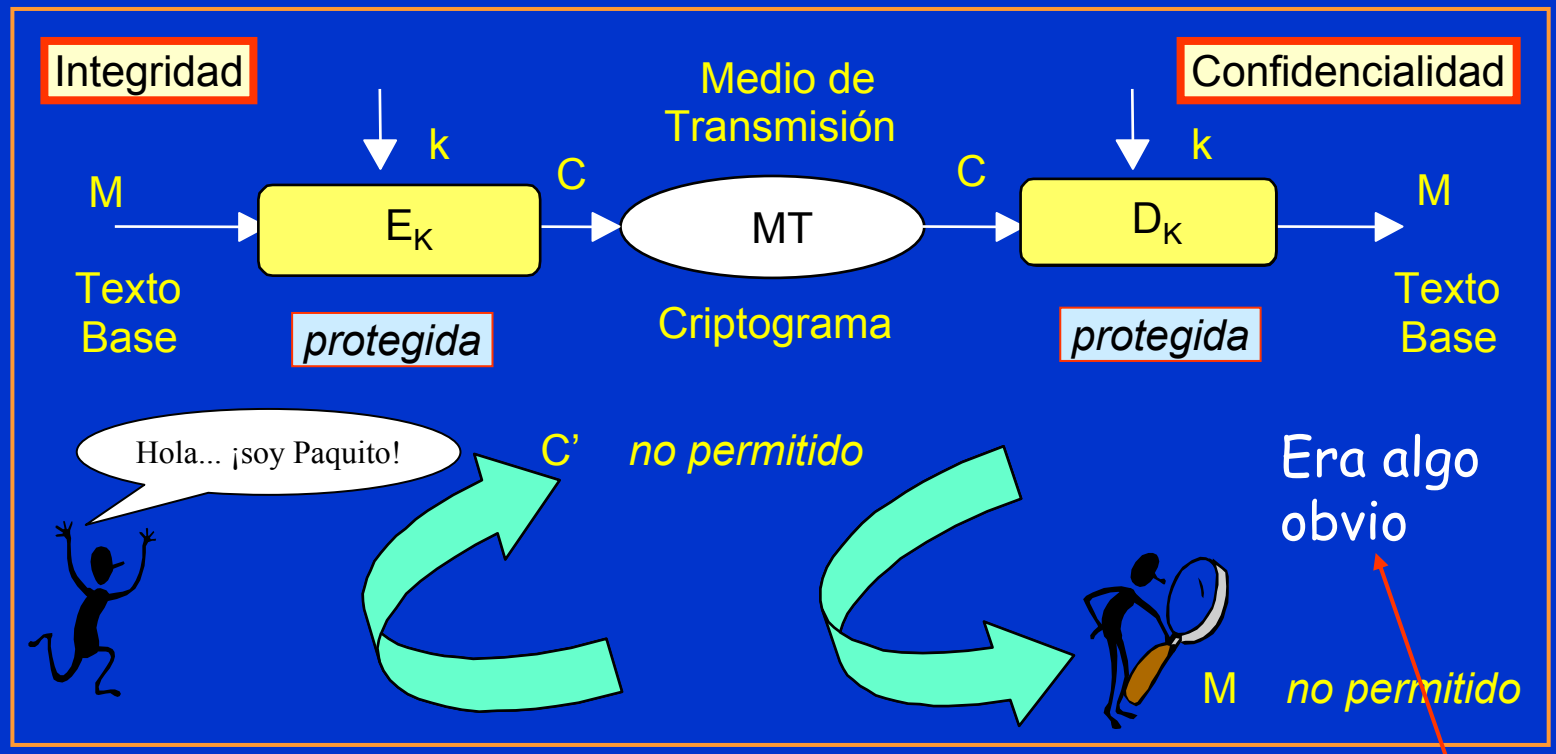
El criptoanalista no podrá descifrar el criptograma C o cualquier otro texto cifrado bajo la transformación E_K.

Integridad con clave secreta



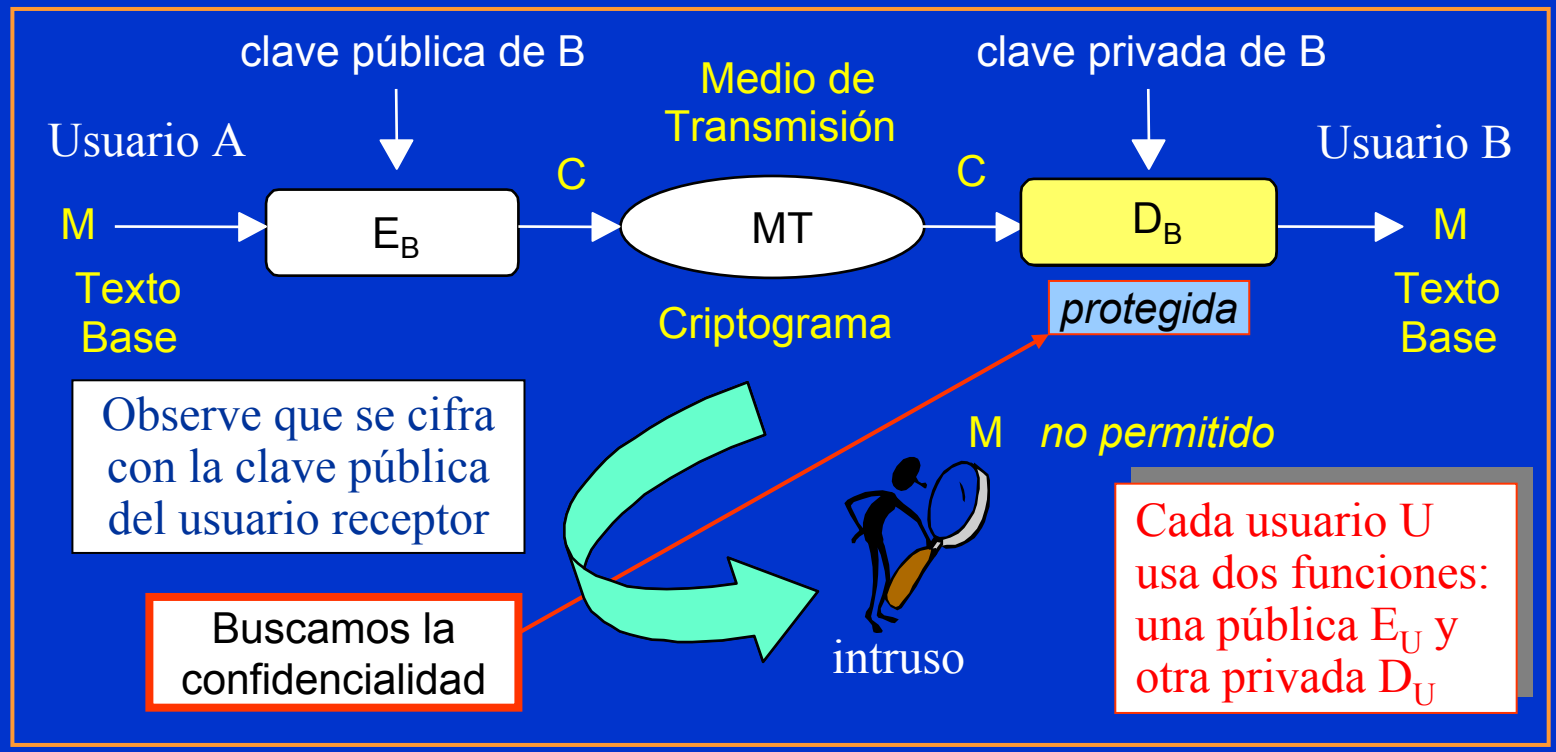
El criptoanalista no podrá cifrar un texto en claro M' y enviarlo al destinatario como $C' = E_K(M')$.

Resumen para sistemas de clave secreta



La confidencialidad y la integridad se lograrán simultáneamente si se protege la clave secreta.

Confidencialidad con clave pública

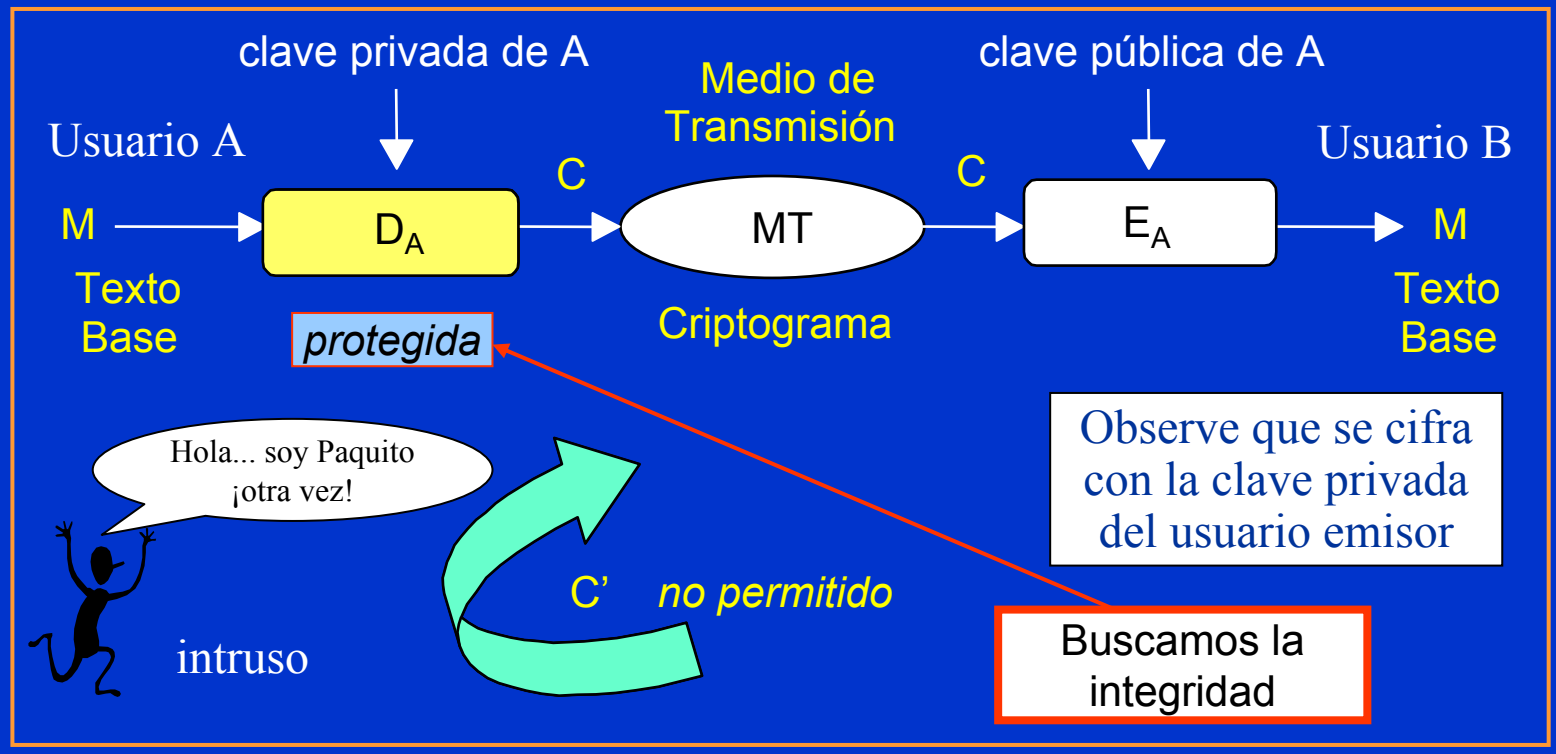


$$C = E_B(M)$$

$$M = D_B(C) = D_B(E_B(M))$$

D_B y E_B son operaciones inversas dentro de un cuerpo

Integridad con clave pública

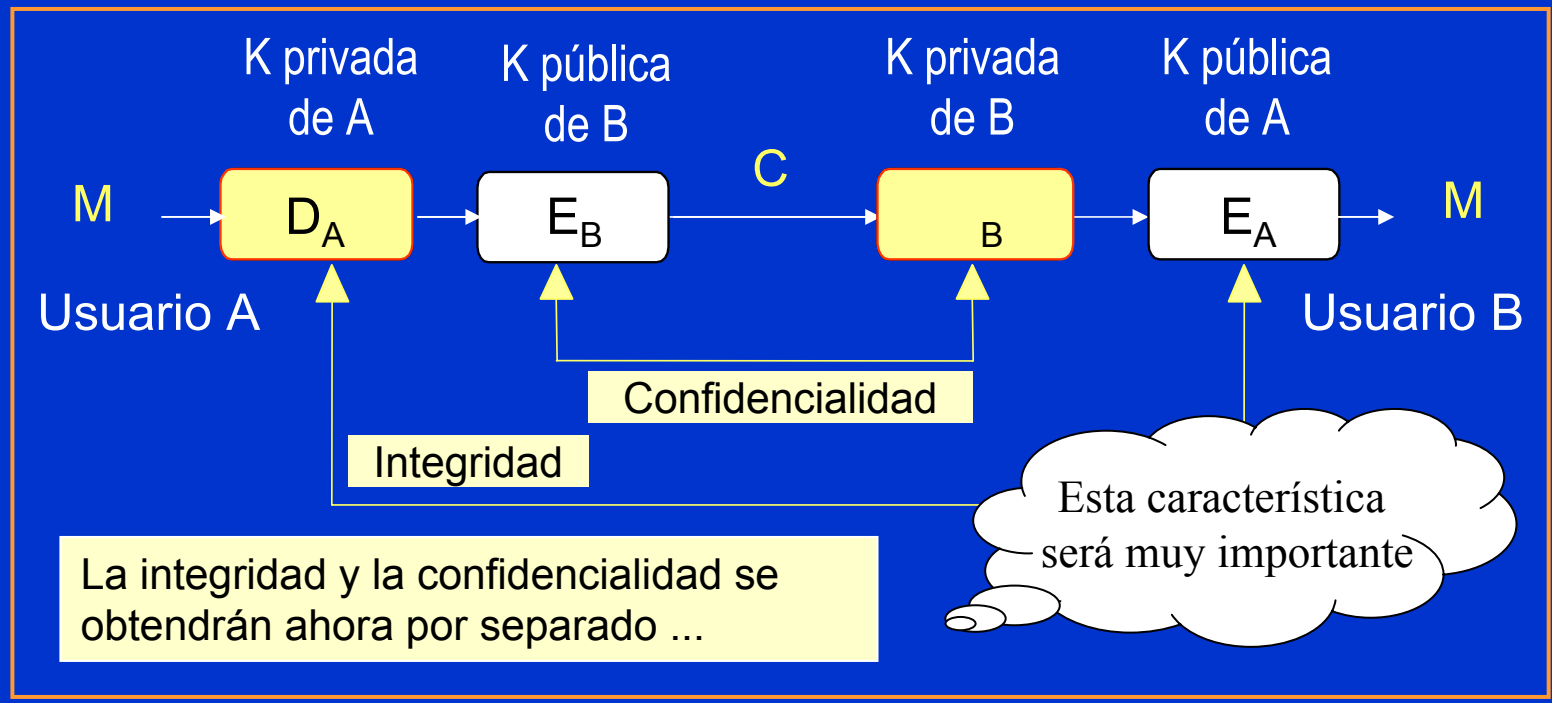


$$C = D_A(M)$$

$$M = E_A(C) = E_A(D_A(M))$$

D_A y E_A son operaciones inversas dentro de un cuerpo

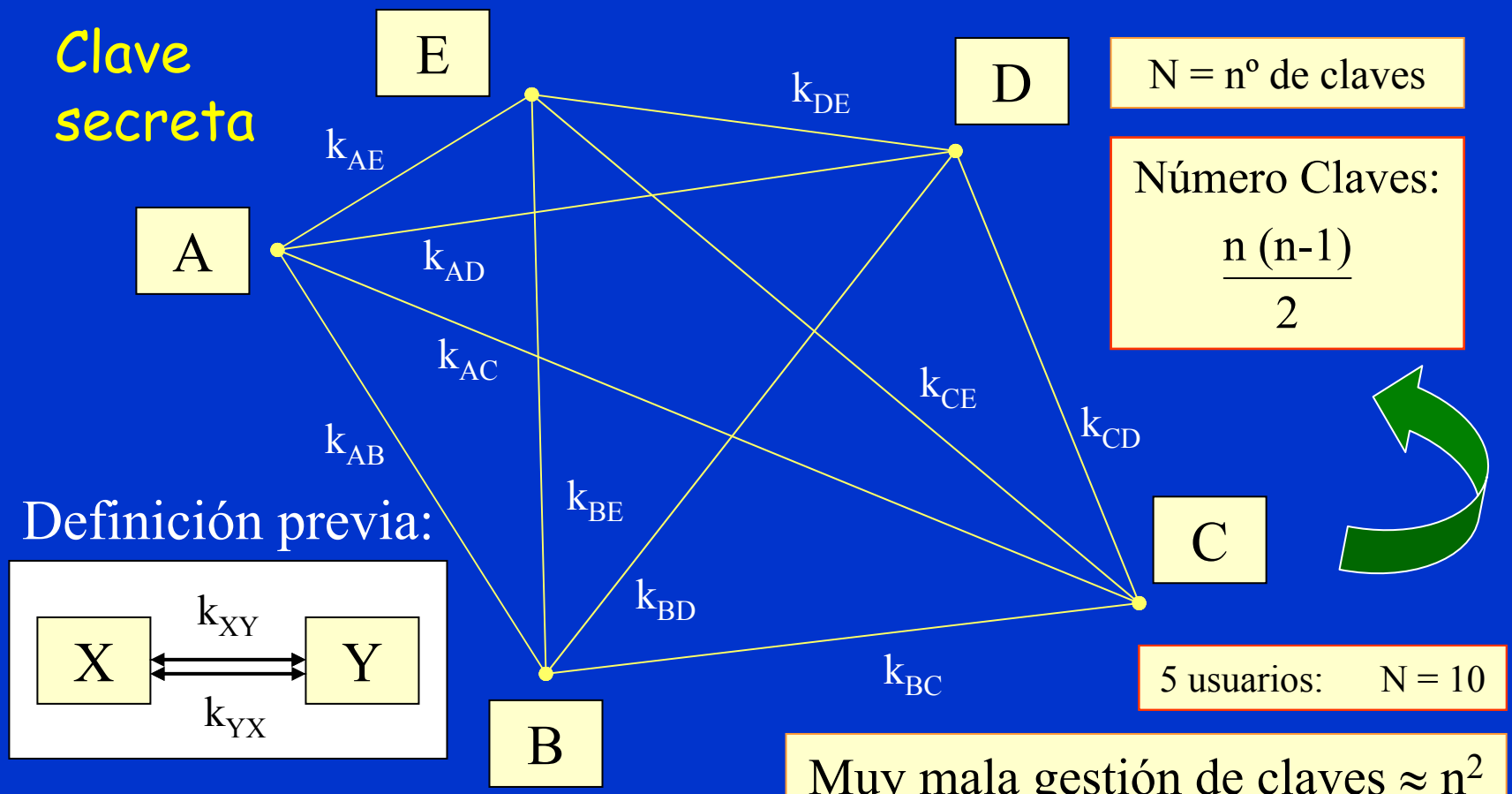
Resumen para sistemas con clave pública



$C = E_B(D_A(M))$ Cifrado del mensaje con firma digital

$M = E_A(D_B(C))$ Descifrado y comprobación de firma

La gestión de las claves secretas



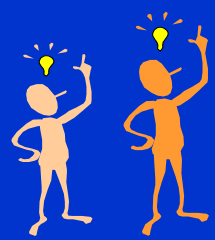
La solución híbrida

¿Es entonces la clave pública la solución a todos nuestros problemas?

¡ NO !

- Tendrá como inconveniente principal (debido a las funciones de cifra empleadas) una tasa o velocidad de cifra mucho más baja que la de los criptosistemas de clave secreta.

¿Solución?



Sistemas de cifra híbridos: los esquemas actuales de protocolos seguros en Internet funcionan así.

Fin del Tema 1

Cuestiones y ejercicios (1 de 2)

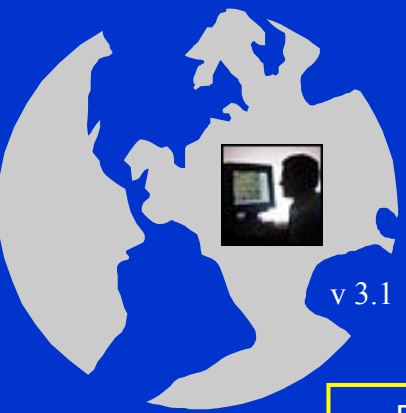
1. Un empleado poco satisfecho ha robado varios discos duros de muy alta calidad con datos de la empresa. ¿Qué importa más, el costo de esos discos o el valor de los datos? Justifique su respuesta.
2. En una empresa se comienza a planificar estrategias de acceso a las dependencias, políticas de backup, de protección de los equipos ante fuego, agua, etc. ¿Eso es seguridad física o lógica? ¿Por qué?
3. En nuestra empresa alguien usa software pirata. ¿Es una amenaza de interrupción, interceptación, modificación o de generación?
4. Una clave de sesión en Internet para proteger una operación de cifra dura 45 segundos. Si alguien intercepta el criptograma, ¿debemos preocuparnos si sabemos que la próxima vez la clave será otra?
5. Si se prueban todas las combinaciones posibles de una clave para romper un criptograma, ¿qué tipo de ataque estamos realizando?

Cuestiones y ejercicios (2 de 2)

6. Si protegemos una clave en el extremo emisor, ¿qué buscamos, la confidencialidad o la integridad? ¿Y si es en el extremo receptor?
7. ¿Por qué en un sistema simétrico se obtiene la confidencialidad y la integridad al mismo tiempo protegiendo la clave?
8. Explique qué significa que en un sistema de cifra asimétrica se obtengan la confidencialidad y la integridad por separado.
9. Si se cifra un mensaje con la clave privada del emisor, ¿qué se obtiene? ¿Y si el emisor cifra con la clave pública del receptor?
10. ¿Tiene sentido que el emisor cifre de forma asimétrica con su clave pública? ¿Qué logramos con ello? ¿Para qué serviría?
11. Queremos comunicarnos 10 usuarios con un sistema de cifra de clave secreta única entre cada dos miembros. ¿Cuántas claves serán necesarias? ¿Es eficiente el sistema? ¿Y si hay un usuario más?

Tema 2
Calidad de la Información y Virus

Seguridad Informática y Criptografía



Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 25 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Nota del autor

El contenido de este tema corresponde a los primeras apuntes del autor.
Tenga en cuenta entonces que se trata de un borrador, simplemente unas notas sueltas que tratan genéricamente este tema y la única intención de mantener este capítulo en el libro es porque algún día será ampliado, actualizado y tratado como se merece.

¿Qué es la información?

- El concepto en ingeniería:
  - Estudio de las estadísticas y características del lenguaje que nos permitirá su análisis desde un punto de vista matemático, científico y técnico.
- El concepto en la empresa:
  - Conjunto de datos propios que se gestionan y mensajes que se intercambian personas y/o máquinas dentro de una organización.

Teoría de la Información

- El estudio hecho por Claude Shannon en años posteriores a la 2ª Guerra Mundial ha permitido:
  - Cuantificar la cantidad de información.
  - Medir la entropía de la información.
  - Definir un sistema con secreto perfecto.
  - Calcular la redundancia y ratio del lenguaje.
  - Encontrar la distancia de unicidad.

  Aunque todo el estudio está orientado hacia los criptosistemas clásicos que cifran letras, se verá en detalle en un capítulo posterior pues permite definir y analizar sistemas con secreto perfecto.

La información en la empresa

- Se entenderá como:
  - Todo el conjunto de datos y ficheros de la empresa.
  - Todos los mensajes intercambiados.
  - Todo el historial de clientes y proveedores.
  - Todo el historial de productos, ... etc.
  - En definitiva, el *know-how* de la organización.
- Si esta información se pierde o deteriora, le será muy difícil a la empresa recuperarse y seguir siendo competitiva $\Rightarrow$ políticas de seguridad.

Importancia de la información

- El éxito de una empresa dependerá de la calidad de la información que genera y gestiona. Así, una empresa tendrá una información de calidad si ésta permite, entre otras características, la confidencialidad, la integridad y disponibilidad.

- La implantación de una política y medidas de seguridad informática en la empresa comienza a tenerse en cuenta sólo a finales de la década pasada. En este nuevo siglo es un factor estratégico en el desarrollo y vida de la misma. Tras los acontecimientos en las torres gemelas del año 2001, varias empresas han desaparecido por haber perdido toda su información. Es una señal de peligro y un aviso.

Vulnerabilidad de la información

- La información (datos) se verá afectada por muchos factores, incidiendo básicamente en los aspectos de confidencialidad, integridad y disponibilidad de la misma.

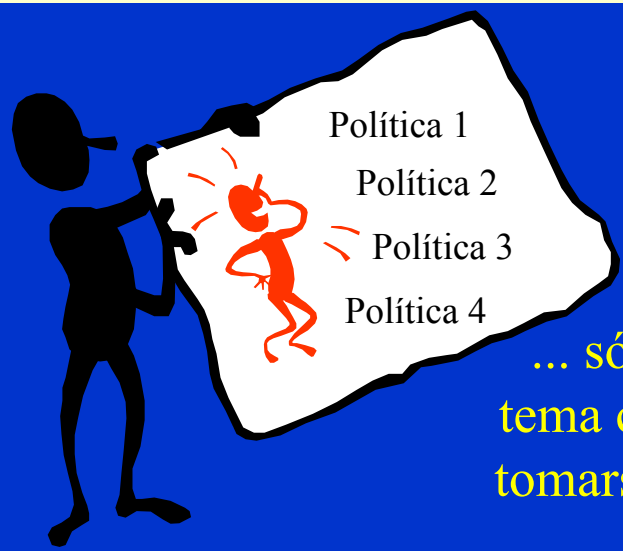
- Desde el punto de vista de la empresa, uno de los problema más importantes puede ser el que está relacionado con el delito o crimen informático, por factores externos e internos.

Implantar políticas de seguridad

Esto se verá agravado por otros temas, entre ellos los aspectos legales y las características de los nuevos entornos de trabajo de la empresa del siglo XXI.

Solución ?

La solución parece muy sencilla: aplicar técnicas y políticas de seguridad...



... sólo ahora el tema comienza a tomarse en serio.

Acciones contra los datos

- Una persona no autorizada podría:
  - Clasificar y desclasificar los datos.
  - Filtrar información.
  - Alterar la información.
  - Borrar la información.
  - Usurpar datos.
  - Hojear información clasificada.
  - Deducir datos confidenciales.

Por lo tanto, deberemos proteger nuestros datos

Protección de los datos

- La medida más eficiente para la protección de los datos es determinar una buena política de copias de seguridad o backups:
  - Copia de seguridad completa
    - Todos los datos (la primera vez).
  - Copias de seguridad incrementales
    - Sólo se copian los ficheros creados o modificados desde el último backup.
  - Elaboración de un plan de backup en función del volumen de información generada
    - Tipo de copias, ciclo de esta operación, etiquetado correcto.
    - Diarias, semanales, mensuales: creación de tablas.
  - Establecer quién cómo y dónde se guardan esos datos.

Hackers y crakers

- Hacker:
  - Definición inicial de los ingenieros del MIT que hacían alardes de sus conocimientos en informática.
  - Pirata Informático.
- Cracker:
  - Persona que intenta de forma ilegal romper la seguridad de un sistema por diversión o interés.

    No existe uniformidad de criterios en su clasificación; no obstante, su acción cada día se vuelve más técnica, sofisticada y debemos implementar medidas para proteger nuestra información ante tales ataques.

Puntos vulnerables en la red

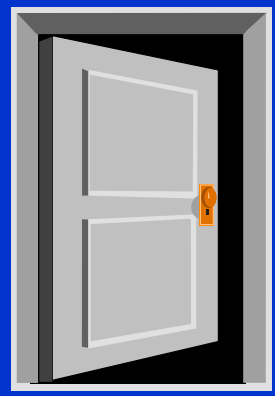
Las empresas relacionadas con las Nuevas Tecnologías de la Información NTIs hacen uso de varias técnicas y herramientas de redes para el intercambio de datos:

- Transferencia de ficheros (ftp)
- Transferencia de datos e información a través de Internet (http)
- Conexiones remotas a máquinas y servidores (telnet)

Todo esto presentará graves riesgos de ataques de hackers y otros delincuentes informáticos, pero ...

¿Dónde está el enemigo?

Por muy organizados que puedan estar estos grupos de vándalos o delincuentes, primero que nada hay que ponerse en el lugar que nos corresponde y no caer en la paranoia. Además, debemos pensar que el peor enemigo puede estar dentro de casa...



La solución sigue siendo la misma: la puesta en marcha de una adecuada política de seguridad en la empresa.

Delitos informáticos

Son acciones que vulneran la confidencialidad, integridad y disponibilidad de la información.

– Ataques a un sistema informático: Veamos algunos

- Fraude
- Malversación
- Robo
- Sabotaje
- Espionaje
- Chantaje
- Revelación
- Mascarada
- Virus
- Gusanos
- Caballos de Troya ... etc.

Fraude y sabotaje

Fraude

Acto deliberado de manipulación de datos perjudicando a una persona física o jurídica que sufre de esta forma una pérdida económica. El autor del delito logra de esta forma un beneficio normalmente económico.

Sabotaje

Acción con la que se desea perjudicar a una empresa entorpeciendo deliberadamente su marcha, averiando sus equipos, herramientas, programas, etc. El autor no logra normalmente con ello beneficios económicos pero pone en jaque mate a la organización.

Chantaje y mascarada

Chantaje

Acción que consiste en exigir una cantidad de dinero a cambio de no dar a conocer información privilegiada o confidencial y que puede afectar gravemente a la empresa, por lo general a su imagen corporativa.

Mascarada

Utilización de una clave por una persona no autorizada y que accede al sistema suplantando una identidad. De esta forma el intruso se hace dueño de la información, documentación y datos de otros usuarios con los que puede, por ejemplo, chantajear a la organización.

Virus y gusanos

Virus

Código diseñado para introducirse en un programa, modificar o destruir datos. Se copia automáticamente a otros programas para seguir su ciclo de vida. Es común que se expanda a través de plantillas, las macros de aplicaciones y archivos ejecutables.

Gusanos

Virus que se activa y transmite a través de la red. Tiene como finalidad su multiplicación hasta agotar el espacio en disco o RAM. Suele ser uno de los ataques más dañinos porque normalmente produce un colapso en la red (p.e. el gusano de Internet de Robert Morris Jr.).

Caballos de Troya

Caballos de Troya

Virus que entra al ordenador y posteriormente actúa de forma similar a este hecho de la mitología griega. Así, parece ser una cosa o programa inofensivo cuando en realidad está haciendo otra y expandiéndose. Ejemplo: el huevo de Pascua de Windows 95.

Y hay muchos más delitos. Incluso aparecerán nuevos delitos y ataques a los sistemas informáticos y redes que a fecha de hoy no sabemos cómo serán ni qué vulnerabilidad atacarán... Este enfrentamiento entre el "bien" y el "mal" es inevitable en un sistema abierto ... y las comunicaciones hoy son así.

Virus informáticos

Nota del autor

- Las próximas diapositivas son una breve y elemental introducción al tema de los virus informáticos, orientado además sólo al mundo de los PCs y del llamado entorno Windows. No pretende ser ni mucho menos un documento que trate este tema con la profundidad que debería hacerse y se merece.
- Mucha gente cataloga a éste como un tema menor, sin embargo dentro de las empresas es uno de los mayores problemas con los que se enfrentan los responsables de seguridad informática.
- Se incluye aquí este apartado como un factor más a tener en cuenta en cuanto a la calidad de la información que manejamos, dado que ésta puede verse afectada por este tipo de ataques tan comunes hoy en día por ejemplo vía e-mail.

Historia y tipos de virus

- Primer ejemplo: John von Neuman (1949)
- Primer virus: M. Gouglas de Bell Laboratories crea el Core War en 1960.
- Primeros ataques a PCs entre 1985 y 1987:
  - Virus Jerusalem y Brain.

- Inofensivos (pelota, letras, etc.)
  - Sólo molestan y entorpecen el trabajo pero no destruyen información. Podrían residir en el PC.
- Malignos (Viernes 13, Melissa, Nimbda, etc.)
  - Destruyen los datos y afectan a la integridad y la disponibilidad del sistema. Hay que eliminarnos.

Transmisión de un virus

- Se transmiten sólo mediante la ejecución de un programa. Esto es muy importante recordarlo.
- El correo electrónico por definición no puede contener virus al ser sólo texto. No obstante, muchas veces contienen archivos añadidos o los visualizadores ejecutan código en el cliente de correo del usuario y éstos pueden tener incluido un virus. Ahí está el peligro.
- El entorno web es mucho más peligroso. Un enlace puede lanzar un programa en Java u otro lenguaje que se ejecute y afecte el disco duro.

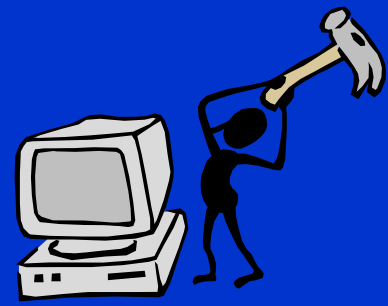
Tipos de ataque de un virus

- Están aquellos que infectan a programas con extensión exe, com y sys por ejemplo.
  - Residen en memoria al ejecutarse el huésped y de ahí se propagan a otros archivos.
- Y también aquellos que infectan el sistema y el sector de arranque y tablas de entrada (áreas determinadas del disco).
  - Se instalan directamente allí y por lo tanto residen en memoria.

Algunas medidas básicas de prevención

- Proteger los discos extraíbles con la pestaña, una protección de tipo hardware muy elemental.
- Instalar un antivirus y actualizarlo al menos una vez al mes; recomendable cada 15 días.
- Ejecutar el antivirus de vez en cuando al disco duro (por ejemplo una vez al mes) y siempre a los disquetes y archivos que se descargan de Internet.
- Usar siempre software legal, con licencia.
- Controlar el acceso de extraños al computador.

¿Qué hacer en caso de estar infectado?



- Detener las conexiones remotas.
- No mover el ratón ni activar el teclado.
- Apagar el sistema y desconectarlo.
- Arrancar con un disquete de arranque o emergencia protegido y ejecutar luego un programa antivirus.
- Hacer copia de seguridad de ficheros.
- Formatear el disco duro a bajo nivel si no queda otra solución ☹.
- Instalar nuevamente el sistema operativo y restaurar las copias de seguridad.

Fin del Tema 2

Cuestiones y ejercicios

1. ¿Qué diferencia hay entre el concepto de información y su calidad según lo entienda una empresa o los estudios de ingeniería?
2. ¿Por qué se dice que la información de una empresa es su activo más valioso? Compare este activo con el personal de la misma y póngase en situaciones en las que ambos se pierden, ¿qué situación podría ser es más perjudicial para la continuidad de dicha empresa?
3. Como supervisores de seguridad hemos detectado que alguien está realizando acciones no lícitas, por ejemplo copias no autorizadas de información. ¿Qué actitud debemos tomar?
4. ¿Qué medidas serían la más adecuadas de cara a minimizar el ataque por virus en nuestra empresa?
5. Si deseamos que nuestra empresa esté debidamente protegida tanto física como lógicamente, ¿qué debemos hacer?

Tema 3
Introducción a la Seguridad Física

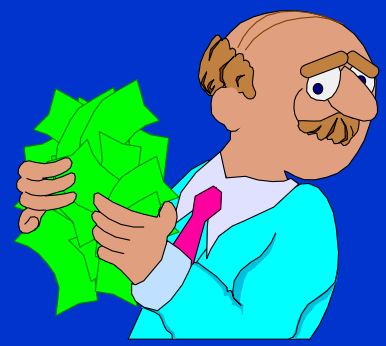
Seguridad Informática y Criptografía

Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 37 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Nota del autor



El contenido de este archivo corresponde sólo a un primer borrador en relación con el tema de la seguridad física. Actualmente ésta tiene una importancia igual y en algunos casos superior a la seguridad lógica. Los acontecimientos acaecidos a comienzos de este siglo han puesto al descubierto cuán vulnerable puede ser un sistema ante un ataque; por ello se han actualizado algunos temas pero este documento sigue siendo un borrador de trabajo.

Seguridad Física

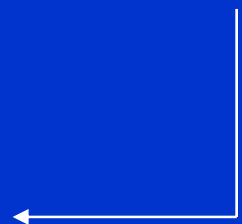
Los datos deben protegerse aplicando:

- Seguridad Lógica
  - Uso de herramientas de protección de la información en el mismo medio en el que se genera o transmite.
  - Protocolos de autenticación entre cliente y servidor.
  - Aplicación de normativas.
- Seguridad Física
  - Procedimientos de protección física del sistema: acceso personas, incendio, agua, terremotos, etc.
  - Medidas de prevención de riesgos tanto físicos como lógicos a través de una política de seguridad, planes de contingencia, aplicación de normativas, etc.

La seguridad Física en entornos de PCs

- Anclajes a mesas de trabajo.
- Cerraduras.
- Tarjetas con alarma.
- Etiquetas con adhesivos especiales.
- Bloqueo de disquetera.
- Protectores de teclado.
- Tarjeta de control de acceso al hardware.
- Suministro ininterrumpido de corriente.
- Toma de tierra.
- Eliminación de la estática... etc.

Temas a tener en cuenta en un entorno PC



Análisis de riesgo: plan estratégico

- Es el proceso de identificación y evaluación del riesgo a sufrir un ataque y perder datos, tiempo y horas de trabajo, comparándolo con el costo que significaría la prevención de este suceso.

- Su análisis no sólo nos lleva a establecer un nivel adecuado de seguridad, sino que permite conocer mejor el sistema que vamos a proteger.

Información del análisis de riesgo

- Información que se obtiene en un análisis de riesgo:
  - Determinación precisa de los recursos sensibles de la organización.
  - Identificación de las amenazas del sistema.
  - Identificación de las vulnerabilidades específicas del sistema.
  - Identificación de posibles pérdidas.
  - Identificación de la probabilidad de ocurrencia de una pérdida.
  - Derivación de contramedidas efectivas.
  - Identificación de herramientas de seguridad.
  - Implementación de un sistema de seguridad eficiente en costes y tiempo.

Ecuación básica del análisis de riesgo

¿ B > P * L ?

- B: Carga o gasto que significa la prevención de una pérdida específica por vulnerabilidad.
- P: Probabilidad de ocurrencia de esa pérdida específica.
- L: Impacto total de dicha pérdida específica.

¿Cuándo y cuánto invertir en seguridad?

Si $B \leq P * L$

Hay que implementar una medida de prevención.

Si $B > P * L$

No es necesaria una medida de prevención.

... al menos matemáticamente. No obstante, siempre puede ocurrir una desgracia que esté fuera de todo cálculo. Por ejemplo, el ataque a las torres gemelas y sus posteriores consecuencias informáticas no estaba contemplado en ningún plan contingencia...

Efectividad del coste de la medida

- Las medidas y herramientas de control han de tener menos coste que el valor de las posibles pérdidas y el impacto de éstas si se produce el riesgo temido.

- Ley básica: el costo del control ha de ser menor que el activo que protege. Algo totalmente lógico y que tanto los directivos como los responsables de seguridad de la empresa deberán estimar de forma adecuada a su realidad.

El factor L en la ecuación de riesgo

Factor L (en $B \leq P * L$)

- El factor de impacto total L es difícil de evaluar. Incluye daños a la información, a los equipos, pérdidas por reparación, por volver a levantar el sistema, pérdidas por horas de trabajo, etc.
- Siempre habrá una parte subjetiva.
- La pérdida de datos puede llevar a una pérdida de oportunidades por el llamado efecto cascada.
- En la organización debe existir una comisión especializada interna o externa que sea capaz de evaluar todas las posibles pérdidas y cuantificarlas.

El factor P en la ecuación de riesgo

Factor P (en $B \leq P * L$)

- El factor P está relacionado con la determinación del impacto total L y depende del entorno en el que esté la posible pérdida. Como este valor es difícil de cuantificar, dicha probabilidad puede asociarse a una tendencia o frecuencia conocida.
  - Conocido P para un L dado, se obtiene la probabilidad de pérdida relativa de la ocurrencia P*L que se comparará con B, el peso que supone implantar la medida de prevención respectiva.

El factor B en la ecuación de riesgo

Factor B (en $B \leq P * L$)

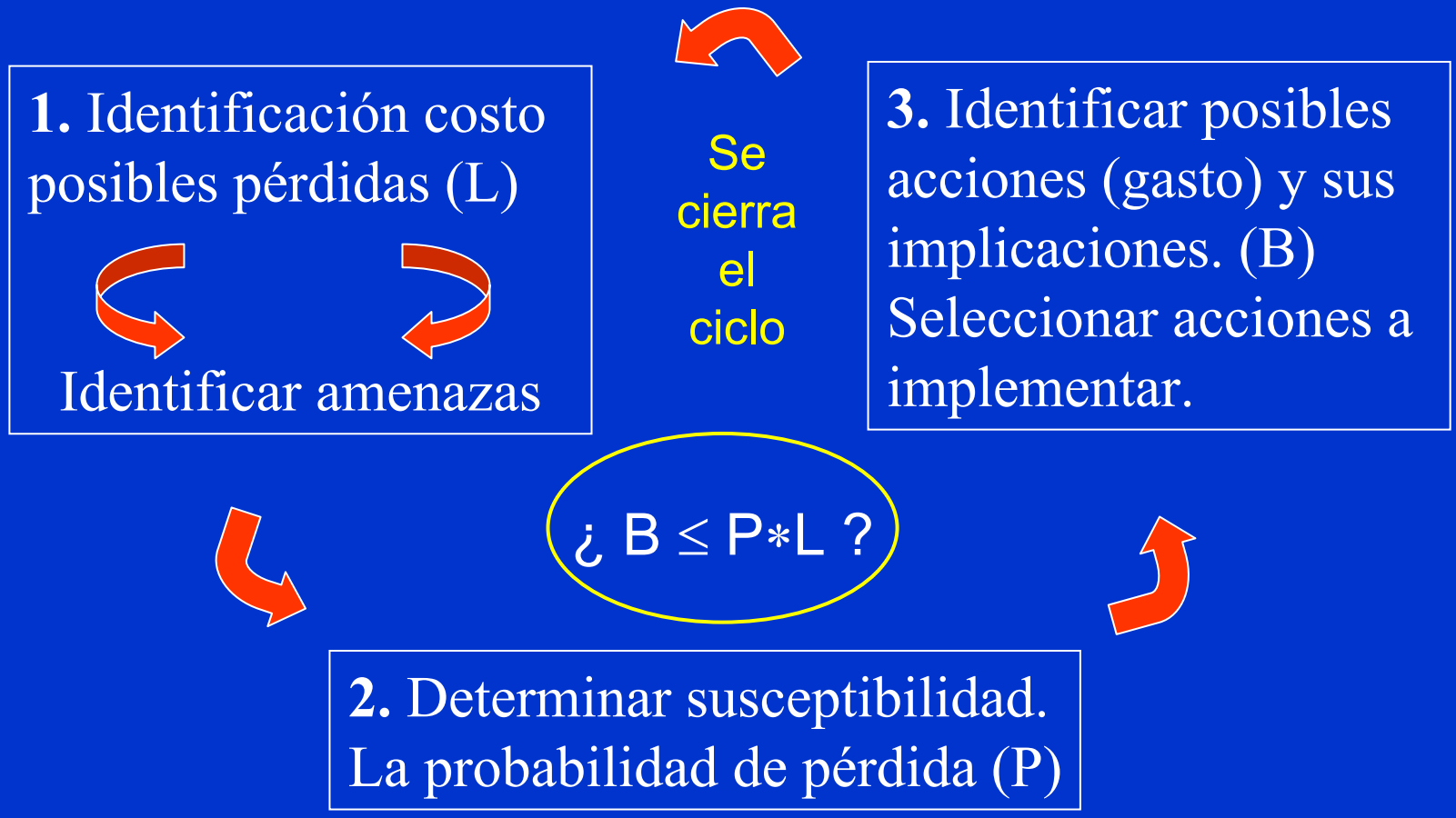
- Indica qué se requiere para prevenir una pérdida. Es la cantidad de dinero que vamos a disponer para mitigar la posible pérdida.
  - Ejemplo: la carga de prevención para que un sistema informático minimice el riesgo de que sus servidores sean atacados desde fuera incluye la instalación de software y hardware adecuado, un cortafuegos, un sistema de detección de intrusos, una configuración de red segura, una política de seguimiento de accesos y de passwords, personal técnico cualificado, etc. Todo ello importa una cantidad de dinero específica.

Cuantificación de la protección

$$¿\ B \leq P * L\ ?$$

- ¿Cuánta protección es necesaria?
  - En nuestro ejemplo: qué configuración de red usar, en qué entorno trabajar, qué tipo de cortafuegos, etc. Eso dependerá del novel de seguridad que nuestra empresa desee o crea oportuno.
- ¿De qué forma nos protegeremos?
  - Una casa puede protegerse con puertas, cerraduras, barras en ventanas, sistemas de alarmas, etc.
  - En un sistema informático podemos aplicar medidas físicas, políticas de seguridad de accesos, planes de contingencia y recuperación, cortafuegos, cifrado de la información, firmas, pasarelas seguras, etc.

Pasos en un análisis de riesgos



Algunas políticas de seguridad

- Políticas administrativas
  - Procedimientos administrativos.
- Políticas de control de acceso
  - Privilegios de acceso del usuario o programa.
- Políticas de flujo de información
  - Normas bajo la cuales se comunican los sujetos dentro del sistema.

Aspectos administrativos

- Políticas administrativas
  - Se establecen aquellos procedimientos de carácter administrativo en la organización como por ejemplo en el desarrollo de programas: modularidad en aplicaciones, revisión sistemática, etc.
  - Se establecen responsabilidades compartidas por todos los usuarios, cada uno en su nivel.

Control de accesos

- Políticas de control de acceso
  - Política de menor privilegio
    - Acceso estricto a objetos determinados, con mínimos privilegios para los usuarios.
  - Política de compartición
    - Acceso de máximo privilegio en el que cada usuario puede acceder a todos los objetos.
  - Granularidad
    - Número de objetos accesibles. Se habla entonces de granularidad gruesa y fina.

Control de flujo

- Políticas de control de flujo
  - La información a la que se accede, se envía y recibe por:
    - ¿Canales claros o canales ocultos? ¿Seguros o no?
  - ¿Qué es lo que hay que potenciar?
    - ¿La confidencialidad o la integridad?
    - ¿La disponibilidad? ... ¿El no repudio?
    - Según cada organización y su entorno de trabajo y servicios ofrecidos, habrá diferencias. En algunos sistemas primarán unos más que otros, en función de cuán secreta es la información que procesan.

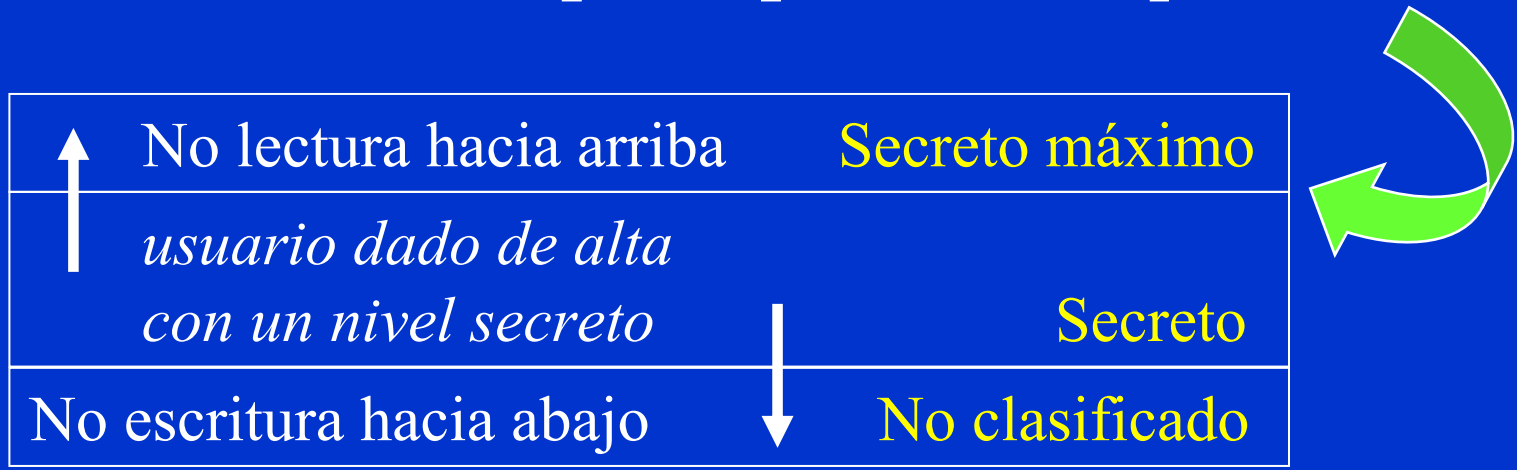
Modelos de seguridad

- Modelo de Bell LaPadula (BLP)
  - Rígido. Confidencialidad y con autoridad.
- Modelo de Take-Grant (TG)
  - Derechos especiales: tomar y otorgar.
- Modelo de Clark-Wilson (CW)
  - Orientación comercial: integridad.
- Modelo de Goguen-Meseguer (GM)
  - No interferencia entre usuarios.
- Modelo de Matriz de Accesos (MA)
  - Estados y transiciones entre estados
    - Tipo Graham-Dennig (GD)
    - Tipo Harrison-Ruzzo-Ullman (HRU)

Se definirán brevemente en próxima diapositiva

Modelo de Bell LaPadula BLP

- La escritura hacia abajo está prohibida.
- La lectura hacia arriba está prohibida.
- Es el llamado principio de tranquilidad.



Modelo de Take Grant TG

- Se describe mediante grafos orientados:
  - el vértice es un objeto o sujeto.
  - un arco es un derecho.
- Se ocupa sólo de aquellos derechos que pueden ser transferidos.

Modelo de Clark Wilson CW

- Basado en políticas de integridad
  - Elementos de datos restringidos.
    - sobre éstos debe hacerse un chequeo de consistencia.
  - Elementos de datos no restringidos.
  - Procedimientos de transformación.
    - trata los dos elementos.
  - Procedimientos de verificación de integridad.

Criterios y normativas de seguridad

- Criterio de evaluación TSEC
  - Trusted Computer System Evaluation Criteria, también conocido como Orange Book.
- Criterio de evaluación ITSEC
  - Information Technology Security Evaluation Criteria.
- Criterio de evaluación CC
  - Common Criteria: incluye los dos anteriores.
- Ley Orgánica de Protección de Datos LOPD (1999, España)
  - Establece un conjunto de medidas de seguridad de debido cumplimiento por parte de empresas y organismos.
- Normativa internacional 17799
  - Desarrolla un protocolo de condiciones mínimas de seguridad informática de amplio espectro.

La ley de seguridad informática en España

- Ley Orgánica de Protección de Datos LOPD se desarrolla en España en el año 1999 y comienza a aplicarse ya en el siglo XXI.
- Se crea una Agencia de Protección de datos APD que debe velar por el cumplimiento de esta ley mediante la realización de auditorías, al menos cada dos años. La APD la forman 9 personas.
- Se definen las funciones del Responsable de Fichero y del Encargado de Tratamiento.
- Las faltas se clasifican como leves, graves y muy graves con multas de 60.000, 300.000 y 600.000 €.
- En el Real Decreto 994/1999 sobre "Medidas de seguridad de los ficheros automatizados que contengan datos de carácter personal" se definen las funciones del Responsable de Seguridad.
- Establece un conjunto de procedimientos de obligado cumplimiento de forma que además de proteger la privacidad de las personas, se cumplan los principios de la seguridad informática física y lógica.

La normativa 17799

Código de buenas prácticas para la Gestión de la Seguridad de la Información: PNE-ISO/IEC 17799 (Proyecto de Norma Española)

- Antecedentes
- Introducción
- Objeto y campo de la aplicación
- Términos y definiciones
- Política de seguridad
- Aspectos organizativos para la seguridad
- Clasificación y control de los archivos
- Seguridad ligada al personal
- Seguridad física y del entorno
- Gestión de comunicaciones y operaciones
- Control de accesos
- Desarrollo y mantenimiento de sistemas
- Gestión de continuidad del negocio
- Conformidad

En 70 páginas y diez apartados, presenta unas normas, recomendaciones y criterios básicos para establecer unas políticas de seguridad. Éstas van desde los conceptos de seguridad física hasta los de seguridad lógica. Parte de la norma elaborada por la British Standards Institution BSI, adoptada por International Standards Organization ISO y la International Electronic Commission IEC.

Planes de contingencia

- Un Plan de Contingencia consiste en un análisis pormenorizado de las áreas que componen nuestra organización que nos servirá para establecer una política de recuperación ante un desastre.
  - Es un conjunto de datos estratégicos de la empresa y que se plasma en un documento con el fin de protegerse ante eventualidades.
- Además de aumentar su seguridad, con un plan estratégico la empresa también gana en el conocimiento de fortalezas y debilidades.
- Pero si no lo hace, se expone a sufrir una pérdida irreparable mucho más costosa que la implantación de este plan.

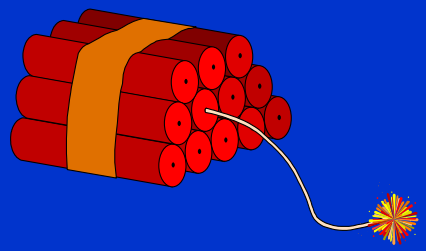
Desastres naturales y su prevención

- Desastres naturales
  - Huracán
  - Tormenta
  - Inundación
  - Tornado
  - Vendaval
  - Incendio
  - Otros

- Medidas prevención
  - Emplazamientos adecuados
  - Protección fachadas, ventanas, puertas

Vandalismo informático y su prevención

- Terrorismo
- Sabotaje
- Robo



- Virus
- Programas malignos

- Medidas de prevención
  - Fortificación entradas
  - Guardia Jurado
  - Patrullas
  - Circuito cerrado TV
  - Control de accesos
  - Protección de software y hardware con antivirus, cortafuegos, etc.

Amenazas del agua y su prevención

- Amenazas
  - Inundaciones por causas propias de la empresa
  - Inundaciones por causas ajenas
  - Pequeños incidentes personales (botella de agua, taza con café)
- Medidas prevención
  - Revisar conductos de agua.
  - Localizar la sala con los equipos más caros en un sitio libre de estos problemas.
  - Instalar sistemas de drenaje de emergencia.
  - Concienciar empleados.

Amenazas del fuego y su prevención

- Amenazas
  - Una mala instalación eléctrica.
  - descuidos personales como fumar en la sala de ordenadores.
  - Papeleras mal ubicadas en la que se tira un cigarrillo no apagado.
  - Vulnerabilidades del sistema por humo.
- Medidas prevención
  - Detector humo y calor.
  - Materiales ignífugos.
  - Almacén de papel separado de máquinas.
  - Estado del falso suelo.
  - Extintores revisados.
    - Es la amenaza más temida por su rápido poder destructor.

¿Qué sucede si se produce una catástrofe?

- Las empresas dependen hoy en día de los equipos informáticos y de todos los datos que hay allí almacenados (nóminas, clientes, facturas, ...).
- Dependen también cada vez más de las comunicaciones a través de redes de datos.
- Si falla el sistema informático y éste no puede recuperarse, la empresa puede desaparecer porque no tiene tiempo de salir nuevamente al mercado con ciertas expectativas de éxito, aunque conserve a todo su personal.

Tiempos de recuperación ante desastres

- Período máximo de paro de una empresa sin poner en peligro su supervivencia:
  - Sector Seguros: 5,6 días
  - Sector Fabricación: 4,9 días
  - Sector Industrial: 4,8 días
  - Sector Distribución: 3,3 días
  - Sector Financiero: 2,0 días

Ref. Estudio de la Universidad de Minnesota (1996)

Pérdidas por no contar con un plan

- Pérdida de clientes.
- Pérdida de imagen.
- Pérdida de ingresos por beneficios.
- Pérdida de ingresos por ventas y cobros.
- Pérdida de ingresos por producción.
- Pérdida de competitividad.
- Pérdida de credibilidad en el sector.

Implantación de medidas básicas

- Plan de emergencia
  - Vidas, heridos, activos, evacuación personal.
  - Inventariar recursos siniestrados.
  - Evaluar el coste de la inactividad.

- Plan de recuperación
  - Acciones tendentes a volver a la situación que existía antes del desastre.

Plan de continuidad

- Instalaciones alternativas
  - Oficina de servicios propia.
  - Acuerdo con empresa vendedora de HW y SW.
  - Acuerdo recíproco entre dos o más empresas.
  - Arranque en frío; sala vacía propia.
  - Arranque en caliente: centro equipado.
  - Sistema *Up Start*: caravana, unidad móvil.
  - Sistema *Hot Start*: centro gemelo.

Fin del Tema 3

Cuestiones y ejercicios (1 de 2)

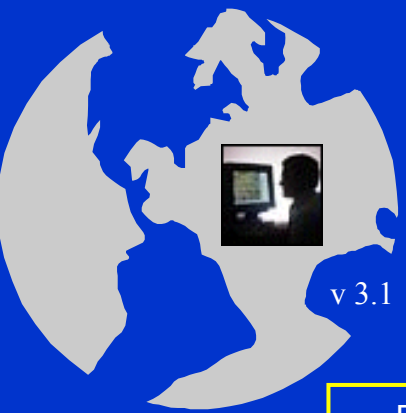
1. ¿Qué es y qué significa hacer un análisis de riesgos?
2. Explique el sentido de las ecuaciones $B > P*L$ y $B \leq P*L$.
3. Tras un estudio, obtenemos $B > P*L$, ¿podemos estar totalmente tranquilos al no utilizar medida alguna de prevención?
4. Explique qué significan los factores L y P en la ecuación $B > P*L$.
5. ¿Cuáles son los pasos a seguir en un análisis de riesgos de acuerdo a los factores de la ecuación de $B > P*L$?
6. En algunos sistemas de gestión de información a veces prima más el elemento confidencialidad, en cambio en otros más el de integridad. Dé algunos ejemplos en que pueda cumplirse al menos en parte este escenario. ¿Qué opina respecto a una transacción electrónica?
7. Comente el modelo de seguridad de Bell Lapadula. ¿Por qué se le llama el modelo de la tranquilidad?

Cuestiones y ejercicios (2 de 2)

8. Ud. es el responsable de seguridad y detecta que un empleado está robando información confidencial, ¿cómo reaccionaría?
9. ¿Cuáles pueden ser las pérdidas en una empresa si no se cuenta con un adecuado Plan de Contingencia y sucede un desastre?
10. ¿Qué es un Plan de Contingencia y por qué es importante?
11. Nuestra empresa está a medias entre el rubro distribución y el de las finanzas. ¿Resulta estratégico tener aquí un Plan de Contingencia?
12. ¿Qué soluciones tenemos para que un banco no se vea afectado por un desastre y pueda seguir trabajando con sus clientes con un tiempo de recuperación bajo o mínimo? ¿Cómo sería su coste?
13. ¿Se pueden prever situaciones extremas como lo acontecido con las torres gemelas? ¿En que tipo de empresas o instituciones no deben descartarse estos extremos? ¿En mi empresa que vende pasteles?

Tema 4
Teoría de la Información

Seguridad Informática y Criptografía



Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 57 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Fundamentos de la Seguridad Informática

Los pilares sobre los que descansa toda la teoría asociada a los criptosistemas son tres:

Lo veremos en ...

- La teoría de la información — este archivo
  - Estudio de la cantidad de información y entropía.
- La teoría de los números — archivo SItema05
  - Estudio de las matemáticas discretas y cuerpos finitos.
- La teoría de la complejidad de los algoritmos
  - Clasificación de los problemas. — archivo SItema06

Teoría de la información

- Información:
  - Conjunto de datos o mensajes inteligibles creados con un lenguaje de representación y que debemos proteger ante las amenazas del entorno, durante su transmisión o almacenamiento, usando entre otras cosas, las técnicas criptográficas.
  - La teoría de la información mide la *cantidad de información* que contiene un mensaje a través del número medio de bits necesario para codificar todos los posibles mensajes con un *codificador óptimo*.

Veremos estas dos definiciones más adelante.

Representación de la información

Puede ser numérica, alfabética, simbólica, por lenguaje.

Ejemplo: 24/01/03 24-01-03 24-1-03 24/01/2003
01/24/03 01-24-03 1-24-03 01-24-2003 ...

- Todos son el día 24 de enero del año 2003.

Vitaminas: B_{12}, C, ...

Grupo sanguíneo: A2 Rh+ ...

Elementos: Fe, Si, Hg ...

Compuestos químicos: H_2O, CO_2 ...

Más común ➡ Lenguaje con código: *"Hoy hace calor"*

¿Qué información nos entrega el mensaje "*Hoy hace calor*"?

Pero ¿qué es la información?

Veremos qué información nos entrega un mensaje dependiendo del contexto en que nos encontremos. Esto puede ser:

a) En función de la extensión del mensaje recibido.

b) En función de la utilidad del mensaje recibido.

c) En función de la sorpresa del mensaje recibido.

d) Dependiendo del entorno de esa sorpresa.

e) En función de la probabilidad de recibir un mensaje.

Este es el entorno del estudio realizado por Claude Shannon orientado a la ingeniería y que aquí nos interesa.

Cantidad de información (caso 1)

En función de la extensión del mensaje

- Ante una pregunta cualquiera, una respuesta concreta y extensa nos entregará mayor información sobre el tema en particular, y diremos que estamos ante una mayor "cantidad de información".

- Pregunta: ¿Hace calor allí? (*una playa en particular*)
  - Respuesta 1: Sí, hace mucho calor.
  - Respuesta 2: Cuando no sopla el viento, el calor allí es inaguantable pues supera los 42 grados a la sombra. ☹

¿Dónde hay una mayor cantidad de información?

Cantidad de información (caso 2)

En función de la utilidad del mensaje

- Ante una pregunta cualquiera, una respuesta más útil y clara nos dejará con la sensación de haber recibido una mayor "cantidad de información".

- Pregunta: ¿Hace calor allí? (*una playa en particular*)
  - Respuesta 1: Sí, sobre 30 grados. 
  - Respuesta 2: Si no hay viento del sur y el mar está en calma, es normal que la temperatura suba bastante.

¿Dónde hay una mayor cantidad de información?

Cantidad de información (caso 3)

En función de la sorpresa del mensaje

- Ante una pregunta cualquiera, una respuesta más inesperada y sorprendente, nos dará la sensación de contener una mayor "cantidad de información".

- Pregunta: ¿Hace calor allí? (*ahora Finlandia en otoño*)
  - Respuesta 1: Sí, muchísimo. Es insoportable. ☺
  - Respuesta 2: En esta época del año, la temperatura es más suave y el tiempo muy agradable.

¿Dónde hay una mayor cantidad de información?

Cantidad de información (caso 4)

Dependencia del entorno (sorpresa)

- Ante una pregunta cualquiera, una respuesta inesperada y sorprendente en el entorno, nos dará la sensación de contener una mayor "cantidad de información".

- Pregunta: ¿Hace calor allí?

(*ahora las mismas respuestas hablan de la temperatura en un horno*)

- Respuesta 1: Sí, muchísimo. Es insoportable.
- Respuesta 2: En esta época del año, la temperatura es más suave y el tiempo muy agradable. 😐?

¿Dónde hay una mayor cantidad de información?

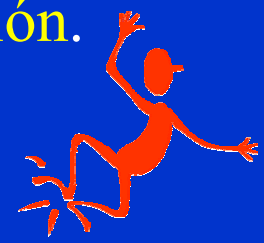
Cantidad de información (caso 5)

En función de la probabilidad de recibir un mensaje

- Este enfoque probabilístico es el que nos interesará en cuanto a la definición de Cantidad de Información.

¿Dónde le da alegría a su cuerpo Macarena?

- Respuesta 1: En un país de Europa.
- Respuesta 2: En una ciudad de España.



¿Por qué?

Respuesta 3: En el número 7 de la calle de la Sierpes en Sevilla, España.

¿Dónde hay una mayor cantidad de información?

Incertidumbre e información

Ante varios mensajes posibles, en principio todos equiprobables, aquel que tenga una menor probabilidad será el que contenga una mayor cantidad de información.

- En el ejemplo anterior:
  - Al ser más extenso el número de calles y sus números en una ciudad que el número de ciudades en España y esto último mayor que los países en Europa, la última respuesta tendrá una mayor incertidumbre. Si suponemos todos los estados equiprobables, entonces la cantidad de información de la respuesta tercera será mayor que las demás.

Concepto de variable aleatoria

- Sea X una variable aleatoria con n estados posibles con $X = x_i$ una ocurrencia iésima:

$$X = \{x_1, x_2, x_3, ..., x_{n-1}, x_n\}$$

$$p_1 = p(x_1),\ p_2 = p(x_2),\ ...,\ p_n = p(x_n)$$

Como:

$$0 \leq p_i \leq 1 \quad \text{para } i = 1, 2, ..., n$$

Entonces:

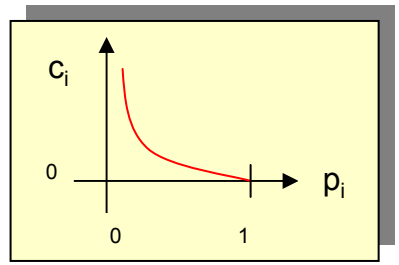
$$\sum_{i=1}^{n} p_i = 1$$

La probabilidad de que ocurra p_1 o p_2 o p_3, etc. será siempre la unidad porque seguro será uno de ellos.

Definición logarítmica

- Definiremos c_i a la cantidad de información del estado i, como el logaritmo en base dos de la probabilidad de que ocurra el estado iésimo.

$$c_i = - \log_2 (p_i)$$



- Logaritmo: $p(x_i) = 1 \Rightarrow$ no hay incertidumbre: $c_i = 0$

  $p(x_i) = 0 \Rightarrow$ máxima incertidumbre: $c_i \rightarrow \infty$

- Signo: $p(x_i) < 1 \Rightarrow \log p(x_i)$ será negativo

- Base: Fenómeno binario $\Rightarrow$ dos estados (bit)

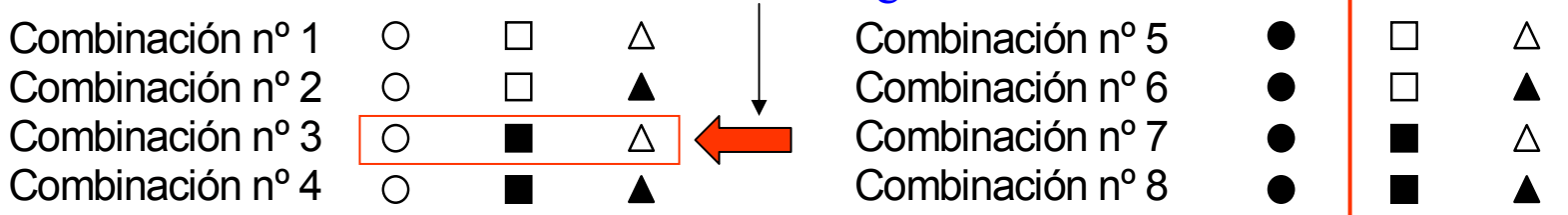
Grado de indeterminación

$$c_i = \frac{\text{Grado de indeterminación previo}}{\text{Grado de indeterminación posterior}}$$

Ejemplo del mago: En una bolsa hay un círculo, un cuadrado y un triángulo: negros o blancos.

Si hay equiprobabilidad entonces $p(x_i) = 1/8$

Esta será la combinación elegida



¿Qué cantidad de información tiene cada uno de los estados?

Solución al ejemplo del mago

Combinación nº 1	○	□	△	Combinación nº 5	●	□	△
Combinación nº 2	○	□	▲	Combinación nº 6	●	□	▲
Combinación nº 3	○	■	△	Combinación nº 7	●	■	△
Combinación nº 4	○	■	▲	Combinación nº 8	●	■	▲

Los 8 estados serán equiprobables: $p(x_i) = 1/8$

Incertidumbre inicial $I_i = 8$

Daremos algunas pistas ✌:

Veamos esto ahora matemáticamente ... →

- Las figuras no son del mismo color: I_i baja de 8 a 6 al descartarse las combinaciones 1 y 8.
- El círculo es blanco: I_i baja de 6 a 3 (descarte 5, 6 y 7).
- Hay dos figuras blancas: I_i baja de 3 a 2 (descarte 4).
- El cuadrado es negro: I_i baja de 2 a 1 (descarte 2.)

Se acaba la incertidumbre pues la solución es la combinación 3.

Solución matemática al ejemplo del mago

- Las figuras no son del mismo color. I_i baja de 8 a 6:

  $c_{i1} = \log (8/6) = \log 8 - \log 6$

- El círculo es blanco. I_i baja de 6 a 3:

  $c_{i2} = \log (6/3) = \log 6 - \log 3$

- Hay dos figuras blancas. I_i baja de 3 a 2:

  $c_{i3} = \log (3/2) = \log 3 - \log 2$

- El cuadrado es negro. I_i baja de 2 a 1:

  $c_{i4} = \log (2/1) = \log 2 - \log 1$

Todas las magnitudes se pueden sumar como escalares:

$$c_i = c_{i1} + c_{i2} + c_{i3} + c_{i4} = \log 8 - \log 1 = \log 8$$

Base del logaritmo

Sean: I_i la indeterminación inicial

I_f la indeterminación final

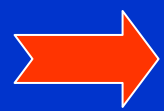
$$c_i = \log (I_i / I_f) = \log I_i - \log I_f$$

La cantidad de información tiene como unidad de medida la de un fenómeno de sólo dos estados, un fenómeno binario. Luego:

$$c_i = \log_b (2/1) = \log_b 2 - \log_b 1$$

– Si $\log_b 2$ debe ser igual a 1 entonces la base b = 2.

– Precisamente a esta unidad se le llama bit (binary digit)

– Ejemplo anterior: $c_i = \log_2 8 = 3$  ¡Sólo 3 preguntas!

Con sólo tres preguntas...

Con sólo tres preguntas "*más o menos inteligentes*" podemos pasar de la incertidumbre total a la certeza:

Pregunta 1: ¿Está entre la opción 1 y la 4? ⇒ Sí

Pregunta 2: ¿Está entre la opción 1 y la 2? ⇒ No

Pregunta 3: ¿Es la opción 4? ⇒ No Se acaba la indeterminación

Combinación nº 1	○	□	△
Combinación nº 2	○	□	▲
Combinación nº 3	○	■	△
Combinación nº 4	○	■	▲

Combinación nº 5	●	□	△
Combinación nº 6	●	□	▲
Combinación nº 7	●	■	△
Combinación nº 8	●	■	▲

Entropía de los mensajes

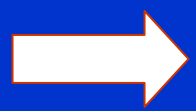
- Si un fenómeno tiene un grado de indeterminación k y sus estados son equiprobables, la probabilidad p de que se dé uno de esos estados será 1/k. Luego:

$$c_i = \log_2 (k/1) = \log_2 [1/(1/k)] = - \log_2 p$$

- Si ahora cada uno de estos estados tiene una probabilidad distinta p_i, la entropía H será igual a la suma ponderada de la cantidad de información:

$$H = - p_1 \log_2 p_1 - p_2 \log_2 p_2 - \ldots - p_k \log_2 p_k$$

$$H = - \sum_{i=1}^{k} p_i \log_2 p_i$$



Ecuación no inmediata

Definición de entropía

- La entropía de un mensaje X, que se representa por H(X), es el valor medio ponderado de la cantidad de información de los diversos estados del mensaje.

$$H(X) = -\sum_{i=1}^{k} p(x_i) \log_2 p(x_i)$$

Esto lo veremos más adelante

- Es una medida de la incertidumbre media acerca de una variable aleatoria y el *número de bits de información*.

El concepto de incertidumbre en H puede aceptarse. Lo que llama la atención ✋ es lo del número de bits de información.

Propiedades de la entropía

a) La entropía es no negativa y se anula si y sólo si un estado de la variable es igual a 1 y el resto 0 (demostración sencilla).

b) La entropía es máxima, mayor incertidumbre del mensaje, cuando todos los valores posibles de la variable X son equiprobables (empíricamente fácil; demostración no directa).

Si hay n estados equiprobables, entonces $p_i = 1/n$.

Luego:

$$H(X) = - \sum_i p_i \log_2 p_i = - n(1/n) \log_2 (1/n) = - (\log_2 1 - \log_2 n)$$

$$H(X)_{máx} = \log_2 n$$

Codificador óptimo

Nos falta encontrar el segundo término pendiente en la definición de cantidad de información: codificador óptimo.

Introduciendo el signo negativo dentro del logaritmo en la expresión de la entropía, ésta nos quedará como:

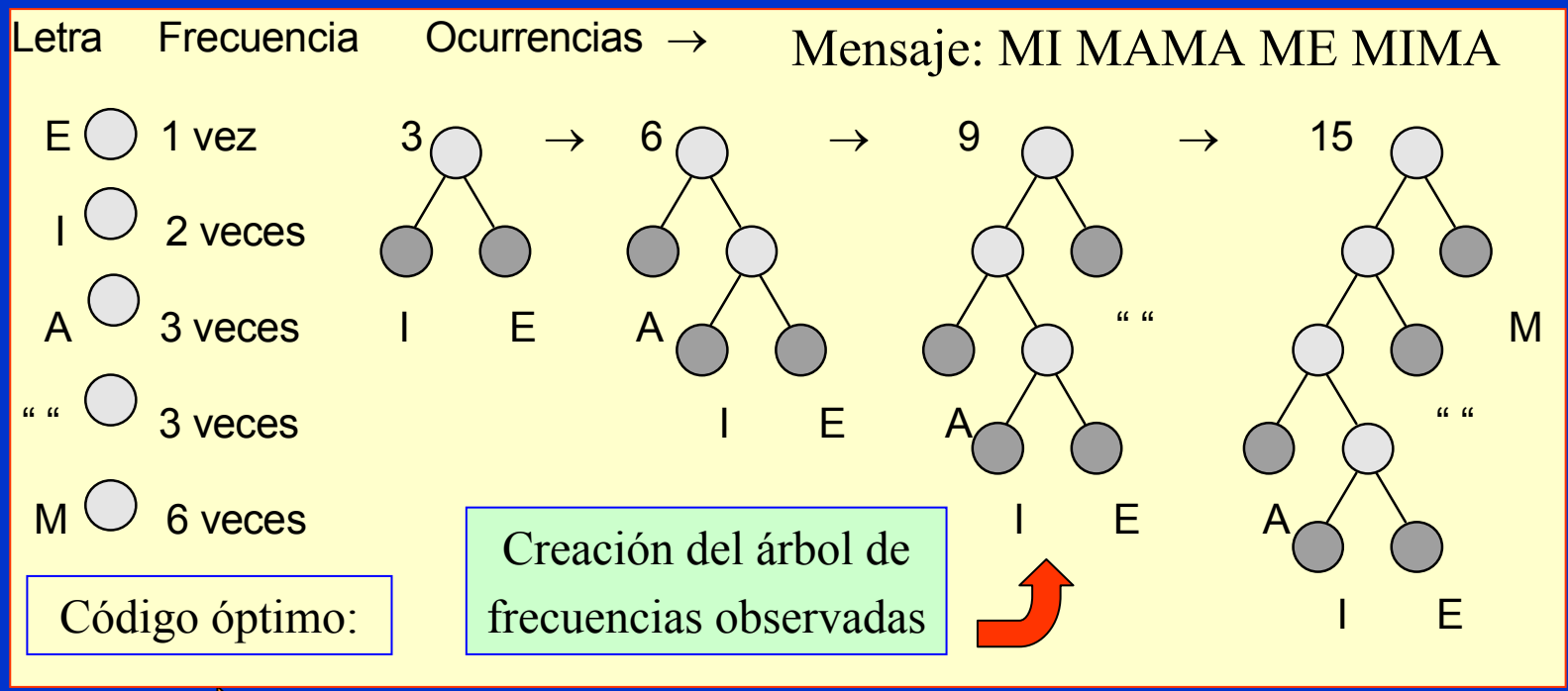
$$H(X) = \sum_i p(x) \log_2 [1/p(x)]$$

Veamos un ejemplo de codificación →

La expresión $\log_2 [1/p(x)]$ representa el número necesario de bits para codificar el mensaje X en un codificador óptimo.

⇒ Codificador óptimo es aquel que para codificar un mensaje X usa el menor número posible de bits.

Codificación con el método de Huffman



M = 1 “ ” = 01 A = 000 I = 0010 E = 0011

Mensaje: 1 0010 01 1 000 1 000 01 1 0011 01 1 0010 1 000 (33 bits)

Pregunta: ¿Con cuántos bits codificaría si se usara ASCII? Saque conclusiones.

Número necesario de bits y entropía

Para el mensaje $M = M_1M_2M_1M_1M_3M_1M_2M_3$ de 8 caracteres se pide calcular el número de bits óptimo de codificación:

Solución:

$p(M_1) = 0.5$; $p(M_2) = 0.25$; $p(M_3) = 0.25$; y obviamente $p(M_i) = 1.0$.

Para M_1: $\log_2 [1/ P(M_1)] = \log_2 2 = 1$ (necesitamos 1 bit)

Para M_2: $\log_2 [1/ P(M_2)] = \log_2 4 = 2$ (necesitamos 2 bits)

Para M_3: $\log_2 [1/ P(M_3)] = \log_2 4 = 2$ (necesitamos 2 bits)

Luego si M_1 se codifica como 0, M_2 como 10 y M_3 como 11, el mensaje M se codificará como: 0 10 0 0 11 0 10 11, es decir se transmiten 12 bits.

Si calcula la entropía de M obtendrá H(M) = 1,5 y al mismo valor se llega con el concepto de número medio de bits: se ha usado 12 bits para codificar un mensaje M de 8 elementos $\Rightarrow$ 12/8 = 1,5 bits por elemento.

Entropía condicional: equivocación de X

Si existe una segunda variable Y que influya sobre X, esto nos entregará importante información adicional.

El resultado más interesante es que...

La entropía se reduce: hay más *orden* y menos *incertidumbre*.

$$H(X/Y) = - \sum_{x,y} p_{(x,y)} \log_2 p_{(x,y)}$$

Donde $p(x,y) = p(y)p(x/y)$ y la relación $p(x/y)$ es la probabilidad de que se obtenga un estado X conocido el valor de Y.

Luego:

$$H(X/Y) = - \sum_{y} p_{(y)} \sum_{x} p_{(x/y)} \log_2 p_{(x/y)}$$

Ejemplo de entropía condicional

Sea $X = \{x_1, x_2, x_3, x_4\}$ con $p(x_i) = 0.25$
Sea ahora $Y = \{y_1, y_2, y_3\}$ con $p(y_1) = 0.5$; $p(y_2) = 0.25$; $p(y_3) = 0.25$
Luego $H(X) = 4 \log_2 4 = 2.0$ y $H(Y) = 2 \log_2 4 + \log_2 2 = 1.5$
Además hay las siguientes dependencias entre X e Y:
Si $Y = y_1 \Rightarrow X = x_1$ o x_2 o x_3 o x_4 (cualquiera con igual probabilidad)
Si $Y = y_2 \Rightarrow X = x_2$ o x_3 (cualquiera con igual probabilidad)
Si $Y = y_3 \Rightarrow X = x_3$ o x_4 (cualquiera con igual probabilidad)

Como $H(X/Y) = -\sum_{y=1}^{y=3} p_{(y)} \sum_{x=1}^{x=4} p_{(x/y)} \log_2 p_{(x/y)}$

$$H(X/Y) = - p(y_1)[p(x_1/y_1)\log_2 p(x_1/y_1) + p(x_2/y_1)\log_2 p(x_2/y_1) + p(x_3/y_1)\log_2 p(x_3/y_1) + p(x_4/y_1)\log_2 p(x_4/y_1)]$$
$$- p(y_2)[p(x_1/y_2)\log_2 p(x_1/y_2) + p(x_2/y_2)\log_2 p(x_2/y_2) + p(x_3/y_2)\log_2 p(x_3/y_2) + p(x_4/y_2)\log_2 p(x_4/y_2)]$$
$$- p(y_3)[p(x_1/y_3)\log_2 p(x_1/y_3) + p(x_2/y_3)\log_2 p(x_2/y_3) + p(x_3/y_3)\log_2 p(x_3/y_3) + p(x_4/y_3)\log_2 p(x_4/y_3)]$$

Calculando, se obtiene $H(X/Y) = 1.0 + 0.25 + 0.25 = 1.5$. La entropía de X ha bajado en medio bit con el conocimiento de su relación con Y.

Importancia de la entropía condicional

Equivocación de la clave k
¿Cuál es la probabilidad de que a un criptograma C le corresponda una cifra con una clave k?

$$H(K/C) = - \sum_{c} p_{(c)} \sum_{k} p_{(k/c)} \log_2 p_{(k/c)}$$

Servirá como un parámetro para la evaluación de la fortaleza de un criptosistema según equivocación de clave y mensaje.

Equivocación del mensaje M
¿Cuál es la probabilidad de que a un criptograma C le corresponda un mensaje en claro M?

$$H(M/C) = - \sum_{c} p_{(c)} \sum_{m} p_{(m/c)} \log_2 p_{(m/c)}$$

Ratio r del lenguaje

- Ratio r
  - Es el número de "bits de información" en cada carácter para mensajes con una longitud igual a N caracteres. Luego, según la definición de entropía, se tiene:

$$r = H(X)/N \quad \text{(bits/letra)}$$

  - Si codificáramos un mensaje letra a letra suponiendo además equiprobabilidad entre las letras, se obtiene la ratio absoluta del lenguaje, R:

$$R = H(X) \qquad \text{castellano} = 27 \text{ letras}$$

$$R_{\text{castellano}} = \log_2 n = \log_2 27 = 4.75 \quad \text{(bits/letra)}$$

Ratio verdadera del lenguaje

- Ratio verdadera
  - Como las letras que aparecen en un texto no tienen igual probabilidad, su frecuencia de aparición es distinta, los lenguajes está muy estructurados, hay bloques de dos palabras (digramas) característicos, trigramas, poligramas, etc., la ratio baja mucho...

$$1.2 < r < 1.5$$

  - A este valor se llega codificando los mensajes con monogramas, digramas, trigramas, etc.

Significado de la ratio del lenguaje

¿Qué significa esto?

– Si un alfabeto consta de L elementos existirán 2^{R*N} mensajes posibles de longitud N, la entropía máxima será $H(X)_{máx} = \log_2 L$, y sólo habrá 2^{r*N} mensajes que tengan sentido.

Importante: No significa que podamos codificar todos los mensajes de 27 caracteres con 2 bits (esto sería imposible). Significa que la información que contiene cada letra es tan sólo de 1.5 bits.

Veamos un ejemplo

Ejemplo de ratio del lenguaje

Un subalfabeto del castellano módulo 27 consta de 5 caracteres A, E, O, S, T todos ellos equiprobables, lo que puede aceptarse como representativo del lenguaje y más o menos cierto. ¿Cuántos mensaje de longitud 4 existen y cuántos con sentido?

Solución:

$R = \log_2 5 = 2{,}3219$, luego existirán $2^{R*4} = 2^{2.3219*4} = 625 = 5^4$

Como $1.2 < r < 1.5$ entonces cabe esperar x mensajes con sentido de longitud 4 del orden: $2^{1.2*4} < x < 2^{1.5*4}$ es decir $27 < x < 64$.

Buscando en un diccionario encontramos 45 palabras, que es precisamente el valor medio (64+27)/2 = 45.

aeta, asas, asea, asee, aseo, ases, asta, atea, atas, ates, ateo, atoa, atoe, atoo, osas, oses, osos, oste, otea, otee, oteo, easo, esas, eses, esos, esta, este esto, etas, tasa, tase, taso, teas, tesa, tese, teso, teta, seas, seso, seta, seto, sosa, sota, sote, soto.

Redundancia del lenguaje

- La redundancia D del lenguaje será la diferencia entre la ratio absoluta y la ratio real:

$$D = R - r$$

$$3.25 < D < 3.55$$

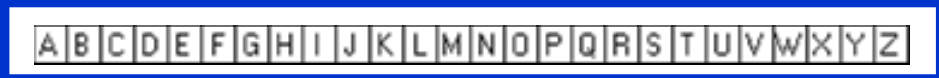
¿Qué significa esto?

– El número de bits extras (*bits redundantes*) necesarios para codificar un mensaje suponiendo un alfabeto de 27 caracteres (codificación con 5 bits puesto que $2^5 = 32$ y $2^4 = 16$) será aproximadamente igual a 3.5.

– D/R será un factor proporcional, luego:

$$68.42 < \%\ \text{Red. Lenguaje (D/R)} < 74.73$$

¿Es nuestro lenguaje redundante?



- El estudio de Shannon demuestra que la estructura del lenguaje produce esta redundancia:
- Diferencia de frecuencias de aparición de las letras.
- Existencia de digramas comunes.
- Existencia de trigramas comunes.
- Existencia de poligramas comunes.
- Estructuración de frases y oraciones con sentido.

Y nuestra misión es crear algoritmos que sean seguros y eviten estos ataques.

Esto dará pistas al criptoanalista para atacar un sistema.

Un ejemplo de redundancia (parte 1)

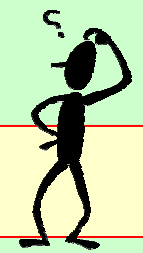
Todos los lenguajes serán redundantes. Esto quiere decir que la misma cantidad de información se puede entregar con menos símbolos o bits.

Sea el siguiente mensaje M = HBNVZNCRC

1ª ayuda:

"En el mensaje original se han quitado las vocales".

Esto nos permite suponer que entre consonantes habrá cero, una, dos o tres vocales, según reglas del lenguaje...



M = __H__B__N__V__Z__N__C__R__C__

Un ejemplo de redundancia (parte 2)

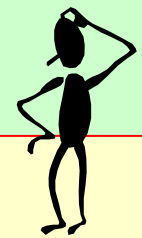
Teníamos el mensaje M = HBNVZNCRC y además

M = __H__B__N__V__Z__N__C__R__C__

2ª ayuda:

"El mensaje original contiene cinco palabras".

Esto nos permite limitar el número de mensajes posibles que tengan sentido. En estas condiciones podrían existir muchos mensajes de 5 palabras, aunque no cumpliesen de forma lógica con las reglas del lenguaje. Un ejemplo podría ser...



M = AHÍ BUENO AVE ZONA CERCA

Un ejemplo de redundancia (parte 3)

Teníamos el mensaje M = HBNVZNCRC y además

M = __H__B__N__V__Z__N__C__R__C__

M = AHÍ BUENO AVE ZONA CERCA

3ª ayuda y siguientes:

a) "El mensaje original tiene que ver con un circo".

b) "Corresponde al estribillo de una canción infantil".

c) "Los espacios están en: M = HB N VZ N CRC".

Seguro que habrá adivinado ya el mensaje.... ☺

M = HABÍA UNA VEZ UN CIRCO

Redundancia y entropía condicional

El ejemplo anterior, además de demostrar que todos los lenguajes son redundantes, es un claro exponente de lo que se entiende en la práctica por entropía condicional.

Cada vez que vamos dando nuevas pistas, disminuye la incertidumbre del mensaje hasta que ésta se anula y por lo tanto la entropía es igual a 0 ya que existe un único mensaje posible con probabilidad igual a la unidad.

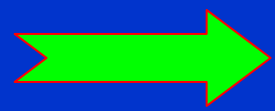
Algo similar ocurre cuando resolvemos un crucigrama y lo anteriormente resuelto nos sirve como pistas para descubrir palabras nuevas. Mientras más palabras tengamos, más fácil se hace avanzar en su resolución.

Secreto de un sistema criptográfico

Shannon midió el secreto de un criptosistema como la incertidumbre del mensaje en claro conocido el criptograma recibido:

Mensajes	$M = \{M_1, M_2, ..., M_3\}$	$\sum_M p(M) = 1$
Criptogramas	$C = \{C_1, C_2, ..., C_3\}$	$\sum_C p(C) = 1$
Claves	$K = \{K_1, K_2, ..., K_3\}$	$\sum_K p(K) = 1$

¿Cuando tendrá nuestro sistema un secreto perfecto?

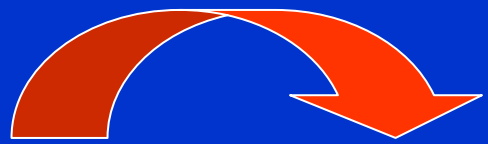


Definiciones previas secreto criptográfico

- $p(M)$: Probabilidad de enviar un mensaje M. Si hay n mensajes M_i equiprobables, $p(M_i) = 1/n$.
- $p(C)$: Probabilidad de recibir un criptograma C. Si cada uno de los n criptogramas recibidos C_i es equiprobable, $p(C_i) = 1/n$.
- $p_M(C)$: Probabilidad de que, a partir de un texto en claro M_i, se obtenga un criptograma C_i.
- $p_C(M)$: Probabilidad de que, una vez recibido un criptograma C_i, éste provenga de un texto claro M_i.

Secreto criptográfico perfecto (1)

Un sistema tiene secreto perfecto si el conocimiento del texto cifrado no proporciona ninguna información acerca del mensaje. Es decir, cuando la probabilidad de acierto al recibir el elemento $i+1$ es la misma que en el estado i.



Secreto perfecto $\Rightarrow$ $p(M) = p_C(M)$

La probabilidad p de enviar un mensaje M con texto en claro p(M) o probabilidad a priori será igual a la probabilidad p de que, conocido un criptograma C, éste se corresponda a un mensaje M cifrado con la clave K. Esta última (probabilidad a posteriori) es $p_C(M)$.

Secreto criptográfico perfecto (2)

La probabilidad p de recibir un texto cifrado C al cifrar un mensaje M usando una clave K será $p_M(C)$. Luego, M debe haberse cifrado con alguna clave K:

$$p_M(C) = \sum_K p(K) \quad \text{donde } E_K(M) = C$$

$$\exists\, k_j \,/\, E_{kj}(M_i) = C_i$$

En el fondo esto viene a significar que para lograr un secreto perfecto, el espacio de claves debe ser al menos de igual tamaño que el espacio de mensajes.

Secreto criptográfico perfecto (3)

La condición necesaria y suficiente del secreto perfecto es que para cualquier valor de M se cumpla que la probabilidad de recibir C, resultado de la cifra de un mensaje M con una clave K, sea la misma que recibir el criptograma C, resultado de la cifra de otro mensaje M' distinto, cifrado con otra clave.

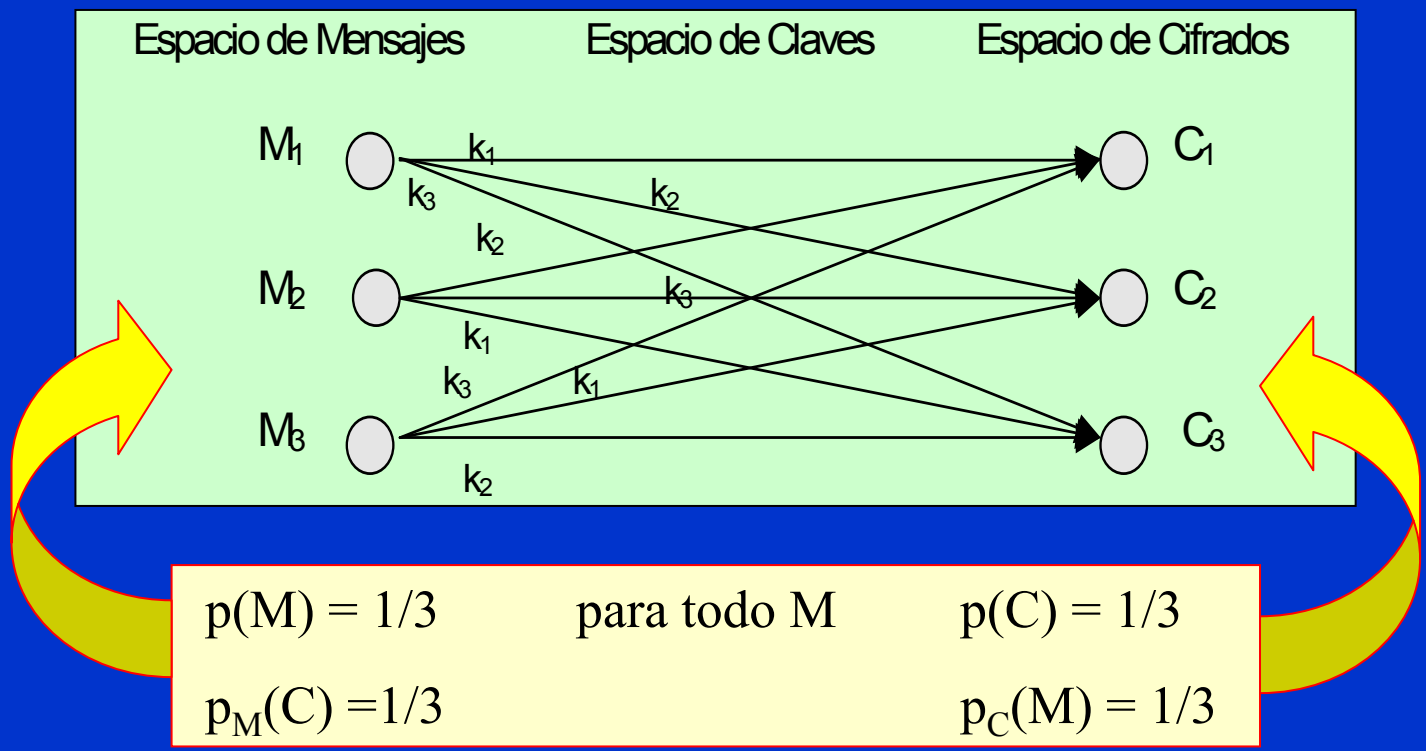
$$p_M(C) = p(C)$$

para todo valor de M

Veamos algunos ejemplos

Cifrador con secreto perfecto

Sea el siguiente escenario:



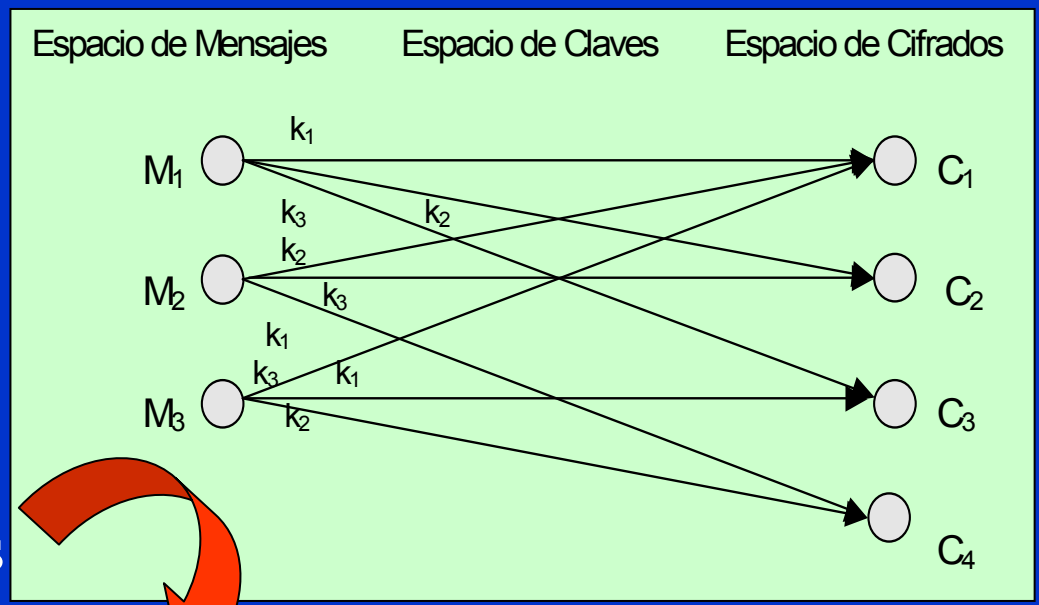
Cifrador sin secreto perfecto (1)

Sea ahora el siguiente escenario:

$p(M_1) = 1/3$

$p(M_2) = 1/3$

$p(M_3) = 1/3$



$p(C_1) = 3/9$

$p(C_2) = 2/9$

$p(C_3) = 2/9$

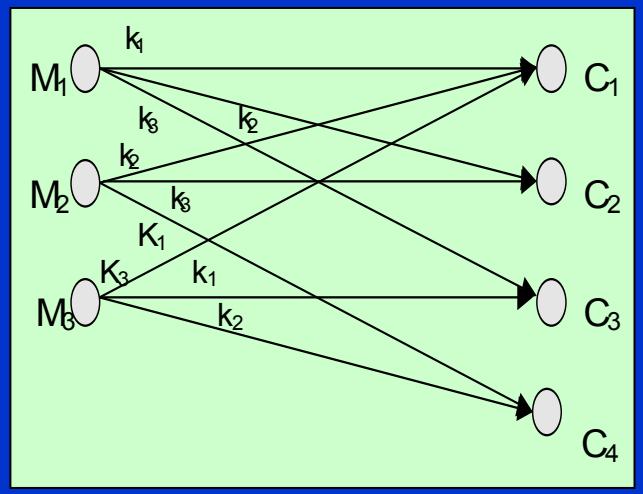
$p(C_4) = 2/9$

Algo más

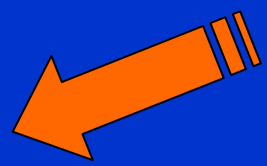
¿Probabilidad de que un mensaje M_i se convierta en un criptograma C_i: $[P_{Mi}(C_i)]$ y que un criptograma C_i sea el resultado de la cifra de un mensaje M_i: $[P_{Ci}(M_i)]$?

Cifrador sin secreto perfecto (2)

Esquema anterior:



$p_{C1}(M_1) = 1/3$	$p_{C1}(M_2) = 1/3$	$p_{C1}(M_3) = 1/3$
$p_{C2}(M_1) = 1/2$	$p_{C2}(M_2) = 1/2$	$p_{C2}(M_3) = 0$
$p_{C3}(M_1) = 1/2$	$p_{C3}(M_2) = 0$	$p_{C3}(M_3) = 1/2$
$p_{C4}(M_1) = 0$	$p_{C4}(M_2) = 1/2$	$p_{C4}(M_3) = 1/2$



$p_{M1}(C_1) = 1/3$	$p_{M1}(C_2) = 1/3$	$p_{M1}(C_3) = 1/3$	$p_{M1}(C_4) = 0$
$p_{M2}(C_1) = 1/3$	$p_{M2}(C_2) = 1/3$	$p_{M2}(C_3) = 0$	$p_{M2}(C_4) = 1/3$
$p_{M3}(C_1) = 1/3$	$p_{M3}(C_2) = 0$	$p_{M3}(C_3) = 1/3$	$p_{M3}(C_4) = 1/3$

Distancia de unicidad

- Se entenderá por Distancia de Unicidad al bloque N de texto cifrado o criptograma mínimo necesario para que se pueda intentar con *ciertas expectativas de éxito* un ataque en búsqueda de la clave usada para cifrar.
- Este valor se obtiene cuando la equivocación de la clave $H_C(K)$ se acerca a cero o tiende a anularse.
- A medida que tenga un criptograma más largo, y por tanto más información, se supone que la tarea de ataque del criptoanalista se va facilitando.
- Se busca el tamaño N de criptograma que permita esperar que la solución de K sea única. Se supondrá un cifrador aleatorio $\Rightarrow$ Modelo de Hellmann

Parámetros del modelo de Hellman (1)

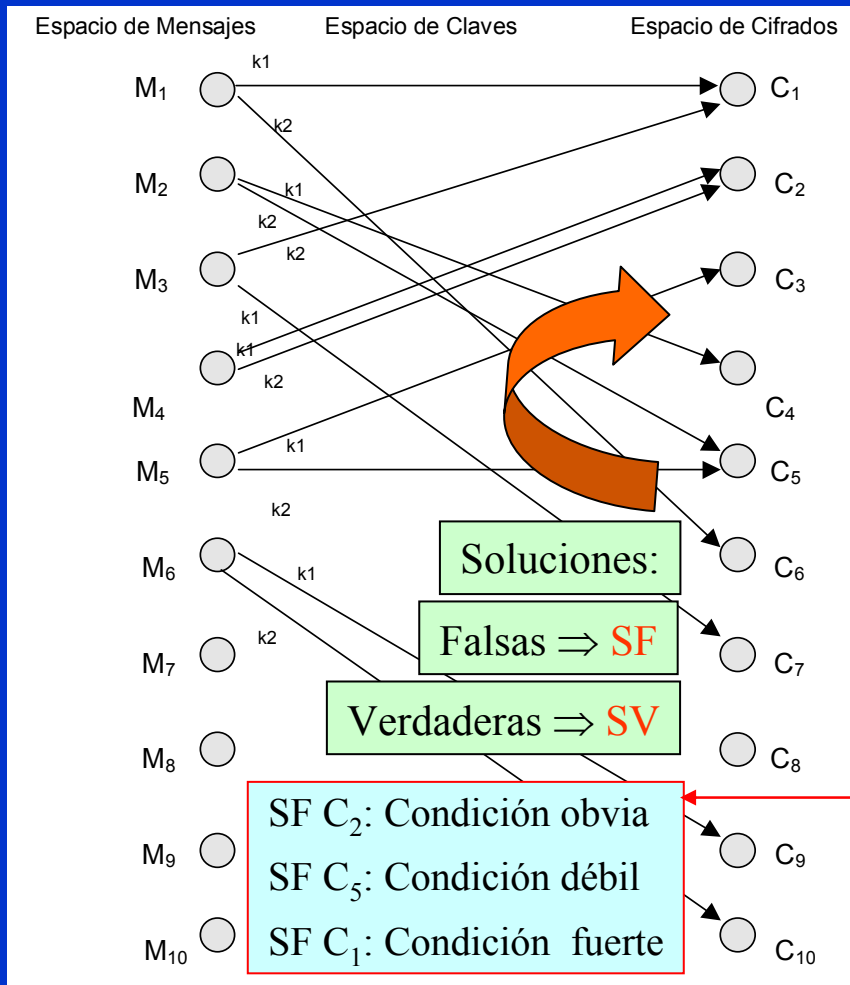
- Existirán 2^{RN} mensajes posibles de longitud N.
- Existirán 2^{rN} mensajes de longitud N con sentido.
- El espacio de mensajes de longitud N se dividirá:
  - Mensajes con sentido: $M_{CS} = 2^{rN}$
  - Mensajes sin sentido: $M_{SS} = 2^{RN} - 2^{rN}$
- Los 2^{rN} mensajes con sentido serán equiprobables siendo su valor $p(M_{CS}) = 1/2^{rN} = 2^{-rN}$
- El resto de mensajes ($2^{RN} - 2^{rN}$) tendrá una probabilidad nula $p(M_{SS}) = 0$.

Parámetros del modelo de Hellman (2)

- Existirán $2^{H(K)}$ claves equiprobables.
- En donde H(K) es la entropía de la clave.
- Con $p(K) = 1/2^{H(K)} = 2^{-H(K)}$
- Con estas claves se cifrarán todos los mensajes con sentido dando lugar a 2^{RN} textos cifrados posibles de longitud N.
- Los criptogramas obtenidos serán equiprobables.

Por sencillez, veremos el modelo de cifrador aleatorio de Hellman sólo con dos claves k_1 y k_2. 

Esquema de cifrador aleatorio de Hellmann



SV: Un criptograma está asociado sólo a un texto en claro con sentido cifrado con una única clave k_i.

SF: Cualquier otra solución de cifra distinta a la anterior.

SV: $C_3 = E_{k1}(M_5)$ $C_4 = E_{k1}(M_2)$

$C_6 = E_{k2}(M_1)$ $C_7 = E_{k1}(M_3)$

$C_9 = E_{k1}(M_6)$ $C_{10} = E_{k2}(M_6)$

SF: $C_2 = E_{k1}(M_4)$ $C_2 = E_{k2}(M_4)$

$C_5 = E_{k2}(M_2)$ $C_5 = E_{k2}(M_5)$

$C_1 = E_{k1}(M_1)$ $C_1 = E_{k2}(M_3)$

Cálculo de la distancia de unicidad (1)

- Para cada solución correcta de un texto M cifrado con una clave k del espacio $2^{H(K)}$, existirán otras $(2^{H(K)}-1)$ claves con la misma probabilidad de entregar una solución falta SF.

Sea q la probabilidad de obtener un mensaje con sentido:

$q = 2^{rN} / 2^{RN} = 2^{(r-R)N} = 2^{-DN}$ Luego:

$SF = (2^{H(K)}-1)\, q = (2^{H(K)}-1)\, 2^{-DN} = 2^{H(K)-DN} - 2^{-DN}$

$SF \approx 2^{H(K)-DN}$ ⟹ $\log_2 SF = H(K) - DN$

Cálculo de la distancia de unicidad (2)

La solución SF = 0 es imposible porque sólo se llega a ella de forma asintótica con un valor de N infinito como se muestra en la diapositiva siguiente.

Se acepta entonces que haya como máximo una sola solución falsa, de ahí su nombre de unicidad, luego:

$$SF = 2^{H(K) - DN} \qquad SF = 1 \quad \Rightarrow \quad H(K) - DN = 0$$

Por lo tanto:

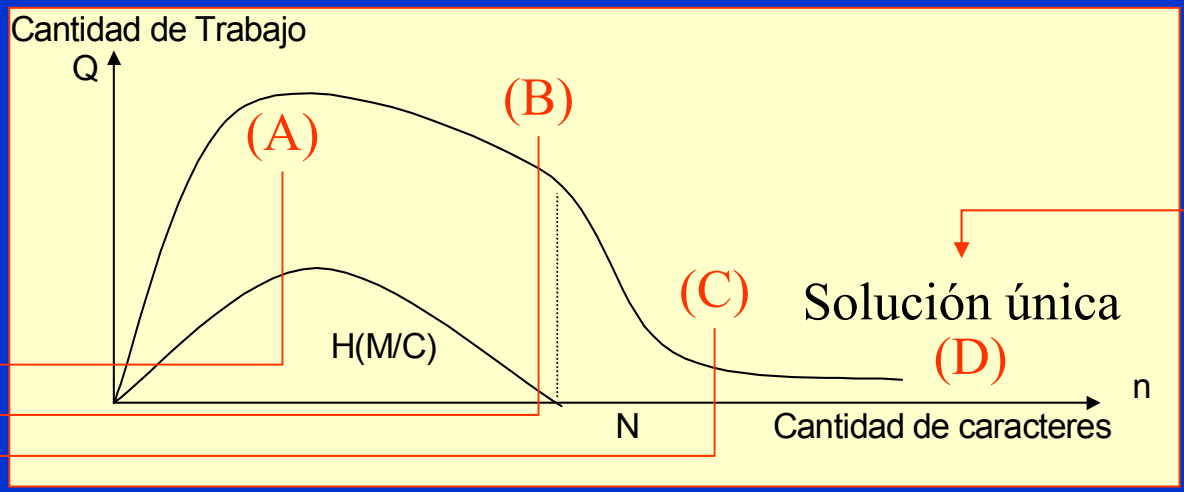
$$N = H(K) / D$$

Número mínimo de bytes o caracteres en C para intentar un ataque.

Ejemplos de distancia de unicidad

- Para el cifrador del César módulo 27 en el que "la clave" es b, todos los posibles desplazamientos de caracteres, $1 \leq b \leq 26$, su entropía $H(X) = \log_2 26 = 4.7$ bits por lo que $N = 4.7/3.4 = 1.4$ caracteres.
- Para el mismo cifrador del César pero con clave, si el alfabeto tiene n caracteres, existirán n! posibles claves. En este caso la entropía de la clave puede aproximarse como $H(X) = \log_2 27! \approx 27 * \log_2 (27/e)$, por lo que $N = 27 * \log2\ (27/2.72)/3.4 = 27.4$ caracteres.
- En el sistema DES la clave verdadera es de 56 bits por lo que su entropía $H(X) = 56$. Si el mensaje sólo contiene letras mayúsculas (27 elementos) podríamos decir que $N = 56/3.4 = 16{,}5$ caracteres.
- Nota: aunque el valor de N sea ahora más bajo no quiere decir en absoluto que el DES sea menos seguro que el cifrador del César con clave. Este último se puede atacar fácilmente con estadísticas del lenguaje muy elementales y el DES no. Además, recuerde que debe contar con un criptograma varias veces mayor que el valor de N si desea que su criptoanálisis tenga alguna posibilidad de éxito.

Cantidad de trabajo Q



(A) Inicialmente hay que hacer un arduo trabajo para obtener algo coherente. Habrá muchas soluciones falsas.

(B) Cuando se tiene una cantidad "adecuada" de texto cifrado, la cantidad de trabajo disminuye. Se descartan algunas soluciones.

(C) Cuando se anula la equivocación de la clave, H(M/C) = 0, disminuyen las soluciones falsas y la solución tiende a ser única.

El uso de técnicas de difusión

Para lograr un mayor secreto en las operaciones de cifra, Shannon propuso usar dos técnicas: difusión y confusión.

Difusión: es la transformación sobre el texto en claro con el objeto de dispersar las propiedades estadísticas del lenguaje sobre todo el criptograma. Se logra con transposiciones.

TRANSPOSICIONES

La transposición consiste básicamente en una permutación, es decir, cambiar los caracteres de lugar según una regla, una función, etc. Por ejemplo el carácter primero se posiciona en el lugar cuarto, el segundo en el lugar tercero, etc.

El uso de técnicas de confusión

Confusión: Transformación sobre el texto en claro con objeto de mezclar los elementos de éste, aumentando la complejidad de la dependencia funcional entre clave y criptograma. Se obtiene a través de sustituciones.

SUSTITUCIONES

La sustitución consiste básicamente modificar la información, es decir, sustituir un carácter por otro de acuerdo a una regla, una función, etc. Por ejemplo cambiar la letra A por la letra M, la letra B por la letra X , etc.

Ambas técnicas serán usadas en los sistemas clásicos y aunque no parezca, también lo usará el cifrador DES.

Fin del Tema 4

Cuestiones y ejercicios (1 de 2)

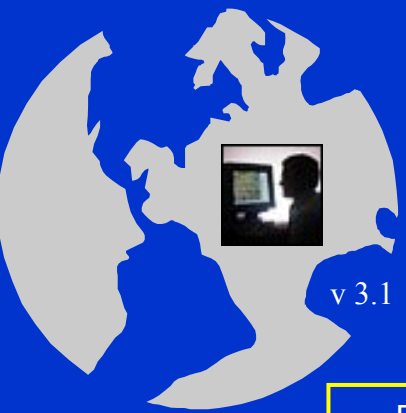
1. Al despertar ponemos la radio y escuchamos noticias que no nos llaman la atención. ¿Por qué decimos que no había información?
2. Justifique la definición logarítmica de cantidad de información, es decir la razón de que $c_i = - \log (p_i)$.
3. ¿Por qué usamos la base 2 en el logaritmo que define c_i?
4. ¿Cuál es el número mínimo –e inteligente- de preguntas que hay que hacer para pasar de la incertidumbre a la certeza en un sistema de n estados equiprobables? ¿Y si no son equiprobables?
5. ¿Por qué la entropía es no nula y se anula sí y sólo si uno de los estados de la variable es igual a la unidad?
6. Codificamos en binario un sistema con 256 estados equiprobables. Si no usamos un codificador óptimo, ¿cuántos bits son necesarios? Mediante un codificador óptimo, ¿usaremos más o menos bits?

Cuestiones y ejercicios (2 de 2)

7. ¿Qué representa la expresión $\log_2 [1/p(x)]$ en la entropía H(X)? Si $p(x_1)=0,6$; $p(x_2)=0,3$; $p(x_3)=0,1$ calcule $\log_2 [1/p(x)]$ ¿qué opina?
8. Definimos un alfabeto con 71 elementos (mayúsculas y minúsculas, minúsculas acentuadas, números, punto y coma). Si estos elementos son equiprobables, ¿cuál es la ratio absoluta de este alfabeto?
9. ¿La ratio verdadera es mayor o menor que la absoluta? ¿Por qué?
10. Un alfabeto consta de 8 elementos equiprobables. ¿Cuántos posibles mensajes de tamaño 4 existen? De éstos, ¿cuántos mensajes podrían tener sentido si esos 8 elementos representan al idioma castellano?
11. ¿Cuándo decimos que un sistema tiene secreto perfecto? En un sistema real, ¿es eso posible? Piense en algún ejemplo y coméntelo.
12. ¿Por qué se dice que hay que minimizar las soluciones falsas SF en el modelo de Hellman para romper la clave? ¿Es la clave k única?

Tema 5
Teoría de los Números

Seguridad Informática y Criptografía



Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 67 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Conceptos básicos de congruencia

- Es la base matemática (matemáticas discretas) en la que se sustentan las operaciones de cifra.
- Concepto de congruencia:
  - Sean dos números enteros a y b: se dice que a es congruente con b en el módulo o cuerpo n (Z_n) si y sólo si existe algún entero k que divide de forma exacta la diferencia (a - b)

$$a - b = k * n$$
$$a \equiv_n b$$
$$a \equiv b \bmod n$$

Operaciones de congruencia en Z_n

¿Es 18 congruente con 3 módulo 5?

¿$18 \equiv 3 \bmod 5$?

Sí, porque: $18-3 = 15 = k*5$ con $k = 3$

¿Cómo se usará esto en criptografía?

Esta operación en Z_n se expresará así:

$18 \bmod 5 = 3$

El valor 3 será el resto o residuo.

El conjunto de números que forman los restos dentro de un cuerpo Z_n serán muy importantes en criptografía.

Propiedades de la congruencia en Z_n

- Propiedad Reflexiva:

$$a \equiv a \bmod n \quad \forall\, a \in Z$$

- Propiedad Simétrica:

$$a \equiv b \bmod n \Rightarrow b \equiv a \bmod n \quad \forall\, a,b \in Z$$

- Propiedad Transitiva:

$$\text{Si} \quad a \equiv b \bmod n \quad \text{y} \quad b \equiv c \bmod n$$
$$\Rightarrow a \equiv c \bmod n \quad \forall\, a,b,c \in Z$$

Propiedades de las operaciones en Z_n (1)

- Propiedad Asociativa:

  $a + (b + c) \bmod n \equiv (a + b) + c \bmod n$

- Propiedad Conmutativa:

  $a + b \bmod n \equiv b + a \bmod n$

  $a * b \bmod n \equiv b * a \bmod n$

- Propiedad Distributiva:

  $a * (b+c) \bmod n \equiv ((a * b) + (a * c)) \bmod n$

Se usará el signo = en vez de ≡ (algo propio de los Campos de Galois)

$$a * (b+c) \bmod n = ((a * b) + (a * c)) \bmod n$$

Propiedades de las operaciones en Z_n (2)

- Existencia de Identidad:

  $a + 0 \bmod n = 0 + a \bmod n = a \bmod n = a$

  $a * 1 \bmod n = 1 * a \bmod n = a \bmod n = a$

- Existencia de Inversos: ✓

  $a + (-a) \bmod n = 0$

  $a * (a^{-1}) \bmod n = 1$ (si $a \neq 0$) ⟶ No siempre existe

- Reducibilidad: ✓

  $(a + b) \bmod n = [(a \bmod n) + (b \bmod n)] \bmod n$

  $(a * b) \bmod n = [(a \bmod n) * (b \bmod n)] \bmod n$

Ambos importantes en criptografía ✓

Conjunto completo de restos CCR

Para cualquier entero positivo n, el conjunto completo de restos será CCR = {0, 1, 2, ... n-1}, es decir:

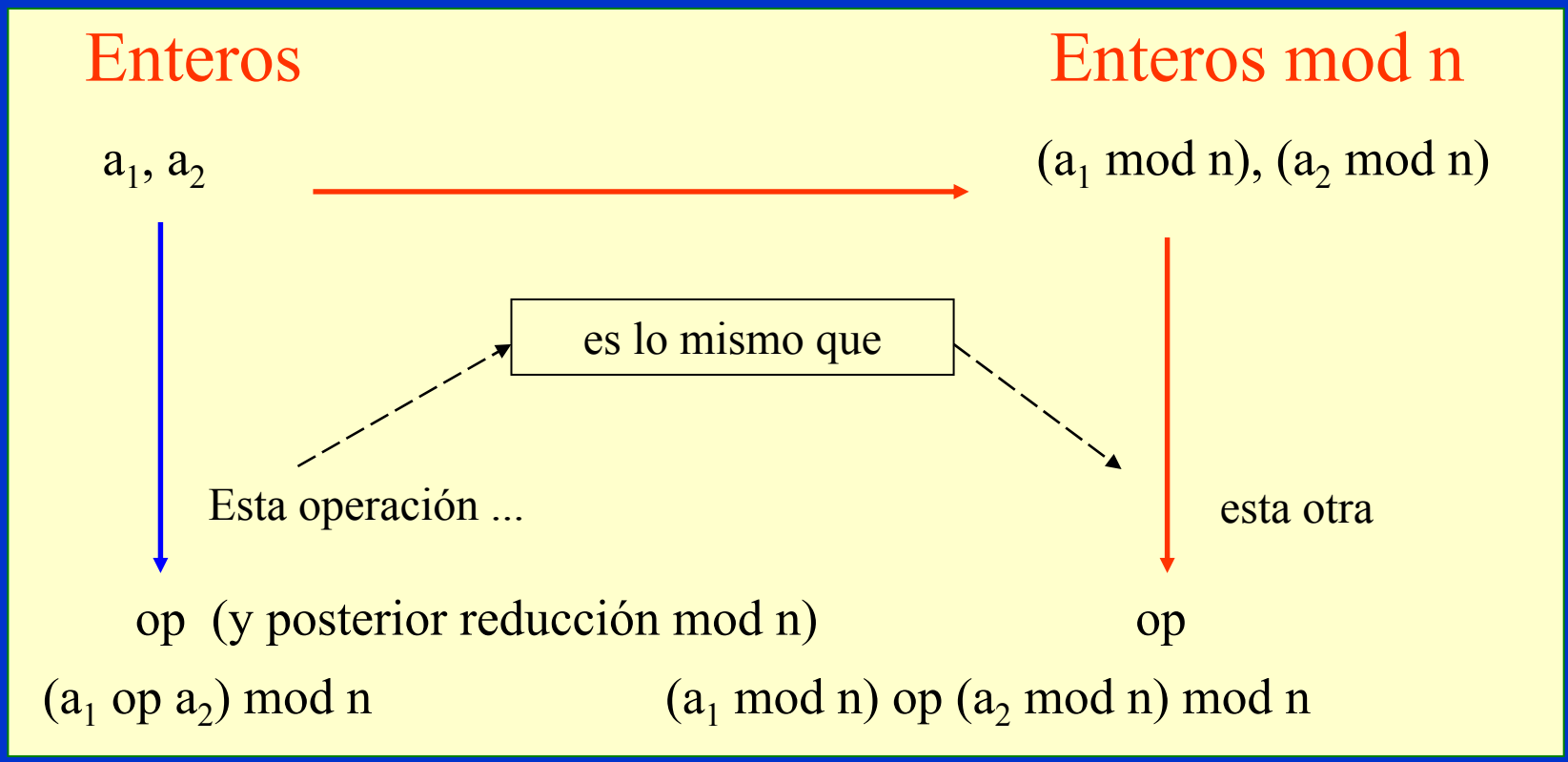
$$\forall\ a \in Z \quad \exists\ !\ r_i \in CCR\ /\ a \equiv r_i \bmod n$$

CCR (11) = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

CCR (6) = {0, 1, 2, 3, 4, 5} = {12, 7, 20, 9, 16, 35}

El segundo conjunto es equivalente: $12 \rightarrow 0$, $7 \rightarrow 1$...

Homomorfismo de los enteros



Un ejemplo de homomorfismo

88 * 93 mod 13

8.184 mod 13

Resultado: 7

Se desbordaría la memoria de nuestro sistema ☹

Ahora ya no se desborda la memoria ☺

Ejemplo: una calculadora capaz de trabajar sólo con tres dígitos ...

Solución por homomorfismo:

88 * 93 mod 13

[(88) mod 13 * (93) mod 13] mod 13

10 * 2 mod 13

20 mod 13 Resultado: 7

se llega a lo mismo, pero...

... y hemos usado siempre números de 3 dígitos. En este caso la operación máxima sería 12*12 = 144, es decir tres dígitos.

Divisibilidad de los números

En criptografía muchas veces nos interesará encontrar el máximo común denominador mcd entre dos números *a* y *b*.

Para la existencia de inversos en un cuerpo n, la base a y el módulo n deberán ser primos entre sí. $\Rightarrow$ mcd (a, n) = 1

Algoritmo de Euclides:

- a) Si x divide a *a* y *b* $\Rightarrow$ $a = x * a'$ y $b = x * b'$
- b) Por lo tanto: $a - k * b = x * a' - k * x * b'$

  $a - k * b = x (a' - k * b')$
- c) Entonces se concluye que x divide a $(a - k * b)$

El máximo común denominador mcd

Como hemos llegado a que x divide a $(a - k * b)$ esto nos permitirá encontrar el mcd (a, b):

Si a > b entonces $a = d_1 * b + r$

(con d_i un entero y r un resto)

Luego mcd (a, b) = mcd (b ,r) $(a > b > r \geq 0)$

porque:

Si b > r entonces $b = d_2 * r + r'$

(con r un entero y r' un resto)

Divisibilidad con algoritmo de Euclides

mcd (148, 40)

$148 = 3 * 40 + 28$

$40 = 1 * 28 + 12$

$28 = 2 * 12 + 4$

$12 = 3 * 4 + 0$

mcd (148, 40) = 4

$148 = 2^2 * 37$

$40 = 2^3 * 5$

Factor común

$2^2 = 4$

No hay factor común

$385 = 5 * 7 * 11$

$78 = 2 * 3 * 13$

Será importante en criptografía

mcd (385, 78)

$385 = 4 * 78 + 73$

$78 = 1 * 73 + 5$

$73 = 14 * 5 + 3$

$5 = 1 * 3 + 2$

$3 = 1 * 2 + 1$

$2 = 2 * 1 + 0$

mcd (385, 78) = 1

Inversión de una operación de cifra

- En criptografía deberá estar permitido invertir una operación para recuperar un cifrado $\Rightarrow$ descifrar.
- Si bien la cifra es una función, en lenguaje coloquial la operación de cifrado sería una "multiplicación" y la operación de descifrado una "división".
- La analogía anterior sólo será válida en el cuerpo de los enteros Z_n con inverso.
- Luego, si en una operación de cifra la función es el valor a dentro de un cuerpo n, deberemos encontrar el inverso $a^{-1} \bmod n$ para descifrar; en otras palabras ...

Inversos en un cuerpo

Si $a * x \bmod n = 1$

x será el inverso multiplicativo (a^{-1}) de a en el módulo n

- No siempre existen los inversos multiplicativos. En realidad lo raro es que existan.
- Por ejemplo, en Z = 2 no existirán inversos pues la única solución para $a*x \bmod 2 = 1$, con a, x = {0, 1}, sería x = 1 para a = 1, una solución trivial. El cero nunca será solución del inverso multiplicativo.

Existencia del inverso por primalidad

$\exists$ inverso a^{-1} en mod n *ssi* mcd (a, n) = 1

Si mcd (a,n) = 1, el resultado de a∗i mod n (para i todos los restos de n) serán valores distintos dentro del cuerpo n.

$$\text{mcd}(a, n) = 1 \Rightarrow \exists\, x \,!\; 0<x<n \;/\; a * x \bmod n = 1$$

Sea: a = 4 y n = 9. Valores de i = {1, 2, 3, 4, 5, 6, 7, 8}

SOLUCIÓN ÚNICA

4∗1 mod 9 = 4	4∗2 mod 9 = 8	4∗3 mod 9 = 3
4∗4 mod 9 = 7	4∗5 mod 9 = 2	4∗6 mod 9 = 6
4∗7 mod 9 = 1	4∗8 mod 9 = 5	Si mcd (a,n) ≠ 1

Inexistencia de inverso (no primalidad)

¿Y si no hay primalidad entre a y n?

Si mcd (a, n) ≠ 1

No existe ningún x que $0 < x < n \ / \ a * x \bmod n = 1$

Sea: a = 3 y n = 6 Valores de i = {1, 2, 3, 4, 5}

$3*1 \bmod 6 = 3$ $3*2 \bmod 6 = 0$ $3*3 \bmod 6 = 3$

$3*4 \bmod 6 = 0$ $3*5 \bmod 6 = 3$

No existe el inverso para ningún resto del cuerpo.

Inversos aditivos y multiplicativos

(A+B) mod 5

B +	0	1	2	3	4
A 0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

0+0 = 0
1∗1 = 1
Es trivial

(A∗B) mod 5

B ∗	0	1	2	3	4
A 0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

- En la operación suma siempre existirá el inverso o valor identidad de la adición (0) para cualquier resto del cuerpo.
- En la operación producto, de existir un inverso o valor de identidad de la multiplicación (1) éste es único. La condición para ello es que el número y el módulo sean primos entre sí. Por ejemplo para n = 4, el resto 2 no tendrá inverso multiplicativo, en cambio el resto 3 sí.

Conjunto reducido de restos CRR

- El conjunto reducido de restos, conocido como CRR de n, es el subconjunto $\{0, 1, \ldots n_i, \ldots n-1\}$ de restos primos con el grupo n.
- Si n es primo, todos los restos serán primos con él.
- Como el cero no es una solución, entonces:

$$CRR = \{1, \ldots, n_i, \ldots n-1\} \ / \ \text{mcd}\ (n_i, n) = 1$$

Ejemplo: CRR mod 8 = {1, 3, 5, 7}

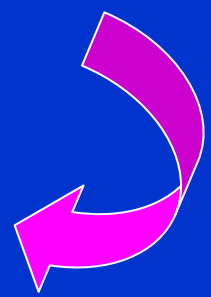
CRR mod 5 = {1, 2, 3, 4}

Utilidad del CRR

¿Qué utilidad tiene esto en criptografía?

El conocimiento del CRR permitirá aplicar un algoritmo para el cálculo del inverso multiplicativo de un número x dentro de un cuerpo o grupo n a través de la función $\phi(n)$, denominada Función de Euler o Indicador de Euler.

Será muy importante tanto en los sistemas simétricos que trabajan en un módulo (con excepción del DES) como en los sistemas asimétricos. En ambos casos la cifra y las claves están relacionadas con el CRR.



Función de Euler $\phi(n)$

- Función $\phi(n)$ de Euler
- Entregará el número de elementos del CRR.
- Podremos representar cualquier número n de estas cuatro formas:
  - a) n es un número primo.
  - b) n se representa como $n = p^k$ con p primo y k entero.
  - c) n es el producto $n = p*q$ con p y q primos.
  - d) n es un número cualquiera (genérico).

$$n = p_1^{e1} * p_2^{e2} * \ldots * p_t^{et} = \prod_{i=1}^{t} p_i^{ei}$$

Veamos cada uno de ellos

Función $\phi(n)$ de Euler ($n = p$)

Caso 1: n es un número primo

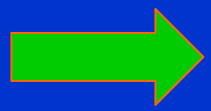
Si n es primo, $\phi(n)$ será igual a CCR menos el 0.

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

Se usará en sistemas ElGamal y DSS

Si n es primo, entonces CRR = CCR - 1 ya que todos los restos de n, excepto el cero, serán primos entre sí.

Ejemplo



CRR(7) = {1,2,3,4,5,6} seis elementos

$\therefore \phi(7) = n - 1 = 7-1 = 6$

$\phi(11) = 11-1 = 10$; $\phi(23) = 23-1 = 22$

Función $\phi(n)$ de Euler ($n = p^k$)

Caso 2: $n = p^k$ (con p primo y k un entero)

$$\phi(n) = \phi(p^k) = p^k - p^{k-1} \qquad \boxed{\phi(p^k) = p^{k-1}(p-1)}$$

De los p^k elementos del CCR, restaremos todos los múltiplos $1*p$, $2*p$, $3*p$, ...$(p^{k-1}-1)*p$ y el cero.

Ejemplo

CRR(16) = {1,3,5,7,9,11,13,15} ocho elementos

$\therefore \phi(16) = \phi(2^4) = 2^{4-1}(2-1) = 2^3*1 = 8$

$\phi(125) = \phi(5^3) = 5^{3-1}*(5-1) = 5^2*4 = 25*4 = 100$

Función $\phi(n)$ de Euler ($n = p*q$) (1)

Caso 3: $n = p*q$ (con p y q primos)

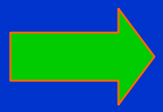
$$\phi(n) = \boxed{\phi(p*q) = \phi(p)*\phi(q) = (p-1)(q-1)}$$

De los $p*q$ elementos del CCR, restaremos todos los múltiplos de $p = 1*p, 2*p, \ldots (q-1)*p$, todos los múltiplos de $q = 1*q, 2*q, \ldots (p-1)*q$ y el cero.

$$\phi(p*q) = p*q - [(q-1) + (p-1) + 1] = \underbrace{p*q - q - p + 1}_{(p-1)(q-1)}$$

Función $\phi(n)$ de Euler ($n = p*q$) (2)

Ejemplo



$CRR(15) = \{1,2,4,7,8,11,13,14\}$ ocho elementos

$\therefore \phi(15) = \phi(3*5) = (3-1)(5-1) = 2*4 = 8$

$\phi(143) = \phi(11*13) = (11-1)(13-1) = 10*12 = 120$

Será una de las funciones más utilizadas ya que es la base del sistema RSA que durante muchos años ha sido un estándar de hecho.

Función $\phi(n)$ de Euler (n = genérico)

Caso 4: $n = p_1^{e1} * p_2^{e2} * \ldots * p_t^{et}$ (p_i son primos)

$$\phi(n) = \prod_{i=1}^{t} p_i^{ei-1} (p_i - 1)$$

(demostración no inmediata)

Ejemplo

$CRR(20) = \{1, 3, 7, 9, 11, 13, 17, 19\}$ ocho elementos

$\therefore \phi(20) = \phi(2^2 * 5) = 2^{2-1}(2-1) * 5^{1-1}(5-1) = 2^1 * 1 * 1 * 4 = 8$

$\phi(360) = \phi(2^3 * 3^2 * 5) = 2^{3-1}(2-1) * 3^{2-1}(3-1) * 5^{1-1}(5-1) = 96$

Teorema de Euler

Dice que si mcd (a,n) = 1 $\Rightarrow$ $a^{\phi(n)} \bmod n = 1$

Ahora igualamos $a*x \bmod n = 1$ y $a^{\phi(n)} \bmod n = 1$

$$\therefore \quad a^{\phi(n)} * a^{-1} \bmod n = x \bmod n$$

$$\therefore \quad x = a^{\phi(n)-1} \bmod n$$

El valor x será el inverso de a en el cuerpo n

Nota: Observe que se ha *dividido* por a en el cálculo anterior. Esto se puede hacer porque mcd (a, n) = 1 y por lo tanto hay un único valor inverso en el cuerpo n que lo permite.

Cálculo de inversos con Teorema Euler

Ejemplo

¿Cuál es el inverso de 4 en módulo 9? $\Rightarrow$ inv (4, 9)

Pregunta: ¿Existe a * x mod n = 4 * x mod 9 = 1?

Como mcd (4, 9) = 1 $\Rightarrow$ Sí ... aunque 4 y 9 no son primos.

$\phi(9) = 6 \quad \therefore \quad x = 4^{6-1} \bmod 9 = 7 \quad \Rightarrow \quad 7*4 = 28 \bmod 9 = 1$

Resulta obvio que: inv (4, 9) = 7 e inv (7, 9) = 4

Teorema de Euler para n = p*q

Si el factor a es primo relativo con n y n es el producto de 2 primos, seguirá cumpliéndose el Teorema de Euler también en dichos primos.

Por ejemplo:

Si $n = p*q \Rightarrow \phi(n) = (p-1)(q-1)$

$\forall$ a / mcd {a, (p,q)} = 1

se cumple que:

$a^{\phi(n)} \bmod p = 1$

$a^{\phi(n)} \bmod q = 1$

En el capítulo dedicado a la cifra con clave pública RSA, relacionaremos este tema con el Teorema del Resto Chino.

Ejemplo Teorema de Euler para n = p*q

Sea $n = p*q = 7*11 = 77$

$\phi(n) = (p - 1)(q - 1) = (7 - 1)(11 - 1) = 6*10 = 60$

Si k = 1, 2, 3, ...

Para a = k*7 $a^{\phi(n)} \bmod n = k*7^{60} \bmod 77 = 56$

Para a = k*11 $a^{\phi(n)} \bmod n = k*11^{60} \bmod 77 = 22$

Para $\forall$ a $\neq$ k*7,11 $a^{\phi(n)} \bmod n = a^{60} \bmod 77 = 1$

Y se cumple que:

Para $\forall$ a $\neq$ k*7,11 $a^{\phi(n)} \bmod p = a^{60} \bmod 7 = 1$

$a^{\phi(n)} \bmod q = a^{60} \bmod 11 = 1$

En caso contrario: $a^{\phi(n)} \bmod p = 0$

$a^{\phi(n)} \bmod q = 0$

Teorema de Fermat

Si el cuerpo de trabajo n es un primo p

$\text{mcd}(a, p) = 1 \quad \Rightarrow \quad a^{\phi(p)} \bmod p = 1$

Entonces $a * x \bmod p = 1$ y $a^{\phi(n)} \bmod p = 1$

Además, en este caso $\phi(p) = p-1$ por lo que igualando las dos ecuaciones de arriba tenemos:

$$\therefore \quad a^{\phi(p)} * a^{-1} \bmod p = x \bmod p$$

$$\therefore \quad x = a^{p-2} \bmod p$$

Luego x será e inverso de a en el primo p.

¿Qué hacemos si no se conoce $\phi(n)$?

- Calcular $a^i \bmod n$ cuando los valores de i y a son grandes, se hace tedioso pues hay que utilizar la propiedad de la reducibilidad repetidas veces.
- Si no conocemos $\phi(n)$ o no queremos usar el teorema de Euler/Fermat, siempre podremos encontrar el inverso de a en el cuerpo n usando el

Algoritmo Extendido de Euclides

Es el método más rápido y práctico

Algoritmo Extendido de Euclides AEE

Si mcd (a, n) = 1 $\Rightarrow$ a * x mod n = 1 $\therefore$ x = inv (a, n)

Luego podemos escribir:

$n = C_1 * a + r_1 \qquad a > r_1$

$a = C_2 * r_1 + r_2 \qquad r_1 > r_2$

$r_1 = C_3 * r_2 + r_3 \qquad r_2 > r_3$

... ...

$r_{n-2} = C_n * r_{n-1} + 1 \qquad r_{n-1} > 1$

$r_{n-1} = C_{n+1} * 1 + 0$

Concluye el algoritmo

Si volvemos hacia atrás desde este valor, obtenemos el inverso de a en el cuerpo n.

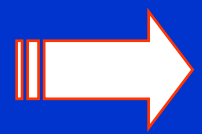
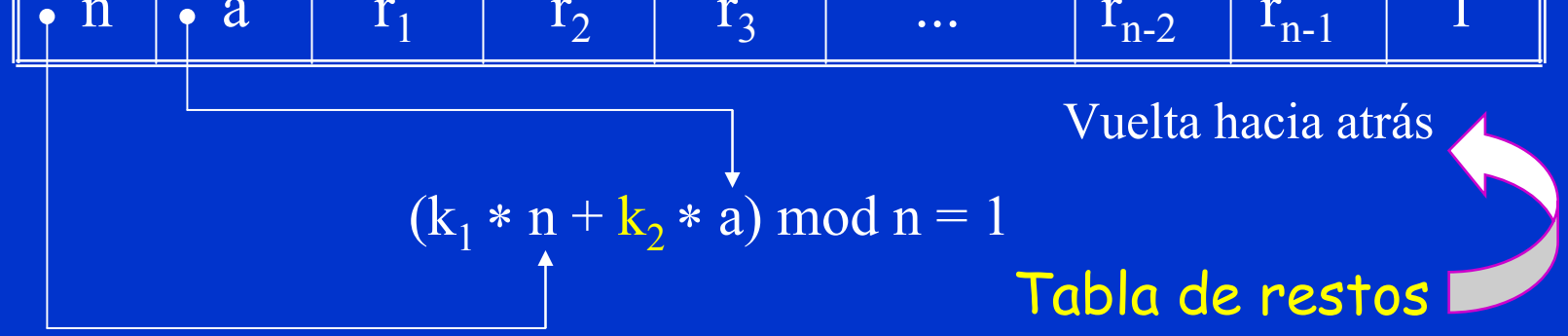


Tabla de restos del AEE

Ordenando por restos desde el valor 1 se llega a una expresión del tipo $(k_1 * n + k_2 * a) \bmod n = 1$, en donde el inverso de a en n lo dará el coeficiente k_2 puesto que $k_1 * n \bmod n = 0$.

	C_1	C_2	C_3	C_4	...	C_{n-1}	C_n	C_{n+1}
n	a	r_1	r_2	r_3	...	r_{n-2}	r_{n-1}	1



Cálculo de inversos con el AEE

Encontrar el inv (9, 25) por el método de restos de Euclides.

a) $25 = 2*9 + 7$

b) $9 = 1*7 + 2$

c) $7 = 3*2 + 1$

d) $2 = 2*1 + 0$

$7 = 25 - 2*9$

$2 = 9 - 1*7$

$1 = 7 - 3*2$

restos

$7 = 25 - 2*9$

$2 = 9 - 1*(25 - 2*9) = 3*9 - 1*25$

$1 = (25 - 2*9) - 3*(3*9 - 1*25)$

$1 = 4*25 - 11*9 \mod 25$

Tabla de Restos

	2	1	3	2	
25	9	7	2	1	0

El inv (9,25) = -11

-11 + 25 = 14

inv (9, 25) = 14

Algoritmo para el cálculo de inversos

Para encontrar x = inv (A, B)

Hacer $(g_0, g_1, u_0, u_1, v_0, v_1, i) = (B, A, 1, 0, 0, 1, 1)$

Mientras $g_i \neq 0$ hacer

- Hacer y_{i+1} = parte entera (g_{i-1}/g_i)
- Hacer $g_{i+1} = g_{i-1} - y_{i+1} * g_i$
- Hacer $u_{i+1} = u_{i-1} - y_{i+1} * u_i$
- Hacer $v_{i+1} = v_{i-1} - y_{i+1} * v_i$
- Hacer i = i+1

Si $(v_{i-1} < 0)$ x = inv (9, 25) = -11+25 = 14

- Hacer $v_{i-1} = v_{i-1} + B$
- Hacer $x = v_{i-1}$

Ejemplo

x = inv (A, B)

x = inv (9, 25)

i	y_i	g_i	u_i	v_i
0	-	25	1	0
1	-	9	0	1
2	2	7	1	-2
3	1	2	-1	3
4	3	1	4	-11
5	2	0	-9	25

Características de inversos en n = 27

Para el alfabeto castellano con mayúsculas (n = 27) tenemos:

x	inv (x, 27)	x	inv (x, 27)	x	inv (x, 27)
1	1	10	19	19	10
2	14	11	5	20	23
4	7	13	25	22	16
5	11	14	2	23	20
7	4	16	22	25	13
8	17	17	8	26	26

$27 = 3^3$ luego no existe inverso para a = 3, 6, 9, 12, 15, 18, 21, 24.

$\text{inv}(x, n) = a \Leftrightarrow \text{inv}(a, n) = x$

$\text{inv}(1, n) = 1$

Inversos en sistemas de cifra clásicos orientados a alfabetos de 27 caracteres.

¿Habrá inversos si mcd (a, n) ≠ 1?

- ¿Pueden existir inversos?
- No, pero...
- Si a * x mod n = b con b ≠ 1 y mcd (a, n) = m, siendo m divisor de b, habrá m soluciones válidas.

Esto no nos sirve en criptografía ... ☹

$6*x \bmod 10 = 4$ $\quad$ $\text{mcd}(6, 10) = 2$

No existe inv (6, 10) pero ... habrá 2 soluciones válidas

$x_1 = 4 \Rightarrow 6*4 \bmod 10 = 24 \bmod 10 = 4$

$x_2 = 9 \Rightarrow 6*9 \bmod 10 = 54 \bmod 10 = 4$

?

Teorema del Resto Chino TRC

Si $n = d_1 * d_2 * d_3 * \ldots * d_t$ con $d_i = p_i^{ei}$ (p primo)

El sistema de ecuaciones:

$$x \bmod d_i = x_i \qquad (i = 1, 2, 3, \ldots t)$$

tiene una solución común en [0, n-1]

$$x = \sum_{i=1}^{t} (n/d_i) * y_i * x_i \bmod n$$

desarrollo

con $y_i = \text{inv}\,[(n/d_i), d_i]$

Ejemplo de aplicación del TRC (1)

Encontrar x de forma que : $12 * x \bmod 3.960 = 36$

Tenemos la ecuación genérica: $a * x_i \bmod d_i = b$

$n = 3.960 \Rightarrow n = 2^3*3^2*5*11 = d_1*d_2*d_3*d_4 = 8*9*5*11$

$a = 12$

$b = 36$

Como $n \Rightarrow d_4$, existirán 4 soluciones de x_i

$a*x_1 \bmod d_1 = b \bmod d_1 \Rightarrow 12*x_1 \bmod 8 = 36 \bmod 8 = 4$

$a*x_2 \bmod d_2 = b \bmod d_2 \Rightarrow 12*x_2 \bmod 9 = 36 \bmod 9 = 0$

$a*x_3 \bmod d_3 = b \bmod d_3 \Rightarrow 12*x_3 \bmod 5 = 36 \bmod 5 = 1$

$a*x_4 \bmod d_4 = b \bmod d_4 \Rightarrow 12*x_4 \bmod 11 = 36 \bmod 11 = 3$

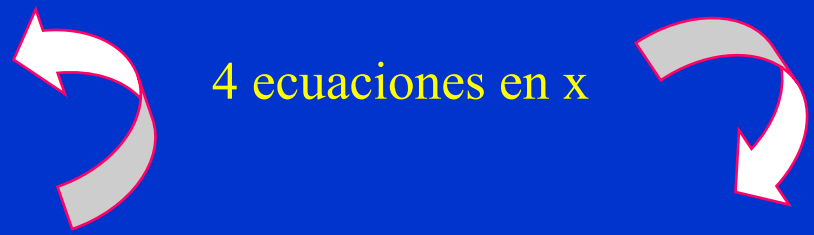
4 ecuaciones en x

Resolviendo para x_i

Ejemplo de aplicación del TRC (2)

$x_1 = 1$ $\quad x_2 = 0$

$x_3 = 3$ $\quad x_4 = 3$



$$12 * x_1 \bmod 8 = 4 \Rightarrow 4 * x_1 \bmod 8 = 4 \Rightarrow x_1 = 1$$

$$12 * x_2 \bmod 9 = 0 \Rightarrow 3 * x_2 \bmod 9 = 0 \Rightarrow x_2 = 0$$

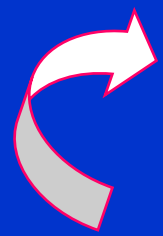
$$12 * x_3 \bmod 5 = 1 \Rightarrow 2 * x_3 \bmod 5 = 1 \Rightarrow x_3 = 3$$

$$12 * x_4 \bmod 11 = 3 \Rightarrow 1 * x_4 \bmod 11 = 3 \Rightarrow x_4 = 3$$

Ejemplo de aplicación del TRC (3)

Resolvemos ahora la ecuación auxiliar del Teorema Resto Chino

$y_i = inv\,[(n/d_i), d_i]$



$y_1 = 7$ $\quad y_2 = 8$

$y_3 = 3$ $\quad y_4 = 7$

$y_1 = inv\,[(n/d_1), d_1] \Rightarrow y_1 = inv[(3960/8),8] = inv\,(495,8)$

$y_2 = inv\,[(n/d_2), d_2] \Rightarrow y_2 = inv[(3960/9),9] = inv\,(440,9)$

$y_3 = inv\,[(n/d_3), d_3] \Rightarrow y_3 = inv[(3960/5),5] = inv\,(792,5)$

$y_4 = inv[(n/d_4), d_4] \Rightarrow y_4 = inv[(3960/11),11] = inv(360,11)$

Ejemplo de aplicación del TRC (4)

$x_1 = 1$ $x_2 = 0$

$x_3 = 3$ $x_4 = 3$

$y_1 = 7$ $y_2 = 8$

$y_3 = 3$ $y_4 = 7$

Aplicando ecuación del Resto Chino para el caso $12 * x \bmod 3.960 = 36$ con $d_1 = 8$, $d_2 = 9$, $d_3 = 5$, $d_4 = 11$:

$$x = \sum_{i=1}^{t} (n/d_i)*y_i*x_i \bmod n$$

$$x = [(n/d_1)y_1x_1 + (n/d_2)y_2x_2 + (n/d_3)y_3x_3 + (n/d_4)y_4x_4]$$

$$x = [495*7*1 + 440*8*0 + 792*3*3 + 360*7*3] \bmod 3.960$$

$$x = [3.465 + 0 + 7.128 + 7.560] \bmod 3.960 = 2.313$$

¿Todo marcha bien en este ejemplo?

¿Es la solución de $12*x \bmod 3.960 = 36$ única?

NO

¿Qué ha sucedido?

Puesto que mcd (a, n) = mcd (12, 3.960) = 12, ya hemos visto en una diapositiva anterior que habrá 12 soluciones válidas.

$x_1 = 3; x_2 = 333; x_3 = 663; x_4 = 993$ ..... $x_8 = \underline{2.313}$ ...

$x_i = 3 + (i-1)*330 \bmod 3.960$ ... hasta llegar a $x_{12} = 3.633$

Observe que x = 2.313, uno de los valores solución, fue el resultado encontrado en el ejercicio anterior.

Otros casos de aplicación del TRC

¿Qué sucede ahora con:

$12*x \bmod 3.960 = 35$?

$12*x \bmod 3.960 = 35$

mcd (a, n) = 12 no es un divisor de b = 35, luego aquí no existe solución.

Teníamos que

$3.960 = 2^3*3^2*5*11$

Encuentre x como ejercicio

$49*x \bmod 3.960 = 1$

¿Qué sucede ahora con:

$49*x \bmod 3.960 = 1$?

Sí existirá x, en este caso es el inverso de 49, y será único ya que $49 = 7*7$ no tiene factores en n.

¿Sólo sirve para esto el TRC?

Calcular el inverso de 49 en el cuerpo 3.960 por medio del Teorema del Resto Chino es algo tedioso ..... ☹ ya lo habrá comprobado ☺.

No obstante, ya habrá comprobado que en este caso el inverso de 49 en el cuerpo 3.960 es x = 889.

¿Para qué sirve entonces este algoritmo?

Entre otras cosas, cuando veamos el sistema de cifra RSA y el tema dedicado a Protocolos Criptográficos, encontraremos una interesante aplicación del Teorema.

Raíz primitiva o generador g de grupo p

- Un generador o raíz primitiva de un número primo p n es aquel que, elevado a todos los restos del cuerpo y reducido módulo n, genera todo el grupo.

  Así, g es un generador si: $\forall \quad 1 \leq a \leq p-1$

  $g^a \bmod p = b$ (con $1 \leq b \leq p-1$, todos los b $\neq$)

Sea p = 3 $\Rightarrow$ CCR = {1,2} (el cero no es solución)

Resto 1: no generará nada porque $1^k \bmod p = 1$

Resto 2: $2^1 \bmod 3 = 2$; $2^2 \bmod 3 = 1$

Luego el 2 es un generador del cuerpo n = 3

¿Cuántas raíces hay en un grupo?

- Existen muchos números dentro del CRR que son generadores del cuerpo ... pero
- Su búsqueda no es fácil ... ¿alguna solución?
- Conociendo la factorización de p-1 $(q_1, q_2, ..., q_n)$ con q_i los factores primos de p-1, diremos que un número g será generador en p si $\forall\ q_i$:

$$g^{(p-1)/qi} \bmod p \neq 1$$

Ejemplo

En cambio...

Si algún resultado es igual a 1, g no será generador.

Búsqueda de raíces primitivas (1)

BÚSQUEDA DE RAÍCES EN EL CUERPO Z_{13}^*

Como $p = 13 \Rightarrow p-1 = 12 = 2^2 * 3$

Luego: $q_1 = 2 \quad q_2 = 3$

Si se cumple $g^{(p-1)/qi} \bmod p \neq 1 \ \forall \ q_i$

entonces g será un generador de p

Generadores en Z_{13}

g: 2,

$2^{(13-1)/2} \bmod 13 = 2^6 \bmod 13 = 12$

$2^{(13-1)/3} \bmod 13 = 2^4 \bmod 13 = 3$

Resto 2

El resto 2 es generador

$3^{(13-1)/2} \bmod 13 = 3^6 \bmod 13 = 1$

$3^{(13-1)/3} \bmod 13 = 3^4 \bmod 13 = 3$

Resto 3

El resto 3 no es generador

Búsqueda de raíces primitivas (2)

Generadores en Z_{13} g: 2, 6, 7,

$4^{(13-1)/2} \bmod 13 = 4^6 \bmod 13 = 1$ Resto 4

$4^{(13-1)/3} \bmod 13 = 4^4 \bmod 13 = 9$ *El resto 4 no es generador*

$5^{(13-1)/2} \bmod 13 = 5^6 \bmod 13 = 12$ Resto 5

$5^{(13-1)/3} \bmod 13 = 5^4 \bmod 13 = 1$ *El resto 5 no es generador*

$6^{(13-1)/2} \bmod 13 = 6^6 \bmod 13 = 12$ Resto 6

$6^{(13-1)/3} \bmod 13 = 6^4 \bmod 13 = 9$ *El resto 6 es generador*

$7^{(13-1)/2} \bmod 13 = 7^6 \bmod 13 = 12$ Resto 7

$7^{(13-1)/3} \bmod 13 = 7^4 \bmod 13 = 9$ *El resto 7 es generador*

Búsqueda de raíces primitivas (3)

Generadores en Z_{13} g: 2, 6, 7, 11

$8^{(13-1)/2} \bmod 13 = 8^6 \bmod 13 = 12$
$8^{(13-1)/3} \bmod 13 = 8^4 \bmod 13 = 1$

Resto 8

El resto 8 no es generador

$9^{(13-1)/2} \bmod 13 = 9^6 \bmod 13 = 1$
$9^{(13-1)/3} \bmod 13 = 9^4 \bmod 13 = 9$

Resto 9

El resto 9 no es generador

$10^{(13-1)/2} \bmod 13 = 10^6 \bmod 13 = 1$
$10^{(13-1)/3} \bmod 13 = 10^4 \bmod 13 = 3$

Resto 10

El resto 10 no es generador

$11^{(13-1)/2} \bmod 13 = 11^6 \bmod 13 = 12$
$11^{(13-1)/3} \bmod 13 = 11^4 \bmod 13 = 3$

Resto 11

El resto 11 es generador

Búsqueda de raíces primitivas (4)

Generadores en Z_{13} g: 2, 6, 7, 11

$12^{(13-1)/2} \bmod 13 = 12^6 \bmod 13 = 1$ Resto 12

$12^{(13-1)/3} \bmod 13 = 12^4 \bmod 13 = 1$ *El resto 12 no es generador*

La tasa de generadores en el grupo p será aproximadamente $\tau = \phi(p-1)/(p-1)$. Por lo tanto por lo general el 30% de los elementos del Conjunto Reducido de Restos de p será un generador en p.

$\tau = \phi(12)/12$

$\tau = 4/12 = 1/3$

Generadores en cuerpos de primos seguros

Un número primo p se dice que es un primo seguro o primo fuerte si: $p = 2*p' + 1$ (con p' también primo).

Por ejemplo:

Si $p' = 11$, luego $p = 2*11 + 1 = 23$ (es primo y es seguro)

En este caso la tasa de números generadores del cuerpo será mayor que en el caso anterior (con p = 13 era del 30%).

Probabilidad: $\tau_{pseguro} = \phi(p-1)/p-1 \approx ½$

Casi la mitad de los números del grupo serán generadores en p.

⇒ Comprobación

Comprobación de generadores en p = 2p'+1

p' = 11; 2p' = 22; p = 2p' + 1 = 23 *primo seguro*

Como 2p' = p - 1 existirán:

$\phi(p')$ = [p'- 1] elementos de orden (p') en el CRR

$\phi(11) = 10 = \{1,2,3,4,5,6,7,8,9,10\}$

$\phi(2p')$ = [p'- 1] elementos de orden (p-1) en el CRR

$\phi(22) = 10 = \{1,3,5,7,9,13,15,17,19,21\}$

$\tau = (p'- 1)/(p-1) = (p'- 1)/2p' \approx ½$

Sigue

Comprobación de generadores en p = 2p'+1

Usando la ecuación $g^{(p-1)/qi} \bmod p$

En este caso con $q_1 = 2$ y $q_2 = 11$

$g^{(23-1)/2} \bmod 23 = g^{11} \bmod 23$

$g^{(23-1)/11} \bmod 23 = g^{2} \bmod 23$

Encontramos los siguientes 10 generadores en p = 23

{5, 7, 10, 11, 14, 15, 17, 19, 20, 21}

Prácticamente la mitad de los valores de CRR que en este caso es igual a 23 – 1 = 22.

Utilidad de la raíz primitiva en criptografía

¿Para qué sirve conocer la raíz primitiva de p?

- La utilidad de este concepto en criptografía lo veremos cuando se estudien los sistemas de clave pública y, en particular, el protocolo de intercambio de claves de Diffie y Hellman.
- También se recurrirá a esta propiedad de los primos cuando estudiemos la firma digital según estándar DSS (ElGamal).

La exponenciación en la cifra asimétrica

- Una de las más interesantes aplicaciones de la matemática discreta en criptografía es la cifra asimétrica en la que la operación básica es una exponenciación $A^B \bmod n$, en donde n es un primo o un producto de primos grandes.
- Esta operación se realizará en el intercambio de clave y en la firma digital.
- ¿Cómo hacer estos cálculos de forma rápida y eficiente, sin tener que aplicar la reducibilidad? Tenemos una solución aplicando un algoritmo de exponenciación rápida.

Un método de exponenciación rápida

- En $x^y \bmod n$ se representa el exponente *y* en binario.
- Se calculan los productos x^{2^j} con j = 0 hasta n-1, siendo n el número de bits que representan el valor *y* en binario.
- Sólo se toman en cuenta los productos en los que en la posición j del valor *y* en binario aparece un 1.

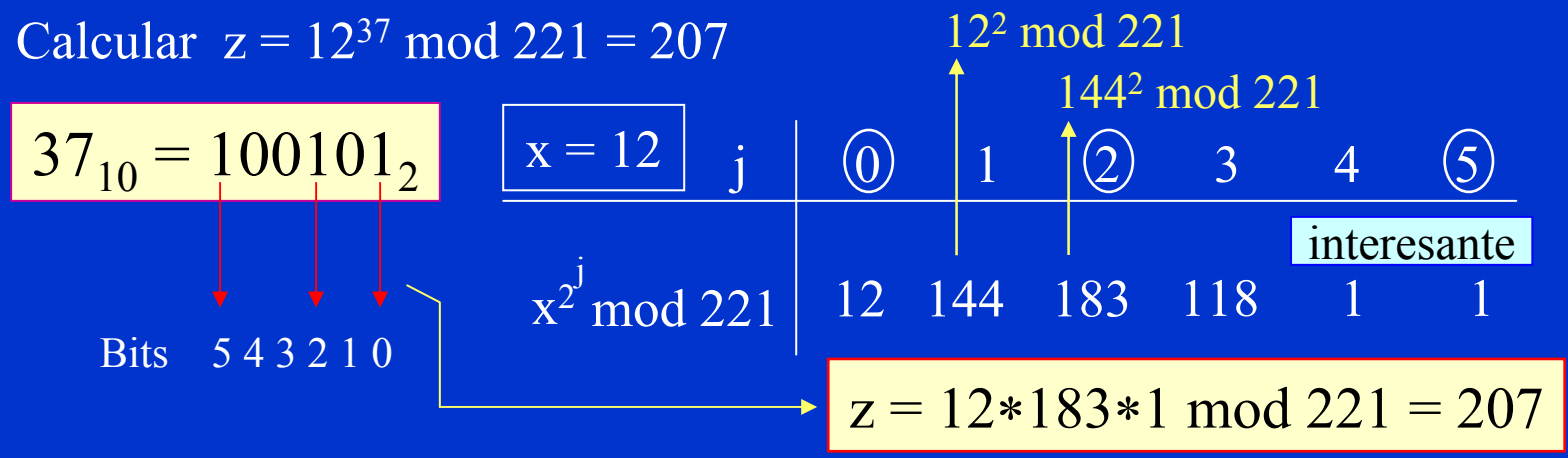
Ejemplo

Calcular $z = 12^{37} \bmod 221 = 207$

12^{37} es un número de 40 dígitos:

8505622499821102144576131684114829934592

Ejemplo de exponenciación rápida



En vez de 36 multiplicaciones y sus reducciones módulo 221 en cada paso ... 72 operaciones...

Hemos realizado cinco multiplicaciones (para j = 0 el valor es x) con sus reducciones módulo 221, más dos al final y su correspondiente reducción. Un ahorro superior al 80% ☺.

Algoritmo de exponenciación rápida

Hallar $x = A^B \bmod n$

- Obtener representación binaria del exponente B de k bits:

  $B_2 \rightarrow b_{k-1}b_{k-2}...b_i...b_1b_0$
- Hacer x = 1
- Para i = k-1, ..., 0 hacer

  $x = x^2 \bmod n$

  Si ($b_i = 1$) entonces

  $x = x*A \bmod n$

Ejemplo: calcule $19^{83} \bmod 91 = 24$

$83_{10} = 1010011_2 = b_6b_5b_4b_3b_2b_1b_0$

x = 1

i=6	b_6=1	$x = 1^2*19 \bmod 91$	= 19	x = 19
i=5	b_5=0	$x = 19^2 \bmod 91$	= 88	x = 88
i=4	b_4=1	$x = 88^2*19 \bmod 91$	= 80	x = 80
i=3	b_3=0	$x = 80^2 \bmod 91$	= 30	x = 30
i=2	b_2=0	$x = 30^2 \bmod 91$	= 81	x = 81
i=1	b_1=1	$x = 81^2*19 \bmod 91$	= 80	x = 80
i=0	b_0=1	$x = 80^2*19 \bmod 91$	= 24	x = 24

19^{83} = 1,369458509879505101557376746718e+106 (calculadora Windows)

Cálculos en campos de Galois (GF)

- Cuando trabajamos en un cuerpo primo p, sabemos que se asegura la existencia de un único inverso multiplicativo. En este caso se dice que estamos trabajando en Campos de Galois GF(p).
- Algunos usos en criptografía:
  - Sistemas de clave pública cuando la operación de cifra es $C = M^e \bmod p$ (cifrador ElGamal) o bien RSA usando el Teorema del Resto Chino para descifrar.
  - Aplicaciones en $GF(q^n)$, polinomios módulo q y de grado n: $a(x) = a_{n-1}*x^{n-1} + a_{n-2}*x^{n-2} + \ \ldots + a_1*x + a_0$: Cifrador de flujo A5, RIJNDAEL, curvas elípticas.

Campos de Galois del tipo $GF(q^n)$

$$a(x) = a_{n-1}*x^{n-1} + a_{n-2}*x^{n-2} + \ \ldots + a_1*x + a_0$$

- Es un polinomio de grado n-1 o menor.
- Los elementos a_i son parte del CCR del módulo q.
- Cada elemento a(x) es un resto módulo p(x), siendo p(x) un polinomio irreducible de grado n (que no puede ser factorizado en polinomios de grado menor que n).
- $GF(2^n)$ es interesante porque CCR(2) = {0, 1} $\Rightarrow$ bits.
- $GF(2^3)$ $\Rightarrow$ 8 elementos o restos polinómicos que son: 0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1, los 8 restos de un polinomio de grado n-1 (n = 3).

Suma en campos de Galois $GF(2^n)$ $\oplus$

Si el módulo de trabajo es 2 (con restos = bits 0 y 1), las operaciones suma y resta serán un OR Exclusivo:

$0 \oplus 1 \bmod 2 = 1$ $\quad$ $1 \oplus 0 \bmod 2 = 1$

$0 \oplus 0 \bmod 2 = 0$ $\quad$ $1 \oplus 1 \bmod 2 = 0$

$CG(2^2)$

$\oplus$	0	1	x	x+1
0	0	1	x	x+1
1	1	0	x+1	x
x	x	x+1	0	1
x+1	x+1	x	1	0

Restos: 0, 1, x, x+1

Como los resultados deberán pertenecer al cuerpo, aplicaremos Reducción por Coeficientes:

Por ejemplo:

$x + (x + 1) = 2x + 1 \bmod 2 = 1$

$1 + 1 = 2 \bmod 2 = 0$

Producto en campos de Galois $GF(2^n)$ $\otimes$

La operación multiplicación puede entregar elementos que no pertenezcan al cuerpo, potencias iguales o mayores que n $\Rightarrow$ Reducción por Exponente.

$CG(2^2)$

$\otimes$	0	1	x	x+1
0	0	0	0	0
1	0	1	x	x+1
x	0	x	x+1	1
x+1	0	x+1	1	x

Restos: 0, 1, x, x+1

Sea el polinomio irreducible de grado n = 2, $p(x) = x^2 + x + 1$

Luego: $x^2 = x + 1$

Por ejemplo:

$(x + 1)*(x + 1) = x^2 + 2x + 1$

$(x + 1)*(x + 1) = (x + 1) + 2x + 1$

$(x + 1)*(x + 1) = 3x + 2 \bmod 2 = x$

Operaciones con campos de Galois en AES

- La suma y multiplicación de polinomios dentro de un cuerpo binario descritas en las dispositivas anteriores, conforman las operaciones básicas del algoritmo de cifra Advanced Encryption Algorithm AES, que con el nombre RIJNDAEL es el estándar mundial desde finales de 2001, desplazando al ya viejo DES.
- En este caso, se trabaja con 8 bits por lo que las operaciones se realizan en $GF(2^8)$. En el capítulo 10 sobre sistemas de cifra con clave secreta encontrará ejemplos de suma y multiplicación polinómica dentro de este cuerpo binario.

Fin del Tema 5

Cuestiones y ejercicios (1 de 3)

1. ¿Qué significa para la criptografía el homomorfismo de los enteros?
2. Si una función de cifra multiplica el mensaje por el valor a dentro del cuerpo n, para qué nos sirve conocer el inverso de a en n?
3. En un cuerpo de cifra n, ¿existen siempre los inversos aditivos y los inversos multiplicativos? ¿Debe cumplirse alguna condición?
4. Si en un cuerpo n el inverso de a es a^{-1}, ¿es ese valor único?
5. Cifraremos en un cuerpo n = 131. ¿Cuál es el conjunto completo de restos? ¿Cuál es el conjunto reducido de restos?
6. Para cifrar un mensaje M = 104 debemos elegir el cuerpo de cifra entre el valor n = 127 y n = 133, ¿cuál de los dos usaría y por qué?
7. ¿Qué nos dice la función $\phi(n)$ de Euler?
8. ¿Qué papel cumple el algoritmo extendido de Euclides en la criptografía? ¿Por qué es importante? ¿En qué se basa?

Cuestiones y ejercicios (2 de 3)

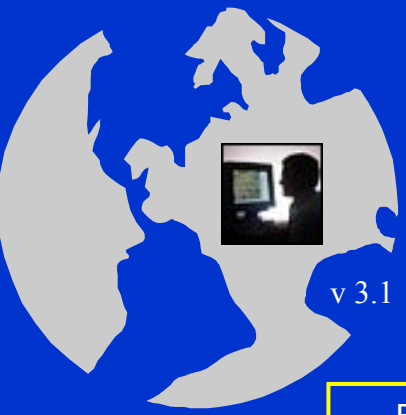
9. Si en el cuerpo n = 37 el inv (21, 37) = 30, ¿cuál es el inv (30, 37)?
10. Usando el algoritmo extendido de Euclides calcule los siguientes inversos: inv (7, 19); inv (21, 52), inv (11, 33), inv (41, 43).
11. ¿Cuántas soluciones x_i hay a la expresión 8*x mod 20 = 12? Explique lo que sucede. ¿Tiene esto interés en criptografía?
12. ¿Qué viene a significar el Teorema del Resto Chino? Aunque aún no lo ha estudiado, ¿le ve alguna utilidad en criptografía?
13. Calcule inv (49, 390) usando el Teorema del Resto Chino.
14. Defina lo que es una raíz primitiva o generador de un cuerpo. ¿Es necesario que ese cuerpo sea un primo?
15. ¿Cuántos generadores podemos esperar en el cuerpo n = 17? Y si ahora n = 7, ¿cuántos generadores habrá? Compruébelo calculando todos los exponentes del conjunto completo de restos de n = 7.

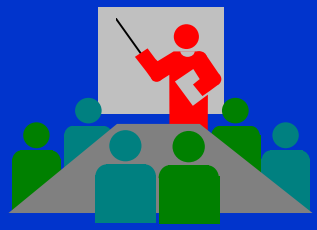
Cuestiones y ejercicios (3 de 3)

16. ¿Cómo se define un primo seguro? ¿Cuántos generadores tiene?
17. A partir de los valores p' = 13, p' = 17, p' = 19 y p' = 23 queremos obtener un primo seguro, ¿con cuál o cuáles de ellos lo logramos?
18. Usando el algoritmo de exponenciación rápida calcule los siguientes valores: 23^{32} mod 51; 100^{125} mod 201; $1.000^{100.000}$ mod 2.500.
19. Compruebe los resultados con la calculadora de Windows. ¿Qué sucede para números muy grandes como el último?
20. ¿Cuántas operaciones básicas se han hecho en cada caso y en cuánto se ha optimizado el cálculo?
21. En GF(2^n) reduzca por coeficientes $5x^5 + x^4 + 2x^3 + 3x^2 + 6x + 2$.
22. Reduzca $(x^3 + 1)(x^2 + x + 1)$ por exponente en GF(2^n) usando como polinomio primitivo $p(x) = x^4 + x + 1$, es decir $x^4 = x + 1$.

Tema 6
Teoría de la Complejidad Algorítmica

Seguridad Informática y Criptografía





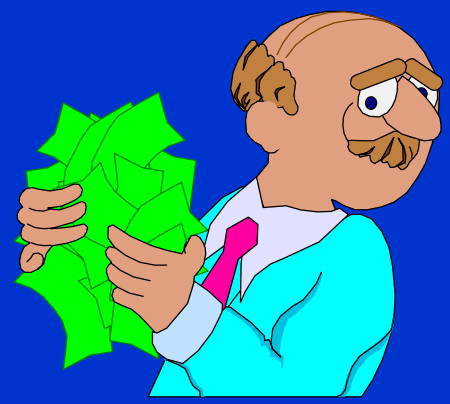
Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 28 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Nota del autor

El contenido de este tema corresponde sólo a una breve introducción a la complejidad de los algoritmos, con el objeto de que el lector pueda hacerse una idea de la importancia de este tema en el análisis y diseño de los algoritmos de cifra y firma digital que se verán en este curso. La fortaleza de estos algoritmos y de los protocolos que los incluyen dependerá de la complejidad asociada al criptoanálisis o ataque de los mismos.



Introducción a la teoría de la complejidad

La teoría de la complejidad de los algoritmos nos permitirá conocer si un algoritmo tiene fortaleza y tener así una idea de su vulnerabilidad computacional.

Complejidad Computacional

Los algoritmos se clasifican según el tiempo de ejecución y en función del tamaño de la entrada.

- Complejidad Polinomial ☺
- Complejidad Exponencial ☹

Esto dará lugar a tipos de "problemas" que nos interesarán.

Operaciones bit en la suma

Si deseamos sumar dos números binarios n y m, ambos de k bits realizaremos k operaciones bit puesto que cada operación básica con los dígitos de una columna es una operación bit.

* Recuerde que 0+0 = 0, 0+1=1, 1+0 = 1, 1+1 = 0 con bit 1 de acarreo. Si un número tiene menos bits, se rellena con ceros por la izquierda.

Ejemplo: encontrar el número de operaciones bit necesarias en la suma en binario de 13+7 $\Rightarrow$ 1101 + 0111 (de k = 4 bits)

```
    1 1 1 1    (acarreo)
      1 1 0 1
+     0 1 1 1
  -------------
  1 0 1 0 0
```

Cada operación básica que hacemos con una columna se conoce como operación bit, luego necesitamos k = 4 operaciones bit.

Operaciones bit en la multiplicación

Para la multiplicación de un número n de k bits por un número m de h bits, el número de operaciones bit será igual a 2∗k∗h. Suponemos que k ≥ h.

* Recuerde que 0x0 = 0, 0x1=0, 1x0 = 0, 1x1 = 1.

Ejemplo: encontrar el número de operaciones bit necesarias en la multiplicación en binario 10x5 ⇒ 1010 x 101 (4 y 3 bits)

```
      1 0 1 0 x 1 0 1
      1 0 1 0
    0 0 0 0
+ 1 0 1 0          (procedemos ahora a sumar)
  1 1 0 0 1 0
```

Como cada operación básica entre dos bits es una operación bit, hemos realizado h∗k = 3∗4 multiplicaciones y luego k∗h = 4∗3 sumas, es decir en total 2∗k∗h = 24 operaciones bit.

La función O(n)

Las operaciones dependerán del tamaño de la entrada por lo que esta complejidad se expresará en términos del tiempo T necesario para el cálculo del algoritmo y del espacio S que utiliza en memoria, y se expresará mediante una función *f* (n), donde n es el tamaño de la entrada.

Esta función será una aproximación pues el resultado exacto dependerá de la velocidad del procesador.

$$f(n) = O(g(n))$$

Ejemplo

$$f = O(n) \quad ssi \quad \exists\, c_o, n_o \,/\, f(n) \leq c_o * g(n)$$

Complejidad de una función f(n)

Si $f(n) = 4n^2 + 2n + 5$ ¿$f = O(n^2)$?

¿se cumple que $c_o * g(n) = c_o * n^2 \geq f(n)$? Sea $c_o = 6$

c_o	n_o	$c_o n_o^2$	$f(n) = 4n^2 + 2n + 5$	¿$c_o * n^2 \geq f(n)$?
6	1	6	11	No
6	2	24	25	No
6	3	54	38	Sí
6	4	96	77	Sí

Se cumple siempre

Luego, la complejidad de $f(n)$ es exponencial.

Tiempos de ejecución

En la expresión O(n) aparecerá el término que domina al crecer el valor de n.

- El tiempo de ejecución de un algoritmo T_1 que realiza $2n+1$ operaciones es de tipo $O(n)$; uno T_2 que realiza $3n^2+n+3$ operaciones será de tipo $O(n^2)$, etc.
- Para realizar la suma de la diapositiva anterior necesitamos $O(n) = O(\log n)$ operaciones bit y para el caso de la multiplicación, éstas serán $O(n*m) = O(\log n * \log m)$ operaciones bit.

+ Operación binaria: n+m (de k bits cada uno)

∗ Operación binaria: n∗m (de k y h bits respectivamente)

Algoritmos de complejidad lineal

- Un algoritmo se dice que tiene tiempo de ejecución polinomial si éste depende polinómicamente del tamaño de la entrada.
- Si la entrada es de tamaño n y t es un entero, el número de operaciones bit será $O(\log^t n)$.

Ejemplos

Si t = 1, el sistema es lineal → Suma

Si t = 2, el sistema es cuadrático → Producto

Si t = 3, el sistema es cúbico → mcd Euclides

Ejemplo de complejidad lineal

Ejemplo: El tiempo de ejecución de un algoritmo es $O(\log^3 n)$. Si doblamos la entrada, ¿en cuánto aumenta este tiempo?

Solución: En el primer caso el tiempo es $O(\log^3 n)$ y en el segundo $O(\log^3 2n)$. Luego para este sistema lineal el tiempo se incrementará en $\log^3 2$ operaciones bit.

Estos son los denominados problemas fáciles y son los que involucrarán un proceso de cifra y descifrado (o firma) por parte del o de los usuarios autorizados.

Algoritmos de complejidad exponencial

- Un algoritmo se dice que tiene tiempo de ejecución exponencial si éste depende exponencialmente del tamaño de la entrada.
- Si la entrada es de tamaño n y t es un entero, el número de operaciones bit será $O(n^t)$.

Para t = 2, será exponencial de orden 2

Para t = 3, será exponencial de orden 3

Ejemplo

Ejemplo de complejidad exponencial

Ejemplo: El tiempo de ejecución de un algoritmo es $O(n^3)$. Si doblamos la entrada, ¿en cuánto aumenta este tiempo?

Solución: En el primer caso el tiempo es $O(n^3)$ y en el segundo $O(2n^3) = O(8n^3)$. Para este sistema exponencial el tiempo se incrementará en 8 operaciones bit.

Estos son los denominados problemas difíciles y son a los que deberá enfrentarse un criptoanalista o atacante que desea romper una cifra o la clave de un usuario.

Comparativas de complejidad

- Los algoritmos polinómicos y exponenciales se comparan por su complejidad $O(n^t)$.
  - Polinómico constante $\Rightarrow$ $O(1)$
  - Polinómico lineal $\Rightarrow$ $O(n)$
  - Polinómico cuadrático $\Rightarrow$ $O(n^2)$
  - Polinómico cúbico $\Rightarrow$ $O(n^3)$ ... etc.
  - Exponencial $\Rightarrow$ $O(d^{h(n)})$

    donde d es una constante y $h(n)$ un polinomio

Si suponemos un ordenador capaz de realizar 10^9 instrucciones por segundo se tiene el cuadro:

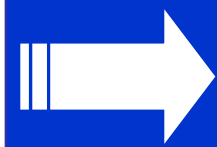


Tabla comparativa de tiempos

Entrada	O(n)	$O(n^2)$	$O(n^3)$	$O(2^n)$
n = 10	10^{-8} seg	10^{-7} seg	10^{-6} seg	10^{-6} seg
$n = 10^2$	10^{-7} seg	10^{-5} seg	10^{-3} seg	$4*10^{13}$ años
$n = 10^3$	10^{-6} seg	10^{-3} seg	1 seg	Muy grande

Incrementos de un orden de magnitud

Computacionalmente imposible

Entrada/10^9: Para n = 100 $\Rightarrow$ $O(n^2) = 100^2/10^9 = 10^{-5}$ seg

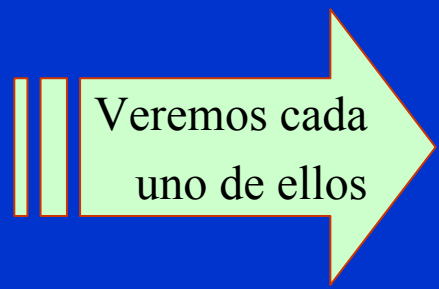
Problemas de tipo NP

En criptografía nos interesan las funciones f(x) de un solo sentido, es decir:

- Fácil calcular f(x) pero muy difícil calcular $f^{-1}(x)$ salvo que conozcamos un secreto o trampa

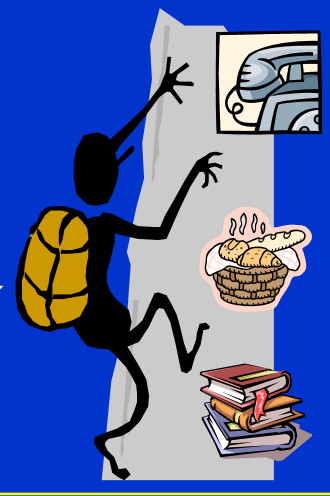
Porque dan lugar a problemas tipo NP, polinomiales no deterministas, computacionalmente difíciles de tratar.

- Problema de la mochila
- Problema de la factorización
- Problema del logaritmo discreto
- Otros ...



El problema de la mochila

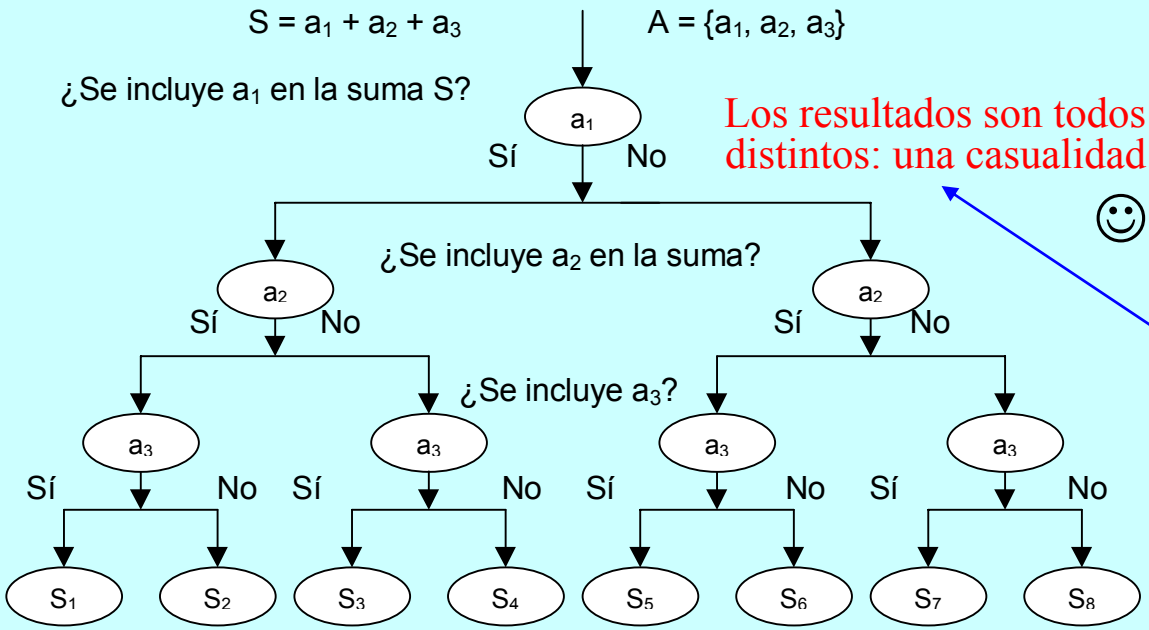
- Es un problema de tipo NP en el que el algoritmo debe realizar en cada paso una selección iterativa entre diferentes opciones.



Enunciado:

Dada una mochila de determinadas dimensiones de alto, ancho y fondo, y un conjunto de elementos de distintos tamaños menores que ella y de cualquier dimensión, ... ¿es posible llenar la mochila (completa) con distintos elementos de ese conjunto sin repetir ninguno de ellos?

Ejemplo del problema de la mochila



$S_1 = a_1+a_2+a_3$ $S_2 = a_1+a_2$ $S_3 = a_1+a_3$ $S_4 = a_1$

$S_5 = a_2+a_3$ $S_6 = a_2$ $S_7 = a_3$ $S_8 = \varnothing$

Sea una mochila con 4 elementos {2, 4, 9, 10}

¿Cuántas sumas posibles hay?

Solución: $2^4 = 16$

∅, 2, 4, 6, 9, 10, 11, 12, 13, 14, 15, 16, 19, 21, 23, 25.

Repita con {2, 4, 6, 10}

Hemos tenido que evaluar $2^3 = 8$ valores $\Rightarrow$ (carácter exponencial)

Interés de las mochilas en criptografía

¿Por qué tiene interés este problema en criptografía?

a) Es de tipo NP completo: su resolución por lo general implica una complejidad exponencial. Luego, será difícil de atacar o criptoanalizar.

b) Existe un caso en el que la resolución es lineal y, si la solución existe, es única. Se da si $A = \{a_1, a_2, a_3, .., a_n\}$ está ordenado de menor a mayor y en donde cada a_i es mayor que la suma de los a_j que le preceden.

Esto dará lugar a los criptosistemas de mochila tramposa que veremos en un próximo capítulo.

El problema de la factorización PFNG

Dado un número n que es el resultado del producto de dos primos n = p*q, se pide encontrar estos factores.

- Cuando el valor n es muy grande, el Problema de la Factorización de Números Grandes PFNG se vuelve computacionalmente intratable.
- No obstante, el caso inverso, dado dos números p y q, encontrar el resultado p*q = n, se trata de un problema de tipo polinomial.
- Este problema se usará en la generación del par de claves del sistema de cifra con clave pública RSA.

Un ejemplo del PFNG

➲ Cálculo fácil o polinomial (función directa)

Calcule "a mano" los siguientes productos de dos primos y tome el tiempo aproximado que tarda en la operación:

a) 13*31 b) 113*131 c) 1.013*1.031 *calcule...*☺

No vale usar calculadora...

¿A qué conclusiones puede llegar ahora?

➲ Cálculo difícil o no polinomial (función inversa)

Usando la criba de Eratóstenes, factorice en dos primos los siguientes números y vuelva a tomar el tiempo empleado:

a) 629 b) 17.399 c) 1.052.627 *calcule...*☹

En el caso a) son primos de 2 dígitos, en c) de 3 y en d) de 4.

Solución al ejemplo anterior

➲ Cálculo fácil o polinomial

a) 13*31 = 403 b) 113*131 = 14.803 c) 1013*1031 = 1.044.403

A medida que aumenta el tamaño de la entrada, el tiempo de cálculo aumenta proporcionalmente con los dígitos.

➲ Cálculo difícil o no polinomial

a) 629 b) 17.399 c) 1.052.627

Un computador experimentará lo mismo....

Aquí resulta evidente que el tiempo de cálculo (da igual que el algoritmo sea éste u otro más depurado y eficaz) aumenta mucho al incrementar en un dígito los números en cuestión.

Solución: a), b) y c) son el producto de los números primos inmediatamente superiores a los de arriba (ver tabla en SItema18).

El problema del logaritmo discreto PLD

Dado un par de enteros α y β que pertenecen al Campo de Galois GF(p), se pide encontrar un entero x de forma que $x = \log_{\alpha} \beta \bmod p$.

Si el valor p es muy grande, el Problema del Logaritmo Discreto PLD es computacionalmente intratable.

No obstante, el caso inverso, dado dos números α y x, encontrar $\beta = \alpha^x \bmod p$ es un problema polinomial.

Este problema se usará en la creación de las claves del sistema de cifra con clave pública ElGamal y en el protocolo de intercambio de clave de Diffie y Hellman.

Un ejemplo del PLD

➲ Cálculo fácil o polinomial (función directa)

Calcule "a mano" las siguientes exponenciaciones mod p y tome el tiempo aproximado que tarda en la operación:

a) $5^4 \bmod 7$ b) $8^{17} \bmod 41$ c) $92^{11} \bmod 251$

$$5^4 = 625$$
$$8^{17} = 2.251.799.813.685.248$$
$$92^{11} = 3.996.373.778.857.415.671.808$$

Nota: Haciendo uso de la propiedad de reducibilidad del capítulo 5, podrá reducir significativamente el tiempo de cálculo. No obstante, este tiempo será de tipo polinomial según el tamaño de la entrada.

Solución al ejemplo anterior

➲ Cálculo difícil o no polinomial (función inversa)

Aunque existen varios algoritmos para este tipo de cálculos (al igual que para la factorización) use la fuerza bruta que se explica a continuación para encontrar los siguientes valores y vuelva a tomar el tiempo empleado:

a) $\log_5 2 \bmod 7$ b) $\log_8 39 \bmod 41$ c) $\log_{92} 217 \bmod 251$

Aplicando fuerza bruta en el 1er caso (la base elevada a todos los restos de p) al final se obtiene que $\log_5 2 \bmod 7 = 4$.

$5^1 \bmod 7 = 5$ $5^2 \bmod 7 = 4$ $5^3 \bmod 7 = 6$

$5^4 \bmod 7 = 2$ $5^5 \bmod 7 = 3$ $5^6 \bmod 7 = 1$

En media deberá recorrer la mitad del espacio... ☹

Logaritmo discreto con α generador

$\log_2 1 \bmod 13 = 0$	$\log_2 2 \bmod 13 = 1$	$\log_2 3 \bmod 13 = 4$
$\log_2 4 \bmod 13 = 2$	$\log_2 5 \bmod 13 = 9$	$\log_2 6 \bmod 13 = 5$
$\log_2 7 \bmod 13 = 11$	$\log_2 8 \bmod 13 = 3$	$\log_2 9 \bmod 13 = 8$
$\log_2 10 \bmod 13 = 10$	$\log_2 11 \bmod 13 = 7$	$\log_2 12 \bmod 13 = 6$

Es decir

$2^0 \bmod 13 = 1$	$2^1 \bmod 13 = 2$	$2^2 \bmod 13 = 4$
$2^3 \bmod 13 = 8$	$2^4 \bmod 13 = 3$	$2^5 \bmod 13 = 6$
$2^6 \bmod 13 = 12$	$2^7 \bmod 13 = 11$	$2^8 \bmod 13 = 9$
$2^9 \bmod 13 = 5$	$2^{10} \bmod 13 = 10$	$2^{11} \bmod 13 = 7$

Se cumplirá además que $a^{p-1} \bmod p = a^0 \bmod p = 1$.

Logaritmo discreto con α no generador

En p=13, el 2 es generador, pero no así el número 3...

Luego

$3^0 \bmod 13 = 1$ $3^1 \bmod 13 = 3$ $3^2 \bmod 13 = 9$

$3^3 \bmod 13 = 1$ $3^4 \bmod 13 = 3$ $3^5 \bmod 13 = 9$

$3^6 \bmod 13 = 1$ $3^7 \bmod 13 = 3$ $3^8 \bmod 13 = 9$

$3^9 \bmod 13 = 1$ $3^{10} \bmod 13 = 3$ $3^{11} \bmod 13 = 9$

$\log_3 1 \bmod 13 = 0$ $\log_3 2 \bmod 13 = NE$ $\log_3 3 \bmod 13 = 1$

$\log_3 4 \bmod 13 = NE$ $\log_3 5 \bmod 13 = NE$ $\log_3 6 \bmod 13 = NE$

$\log_3 7 \bmod 13 = NE$ $\log_3 8 \bmod 13 = NE$ $\log_3 9 \bmod 13 = 2$

$\log_3 10 \bmod 13 = NE$ $\log_3 11 \bmod 13 = NE$ $\log_3 12 \bmod 13 = NE$

NE = no existe

¿Hay más funciones NP?

Existen otros problemas matemáticos que dan lugar a problemas del tipo NP basados en estas funciones unidireccionales (one way functions) pero las dos últimas funciones vistas –factorización de números grandes y logaritmo discreto- son las que más uso tienen, de momento, en la criptografía.

Algunos de ellos se presentarán en el Tema dedicado a los Protocolos Criptográficos.

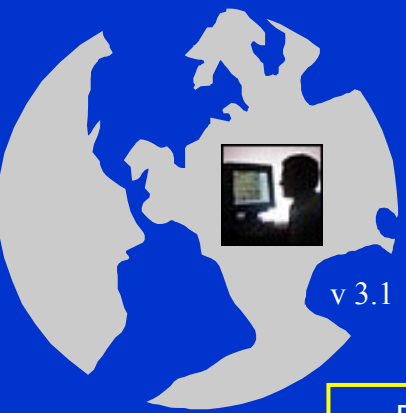
Fin del Tema 6

Cuestiones y ejercicios

1. Deseamos sumar de forma binaria el número 15 (1111) y el número 10 (1010), ambos de k = 4 bits. Haga la suma binaria y verifique que el número de operaciones bit desarrolladas es k = 4.
2. Si multiplicamos en binario 1010*11, donde k = 4 bits y h = 2 bits, compruebe que el número de operaciones bit realizadas es 2*k*h.
3. ¿Por qué son interesantes los problemas de tipo NP en criptografía?
4. Defina el problema de la mochila y su posible utilización en un sistema de cifra.
5. Factorice mentalmente el valor n = 143. Intente hacer lo mismo para n = 1.243. ¿Qué opina ahora del problema de la factorización?
6. A partir de la ecuación $\beta = x^{\alpha} \bmod p$, defina el problema del logaritmo discreto. ¿Qué utilidad tiene en criptografía?

Tema 7
Sistemas de Cifra Clásicos

Seguridad Informática y Criptografía

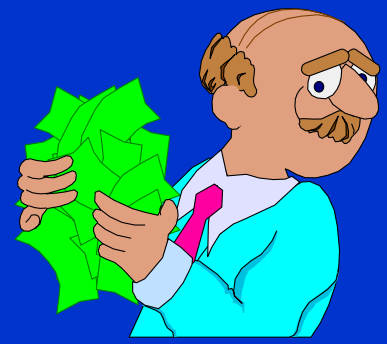


Material Docente de
Libre Distribución

Ultima actualización:17/03/03
Archivo con 40 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Nota del autor



El contenido de este tema corresponde a unos primeros apuntes. No tiene la importancia de los demás temas tratados en este libro, pero si lo desea puede encontrar una explicación más detallada sobre la historia de la criptografía y de los sistemas de cifra clásicos en el libro guía de la asignatura comentado en la presentación de estos apuntes o, en su caso, usando el Libro Electrónico de Criptografía Clásica, software de libre distribución que puede descargar desde Internet como se comenta al final del archivo.

Clasificación histórica de criptosistemas

Los criptosistemas pueden clasificarse según:

a) Su relación con la Historia en:

- Sistemas Clásicos y Sistemas Modernos

No es ésta ni mucho menos la mejor clasificación desde el punto de vista de la ingeniería y la informática ...

No obstante, permitirá comprobar el desarrollo de estas técnicas de cifra hoy en día rudimentarias y en algunos casos simples, desde una perspectiva histórica que es interesante como cultura general para todo ingeniero.

Clasificación actual de los criptosistemas

o bien según:

b) El tratamiento de la información a cifrar en:

- Cifrado en Bloque y Cifrado en Flujo

c) El tipo de clave utilizada en la cifra en:

- Sistema con Clave Secreta y Sistema con Clave Pública

Cifra en flujo	→	Se verá en Temas 8 y 9
Cifra en bloque	→	Se verá en Temas 8 y 10
Cifra con clave secreta	→	Se verá en Temas 10 y 14
Cifra con clave pública	→	Se verá en Temas 11, 12 y 14

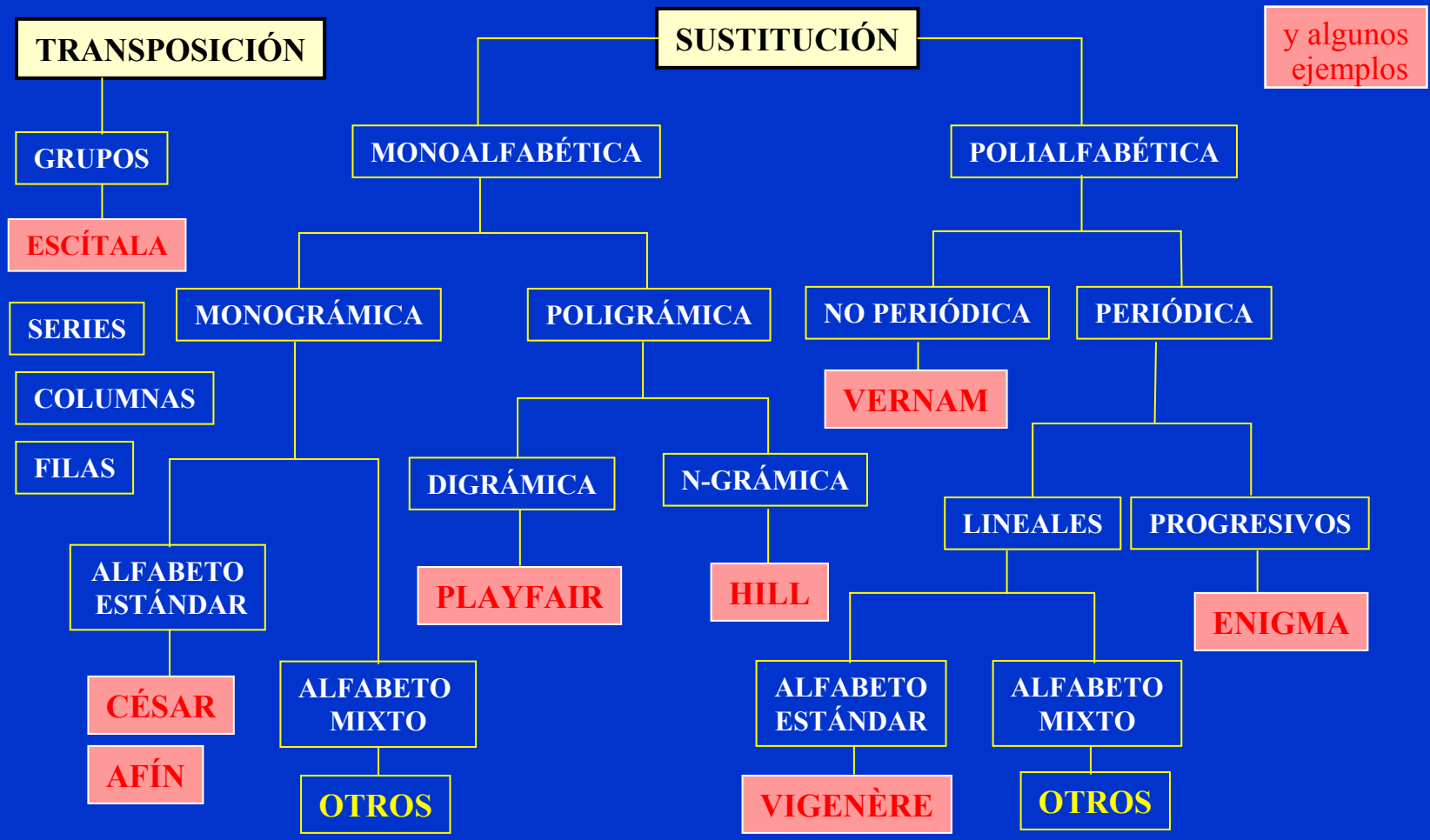
Primera aproximación histórica

- La criptografía es casi tan antigua como las primeras civilizaciones de nuestro planeta.
- Ya en el siglo V antes de J.C. se usaban técnicas de cifra para proteger a la información.
- Se pretendía garantizar sólo la confidencialidad y la autenticidad de los mensajes.
- Los mayores avances se lograron en la Segunda Guerra Mundial: los países en conflicto tenían un gran número de técnicos encargados de romper los mensajes cifrados de los teletipos.

Herramientas de la criptografía clásica

- Tanto máquinas, artilugios de cifra, como los algoritmos que trabajaban matemáticamente dentro de un cuerpo finito n, hacen uso de dos técnicas básicas orientadas a caracteres y que, muchos siglos después, propone Shannon:
  - Sustitución: un carácter o letra se modifica o sustituye por otro elemento en la cifra.
  - Transposición: los caracteres o letras del mensaje se redistribuyen sin modificarlos y según unas reglas, dentro del criptograma. También se le conoce como permutación.

Clasificación de los criptosistemas clásicos



Hitos históricos en la criptografía

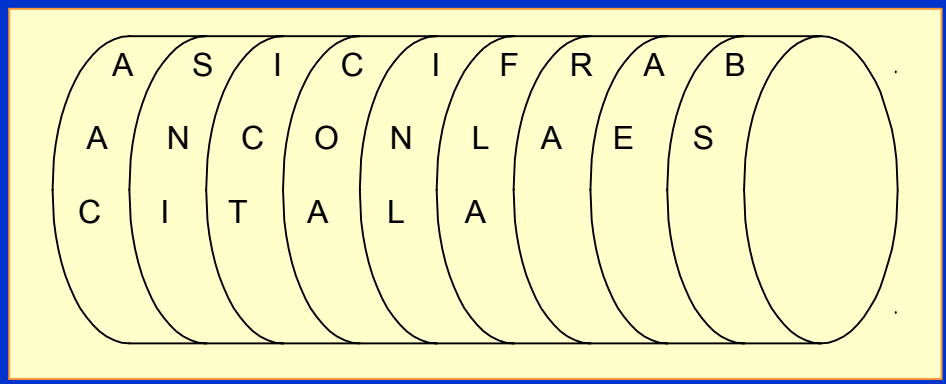
- La criptografía clásica abarca desde tiempos inmemoriales hasta la mitad del siglo XX.
- El punto de inflexión en esta clasificación la marcan tres hechos relevantes:
  - En el año 1948 se publica el estudio de Claude Shannon sobre la Teoría de la Información.
  - En 1974 aparece el estándar de cifra DES.
  - En el año 1976 se publica el estudio realizado por W. Diffie y M. Hellman sobre la aplicación de funciones matemáticas de un solo sentido a un modelo de cifra, denominado cifrado con clave pública.

CIFRADO DIGITAL

Primer cifrador por transposición: escítala

- La escítala era usada en el siglo V a.d.C. por el pueblo griego de los lacedemonios. Consistía en un bastón en el que se enrollaba una cinta de cuero y luego se escribía en ella el mensaje de forma longitudinal.
- Al desenrollar la cinta, las letras aparecen desordenadas.
- La única posibilidad de recuperar el texto en claro pasaba por enrollar dicha cinta en un bastón con el mismo diámetro que el usado en el extremo emisor y leer el mensaje de forma longitudinal. La clave del sistema está en el diámetro del bastón. Se trata de una cifra por transposición pues los caracteres del criptograma son los mismos que en el texto en claro distribuidos de otra forma.

Método de cifra de la escítala



En ese bastón residía la fortaleza de un pueblo. Hoy en día el popular bastón de mando que se le entrega al Alcalde de una ciudad proviene de esos tiempos remotos.

El texto en claro es:

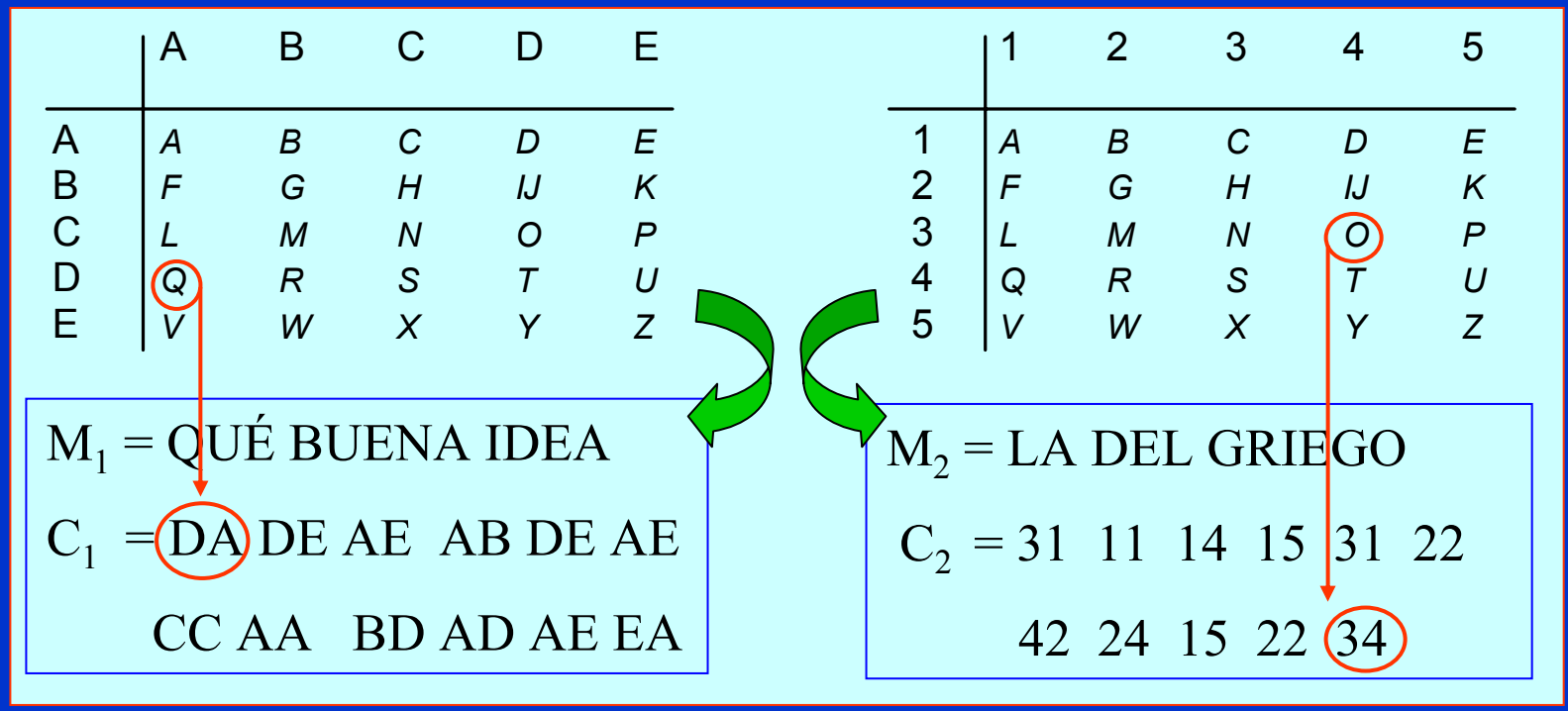
M = ASI CIFRABAN CON LA ESCITALA

El texto cifrado o criptograma será:

C = AAC SNI ICT COA INL FLA RA AE BS

Primer cifrador por sustitución: Polybios

Es el cifrador por sustitución de caracteres más antiguo que se conoce (siglo II a.d.C.) pero como duplica el tamaño del texto en claro, con letras o números, resulta poco interesante.



El cifrador del César

En el siglo I a.d.C., Julio César usa este cifrador, cuyo algoritmo consiste en el desplazamiento de tres espacios hacia la derecha de los caracteres del texto en claro. Es un cifrador por sustitución monoalfabético en el que las operaciones se realizan módulo n, siendo n el número de elementos del alfabeto (en aquel entonces latín).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
M_i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
C_i	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Alfabeto de cifrado del César para castellano mod 27

Ejemplo de cifra del César en mod 27

M_i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
C_i	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

$$C_i = M_i + 3 \bmod 27$$

M = EL PATIO DE MI CASA ES PARTICULAR

C = HÑ SDWLR GH OL FDVD HV SDUWLFXÑDU

Cada letra se cifrará siempre igual. Es una gran debilidad y hace que este sistema sea muy vulnerable y fácil de atacar simplemente usando las estadísticas del lenguaje.

Criptoanálisis del cifrador por sustitución

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Cifrado: $C_i = (M_i + b) \bmod 27$ Descifrado: $M_i = (C_i - b) \bmod 27$

La letra más frecuente del criptograma la hacemos coincidir con la más frecuente del lenguaje, la letra E, y encontramos así b.

C = LZAHL ZBTHW YBLIH XBLKL ILYOH ZLYCH ROKH

Frecuencias observadas en el criptograma: L (7); H (6); Z (3); B (3); Y (3); I (2); K (2); O (2); A (1); T (1); W (1); X (1); C (1); R (1).

Luego, es posible que la letra E del lenguaje (la más frecuente) se cifre como L en el criptograma y que la letra A se cifre como H:

$E + b \bmod 27 = L \Rightarrow b = L - E \bmod 27 = 11 - 4 \bmod 27 = 7$

$A + b \bmod 27 = H \Rightarrow b = H - A \bmod 27 = 7 - 0 \bmod 27 = 7$

M = ESTA ES UNA PRUEBA QUE DEBERIA SER VALIDA

Cifrador por sustitución afín mod 27

Cifrado: $C_i = a*M_i + b \bmod 27$

Descifrado: $M_i = (C_i - b) * a^{-1} \bmod 27$ donde $a^{-1} = \text{inv}(a, 27)$

El factor de decimación a deberá ser primo relativo con el cuerpo n (en este caso 27) para que exista el inverso.

El factor de desplazamiento puede ser cualquiera $0 \leq b \leq 27$.

El ataque a este sistema es también muy elemental. Se relaciona el elemento más frecuente del criptograma a la letra E y el segundo a la letra A, planteando un sistema de 2 ecuaciones. Si el texto tiene varias decenas de caracteres este ataque prospera; caso contrario, puede haber ligeros cambios en esta distribución de frecuencias.

Criptoanálisis a la cifra afín mod 27

C: NAQÑF EKNDP NCIVU FPUAN EJUIP FCNER NFRÑF UNPLN
AFPFQ TFPEI JRTÑE FPKÑI KTAPF LIKIÑ AIPÑU RCUJI
PCIVU CUNER IRLNP TJIAF NEOIÑ CFLNC NLUFA TEF

Caracteres más frecuentes en criptograma: F = 14; N = 13; I = 12

Con E y A las más frecuentes, el ataque falla. En un segundo intento suponemos la letra A más frecuente que la E, luego:

$F = (a*A + b) \bmod 27 \Rightarrow (a*0 + b) \bmod 27 = 5 \Rightarrow b = 5$

$N = (a*E + b) \bmod 27 \Rightarrow (a*4 + 5) \bmod 27 = 13$

Entonces $a = (13-5) * \text{inv}(4, 27) \bmod 27 = 8 * 7 \bmod 27 = 2$

Luego $C_i = (2*M_i + 5) \bmod 27 \Rightarrow M_i = (C_i - 5)*\text{inv}(2, 27)$. luego:

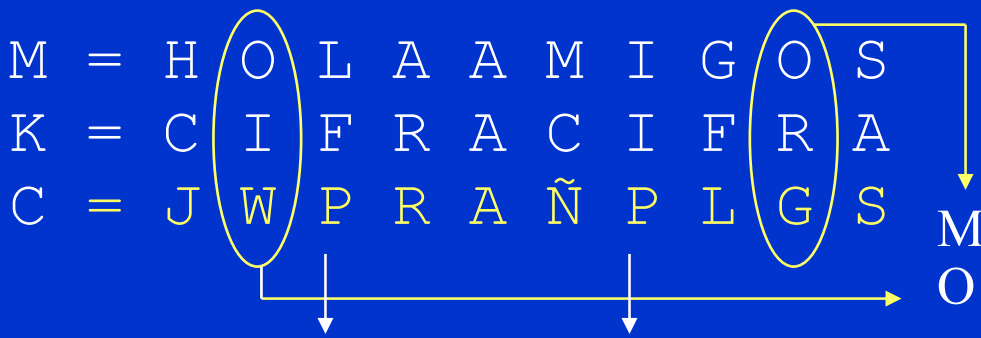
M: EL GRAN PEZ SE MOVÍA SILENCIOSAMENTE A TRAVÉS DE LAS AGUAS NOCTURNAS, PROPULSADO POR LOS RÍTMICOS MOVIMIENTOS DE SU COLA EN FORMA DE MEDIA LUNA. (Primer párrafo del libro "Tiburón" de P. Benchley).

El cifrador de Vigenère

Este cifrador polialfabético soluciona la debilidad del cifrado del César de que una letra se cifre siempre igual. Usa una clave K de longitud L y cifra carácter a carácter sumando módulo n el texto en claro con los elementos de esta clave.

$$C_i = M_i + K_i \bmod 27$$

Sea K = CIFRA y el mensaje M = HOLA AMIGOS



sumando mod 27...

Más de un alfabeto: la letra O se cifra de forma distinta.

Observe que el criptograma P se obtiene de un texto L y de un texto I.

¿Es Vigenère un algoritmo seguro?

Si la clave de Vigenère tiene mas de 6 caracteres distintos, se logra una distribución de frecuencias en el criptograma del tipo normal, es decir más o menos plana, por lo que se difumina la redundancia del lenguaje.

Aunque pudiera parecer que usando una clave larga y de muchos caracteres distintos y por tanto varios alfabetos de cifrado, Vigenère es un sistema de cifra seguro, esto es falso.

La redundancia del lenguaje unido a técnicas de criptoanálisis muy sencillas, como los métodos de Kasiski y del Índice de Coincidencia, permiten romper la cifra y la clave de una manera muy fácil y con mínimos recursos. Veamos un ataque por el método de Kasiski.

Ataque por el método de Kasiski

- El método de Kasiski consiste en buscar repeticiones de cadenas de caracteres en el criptograma. Si estas cadenas son mayores o iguales a tres caracteres y se repiten más de una vez, lo más probable es que esto se deba a cadenas típicas del texto en claro (trigramas, tetragramas, etc., muy comunes) que se han cifrado con una misma porción de la clave.
- Si se detectan estas cadenas, la distancia entre las mismas será múltiplo de la longitud de la clave. Luego, el máximo común divisor entre esas cadenas es un candidato a ser la longitud de la clave, digamos L.
- Dividimos el criptograma en L subcriptogramas que entonces han sido cifrados por una misma letra de la clave y en cada subcriptograma hacemos un ataque simple ahora de tipo estadístico monoalfabético.
- La idea es buscar ahora a través de los tres caracteres más frecuentes en cada subcriptograma las posiciones relativas de las letras A, E y O que en castellano están separadas por 4 y 11 espacios. La letra de la posición que ocupe la letra A (A = 0) será entonces la letra clave correspondiente.

Cadenas repetidas en ataque de Kasiski

Sea el criptograma C de 404 caracteres que vamos a criptoanalizar el siguiente:

```
PBVRQ VICAD SKAÑS DETSJ PSIED BGGMP SLRPW RÑPWY EDSDE ÑDRDP CRCPQ MNPWK
UBZVS FNVRD MTIPW UEQVV CBOVN UEDIF QLONM WNUVR SEIKA ZYEAC EYEDS ETFPH
LBHGU ÑESOM EHLBX VAEEP UÑELI SEVEF WHUNM CLPQP MBRRN BPVIÑ MTIBV VEÑID
ANSJA MTJOK MDODS ELPWI UFOZM QMVNF OHASE SRJWR SFQCO TWVMB JGRPW VSUEX
INQRS JEUEM GGRBD GNNIL AGSJI DSVSU EEINT GRUEE TFGGM PORDF OGTSS TOSEQ
OÑTGR RYVLP WJIFW XOTGG RPQRR JSKET XRNBL ZETGG NEMUO TXJAT ORVJH RSFHV
NUEJI BCHAS EHEUE UOTIE FFGYA TGGMP IKTBW UEÑEN IEEU.
```

Entre otras, se observan las siguientes cadenas (subrayadas) en el criptograma:

- 3 cadenas GGMP, separadas por 256 y 104 posiciones.
- 2 cadenas YEDS, separadas por 72 espacios.
- 2 cadenas HASE, separadas por 156 espacios.
- 2 cadenas VSUE, separadas por 32 espacios.

Luego el período de la clave puede ser mcd (256, 104, 72, 156, 32) = 4. La clave tendrá cuatro caracteres, por lo tanto tomaremos del criptograma el carácter 1º, el 5º, el 9º, etc. para formar el primer subcriptograma C_A; luego el 2º, el 6º, el 10º, etc. para formar el subcriptograma C_B, y así hasta el subcriptograma C_D.

Paso a cifrado monoalfabético en Kasiski

Tenemos 4 subcriptogramas que han sido cifrados con la misma letra de la clave:

C_A = PQAAEPDMRÑEEDCNUSRIECNIONSAAETLUOLAUIEULMNIIEAAOOLU
MNARSOMRSISERNAISIRTMDTOORLIORRENENOAVSNIAEOFAMTEI

C_B = BVDÑTSBPPPDÑPPPBFDPQBUFNUEZCDFBÑMBEÑSFNPBBÑBÑNMKDPF
QFSJFTBPUNJMBNGDUNUFPFSSÑRPFTPJTBTETTJFUBSUTFTPBÑE

C_C = VISSSIGSWWSDCQWZNMWVOEQMVIYESPHEEXEEEWMQRPMVISTMSWO
MOEWQWJWEQEGDISSETEGOOSETYWWGQSXLGMXOHHECEEIGGIWEE

C_D = RCKDJEGLRYDRRMKVVTUVVDLWRKEYEHGSHVPLVHCPRVTVDJJDEIZ
VHSRCVGVXRUGGLJVEGEGRGTQGVJXGRKRZGUJRRVJHHUEYGKUNU

La frecuencia relativa observada en cada uno de los subcriptogramas es:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
C_A	11	0	2	3	12	1	0	0	11	0	0	5	6	9	1	10	2	1	9	7	4	5	1	0	0	0	0
C_B	0	14	1	6	4	12	1	0	0	4	1	0	3	6	8	6	14	2	1	6	9	7	1	0	0	0	1
C_C	0	0	1	2	18	0	7	3	7	1	0	1	7	1	0	0	2	6	1	12	3	0	3	12	3	2	1
C_D	0	0	3	5	7	0	12	6	1	7	5	4	1	1	0	6	2	1	13	2	3	7	14	0	2	3	2

La letra más frecuente del subcriptograma debería corresponder a la letra E del texto en claro, la segunda a la letra A y la tercera a la letra O. ⟶

La regla AEO en el ataque de Kasiski

- Si la posición relativa de la letra A es el valor 0, entonces la letra E está cuatro espacios a la derecha de la A (m+4 mod 27) y la letra O está 15 espacios a la derecha de la letra A (m+15 mod 27).
- Buscaremos en cada subcriptograma C_i las tres letras más frecuentes y que cumplan además con esta distribución.
- Es suficiente contar con estas tres letras para que el ataque prospere. No obstante, podemos afinar más el ataque si tomamos en cuenta la siguiente letra frecuente en castellano (S) en posición (m+19) mod 27.

En el ejemplo para C_A se observa que la única solución que cumple con esto es la que coincide la AEO (11, 12, 10) luego la letra clave sería la A. Para C_B elegimos BFP (14, 12, 14) por lo que la letra clave sería B. Para C_C elegimos EIS (18, 7, 12) por lo que la letra clave sería E. Para C_D elegimos RVG (13, 14, 12) por lo que la letra clave sería R.

La clave será K = ABER y M = "Para que la cosa no me sorprenda...". ✌

El índice de coincidencia IC

Cuando encontramos una longitud L de la clave por el método de Kasiski y rompemos el criptogama en L subcriptogamas, podemos comprobar que cada uno de ellos se trata efectivamente de un cifrado monoalfabético aplicando el concepto del índice de coincidencia IC.

$$IC = \sum_{i=0}^{26} p_i^{\,2} \qquad \text{para castellano mod 27: } IC = p_A^{\,2} + p_B^{\,2} + \ldots + p_Z^{\,2} = 0{,}072$$

Aunque el estudio de este índice IC queda fuera del contexto de estos apuntes, como para el castellano mod 27 el IC = 0,072, en el ataque de Kasiski se comprueba que para cada subcriptograma su IC esté cercano a este valor. Si el IC es menor que 0,5 es muy probable que no estemos ante un cifrador monoalfabético sino uno polialfabético de periodo 2 o mayor.

En el ejemplo anterior, una vez roto el criptograma en cuatro tenemos: $IC_{CA} = 0{,}070$; $IC_{CB} = 0{,}073$; $IC_{CC} = 0{,}075$; $IC_{CD} = 0{,}065$. 

Cifrador poligrámico de Playfair

Los cifrados anteriores se hacían carácter a carácter, es decir eran monográmicos. Para aumentar la seguridad de la cifra podemos cifrar por poligramas, bloques de caracteres.

Un cifrador inventado a finales del siglo XIX es el de Playfair que trabaja con una matriz de 5x5 letras, cifrando por digramas. Si el texto en claro tiene un número impar de elementos, se rellena con una letra establecida, por ejemplo x.

A	B	C	D	E
F	G	H	I/J	K
L	M	N/Ñ	O	P
Q	R	S	T	U
V	W	X	Y	Z

- Si M_1M_2 están en la misma fila, C_1C_2 son los dos caracteres de la derecha.
- Si M_1M_2 están en la misma columna, C_1C_2 son los dos caracteres de abajo.
- Si M_1M_2 están en filas y columnas distintas, C_1C_2 son los dos caracteres de la diagonal, desde la fila de M_1.

Ejemplo de cifra con Playfair

Si la clave K = BEATLES, eliminando la letra Ñ, se pide cifrar el mensaje M = With a little help from my friends.

B	E	A	T	L
S	C	D	F	G
H	I	K	M	N
O	P	Q	R	U
V	W	X	Y	Z

Se rompe la doble MM agregando una X y se rellena al final con X

M = WI TH AL IT TL EH EL PF RO MX MY FR IE ND SX
C = EP BM TB ME LB BI AB RC UP KY RT MY PC KG DV

Estos sistemas también son criptoanalizables pues en el criptograma C persisten algunas propiedades del lenguaje, en este caso la distribución de digramas típicos del castellano como por ejemplo en, de, mb, etc.

El cifrador de matrices de Hill

En 1929 Lester Hill propone un sistema de cifra usando una matriz como clave, cifrando Ngramas de forma que:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ .. \\ C_N \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} & \dots & k_{1N} \\ k_{21} & k_{22} & k_{23} & \dots & k_{2N} \\ k_{31} & k_{32} & k_{33} & \dots & k_{3N} \\ .. & .. & .. & & .. \\ k_{N1} & k_{N2} & k_{N3} & \dots & k_{NN} \end{pmatrix} X \begin{pmatrix} M_1 \\ M_2 \\ M_3 \\ .. \\ M_N \end{pmatrix} \quad \text{mod n}$$

La matriz clave K debe tener inversa K^{-1} en el cuerpo de cifra n. Luego, como $K^{-1} = T_{ADJ(K)}/|K|$ mod n, en donde ADJ(K) es la matriz adjunta, T es la traspuesta y |K| el determinante, este último valor |K| no podrá ser cero ni tener factores en común con n puesto que está en el denominador (concepto de inverso).
Si el texto en claro no es múltiplo del bloque N, se rellena con caracteres predeterminados, por ejemplo la letra X o la Z.

Ejemplo de cifrado de Hill

Sea M = AMIGO CONDUCTOR y la clave K la que se muestra:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 16 & 4 & 11 \\ 8 & 6 & 18 \\ 15 & 19 & 15 \end{pmatrix} \times \begin{pmatrix} 0 \\ 12 \\ 8 \end{pmatrix} \bmod 27$$

K = PELIGROSO será la clave simbólica. Se cifrará el primer trigrama: AMI = 0, 12, 8.

M = AMI GOC OND UCT ORZ

$C_1 = (16*0 + 4*12 + 11*8) \bmod 27 = 136 \bmod 27 = 1 = B$

$C_2 = (8*0 + 6*12 + 18*8) \bmod 27 = 216 \bmod 27 = 0 = A$

$C_3 = (15*0 + 19*12 + 15*8) \bmod 27 = 348 \bmod 27 = 24 = X$

C = BAX PMA BJE XAF EUM (compruebe Ud. los otros trigramas)

Para descifrar encontramos $K^{-1} = \text{inv}(K, 27) = K^{-1} = T_{ADJ(K)}/|K| \bmod n$

$|K| = 16(6*15 - 19*18) - 4(8*15 - 15*18) + 11(8*19 - 15*6) \bmod 27 = 4$

Encontramos luego la matriz adjunta de K, la trasponemos cambiando filas por columnas y la multiplicamos por inv (|K|, 27) = inv (4, 27) = 7 con lo que se obtiene la matriz que se indica (hágalo Ud.)

Ejemplo de descifrado de Hill

$$\begin{pmatrix} M \end{pmatrix} = \begin{pmatrix} K^{-1} \end{pmatrix} \text{ x } \begin{pmatrix} C \end{pmatrix} \bmod n \quad \text{y} \quad K^{-1} = \begin{pmatrix} 18 & 26 & 15 \\ 24 & 6 & 13 \\ 11 & 24 & 10 \end{pmatrix}$$

C = BAXPMABJEXAFEUM y la clave K^{-1} es la que se muestra:

$$\begin{pmatrix} M_1 \\ M_2 \\ M_3 \end{pmatrix} = \begin{pmatrix} 18 & 26 & 15 \\ 24 & 6 & 13 \\ 11 & 24 & 10 \end{pmatrix} \text{ X } \begin{pmatrix} 1 \\ 0 \\ 24 \end{pmatrix} \bmod 27$$

Descifrado del primer trigrama del criptograma: BAX = 1, 0, 24.

C = BAX PMA BJE XAF EUM

$M_1 = (18*1 + 26*0 + 15*24) \bmod 27 = 378 \bmod 27 = 0 = A$

$M_2 = (24*1 + 6*0 + 13*24) \bmod 27 = 336 \bmod 27 = 12 = M$

$M_3 = (11*1 + 24*0 + 10*24) \bmod 27 = 251 \bmod 27 = 8 = I$

M = AMI GOC OND UCT ORZ (compruebe Ud. los otros trigramas)

¿Es seguro el cifrador de Hill?

Si con el sistema de Hill se cifran bloques de 8 caracteres, incluso en un cuerpo tan pequeño como n = 27 el espacio de claves aumenta de forma espectacular, comparable con DES.

Si el módulo de cifra es un primo p, entonces el número de claves válidas es cercano al máximo posible: p^x donde $x = d^2$, con d el tamaño de N-grama o de la matriz clave.

No obstante, el sistema no es seguro. Debido a su linealidad será muy fácil hacer un ataque con texto claro conocido según el método de Gauss Jordan y encontrar así la matriz clave K. Esto es debido a que aparecen los llamados vectores unitarios en el criptograma o en el texto en claro, o bien los obtenemos aplicando este método.

Ataque al cifrado de Hill por Gauss Jordan

El método consiste en escribir una matriz 2N-grámica con los elementos del texto en claro y los elementos del criptograma. En esta matriz realizamos operaciones lineales (multiplicar filas por un número y restar filas entre sí) con el objeto de obtener los vectores unitarios.

Por ejemplo podemos romper la matriz clave K teniendo:

```
M = ENU NLU GAR DEL AMA NCH ADE CUY ONO ...
C = WVX IDQ DDO ITQ JGO GJI YMG FVC UÑT ...
```

$$\left(\begin{array}{ccc|ccc} E & N & U & W & V & X \\ N & L & U & I & D & Q \\ G & A & R & D & D & O \\ D & E & L & I & T & Q \\ A & M & A & J & G & O \\ N & C & H & G & J & I \\ A & D & E & Y & M & G \\ C & U & Y & F & V & C \\ O & N & O & U & Ñ & T \end{array}\right) = \left(\begin{array}{ccc|ccc} 4 & 13 & 21 & 23 & 22 & 24 \\ 13 & 11 & 21 & 8 & 3 & 17 \\ 6 & 0 & 18 & 3 & 3 & 15 \\ 3 & 4 & 11 & 8 & 20 & 17 \\ 0 & 12 & 0 & 9 & 6 & 15 \\ 13 & 2 & 7 & 6 & 9 & 8 \\ 0 & 3 & 4 & 25 & 12 & 6 \\ 2 & 21 & 25 & 5 & 22 & 2 \\ 15 & 13 & 15 & 21 & 14 & 20 \end{array}\right)$$

Operaciones en la matriz de Gauss Jordan

Vamos a dejar en la primera columna un número uno en la fila primera y todas las demás filas un cero. Luego multiplicamos el vector (4 13 21 | 23 22 24) por el inv (4, 27) = 7. Así obtenemos 7(4 13 21 | 23 22 24) mod 27 = (1 10 12 | 26 19 6). Si esto no se puede hacer con la primera fila movemos los vectores. Hecho esto vamos restando las filas respecto de esta primera como se indica:

4	13	21	23	22	24
13	11	21	8	3	17
6	0	18	3	3	15
3	4	11	8	20	17
0	12	0	9	6	15
13	2	7	6	9	8
0	3	4	25	12	6
2	21	25	5	22	2
15	13	15	21	14	20

a) 2^a fila = 2^a fila – 13∗1^a fila mod 27
b) 3^a fila = 3^a fila – 6∗1^a fila mod 27
c) 4^a fila = 4^a fila – 3∗1^a fila mod 27
d) 5^a fila ya tiene un 0
e) 6^a fila = 6^a fila – 13∗1^a fila mod 27
f) 7^a fila ya tiene un 0
g) 8^a fila = 8^a fila – 2∗1^a fila mod 27
h) 9^a fila = 9^a fila – 15∗1^a fila mod 27

Matriz clave de Hill criptoanalizada

Repetimos este procedimiento ahora para algún vector en cuya segunda columna tenga un número con inverso en 27 y lo mismo para la tercera columna, moviendo si es preciso los vectores.

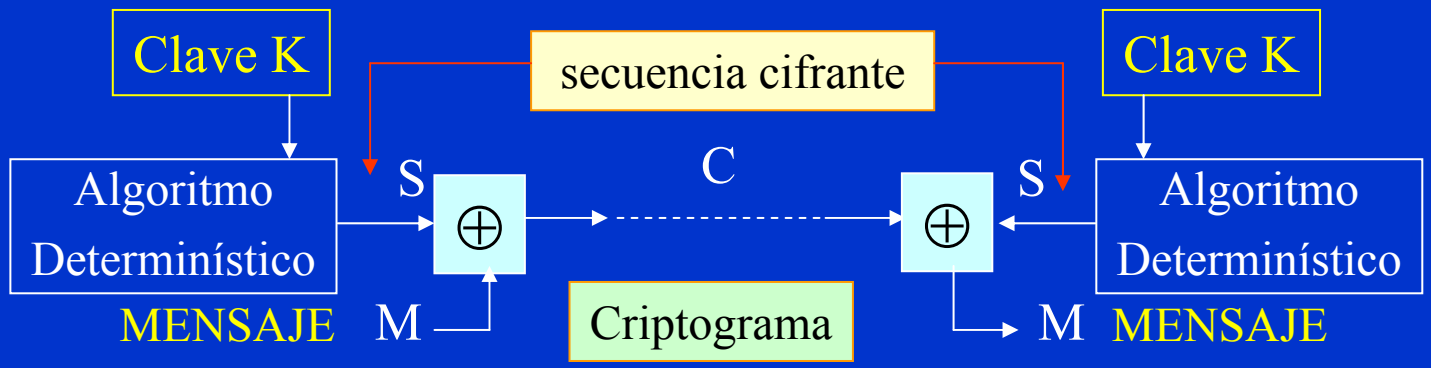
Como la mitad izquierda de la matriz 2N era el texto el claro, la parte derecha de la matriz con vectores unitarios corresponderá a la traspuesta de la clave.

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 2 & 5 & 7 \\ 0 & 1 & 0 & 3 & 5 & 8 \\ 0 & 0 & 1 & 4 & 6 & 9 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \Rightarrow \quad K = \begin{pmatrix} 2 & 3 & 4 \\ 5 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Compruebe que la clave es la utilizada en este cifrado.

El cifrador de Vernam

- En 1917 Gilbert Vernam (MIT) propone un cifrador por sustitución binaria con clave de un solo uso, basado en el código Baudot de 5 bits:
  - La operación de cifra es la función XOR.
  - Usa una secuencia cifrante binaria y aleatoria S que se obtiene de una clave secreta K compartida por emisor y receptor.
  - El algoritmo de descifrado es igual al de cifrado por la involución de la función XOR.
  - La clave será tan larga o más que el mensaje y se usará una sola vez.



Ejemplo de cifrado de Vernam

Usando el código Baudot (véase capítulo 18) se pide cifrar el mensaje M = BYTES con la clave K = VERNAM.

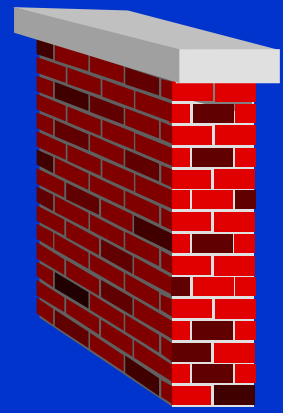
Solución:

$$B \oplus V = 11001 \oplus 11110 = 00111 = U$$
$$Y \oplus E = 10101 \oplus 00001 = 10100 = H$$
$$T \oplus R = 10000 \oplus 01010 = 11010 = G$$
$$E \oplus N = 00001 \oplus 01100 = 01101 = F$$
$$S \oplus A = 00101 \oplus 00011 = 00110 = I$$

C = UHGFI

El sistema de Vernam es el único que es matemáticamente seguro e imposible de criptoanalizar ya que la clave se usa una sola vez (*one time pad*), es aleatoria y tanto o más larga que el propio mensaje.

En construcción ...



ESTIMADO/A LECTOR/A:

AL IGUAL QUE EN CIERTAS PÁGINAS WEB, ESTE CAPÍTULO ESTARÁ EN CONSTRUCCIÓN DURANTE MUCHO TIEMPO... ☹ ... mis disculpas!

NO OBSTANTE, SE SEGUIRÁN PUBLICANDO LOS TEMAS SIGUIENTES DEL CURSO EN DIAPOSITIVAS SOBRE TÉCNICAS DE CIFRA MODERNAS, DE MAYOR INTERÉS ACADÉMICO ☺ DE ACUERDO CON EL AVANCE DE LA ASIGNATURA DE SEGURIDAD INFORMÁTICA.

Pero si está muy interesado en el tema...

El libro electrónico de criptografía clásica

Si está interesado en estos temas de criptografía clásica e historia de las máquinas de cifrar, puede descargar el Libro Electrónico de Criptografía Clásica hecho en ToolBook desde el servidor de la Red Temática Iberoamericana de Criptografía y Seguridad de la Información, CriptoRed.

http://www.criptored.upm.es/paginas/software.htm#propio

El archivo de instalación tiene 5.17 MB y es de libre distribución como gran parte del software y documentos de este servidor.

Su uso es verdaderamente sencillo. Podrá además comprobar y practicar con estos algoritmos de cifra clásica puesto que el libro incluye una sección con software específico para ello.

También puede descargarse desde el mismo servidor un programa para prácticas denominado CriptoClásicos.

El libro de la asignatura y los exámenes

También puede seguir con amplios detalles estos sistemas de cifra clásicos en el libro de la asignatura Seguridad Informática "Aplicaciones Criptográficas", 2ª edición, Junio de 1999 Departamento de Publicaciones - Escuela Universitaria de Informática - Universidad Politécnica de Madrid (España) I.S.B.N.: 84-87238-57-2 Depósito Legal: M-24709-1999

Y en los casi 20 exámenes de dicha asignatura (incluyen siempre un ejercicio básico sobre este tipo de cifra) y sus soluciones que podrá encontrar en el servidor Web de CriptoRed

http://www.criptored.upm.es/paginas/docencia.htm#examenes

Fin del Tema 7

Cuestiones y ejercicios (1 de 3)

LAS SIGUIENTES PREGUNTAS ESTÁN RELACIONADAS CON ESTOS APUNTES, EL LIBRO ELECTRÓNICO DE CRIPTOGRAFÍA CLÁSICA Y EL SOFTWARE DE PRÁCTICAS CRIPTOCLÁSICOS QUE SE HA COMENTADO.

1. ¿Qué significa cifrar por sustitución y qué por transposición?
2. ¿Por qué que el método escítala es un cifrado por permutación?
3. ¿Cuál es la peor debilidad que tiene el sistema de cifra del César?
4. Ciframos el mensaje M = HOLA QUE TAL con un desplazamiento de 6 caracteres, ¿cuál es el criptograma? ¿Y si desplazamos 27?
5. ¿Por qué no podemos cifrar en el cuerpo n = 27 con la función de cifra C = (12M + 5) mod n? ¿Qué condición deberá cumplirse?
6. ¿Cómo podríamos atacar un sistema de cifra tipo César? ¿Y si la cifra es de tipo afín como el de la pregunta anterior?

Cuestiones y ejercicios (2 de 3)

7. Cifre el mensaje M = VAMOS A VER con un sistema afín siendo el valor a = 5 y b = 2 usando sólo operaciones modulares.
8. En un sistema de cifra de Vigenère la clave a usar puede ser CERO o bien COMPADRE, ¿cuál de las dos usaría y por qué?
9. Cifre según Vigenère el mensaje M = UNA PRUEBA con la clave K = OLA sin usar la tabla, sólo con operaciones modulares.
10. ¿Por qué se dice que Vigenère es un cifrador polialfabético?
11. ¿Cómo podríamos atacar un cifrado polialfabético periódico?
12. Cifre con el método de Vernam binario en mensaje M = VIDA y clave K = TACO suponiendo texto ASCII. ¿Si la clave se cambia en cada cifra y es aleatoria, cómo se comporta este cifrador?
13. ¿Qué significa cifrar por homófonos? ¿Qué es el cifrado de Beale?

Cuestiones y ejercicios (3 de 3)

14. Indique las máquinas de cifrar que se usaron en la Segunda Guerra Mundial y diga de forma sencilla cómo funcionaban.
15. Se cifra por permutaciones usando para ello una distribución en columnas con clave. ¿Qué similitud tendrá luego este sistema de cifra con algunas operaciones hechas en el DES?
16. Cifre con el cifrador digrámico de Hill el mensaje ADIOS AMIGO. ¿Qué matriz simbólica puede usar: GATO, GOTA, MISA o MESA?
17. Cifre y descifre con la matriz trigrámica simbólica PELIGROSO el mensaje HOY ES UN HERMOSO DIA.
18. Si la clave es tan grande, ¿es segura la cifra de Hill? ¿Por qué?
19. ¿Qué significan los vectores unitarios? ¿Es fácil encontrarlos?
20. ¿Cómo funciona el ataque de Gauss Jordan? Obtenga la matriz clave del ejercicio 17 mediante Gauss Jordan.

Tema 8
Sistemas de Cifra Modernos

Seguridad Informática y Criptografía

Material Docente de
Libre Distribución

Ultima actualización: 03/02/03
Archivo con 33 diapositivas

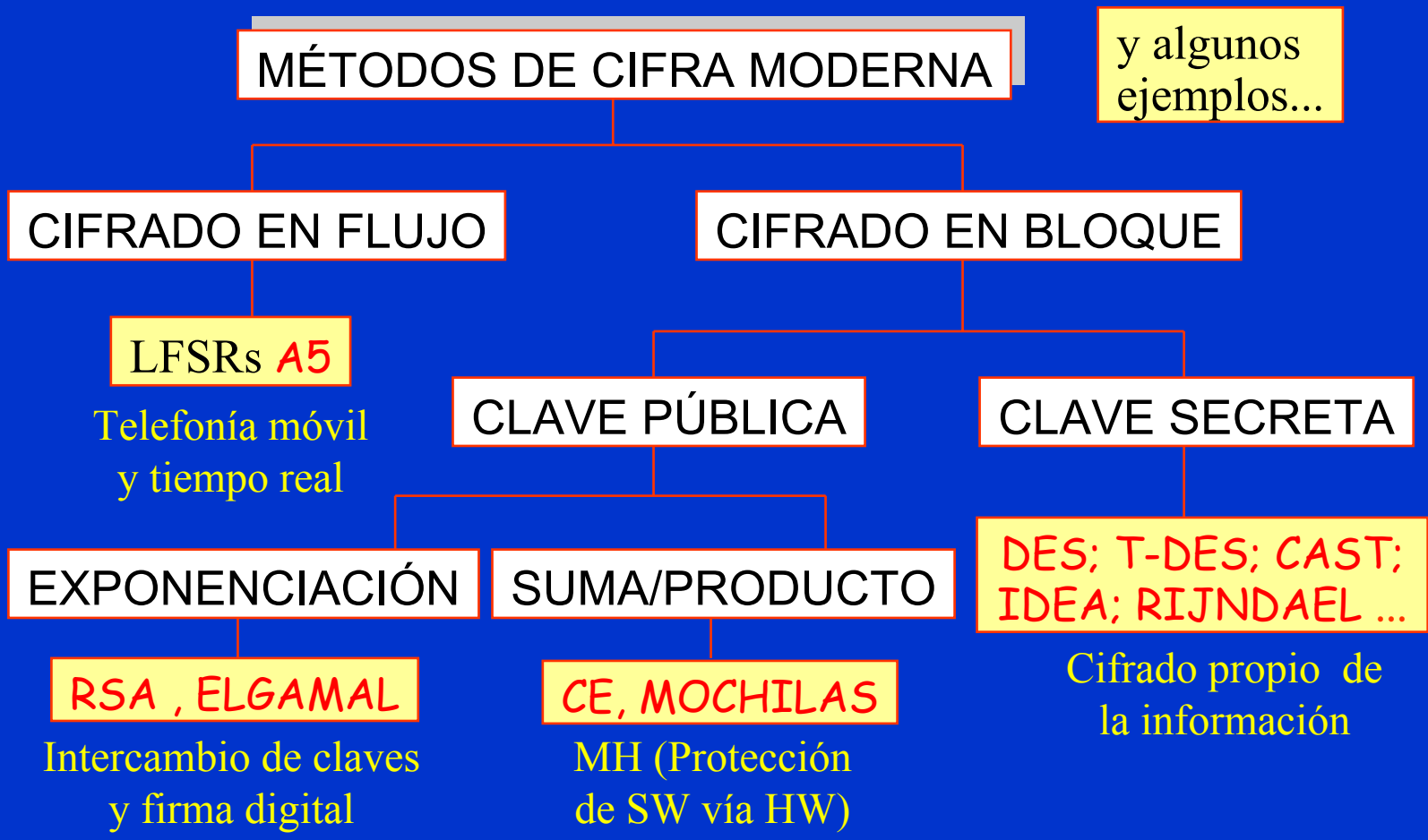
Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Conceptos elementales

Un par de ideas básicas

- Los criptosistemas modernos, cuya cifra en bits está orientada a todos los caracteres ASCII o ANSI usan por lo general una operación algebraica en Z_n, un cuerpo finito, sin que necesariamente este módulo deba corresponder con el número de elementos del alfabeto o código utilizado. Es más, nunca coinciden; siempre será mucho mayor el cuerpo de trabajo que el alfabeto.
- Su fortaleza está en la imposibilidad computacional de descubrir una clave secreta única, en tanto que el algoritmo de cifra es (o debería ser) público.

Clasificación de los criptosistemas



Introducción al cifrado de flujo

Usa el concepto de cifra propuesto por Vernam, que cumple con las ideas de Shannon sobre sistemas de cifra con secreto perfecto, esto es:

a) El espacio de las claves es igual o mayor que el espacio de los mensajes.

b) Las claves deben ser equiprobables.

c) La secuencia de clave se usa una sola vez y luego se destruye (sistema *one-time pad*).

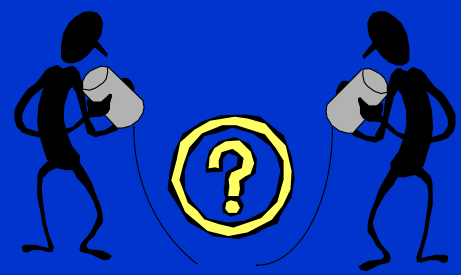
DUDA: ¿Es posible satisfacer la condición a)?

Espacio de claves y del mensaje

¿Espacio de Claves ≥ Espacio de Mensajes?

1) La secuencia de bits de la clave deberá enviarse al destinatario a través de un canal que sabemos es inseguro (recuerde que aún no conoce el protocolo de intercambio de clave de Diffie y Hellman).
2) Si la secuencia es "infinita", desbordaríamos la capacidad del canal de comunicaciones.

¿Qué solución damos a este problema?



El concepto de semilla en un generador

Si por el canal supuestamente seguro enviamos esa clave tan larga ... ¿por qué entonces no enviamos directamente el mensaje en claro y nos dejamos de historias? ☺

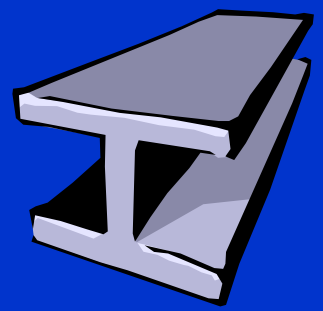
La solución está en generar una secuencia de tipo pseudoaleatoria con un algoritmo determinístico a partir de una semilla de sólo unas centenas de bits. Podremos generar así secuencias con períodos del orden de 2^n, un valor ciertamente muy alto. Esta semilla es la que se envía al receptor mediante un sistema de cifra de clave pública y un algoritmo de intercambio de clave y no sobrecargamos el canal.

Técnica de cifra en flujo

- ✓ El mensaje en claro se leerá bit a bit.
- ✓ Se realizará una operación de cifra, normalmente la función XOR, con una secuencia cifrante de bits S_i que debe cumplir ciertas condiciones:
  - Un período muy alto.
  - Aleatoriedad en sus propiedades.

Lo veremos en el capítulo 9...

Introducción a la cifra en bloque



El mensaje se agrupa en bloques, por lo general de 8 bytes, antes de aplicar el algoritmo de cifra a cada bloque de forma independiente con la misma clave.

Cifrado con Clave Secreta

Hay algunos algoritmos muy conocidos por su uso en aplicaciones bancarias (DES), correo electrónico (IDEA, CAST) y comercio electrónico (Triple DES).

No obstante, tienen tres puntos débiles.

Debilidades de la cifra con clave secreta

a) Mala gestión de claves. Crece el número de claves secretas en un orden igual a n^2 para un valor n grande de usuarios 👎.

b) Mala distribución de claves. No existe posibilidad de enviar, de forma segura, una clave a través de un medio inseguro 👎.

c) No tiene firma digital. Aunque sí será posible autenticar el mensaje mediante una marca, no es posible firmar digitalmente el mensaje 👎.

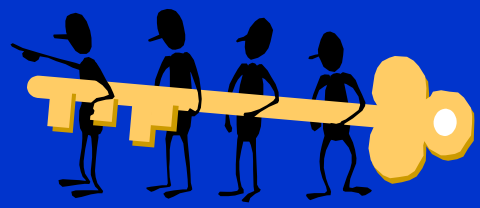
¿Por qué usamos clave secreta?

a) Mala gestión de claves 👎
b) Mala distribución de claves 👎
c) No tiene firma digital 👎

¿Tiene algo de bueno la cifra en bloque con clave secreta?

Sí: la velocidad de cifra es muy alta 👍 y por ello se usará para realizar la función de cifra de la información. Además, con claves de sólo unas centenas de bits obtendremos una alta seguridad pues su no linealidad y algoritmo hace que el único ataque que puede prosperar es de el de la fuerza bruta.

Cifrado en bloque con clave pública



- Comienza a ser ampliamente conocido a través de su aplicación en los sistemas de correo electrónico seguro (PGP y PEM) permitiendo cifrar e incluir una firma digital adjunta al documento o e-mail enviado y también en navegadores Web.
- Cada usuario tiene dos claves, una secreta o privada y otra pública, inversas dentro de un cuerpo.
- Usan las funciones unidireccionales con trampa.

Funciones unidireccionales con trampa

Son funciones matemáticas de un solo sentido (*one-way functions*) y que nos permiten usar la función en sentido directo o de cálculo fácil para cifrar y descifrar (usuarios legítimos) y fuerza el sentido inverso o de cálculo difícil para aquellos impostores, hackers, etc. que lo que desean es atacar o criptoanalizar la cifra.

$f(M) = C$ es *siempre fácil.*

$f^{-1}(C) = M$ es *difícil salvo que se tenga la trampa.*

Funciones con trampa más usadas

Problema de la factorización

Cálculo directo: producto de dos primos grandes $p*q = n$
Cálculo inverso: factorización de número grande $n = p*q$

Problema del logaritmo discreto

Cálculo directo: exponenciación discreta $\beta = \alpha^x \bmod n$
Cálculo inverso: logaritmo discreto $x = \log_\alpha \beta \bmod n$

Otras funciones con trampa

Problema de la mochila

Cálculo directo: sumar elementos de mochila con trampa
Cálculo inverso: sumar elementos de mochila sin trampa

Problema de la raíz discreta

Cálculo directo: cuadrado discreto $x = a*a \bmod n$
Cálculo inverso: raíz cuadrada discreta $a = \sqrt{x} \bmod n$

Cifrado con clave pública de destino

origen

NUESTROS PROTAGONISTAS

destino

Benito

Claves: e_B, n_B, d_B

e_B, n_B: públicas

d_B: privada

e_B y d_B son inversas dentro de un cuerpo Z_B

Si Benito realiza la operación con las claves públicas de Adela (e_A, n_A), la información que se transmite mantiene la confidencialidad: sólo ella puede verla.

$C = E_{eA}(M) \bmod n_A$

Adela

Claves: e_A, n_A, d_A

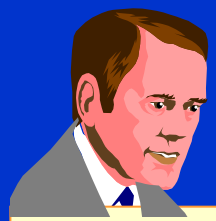
e_A, n_A: públicas

d_A: privada

e_A y d_A son inversas dentro de un cuerpo Z_A

Operación de cifra con clave de destino

Cifrado:



Benito envía un mensaje M a Adela

Benito

Claves: e_B, n_B, d_B

$C = E_{eA}(M) \bmod n_A$

Adela

Claves: e_A, n_A, d_A

Claves públicas

Clave privada

Descifrado:

$M = E_{dA}[E_{eA}(M)] \bmod n_A$

E_{dA} y E_{eA} son inversos

Se obtiene *confidencialidad* del mensaje

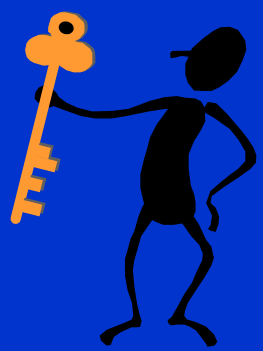
¿Y si usamos la clave pública de origen?

Si en vez de utilizar la clave pública de destino, el emisor usa su propia clave pública, la cifra no tiene sentido bajo el punto de vista de sistemas de clave pública ya que sólo él o ella sería capaz de descifrar el criptograma (deshacer la operación de cifra) con su propia clave privada.

Esto podría usarse para cifrar de forma local uno o varios ficheros, por ejemplo, pero para ello ya están los sistemas de clave secreta, mucho más rápidos y, por tanto, más eficientes.

¿Y si usamos la clave privada de origen?

Si ahora el emisor usa su clave privada en la cifra sobre el mensaje, se obtiene una firma digital que le autentica como emisor ante el destinatario y, además, a este último le permitirá comprobar la integridad del mensaje.



Veamos antes un ejemplo de algoritmo que usa un par de claves entre dos usuarios... →

Obviamente, el emisor nunca podrá realizar la cifra del mensaje M con la clave privada del receptor.

El algoritmo del mensaje en la caja

PROTOCOLO: A envía a B un mensaje M

1 A pone el mensaje M en la caja, la cierra con su llave y la envía a B.

2 B recibe la caja, la cierra con su llave y envía a A la caja con las dos cerraduras .

3 A recibe la caja, quita su llave y devuelve a B la caja sólo con la cerradura de B .

4 B recibe la caja, quita su cerradura y puede ver el mensaje M que A puso en el interior de la caja.

¿Todo bien en el algoritmo de la caja?

Durante la transmisión, el mensaje está protegido de cualquier intruso por lo que existe integridad del mensaje y hay protección contra una ataque pasivo.

Pero el usuario B no puede estar seguro si quien le ha enviado el mensaje M es el usuario A o un impostor. Por lo tanto el algoritmo así implementado no nos permite comprobar la autenticidad del emisor pues no detecta la suplantación de identidad. No obstante... →

Modificando algo el algoritmo anterior, podremos asegurar tanto la integridad del mensaje como la autenticidad de emisor.

Cifrado con clave privada del origen

origen

Benito

Claves: e_B, n_B, d_B

e_B, n_B: públicas

d_B: privada

e_B y d_B son inversas dentro de un cuerpo Z_B

Si ahora Benito realiza la operación de cifra con su clave privada d_B en el cuerpo n_B Adela será capaz de comprobar esa cifra ya que posee (entre otras) la clave pública de Benito. Comprueba así tanto la autenticidad del mensaje como del autor.

$$C = E_{dB}(M) \bmod n_B$$

destino

Adela

Claves: e_A, n_A, d_A

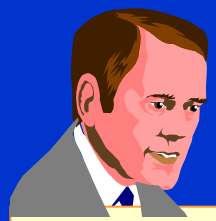
e_A, n_A: públicas

d_A: privada

e_A y d_A son inversas dentro de un cuerpo Z_A

Operación de cifra con clave de origen

Cifrado:



Benito firma un mensaje M a Adela

Benito

Adela

Claves: e_B, n_B, d_B

$C = E_{dB}(M) \bmod n_B$

Claves: e_A, n_A, d_A

Clave privada

Claves públicas

Descifrado:

$M = E_{eB}[E_{dB}(M)] \bmod n_B$

E_{dB} y E_{eB} son inversos

Se comprueba la *integridad* del origen

Uso de la criptografía asimétrica

- Estas dos operaciones de cifra son posibles debido a la característica intrínseca de los sistemas de clave pública: el uso de una clave privada (secreta) inversa de una pública.

  ¿Qué aplicación tendrán entonces los sistemas de criptografía de clave pública o asimétrica?
- Usando la clave pública del destino se hará el intercambio de claves de sesión de una cifra con sistemas simétricos (decenas a centenas de bits).
- Usando la clave privada de origen, se firmará digitalmente un resumen (decenas a centenas de bits) del mensaje obtenido con una función hash.

La gestión de claves

Gestión de claves

Clave Secreta	Clave Pública
Hay que memorizar un número muy alto de claves: $\rightarrow n^2$.	Sólo es necesario memorizar la clave privada del emisor.

En cuanto a la gestión de claves, serán mucho más eficientes los sistemas de cifra asimétricos pues los simétricos no permiten una gestión lógica y eficiente de estas claves.

El espacio de claves

Longitud y espacio de claves

Clave Secreta	Clave Pública
Debido al tipo de cifrador usado, la clave será del orden de la centena de bits.	Por el algoritmo usado en la cifra, la clave será del orden de los miles de bits.

≥ 128

≥ 1024

En cuanto al espacio de claves, no son comparables los sistemas simétricos con los asimétricos. Para atacar un sistema asimétrico no se buscará en todo el espacio de claves como debería hacerse en los sistemas simétricos.

La vida de las claves

Vida de una clave	
Clave Secreta	Clave Pública
La duración es muy corta. Normalmente se usa como una clave de sesión.	La duración de la clave pública, que la entrega y gestiona un tercero, suele ser larga.

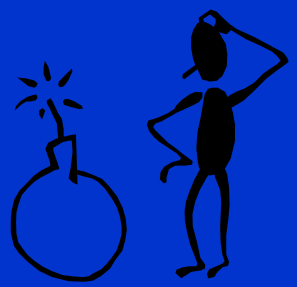
Segundos o minutos

En cuanto a la vida de una clave, en los sistemas simétricos ésta es muchísimo menor que las usadas en los asimétricos. La clave de sesión es aleatoria, en cambio la asimétrica es propia del usuario.

Meses o un año

Vida de la clave y principio de caducidad

Si en un sistema de clave secreta, ésta se usa como clave de una sesión que dura muy poco tiempo... y en este tiempo es imposible romperla...



¿para qué preocuparse entonces?

La confidencialidad de la información tiene una caducidad. Si durante este tiempo alguien puede tener el criptograma e intentar un ataque por fuerza bruta, obtendrá la clave (que es lo menos importante) ...

¡pero también el mensaje secreto! ... puede ser muy peligroso.

El problema de la autenticación

Condiciones de la autenticidad:

a) El usuario A deberá protegerse ante mensajes dirigidos a B que un tercer usuario desconocido C introduce por éste. Es la suplantación de identidad o problema de la autenticación del emisor.

b) El usuario A deberá protegerse ante mensajes falsificados por B que asegura haberlos recibido firmados por A. Es la falsificación de documento o problema de la autenticación del mensaje.

La autenticación de origen y de destino

Autenticación

Clave Secreta	Clave Pública
Se puede autenticar al mensaje pero no al emisor de forma sencilla y clara.	Al haber una clave pública y otra privada, se podrá autenticar el mensaje y al emisor.

En cuanto a la autenticación, los sistemas simétricos tienen una autenticación más pesada y con una tercera parte de confianza. Los asimétricos permiten una firma digital verdadera, eficiente y sencilla.

La velocidad de cifra

Velocidad de cifra

Clave Secreta	Clave Pública
La velocidad de cifra es muy alta. Es el algoritmo de cifra del mensaje.	La velocidad de cifra es muy baja. Se usa para el intercambio de clave y la firma digital.

Cientos de M Bytes/seg en HW

En cuanto a la velocidad de cifra, los sistemas simétricos son de 100 a 1.000 veces más rápidos que los asimétricos. En SW la velocidad de cifra es más baja.

Cientos de K Bytes/seg en HW

Resumen cifra simétrica v/s asimétrica

Cifrado Simétrico

- Confidencialidad
- Autenticación parcial
- Sin firma digital
- Claves:
  - Longitud pequeña
  - Vida corta
  - Número elevado
- Velocidad alta

Cifrado Asimétrico

- Confidencialidad
- Autenticación total
- Con firma digital
- Claves:
  - Longitud grande
  - Vida larga
  - Número reducido
- Velocidad baja

Fin del Tema 8

Cuestiones y ejercicios (1 de 2)

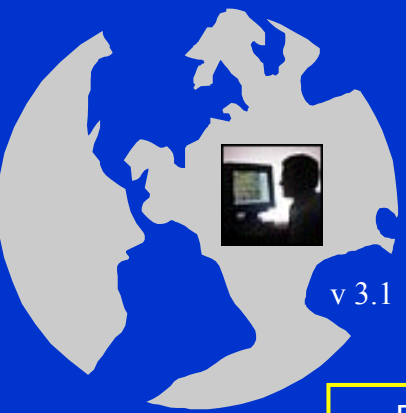
1. En un sistema de cifra se usa un cuerpo de trabajo n. ¿Cómo es el tamaño de ese cuerpo comparado con el tamaño del alfabeto usado?
2. ¿Cómo se clasifican los criptosistemas en función del tratamiento que hacemos del mensaje a cifrar?
3. ¿Cómo se clasifican los criptosistemas en función de tipo de clave que se usa en ambos extremos, emisor y receptor?
4. ¿Por qué se dice que un sistema es simétrico y el otro asimétrico?
5. ¿Es posible cumplir 100% con la condición del cifrado de Vernam?
6. ¿Por qué en los cifradores de flujo se usa la misma función XOR en el extremo emisor y en el extremo receptor? ¿Son inversas aquí las claves usadas para cifrar y descifrar?
7. Indique y comente algunas debilidades de los sistemas de cifra en bloque con clave secreta.

Cuestiones y ejercicios (2 de 2)

8. Si ciframos un número con la clave pública del usuario receptor, ¿qué cree Ud. que estamos haciendo?
9. ¿Por qué decimos que en un sistema asimétrico la gestión de claves es mucho mejor que en un sistema simétrico?
10. Nos entregan un certificado digital (certificación de clave pública) de 512 bits. ¿Es hoy en día un valor adecuado? ¿Por qué sí o o?
11. ¿Por qué decimos que con un sistema asimétrico es muy fácil generar una firma digital en emisión y comprobarla en destino?
12. Compare los sistemas simétricos y asimétricos en cuanto a su velocidad de cifra.
13. ¿Qué es un cifrado híbrido? ¿Por qué y cómo se usa la cifra híbrida en el intercambio de información segura por ejemplo en Internet?
14. ¿Qué relación hay entre vida de una clave y principio de caducidad?

Tema 9
Sistemas de Cifra en Flujo

Seguridad Informática y Criptografía



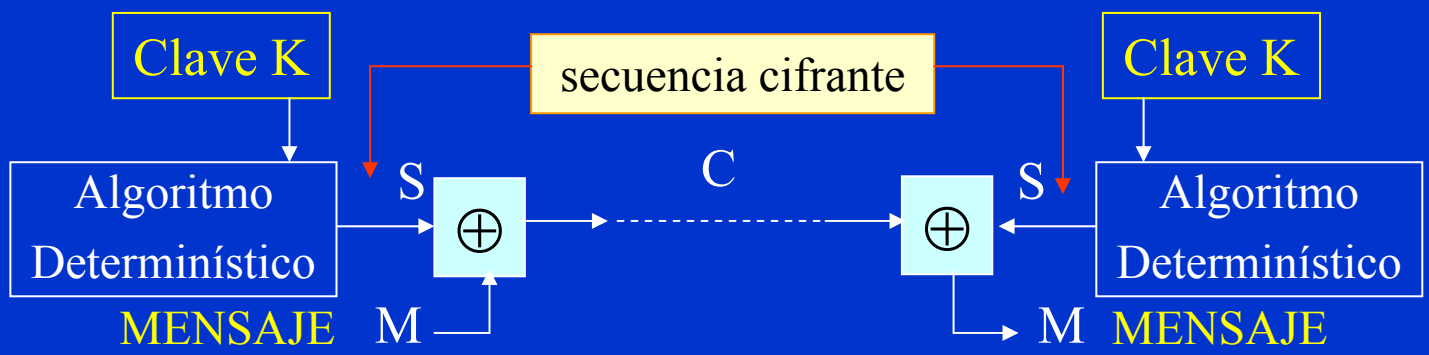
Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 48 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Cifrador de flujo básico

- Siguiendo la propuesta de cifrador hecha en 1917 por Vernam, los cifradores de flujo (clave secreta) usan:
  - Una cifra basada en la función XOR.
  - Una secuencia cifrante binaria y aleatoria S que se obtiene de una clave secreta K compartida por emisor y receptor.
  - Un algoritmo de descifrado que es igual al de cifrado por la involución de la función XOR.



Características de la secuencia cifrante S_i

Condiciones para una clave segura

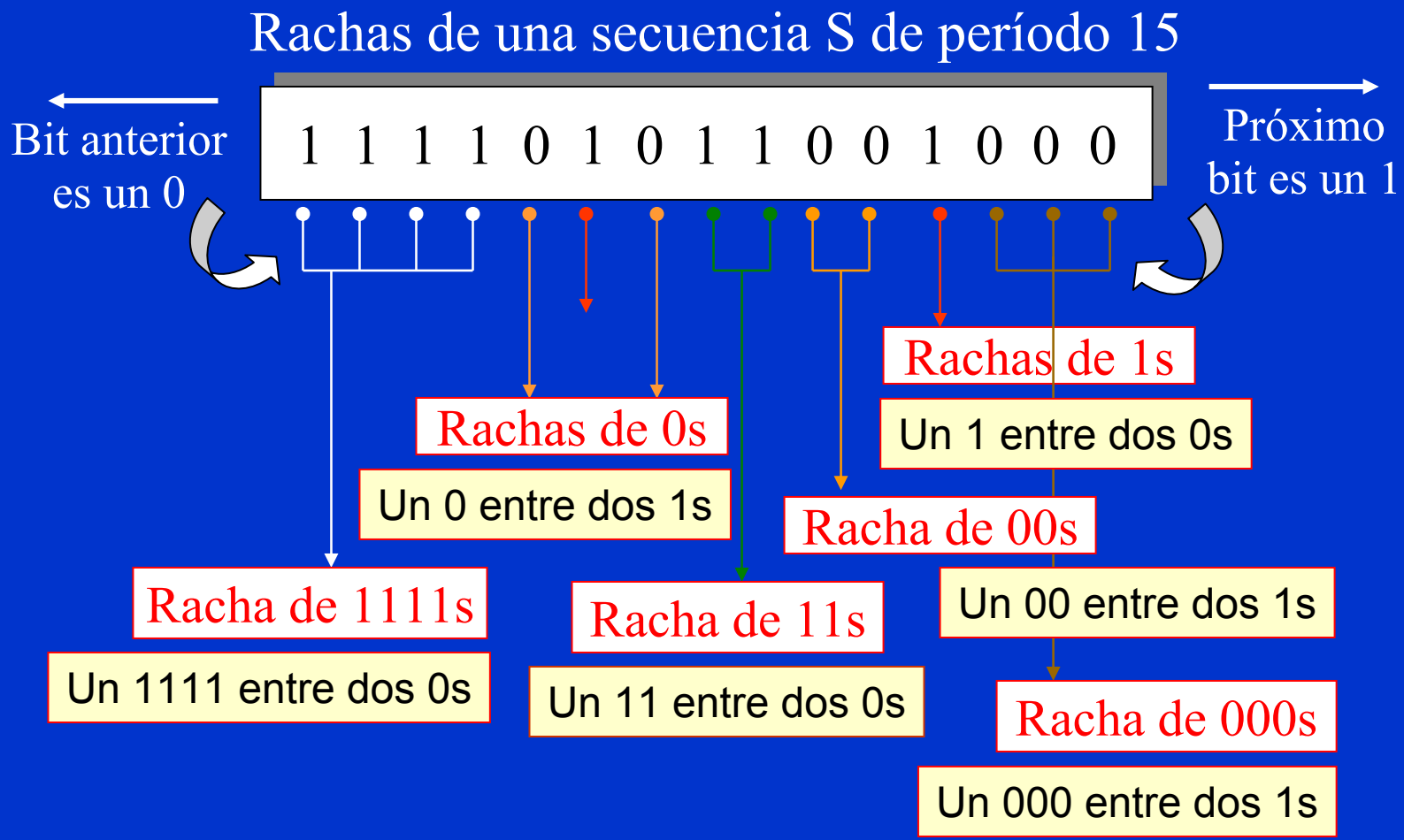
- Período:
  - La clave deberá ser tanto o más larga que el mensaje. En la práctica se usará una semilla de unos 120 a 250 bits para generar períodos superiores a 10^{35}.
- Distribución de bits:
  - Distribución uniforme de unos y ceros que represente una secuencia pseudoaleatoria (Postulados *Golomb*).

Rachas y AC(k)

Rachas de dígitos: uno o más bits entre dos bits distintos.

Función de Autocorrelación Fuera de Fase AC(k): desplazamiento de k bits sobre la misma secuencia S.

Rachas de dígitos en una secuencia



Distribución de las rachas de dígitos

Las rachas, es decir la secuencia de dígitos iguales entre dos dígitos distintos, deberán seguir una distribución estadística de forma que la secuencia cifrante S_i tenga un comportamiento de clave aleatoria o pseudoaleatoria.

Para que esto se cumpla, es obvio que habrá más rachas cortas que rachas largas como en el ejemplo anterior.

Como veremos más adelante, esta distribución seguirá una progresión geométrica. Por ejemplo una secuencia S_i podría tener 8 rachas de longitud uno, 4 de longitud dos, 2 de longitud tres y 1 de longitud cuatro.

Autocorrelación fuera de fase AC(k)

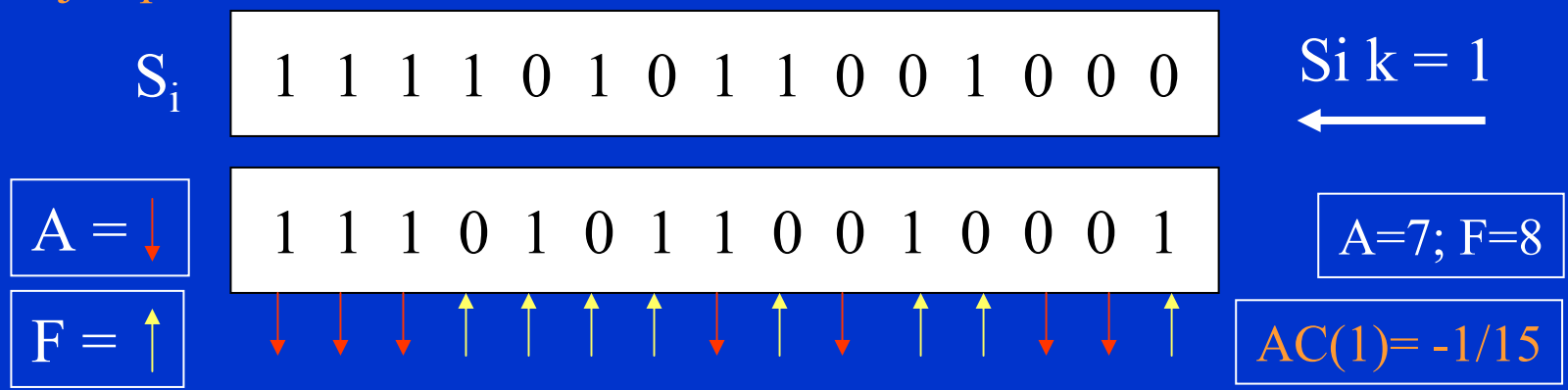
Función de Autocorrelación:

– Autocorrelación AC(k) fuera de fase de una secuencia S_i de período T desplazada k bits a la izquierda:

$$AC(k) = (A - F) / T$$

Aciertos ⇒ bits iguales Fallos ⇒ bits diferentes

Ejemplo



Autocorrelación fuera de fase constante

S_i | 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0

Como ejercicio, compruebe que para esta secuencia cifrante S_i la Autocorrelación Fuera de Fase AC(k) para todos los valores de k $(1 \leq k \leq 14)$ es constante e igual a -1/15. Esto será importante para la calidad de la clave.

Para que una secuencia cifrante sea considerada segura, además de cumplir con la distribución de rachas, deberá tener una AC(k) constante como veremos más adelante.

Imprevisibilidad e implementación de S_i

- Imprevisibilidad:
  - Aunque se conozca una parte de la secuencia S_i, la probabilidad de predecir el próximo dígito no debe ser superior al 50%.
  - Esto se define a partir de la Complejidad Lineal.
- Facilidad de implementación:
  - Debe ser fácil construir un generador de secuencia cifrante con circuitos electrónicos y chips, con bajo coste, alta velocidad, bajo consumo, un alto nivel de integración, etc.

Primer postulado de Golomb G1

Postulado G1:

– Deberá existir igual número de ceros que de unos. Se acepta como máximo una diferencia igual a la unidad.

S_1 | 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0 |

En la secuencia S_1 de 15 bits, hay 8 unos y 7 ceros. Luego sí cumple con el postulado G1.

S_2 | 1 1 0 1 0 1 0 1 0 0 0 1 0 0 0 1 |

En la secuencia S_2 de 16 bits, hay 7 unos y 9 ceros. Luego no cumple con el postulado G1.

Significado del postulado G1

S_i	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0

¿Qué significa esto?

Si una secuencia S_i cumple con G1, quiere decir que la probabilidad de recibir un bit 1 es igual a la de recibir un bit 0, es decir un 50%.

Por lo tanto, a lo largo de una secuencia S_i, independientemente de los bits recibidos con anterioridad, en media será igualmente probable recibir un “1” que un “0”, pues hay una mitad de valores uno y otra mitad de valores cero.

Segundo postulado de Golomb G2

Postulado G2:

– En un período T, la mitad de las rachas de S_i serán de longitud 1, la cuarta parte de longitud 2, la octava parte de longitud 3, etc.

S_i	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0

Las rachas de esta secuencia están en una diapositiva anterior

En la secuencia S_i de 15 bits, hay 4 rachas de longitud uno, 2 rachas de longitud dos, 1 racha de longitud tres y 1 racha de longitud cuatro. Este tipo de distribución en las rachas para períodos impares, es típica de las denominadas *m-secuencias* como veremos más adelante en el apartado generadores LFSR.

Significado del postulado G2

S_i	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0

¿Qué significa esto?

Si una secuencia S_i cumple con G2, quiere decir que la probabilidad de recibir un bit 1 ó 0 después de haber recibido un 1 o un 0 es la misma, es decir un 50%.

Es decir, recibido por ejemplo un "1", la cadena "10" es igualmente probable que la cadena "11". Lo mismo sucede con un 0 al comienzo, un 00, 01, 10, 11, 000, 001, etc. Existe una equiprobabilidad también en función de los bits ya recibidos.

Como comprobaremos más adelante, esto va a significar que la secuencia pasa por todos sus estados, es decir todos sus restos.

Tercer postulado de Golomb G3 (1)

Postulado G3:

- La autocorrelación AC(k) deberá ser constante para todo valor de desplazamiento de k bits.

S_i | 0 1 1 1 0 1 0 0 | ⇐ Secuencia original

⟵ Desplazamiento de un bit a la izquierda

k=1	1 1 1 0 1 0 0 0	AC(1) = (4-4)/8 = 0
k=2	1 1 0 1 0 0 0 1	AC(2) = (4-4)/8 = 0
k=3	1 0 1 0 0 0 1 1	AC(3) = (2-6)/8 = -1/2
k=4	0 1 0 0 0 1 1 1	AC(4) = (4-4)/8 = 0

sigue

Tercer postulado de Golomb G3 (2)

S_i	0 1 1 1 0 1 0 0	Secuencia original
k=5	1 0 0 0 1 1 1 0	AC(5) = (2-6)/8 = -1/2
k=6	0 0 0 1 1 1 0 1	AC(6) = (4-4)/8 = 0
k=7	0 0 1 1 1 0 1 0	AC(7) = (4-4)/8 = 0
k=8	0 1 1 1 0 1 0 0	Secuencia original en fase

La secuencia S_i = 01110100 de 8 bits no cumple con G3.

S_i = 10101100 sí cumple. Compruebe que AC(k) = - ½.

Significado del postulado G3

S_i	0 1 1 1 0 1 0 0	No cumple con G3
S_i	1 0 1 0 1 1 0 0	Sí cumple con G3

¿Qué significa esto?

Si una secuencia cumple con el postulado G3 quiere decir que, independientemente del trozo de secuencia elegido por el atacante, no habrá una mayor cantidad de información que en la secuencia anterior. Así, será imposible aplicar ataques estadísticos a la secuencia recibida u observada al igual como operábamos, por ejemplo y guardando las debidas distancias, con el sistema Vigenère y el ataque de Kasiski.

Generador de congruencia lineal

$x_{i+1} = (a * x_i \pm b)(\text{mod } n)$ secuencia cifrante

- Los valores a, b, n caracterizan al generador y se utilizan como clave secreta.
- El valor x_0 se conoce como semilla; es el que inicia el proceso generador de la clave X_i.
- La secuencia se genera de i = 0 hasta i = n-1.
- Tiene como debilidad que resulta relativamente fácil atacar la secuencia, de forma similar a los cifradores afines de la criptografía clásica.

Ejemplo generador de congruencia lineal

Sea:
$a = 5 \quad b = 1$
$n = 16 \quad x_0 = 10$

$$x_{i+1} = (a * x_i \pm b) \pmod{n}$$

Pero...

$S_i = 10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5$

$x_1 = (5*10+1) \bmod 16 = 3$ $\quad x_2 = (5*3+1) \bmod 16 = 0$
$x_3 = (5*0+1) \bmod 16 = 1$ $\quad x_4 = (5*1+1) \bmod 16 = 6$
$x_5 = (5*6+1) \bmod 16 = 15$ $\quad x_6 = (5*15+1) \bmod 16 = 12$
$x_7 = (5*12+1) \bmod 16 = 13$ $\quad x_8 = (5*13+1) \bmod 16 = 2$
$x_9 = (5*2+1) \bmod 16 = 11$ $\quad x_{10} = (5*11+1) \bmod 16 = 8$
$x_{11} = (5*8+1) \bmod 16 = 9$ $\quad x_{12} = (5*9+1) \bmod 16 = 14$
$x_{13} = (5*14+1) \bmod 16 = 7$ $\quad x_{14} = (5*7+1) \bmod 16 = 4$
$x_{15} = (5*4+1) \bmod 16 = 5$ $\quad x_{16} = (5*5+1) \bmod 16 = 10$

¿Algo falla en este generador?

$$x_{i+1} = (a * x_i \pm b) (\mathrm{mod}\ n)$$

Ejercicios

¿Qué sucede si
$a = 11 \quad b = 1$
$n = 16 \quad x_0 = 7$?

¿Qué sucede si
$a = 5 \quad b = 2$
$n = 16 \quad x_0 = 10$?

¿Qué sucede si
$a = 5 \quad b = 2$
$n = 16 \quad x_0 = 1$?

¿Qué sucede si
$a = 4 \quad b = 1$
$n = 16 \quad x_0 = 10$?

Saque sus propias conclusiones.

Como habrá comprobado, este tipo de generadores de secuencia cifrante no son criptográficamente nada interesantes.

Una vez hecho esto personalmente, pase a la siguiente diapositiva.

Debilidad en este tipo de generadores

$S_i = (11*7 + 1) \bmod 16$

$S_i = 15, 7$

El período que se genera es sólo de tamaño dos ... ☹

$S_i = (5*10 + 2) \bmod 16$

$S_i = 4, 6, 0, 2, 12, 14, 8, 10$

$S_i = (5*1 + 2) \bmod 16$

$S_i = 7, 5, 11, 9, 15, 13, 3, 1$

Se obtiene un período muy bajo y sólo valores pares e impares. El primer caso es igual que el de los apuntes pero con b = 2 ... ☹ ☹

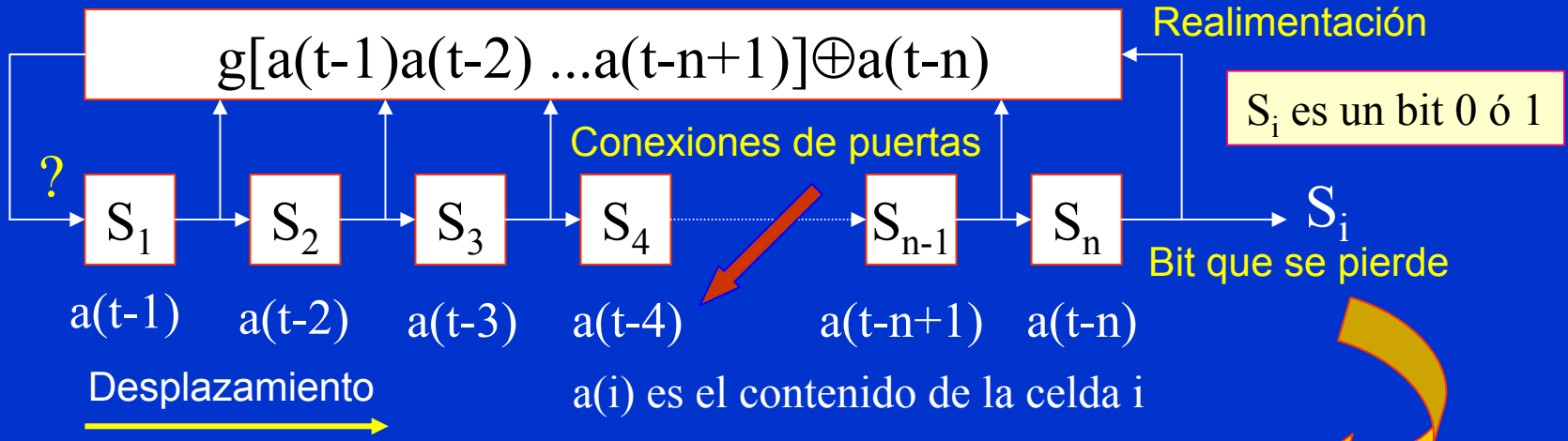
$S_i = (4*10 + 1) \bmod 16$

$S_i = 9, 5, 5, ...$

Peor aún, ya no se genera una secuencia ... ☹ ☹ ☹

Registros de desplazamiento

Generador de secuencia cifrante con registros de desplazamiento



Genera una secuencia con un período máximo 2^n

NLFSR

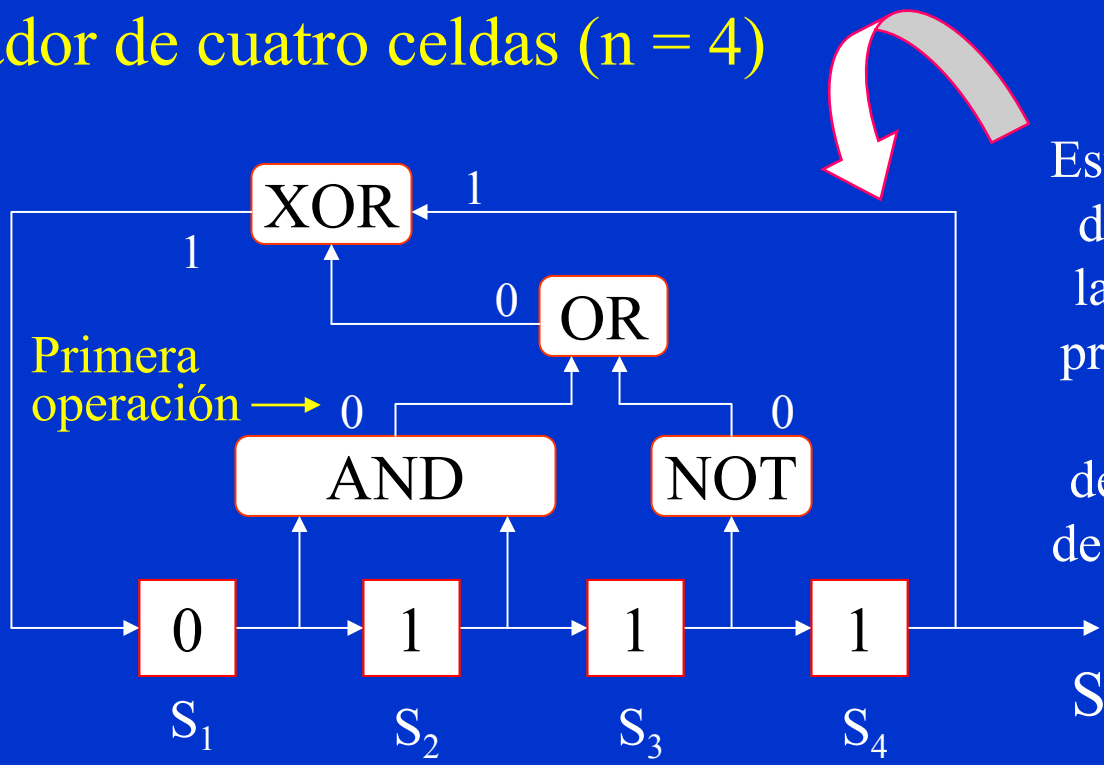
Registros de Desplazamiento Realimentados No Linealmente

Registros de Desplazamiento Realimentados Linealmente

LFSR

Generador NLFSR de 4 celdas (1)

Generador de cuatro celdas (n = 4)

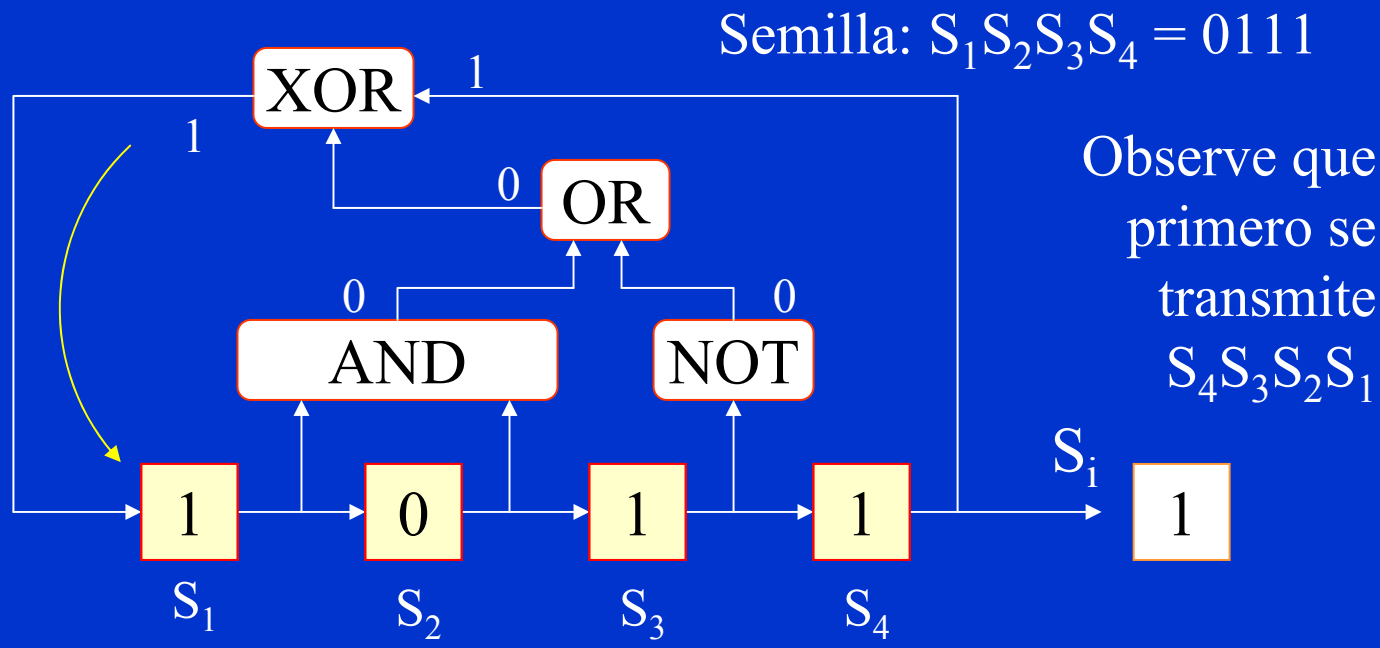


Este es el estado de las celdas y las operaciones previas antes de producirse el desplazamiento de un bit hacia a la derecha.

Sea la semilla: $S_1S_2S_3S_4 = 0111$

Operaciones

Generador NLFSR de 4 celdas (2)



Observe que primero se transmite $S_4S_3S_2S_1$

$S_i = \underline{1110}\ 1100\ 1010\ 0001$. $T_{máx} = 2^n = 2^4 = 16$. Se conoce como secuencia de De Bruijn. El contenido de las celdas pasa por todos los estados posibles: 0000 → 1111.

Generadores lineales LFSR

$$a(t) = C_1a(t-1) \oplus C_2a(t-2) \oplus C_3a(t-3) \oplus \ldots \oplus C_na(t-n)$$

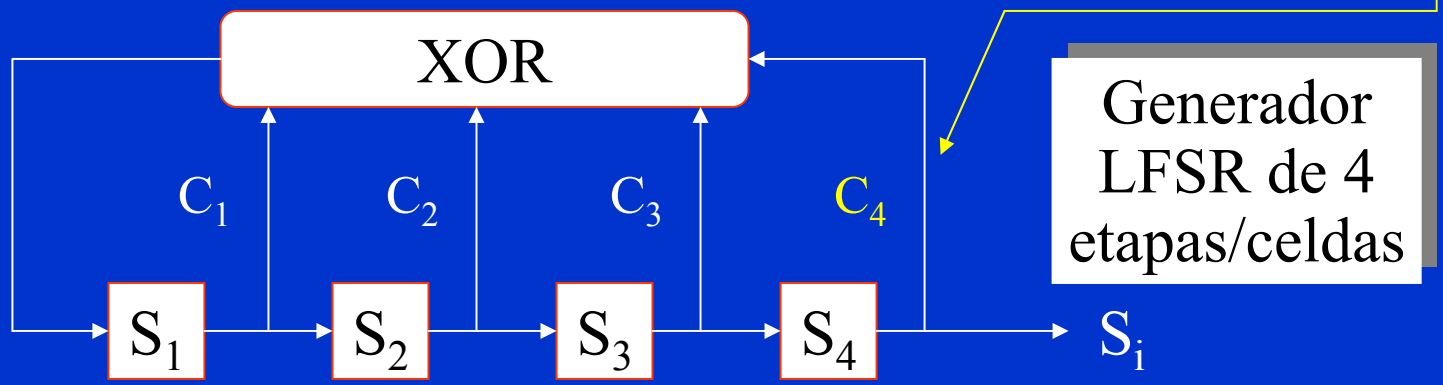
$C_i = \{1,0\} \Rightarrow$ conexión/no conexión celda $\quad C_n = 1$

Función única: XOR

$T_{máx} = 2^n - 1$

Polinomio asociado:

$$f(x) = C_nx^n + C_{n-1}x^{n-1} + \ldots. + C_2x^2 + C_1x + 1$$



Generador LFSR de 4 etapas/celdas

Tipos de generadores lineales LFSR

Observación: como la única función de realimentación de un LFSR es un XOR, no estará permitida la cadena de todos ceros.

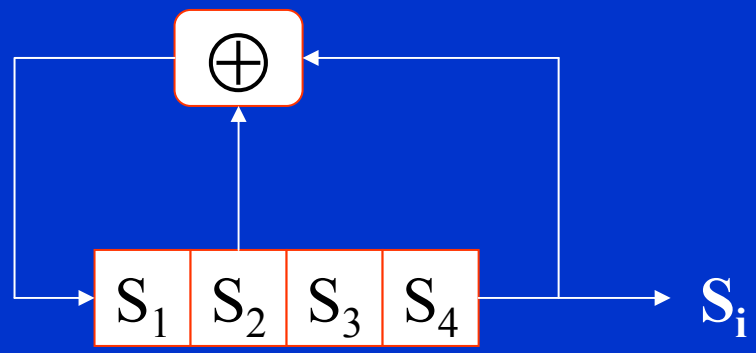
En función del polinomio asociado tendremos:

- LFSR con polinomios factorizables
  - No serán criptográficamente interesantes.
- LFSR con polinomios irreducibles
  - No serán criptográficamente interesantes.
- LFSR con polinomios primitivos
  - Según los postulados de Golomb, este tipo de polinomio que genera todos los estados lineales posibles del cuerpo de trabajo n, será el que nos entregue una secuencia cifrante de interés criptográfico con período $T = 2^n - 1$.

Generador LFSR con f(x) factorizable

Generador f(x) factorizable de cuatro celdas (n = 4)

Sea $f(x) = x^4 + x^2 + 1$



f(x) es factorizable porque:

Sea $f(x_1) = f(x_2) = (x^2+x+1)$

$f(x) = f(x_1) \bullet f(x_2)$

$f(x) = (x^2+x+1) \bullet (x^2+x+1)$

$f(x) = x^4+2x^3+3x^2+2x+1$

Tras la reducción módulo 2

Luego $f(x) = x^4 + x^2 + 1$

Problema

T dependerá de la semilla

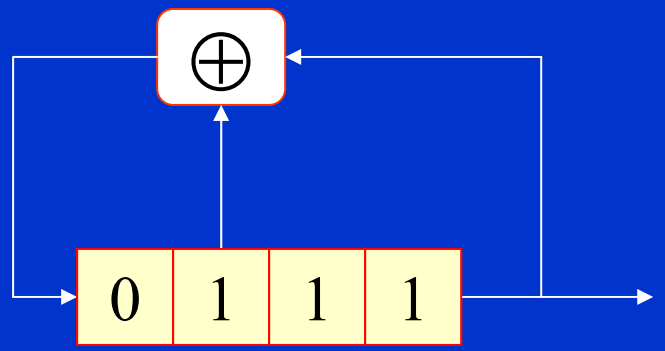
$T \leq 2^n - 1$

Y además, habrá períodos secundarios divisores de T.

Ejemplo de LFSR con f(x) factorizable (1)

$f(x) = x^4 + x^2 + 1$

Sea ahora la semilla:
$S_1S_2S_3S_4 = 0111$



Registro	Bit S_i
0111	1
0011	1
1001	1
1100	0

Registro	Bit S_i
1110	0
1111	1
0111	1

. . . semilla

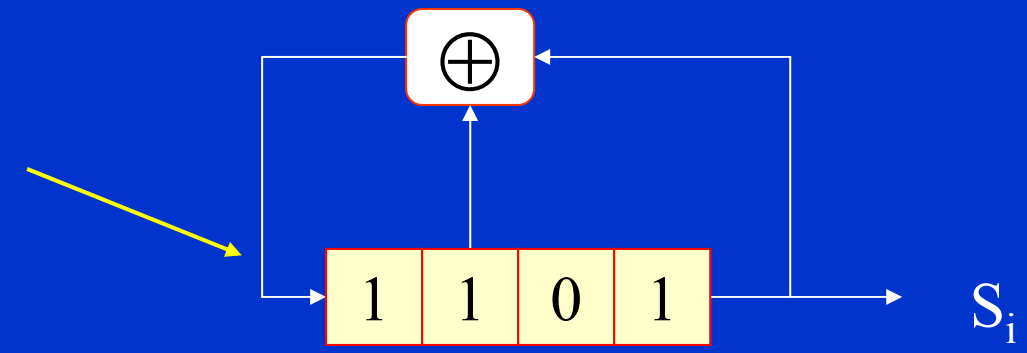
Primer bit: resultado de la operación $S_1 = S_2 \oplus S_4$

$S_i = 111001 \quad T = 6$

Ejemplo de LFSR con f(x) factorizable (2)

$f(x) = x^4 + x^2 + 1$

Sea la semilla:
$S_1S_2S_3S_4 = 1101$



Registro	Bit S_i
1101	1
0110	0
1011	1
1101	1

. . . semilla

Primer bit: resultado de la operación $S_1 = S_2 \oplus S_4$

$S_i = 101$
$T = 3$

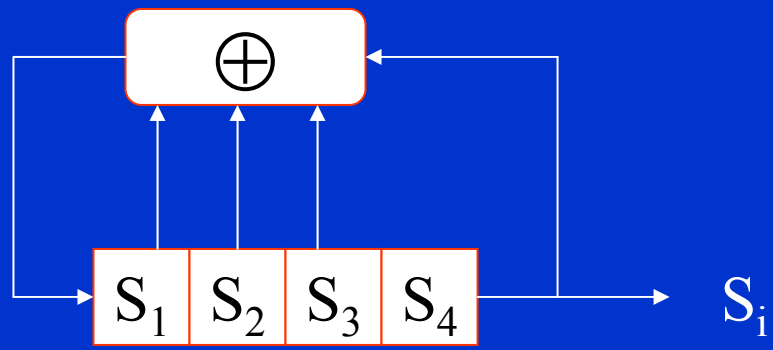
T es un período secundario y en en este caso es incluso menor que la semilla.

Generador LFSR con f(x) irreducible

Generador f(x) irreducible de cuatro celdas (n = 4)

Sea $f(x) = x^4 + x^3 + x^2 + x + 1$

Es imposible factorizar en módulo 2 la función f(x) mediante dos polinomios $f(x_1)$ y $f(x_2)$ de grado menor



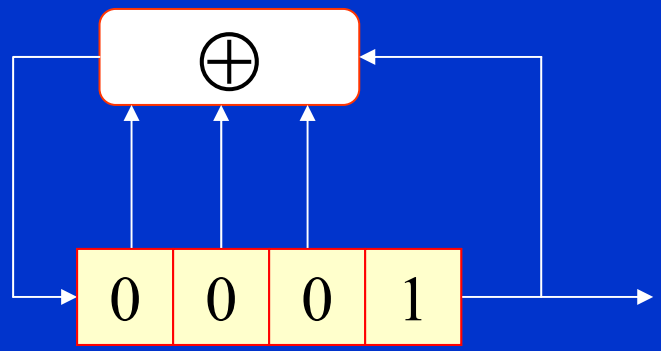
Problema

Ahora T ya no depende de la semilla pero será un factor de $T_{máx} = 2^n - 1$ y no obtendremos un período máximo.

Ejemplo de LFSR con f(x) irreducible

$f(x) = x^4 + x^3 + x^2 + x + 1$

Sea ahora la semilla:
$S_1S_2S_3S_4 = 0001$



Primer bit: resultado de la operación $S_1 = S_2 \oplus S_4$

Registro	Bit S_i
0001	1
1000	0
1100	0
0110	0

Registro	Bit S_i
0011	1
0001	1

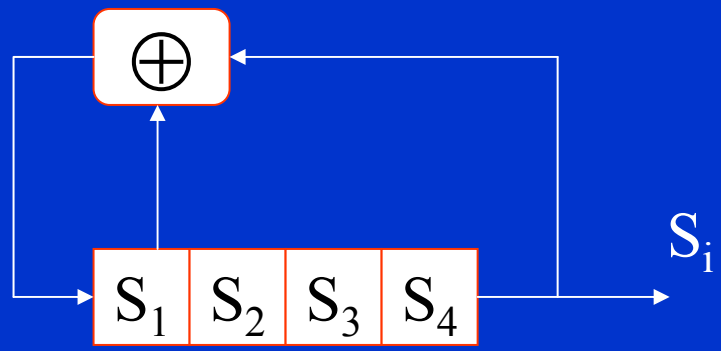
. . . semilla

$S_i = 100011$ $T = 5$ siendo
$T_{máx} = 2^n - 1 = 2^4 - 1 = 15$

Generador LFSR con f(x) primitivo

Generador f(x) primitivo de cuatro celdas (n = 4)

Sea $f(x) = x^4 + x + 1$



f(x) no es factorizable como $f(x_1) \bullet f(x_2)$ en módulo dos. Es además un generador del grupo.

Habrá $\phi(2^n - 1)/n$ polinomios primitivos

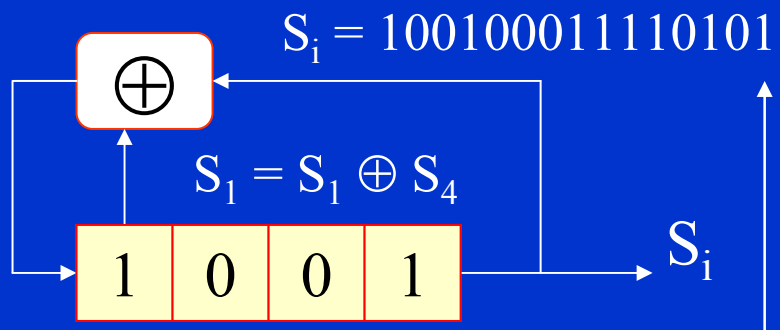
T ya no dependerá de la semilla y será un valor máximo $T_{máx} = 2^n - 1$.
Se generan *m-secuencias*

Ejemplo de LFSR con f(x) primitivo

Generador f(x) primitivo de cuatro celdas (n = 4)

$f(x) = x^4 + x + 1$

$S_1S_2S_3S_4 = 1001$



Registro	Bit S_i				
1001	1				
0100	0	1110	0	1010	0
0010	0	1111	1	1101	1
0001	1	0111	1	0110	0
1000	0	1011	1	0011	1
1100	0	0101	1	1001	T = 15

$T = 2^n - 1$

$T = 2^4 - 1$

$T = 15$

Secuencia S_i de un LFSR con f(x) primitivo

Características

- Secuencia máxima de 2^n - 1 bits
- Cumple con G1:
  - Hay 2n bits 1 y 2n-1 bits 0
- Cumple con G2:

m-secuencia

  - Distribución de rachas de m-secuencia. El vector binario de las celdas pasa por todos los estados excepto la cadena de ceros que está prohibida.
- Cumple con G3:
  - Los aciertos (A) serán iguales a 2^{n-1} - 1

Rachas en S_i de un LFSR con f(x) primitivo

Rachas de Longitud	Rachas de Ceros	Rachas de Unos
1	2^{n-3}	2^{n-3}
2	2^{n-4}	2^{n-4}
...	...	...
p	2^{n-p-2}	2^{n-p-2}
...	...	...
n-2	1	1
n-1	1	0
n	0	1
TOTAL	2^{n-2}	2^{n-2}

Rachas de una m-secuencia

Sin embargo, no es un generador ideal para la cifra porque su Complejidad Lineal es muy baja. →

Debilidad de un LFSR con f(x) primitivo

Como este LFSR genera una secuencia de longitud máxima, ésta será previsible y se puede encontrar la secuencia completa S_i de 2^n - 1 bits ...

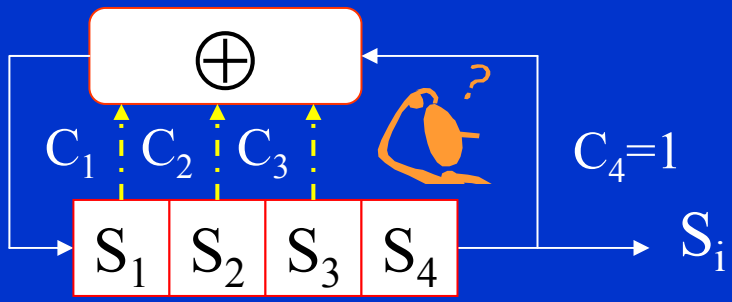
¡ con sólo conocer 2n bits !

Por ejemplo, si en un sistema de 8 celdas con un período 2^8-1 = 255 conocemos 2•8 = 16 bits seremos capaces de encontrar las conexiones de las celdas o valores de C_i y generar así la secuencia completa S_i.

Esta debilidad es la que usa el ataque conocido como algoritmo de Berlekamp-Massey.

Ejemplo de ataque de Berlekamp-Massey

Si conocemos $2*n = 8$ bits $S_1S_2S_3S_4S_5S_6S_7S_8$ de un LFSR de 4 celdas $C_1C_2C_3C_4$, tenemos el sistema de ecuaciones:



$$S_5 = C_1 \cdot S_1 \oplus C_2 \cdot S_2 \oplus C_3 \cdot S_3 \oplus C_4 \cdot S_4$$
$$S_6 = C_1 \cdot S_5 \oplus C_2 \cdot S_1 \oplus C_3 \cdot S_2 \oplus C_4 \cdot S_3$$
$$S_7 = C_1 \cdot S_6 \oplus C_2 \cdot S_5 \oplus C_3 \cdot S_1 \oplus C_4 \cdot S_2$$
$$S_8 = C_1 \cdot S_7 \oplus C_2 \cdot S_6 \oplus C_3 \cdot S_5 \oplus C_4 \cdot S_1$$

Si asignamos valores de esos $2*n = 8$ bits $S_1S_2S_3S_4S_5S_6S_7S_8$ seremos capaces de resolver este sistema

Primero se transmite $S_4S_3S_2S_1$ (semilla) y luego bits $S_5S_6S_7S_8$.

Solución al ataque de Berlekamp-Massey

$$S_5 = C_1 \bullet S_1 \oplus C_2 \bullet S_2 \oplus C_3 \bullet S_3 \oplus C_4 \bullet S_4$$
$$S_6 = C_1 \bullet S_5 \oplus C_2 \bullet S_1 \oplus C_3 \bullet S_2 \oplus C_4 \bullet S_3$$
$$S_7 = C_1 \bullet S_6 \oplus C_2 \bullet S_5 \oplus C_3 \bullet S_1 \oplus C_4 \bullet S_2$$
$$S_8 = C_1 \bullet S_7 \oplus C_2 \bullet S_6 \oplus C_3 \bullet S_5 \oplus C_4 \bullet S_1$$

Si los 8 bits $S_1S_2S_3S_4S_5S_6S_7S_8$ son 1100 1000

$S_1 = 0 \quad S_5 = 1$
$S_2 = 0 \quad S_6 = 0$
$S_3 = 1 \quad S_7 = 0$
$S_4 = 1 \quad S_8 = 0$

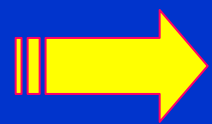
$$1 = C_1 \bullet 0 \oplus C_2 \bullet 0 \oplus C_3 \bullet 1 \oplus C_4 \bullet 1$$
$$0 = C_1 \bullet 1 \oplus C_2 \bullet 0 \oplus C_3 \bullet 0 \oplus C_4 \bullet 1$$
$$0 = C_1 \bullet 0 \oplus C_2 \bullet 1 \oplus C_3 \bullet 0 \oplus C_4 \bullet 0$$
$$0 = C_1 \bullet 0 \oplus C_2 \bullet 0 \oplus C_3 \bullet 1 \oplus C_4 \bullet 0$$

$C_1 = 1 \quad C_2 = 0$
$C_3 = 0 \quad C_4 = 1$

Conclusiones ataque Berlekamp-Massey

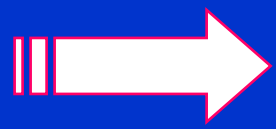
CONCLUSIONES:

- Como se conoce la configuración del generador LFSR, y S_i es una m-secuencia de período $2^n - 1$, entonces por el conjunto de n celdas pasarán todos los restos del campo de Galois de 2^n, excepto la cadena de n ceros que sabemos está prohibida en estos sistemas generadores lineales.
- Para el ejemplo anterior, esto quiere decir que cualquier grupo de 2n = 8 dígitos correlativos nos permite generar la secuencia máxima, en este caso de $2^n = 16$ bits.
- La solución es aumentar la complejidad lineal del generador por ejemplo conectando varios LFRs.

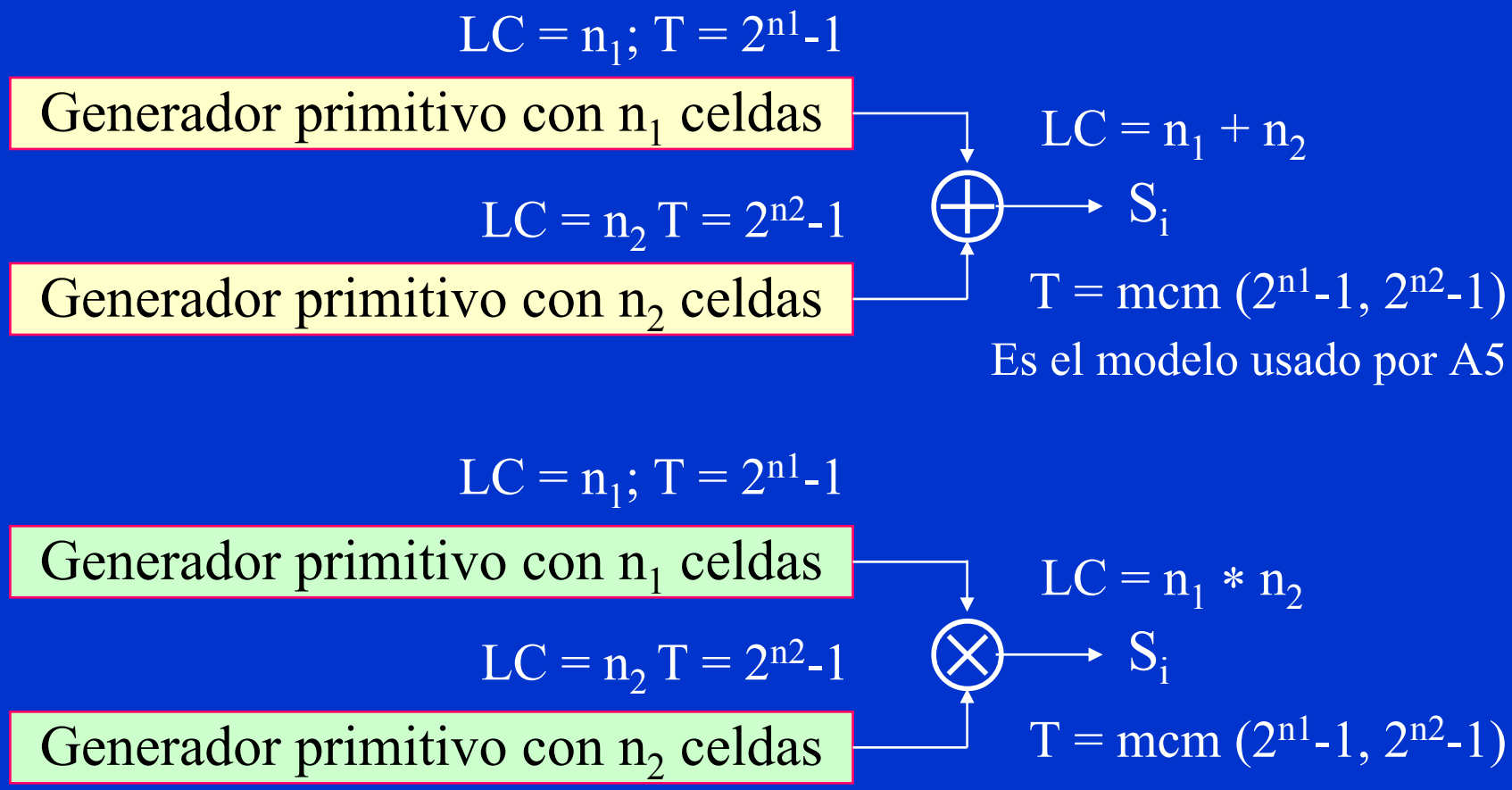


Complejidad lineal LC

- Un LFSR con polinomio primitivo de n celdas tendrá una complejidad lineal LC igual a n; es decir con 2n bits se puede generar la secuencia completa como hemos visto.
- Lo ideal es que si este LFSR entrega una secuencia S_i con un período igual a $2^n - 1$, su LC fuese cercana a este valor.
- Para aumentar esta LC podemos usar:
  - Operaciones no lineales de las secuencias del LFSR
    - Operaciones de suma
    - Operaciones de multiplicación
  - Filtrado no lineal de los estados del LFSR.

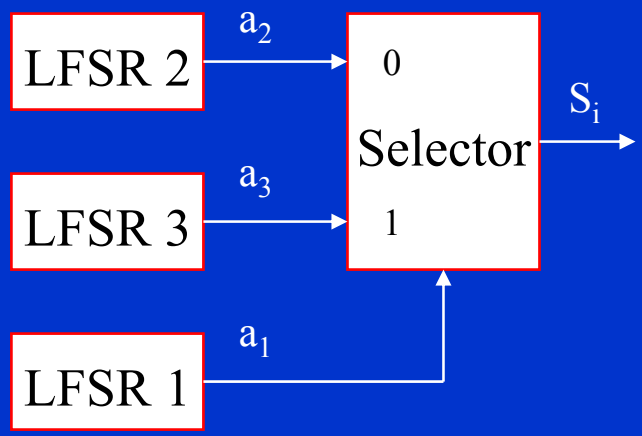


Operaciones no lineales con dos registros



Generadores LFSR con filtrado no lineal

Generador de Geffe



- Si a_1 es un 0 $\Rightarrow$ S_i es el bit de a_2
- Si a_1 es un 1 $\Rightarrow$ S_i es el bit de a_3

Luego: $S_i = a_2 \oplus a_1a_2 \oplus a_1a_3$

$LC = (n_1 + 1)n_2 \oplus n_1n_3$

$T = mcm\ (2^{n1}-1, 2^{n2}-1, 2^{n3}-1)$

Se mejora la LC e incluso se aumenta si ponemos más LFSRs pero este generador es débil ante ataques por correlación de bits.

Existen una infinidad de esquemas en esta línea, entre ellos los de Beth-Piper, Jennings, Gollmann y Massey-Rueppel. Encontrará mayor información consultando la bibliografía del tema 18.

Algoritmos de cifrado en flujo

Sistemas más conocidos:

- A5:
  - Algoritmo no publicado propuesto en 1994. Versiones A5/1 fuerte (Europa) y A5/2 débil (exportación).
- RC4:
  - Algoritmo de RSA Corp. (*Rivest Cipher* #4) desarrollado en el año 1987, usado en Lotus Notes y luego en el navegador de Netscape desde 1999. No es público.
- SEAL:
  - Algoritmo propuesto por IBM en 1994.

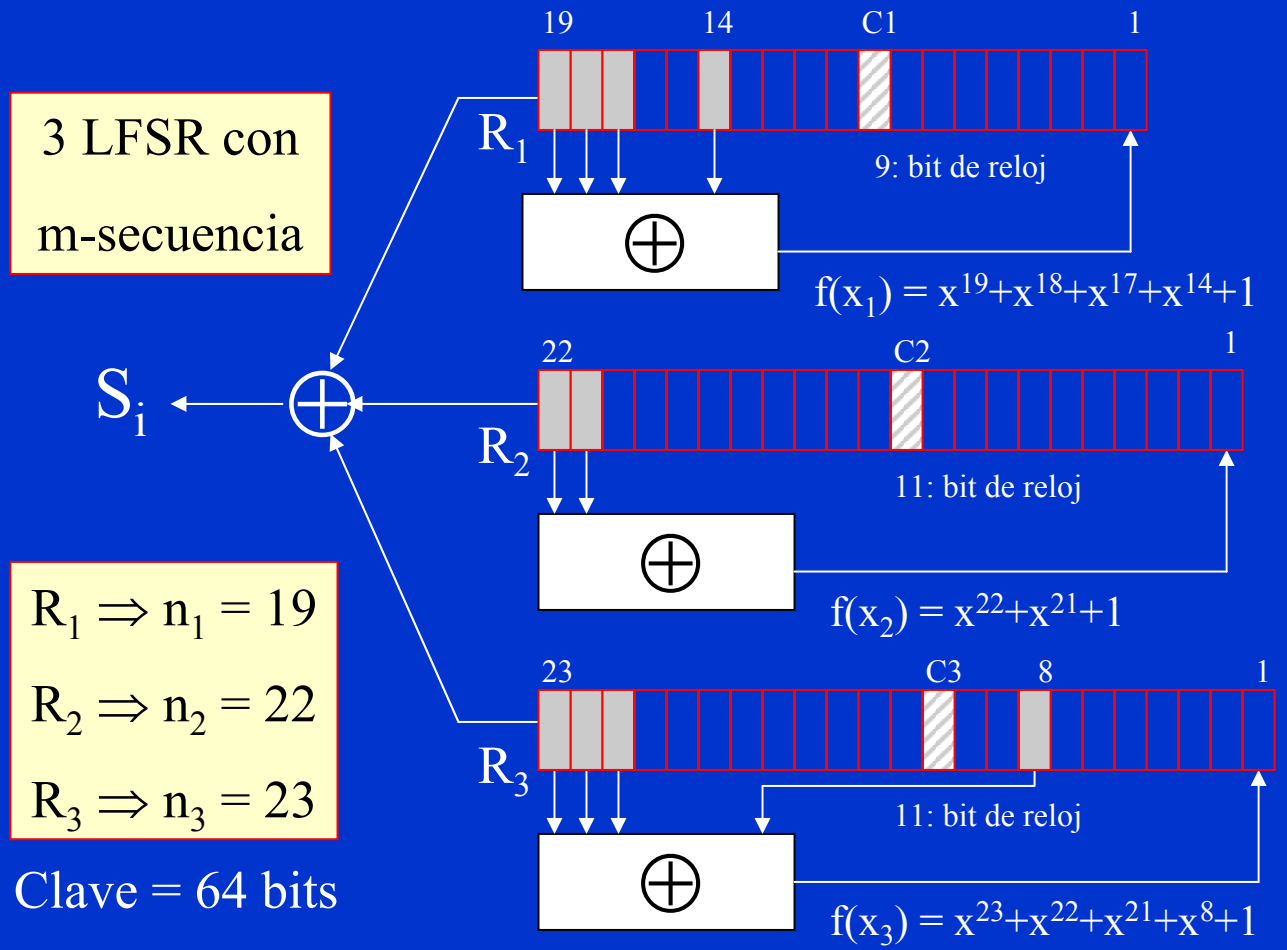
El algoritmo de cifra A5

El uso habitual de este algoritmo lo encontramos en el cifrado del enlace entre el abonado y la central de un teléfono móvil (celular) tipo GSM.

Con cerca de 130 millones de usuarios en Europa y otros 100 millones de usuarios en el resto del mundo, el sistema ha sucumbido a un ataque en diciembre de 1999 realizado por Alex Biryukov y Adi Shamir.

Esta es una consecuencia inevitable en el mundo de la criptografía cuando los desarrolladores de algoritmos no hacen público el código fuente.

Esquema del algoritmo de cifra A5/1



Una función mayoría entre C1, C2 y C3 hace que sólo los registros en los que coincide el bit con ese valor produzcan desplazamiento. En cada paso habrá dos o tres registros en movimiento.

Consideraciones sobre el período de A5/1

El período T vendrá dado por el máximo común múltiplo de los tres períodos individuales:

$$T = mcm\ (2^{n1} - 1, 2^{n2} - 1, 2^{n3} - 1)$$

Como n_1, n_2 y n_3 son primos entre sí, también lo serán los valores $(2^{n1} -1)$, $(2^{n2} - 1)$ y $(2^{n3} - 1)$. Luego el período T será el producto:

$$T = T_1 * T_2 * T_3$$

Entonces $T = (2^{19}-1)(2^{22}-1)(2^{23}-1) = 524.287 * 4.194.303 * 8.388.607$

$T = 18.446.702.292.280.803.327 < 2^{64}$ que es un valor demasiado bajo incluso para mediados de la década pasada. ☹

Registros y función mayoría en A5/2

- Usa los mismos tres registros de desplazamiento con polinomio primitivo que A5/1:
  - $f(x_1) = x^{19} + x^{18} + x^{17} + x^{14} + 1$
  - $f(x_2) = x^{22} + x^{21} + 1$
  - $f(x_3) = x^{23} + x^{22} + x^{21} + x^{8} + 1$
- Además, usa un cuarto registro R_4 con un polinomio primitivo:
  - $f(x_4) = x^{17} + x^{12} + 1$
- Usa cuatro copias de una función mayoría F para cada uno de los cuatro registros que se define como:
  - $F(x_1,x_2,x_3) = x_1x_2 + x_1x_3 + x_2x_3$

Otras operaciones de A5/2

- En R_1 las entradas a la función F_1 son las celdas 13, 15 y 16.
- En R_2 las entradas a la función F_2 son las celdas 10, 14 y 17.
- En R_3 las entradas a la función F_3 son las celdas 14, 17 y 19.
- En R_4 las entradas a la función F_4 son las celdas 4, 8 y 11. La salida de esta copia determina qué registros de R_1,R_2,R_3 se desplazarán en el ciclo.

- Complementación de celdas y sumas en salida de F:
  - En R_1 se complementa la celda 15 y se suma la celda 19 a F.
  - En R_2 se complementa la celda 17 y se suma la celda 22 a F.
  - En R_3 se complementa la celda 14 y se suma la celda 23 a F.

Fin del Tema 9

Cuestiones y ejercicios (1 de 3)

1. ¿Por qué en los sistemas de cifra en flujo se usa una función XOR tanto en emisor como en receptor? ¿Son las claves inversas?
2. Si tenemos una clave de 16 bits, ¿cuál es el período máximo que podremos lograr en una secuencia cifrante? ¿Por qué?
3. ¿Qué rachas encuentra en la secuencia 1101001111010100?
4. ¿Por qué es lógico esperar más rachas cortas que rachas largas?
5. Si en una secuencia cifrante se observa una correlación fuera de fase no constante, ¿qué significa? ¿Podríamos hacer un ataque similar al que permite romper el sistema de cifra polialfabético Vigenère?
6. A nivel de probabilidades de ocurrencia de bits, ¿qué significan los postulados de Golomb G1 y G2?
7. ¿Qué significa que una secuencia cifrante pase por todos los estados o restos posibles? ¿Cuál sería en este caso su período?

Cuestiones y ejercicios (2 de 3)

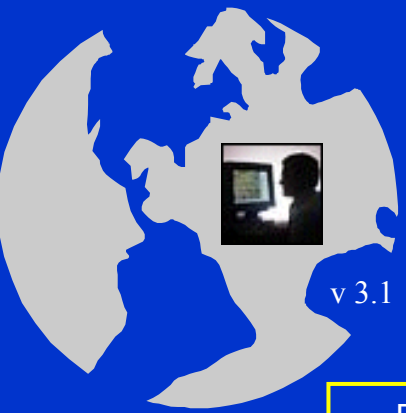
8. ¿Qué secuencias se obtiene con un generador de congruencia lineal en el que a = 7, b = 4 y n = 8? ¿Y si ahora hacemos a = 3?
9. ¿Qué tipo de ataque podríamos intentar para romper una secuencia cifrante obtenida con un generador de congruencia lineal?
10. ¿Por qué en un registro de desplazamiento siempre se realimenta el bit de la última celda, bit que sale a línea?
11. En el generador NLFSR de los apuntes si se cambia la función AND por una OR, ¿qué sucede con la secuencia? Saque conclusiones.
12. Decimos que los generadores LFSR tienen asociado un polinomio $f(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_2 x^2 + a_1 x + 1$, donde a_i toma los valores de 0 ó 1. ¿Por qué f(x) termina en 1?
13. ¿Qué polinomio elegiría para un LFSR, un polinomio factorizable, uno irreducible o uno primitivo? ¿Por qué?

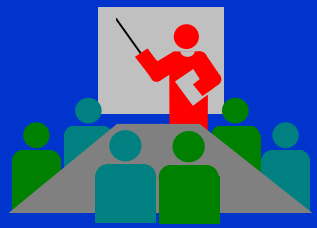
Cuestiones y ejercicios (3 de 3)

14. ¿Por qué no está permitida la cadena de n ceros en un generador LFSR de n etapas y en cambio sí en los NLFSR?
15. En el generador LFSR con f(x) primitivo de 4 etapas, la secuencia pasa por todos los restos excepto 0000. Si usamos hora una semilla distinta, ¿debemos hacer otra vez todos los cálculos o no? ¿Por qué?
16. Justifique la distribución de las rachas en una m-secuencia. ¿Por qué tiene ese comportamiento extraño al final de las rachas largas?
17. ¿Qué debilidad de los sistemas LFSR con polinomios primitivos usa el algoritmo de Berlekamp-Massey para realizar el ataque?
18. En un ataque de Berlekamp-Massey, ¿importa en qué posición de la secuencia se encuentran los 2n bits? ¿Por qué?
19. ¿Por qué cree que el algoritmo A5 es débil? ¿Cuál fue el peor error cometido por sus creadores?

Tema 10
Cifrado Simétrico en Bloque

Seguridad Informática y Criptografía



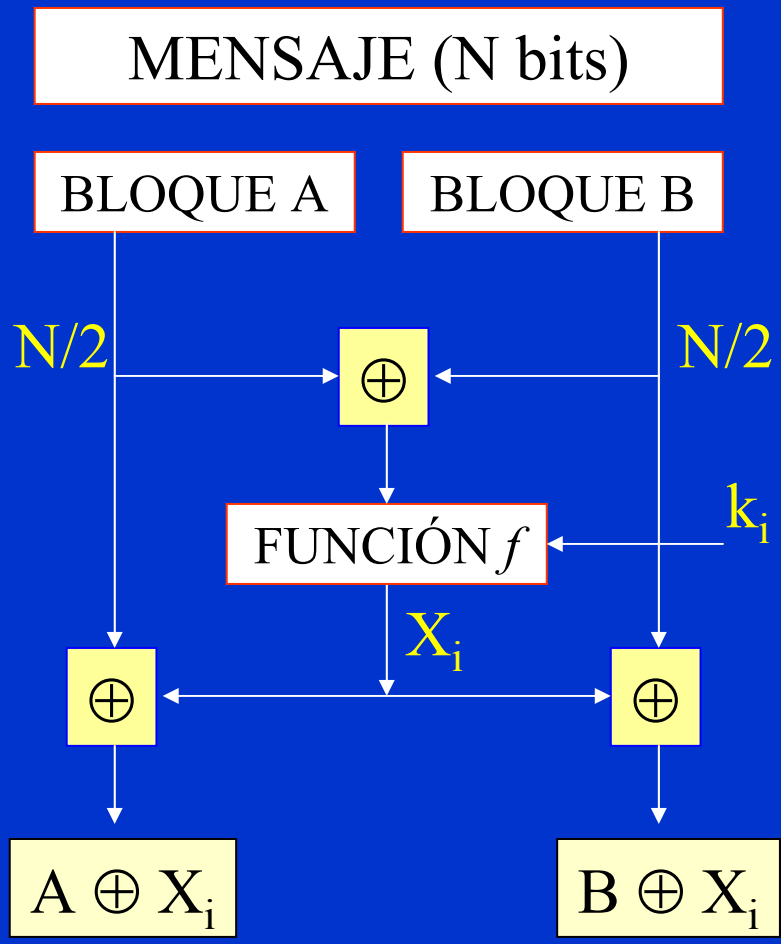


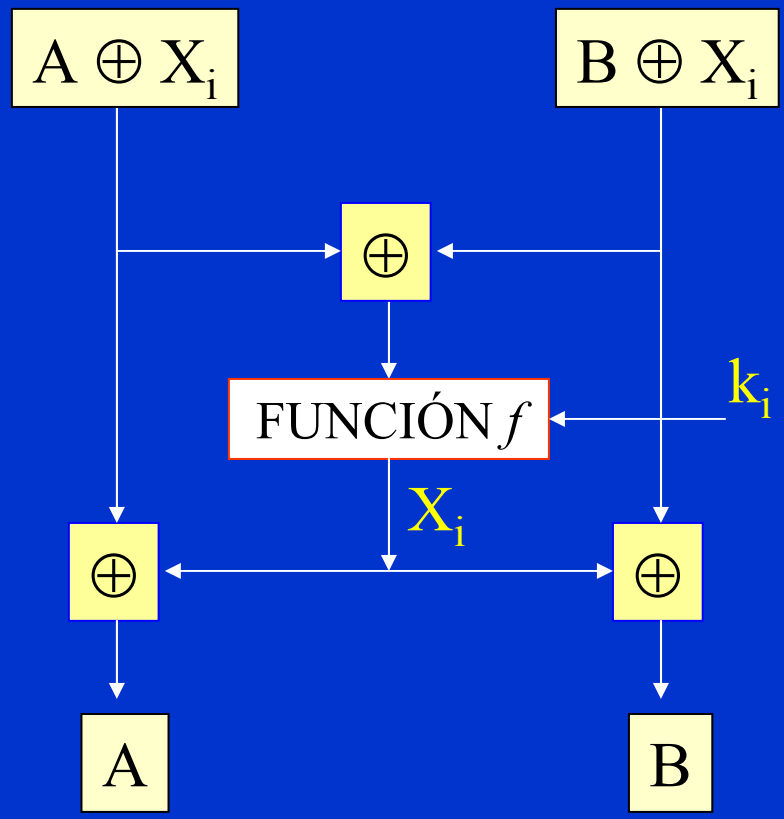
Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 87 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Cifrado y descifrado genérico en bloque

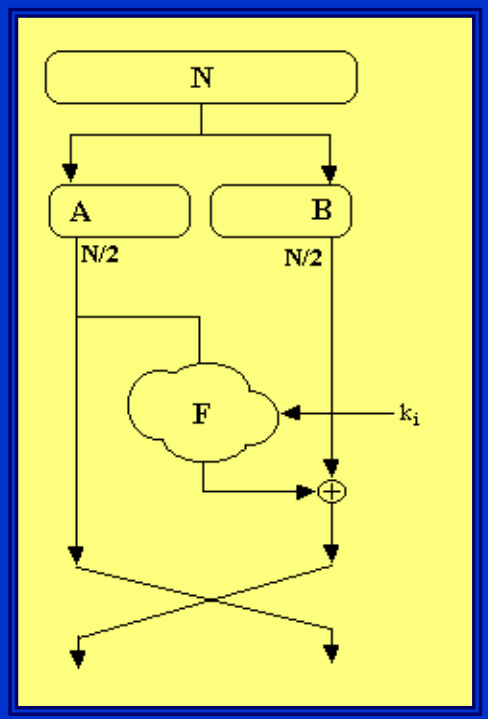




Puesto que $Y \oplus X_i \oplus X_i = Y$

Cifrado tipo Feistel

Horst Feistel: inventor (IBM) del algoritmo LUCIFER a comienzos de los años 70. El algoritmo fue utilizado por el Reino Unido. En 1974 se propone a la NSA como estándar y en ese año dará origen al DES.



- Dado un bloque de N bits (típico 64) éste se dividirá en dos mitades.
- Existirá una función unidireccional F (muy difícil de invertir).
- Se realizan operaciones con la clave k_i sólo con una mitad del bloque, y se permutan en cada vuelta las dos mitades, operación que se repite durante n vueltas.

Un ejemplo básico de cifrado tipo Feistel

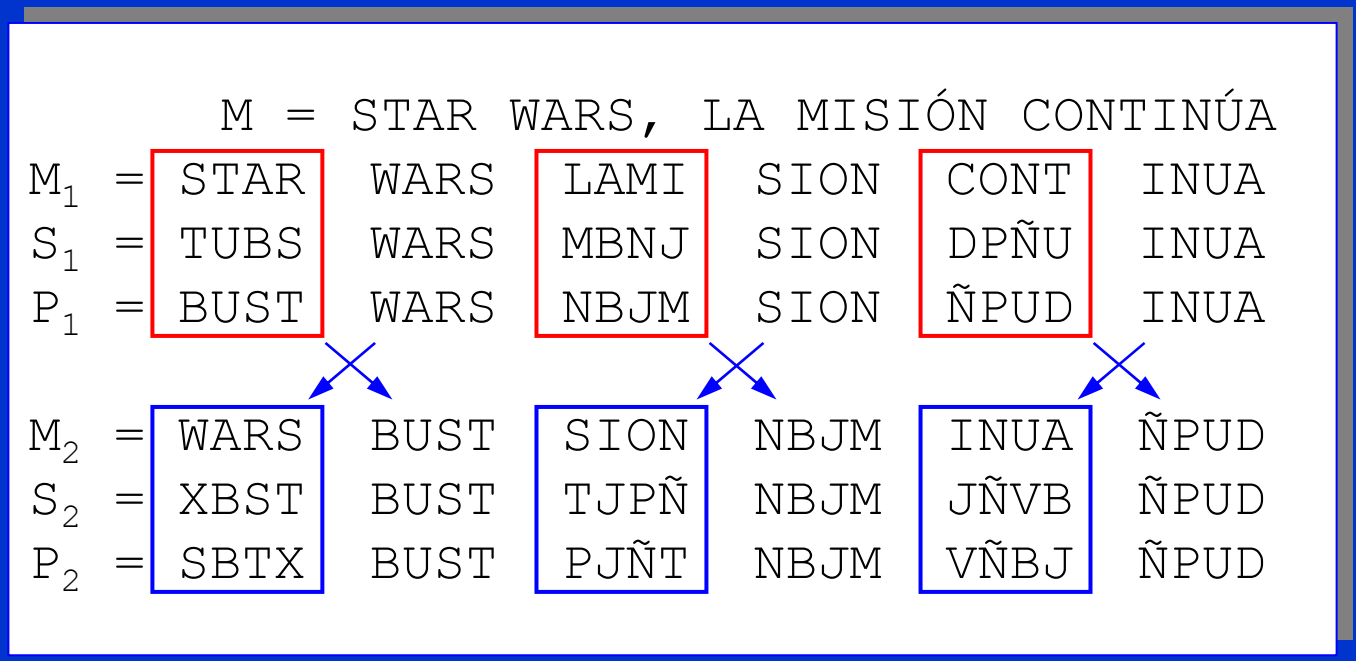
El algoritmo usará bloques de tamaño 8 caracteres. Tendrá dos vueltas y en cada vuelta realizará una operación de sustitución S y una permutación P sobre la 1ª mitad.

Sustitución: $C_i = (M_i + 1) \bmod 27$

Permutación: $C_i = \Pi_{3241}$ (el carácter 1º pasa a la 4ª posición en el criptograma, el 4º a la 3ª, el 2º a la 2ª y el 3º a la 1ª)

Mensaje: M = STAR WARS, LA MISIÓN CONTINÚA

Cifrado tipo Feistel en cuerpo n = 27



S_i: +1 mod 27

P_i: Π_{3241}

Primera vuelta

Segunda vuelta

C = SBTX BUST PJÑT NBJM VÑBJ ÑPUD

Aunque le parezca increíble, el DES hará prácticamente lo mismo trabajando con bits y con funciones un poco más “complejas”.

Cifradores de bloque más conocidos

Algoritmo	Bloque (bits)	Clave (bits)	Vueltas
Lucifer	128	128	16
DES	64	56	16
Loki	64	64	16
RC2	64	variable	--
CAST	64	64	8
Blowfish	64	variable	16
IDEA	64	128	8

Skipjack	64	80	32
RIJNDAEL	128	128 o más	flexible

Características de estos algoritmos

- Lucifer: algoritmo original tipo Feistel que dará lugar al DES.
- DES: algoritmo tipo Feistel que se convirtió en estándar durante casi treinta años. Hoy es vulnerable por su longitud de clave.
- Loki: algoritmo australiano similar al DES, tipo Feistel.
- RC2: algoritmo propuesto por Ron Rivest y que se incluye en navegadores de Internet desde 1999.
- CAST: algoritmo tipo Feistel que se ofrece como cifrador por defecto en últimas versiones de PGP.
- Blowfish: algoritmo tipo Feistel propuesto por Bruce Schneier.
- IDEA: algoritmo europeo usado en el correo electrónico PGP.
- Skipjack: propuesta de nuevo estándar en USA a finales de los 90 para comunicaciones oficiales (tiene puerta trasera).
- RIJNDAEL: nuevo estándar mundial desde finales de 2001.

Otros cifradores de bloque

Algoritmo	Bloque (bits)	Clave (bits)	Vueltas
Twofish	128	variable	variable
Khufu	64	512	16, 24, 32
Khafre	64	128	más vueltas
Gost	64	256	32
RC5	variable	variable	variable
SAFER 64	64	64	8
Akelarre	variable	variable	variable
FEAL	64	64	32

De éstos, los más conocidos son Twofish -uno de los candidatos a AES y que lo encontraremos en últimas versiones de PGP- y RC5.

Características de estos algoritmos

- Twofish: Propuesto por Bruce Schneir después de Blowfish, de tipo Feistel, diseño simple, sin claves débiles y multiplataforma.
- Khufu: algoritmo propuesto por Ralph Merkle con una clave generada con un sistema de "cajas" S.
- Khafre: algoritmo propuesto por Ralph Merkle en el que la clave ya no depende de las cajas S.
- Gost: algoritmo similar al DES con cajas S secretas propuesto en la Unión Soviética.
- RC5: algoritmo propuesto por Ron Rivest; realiza operaciones or exclusivo, suma modular y desplazamiento de bits.
- SAFER 64: algoritmo propuesto por James Massey.
- Akelarre: algoritmo español propuesto en 1996 por el CSIC, Consejo Superior de Investigaciones Científicas.
- FEAL: algoritmo propuesto en Japón.

Algunas tasas de cifra comparativas

Velocidad de cifra de algoritmos en un PC 486 a 33 MHz

Algoritmo	Kbytes/seg	Algoritmo	Kbytes/seg
DES	35	Triple DES	12
IDEA	53	FEAL (32 v)	91
Khufu (16 v)	221	Khufu (32 v)	115
RC5 (8 v)	127	RC5 (16 v)	65
SAFER (6 v)	81	SAFER (12 v)	41
Blowfish (12 v)	182	Blowfish (20 v)	110

Fuente: Criptografía Digital. Fundamentos y Aplicaciones. José Pastor y Miguel Angel Sarasa, Prensas Universitarias de Zaragoza (1998).

Estos son sólo valores indicativos ya que la velocidad del PC es algo baja (extrapolar).

Algoritmos DES, IDEA y AES

Profundizaremos en estas diapositivas en los algoritmos DES, Triple DES, IDEA y RIJNDAEL. ¿Por qué?

- DES es un cifrador de Feistel, ha sido un estándar y en aplicaciones bancarias se seguirá usando durante tiempo.
- DES es de muy fácil comprensión y usa cajas S como varios algoritmos más modernos y el actual estándar AES.
- Triple DES sigue siendo un estándar en e-commerce.
- IDEA es un algoritmo seguro que hace uso de los conceptos de inversos en un cuerpo finito, como todos los algoritmos de cifra modernos, y se usa entre otros en la aplicación PGP.
- RIJNDAEL es el nuevo estándar de cifra avanzada, AES.

Modos de cifra

Todos los algoritmos pueden usarse aplicando diversos modos de cifra, entre ellos:

ECB: Electronic CodeBook (libro electrónico de códigos)
CBC: Cipher Block Chaining (encadenamiento de bloques)
CFB: Cipher FeedBack (realimentación de bloques)
OFB: Output FeedBack (realimentación bloque de salida)

Analizaremos cada uno de ellos para el caso del DES, aunque el estudio es extensible a todos los demás ya que en estos modos el cifrador se considera una caja negra.

Data Encryption Standard DES

DES (Data Encryption Standard) ha sido el estándar utilizado mundialmente durante 25 años, generalmente en la banca. Hoy presenta signos de envejecimiento y ha sucumbido a los diversos criptoanálisis que contra él se viene realizando hace ya años.

FECHAS DE INTERÉS

1973: En EEUU la NBS National Bureaux of Standards llama a concurso público para buscar un algoritmo criptográfico estándar.
1974: La NSA National Security Agency declara desierto el primer concurso, publica unas segundas especificaciones y elige Lucifer, algoritmo original de IBM (años 70) con variaciones.

1976: El DES se adopta como estándar y se autoriza para ser utilizado en las comunicaciones no clasificadas del gobierno.

Especificaciones del algoritmo DES

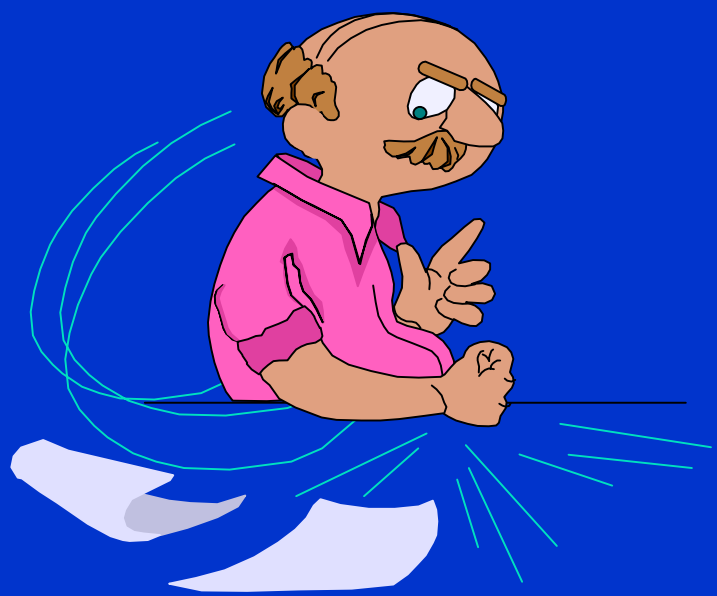
Especificaciones del concurso

- El nivel de seguridad computacional debe ser alto.
- El algoritmo debe ser fácil de entender y deberá estar especificado en todos sus detalles.
- La seguridad del sistema no debe verse afectada por la publicación y divulgación del algoritmo.
- Debe estar disponible para cualquier usuario.
- Deberá poder usarse en diferentes aplicaciones.
- Fabricación con dispositivos electrónicos de bajo costo.
- Se debe poder usar como validación.
- Debe ser exportable.

No se cumplen en 1973 pero sí en 1974, aunque ...

El papel de la NSA en el DES

La NSA impone una limitación en la longitud de la clave:

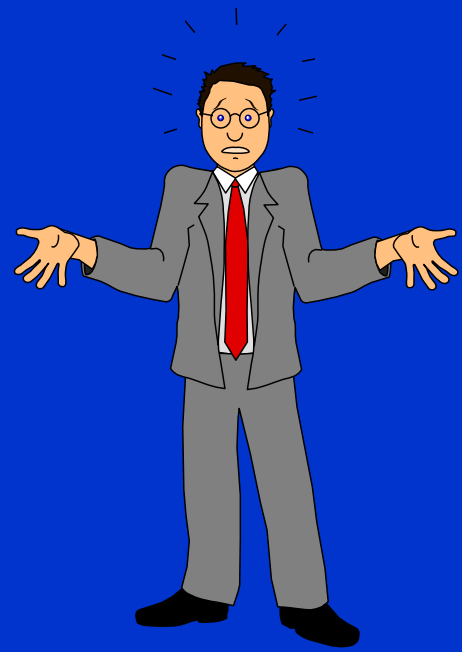


De los 128 bits de Lucifer, NSA deja la clave en 64 bits. Al final, la clave sólo son 56 bits efectivos puesto que al ser datos de 8 bits, se conoce el bit de paridad.

Luego, el espacio de claves es $2^{56} = 7.2 \cdot 10^{16}$, tan sólo 72 mil billones de valores.

¿Por qué esta reducción?

Hay distintas versiones sobre esta reducción del espacio de claves: una habla de la dificultad de diseñar chips capaces de operar de forma eficiente con una clave de 128 bits en esos años; la otra sobre una política de seguridad interna para proteger información sensible ante ataques externos y ser capaces, no obstante, de practicar criptoanálisis en un tiempo razonable.



Especificaciones técnicas finales del DES

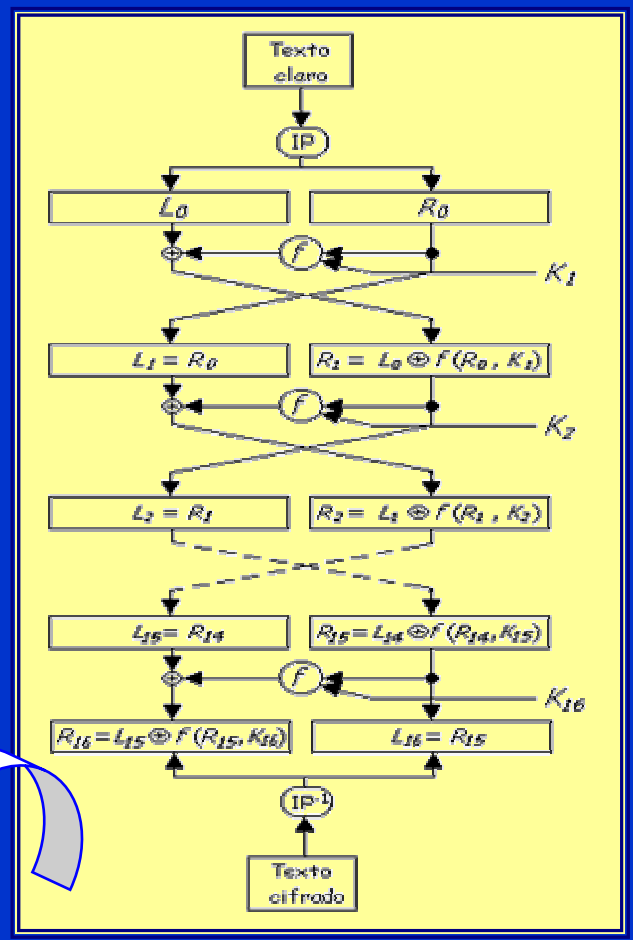
- Bloque a cifrar: 64 bits
- Clave: 8 bytes (con paridad, no caracteres ASCII)
- Normas ANSI:
  - X3.92: Descripción del algoritmo.
  - X3.108: Descripción de los modos de operación (ECB, CBC, OFB).
- Fácil implementación en un circuito integrado.

Veremos su descripción y modos de operación.

Visión general del DES

- Cifrador de bloque
- Tipo Feistel
- Longitud de clave de 56 bits
- Realiza 16 vueltas.
- La cifra del bloque central usa técnicas de sustituciones y permutaciones.
- Para poder realizar las sumas or exclusivo, usará permutaciones con expansión y compresión para igualar el número de bits.

En el descifrado se aplican claves y desplazamientos en sentido inverso



Permutación inicial del DES: tabla IP

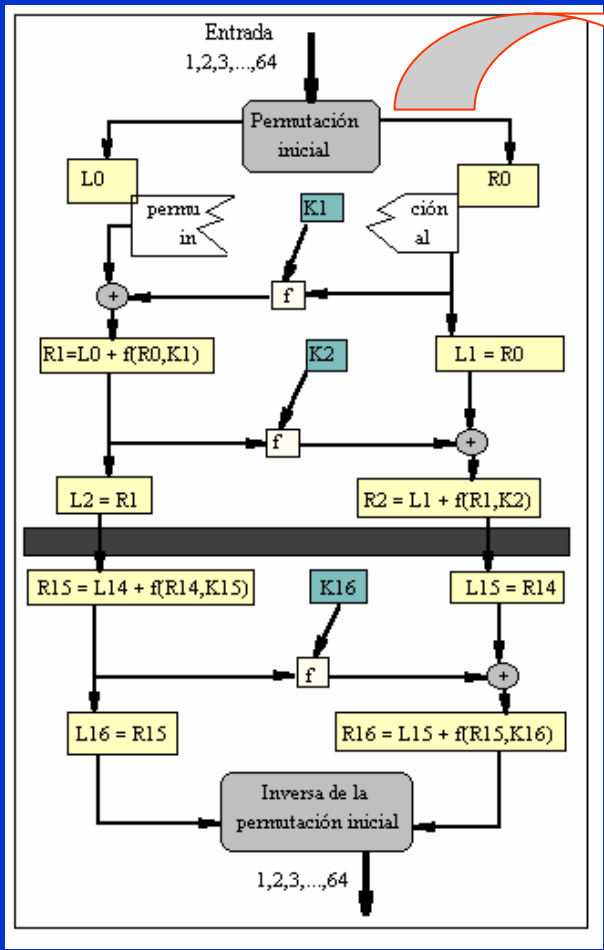


Tabla IP sobre bloque de texto (no tiene interés criptográfico)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

El bit 1 se lleva a la posición 40

Bloques izquierdo y derecho de texto

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

L_0 = 58 50 42 34 26 18 10 02 60 52 44 36
28 20 12 04 62 54 46 38 30 22 14 06
64 56 48 40 32 24 16 08

R_0 = 57 49 41 33 25 17 09 01 59 51 43 35
27 19 11 03 61 53 45 37 29 21 13 05
63 55 47 39 31 23 15 07

Observe la distribución correlativa que existe entre los bits del bloque izquierdo L_0 y del bloque derecho R_0 de texto. Este tipo de distribución de los bits en tablas, a simple vista caprichosa, será muy común en el DES.

Permutación final del DES: tabla IP^{-1}

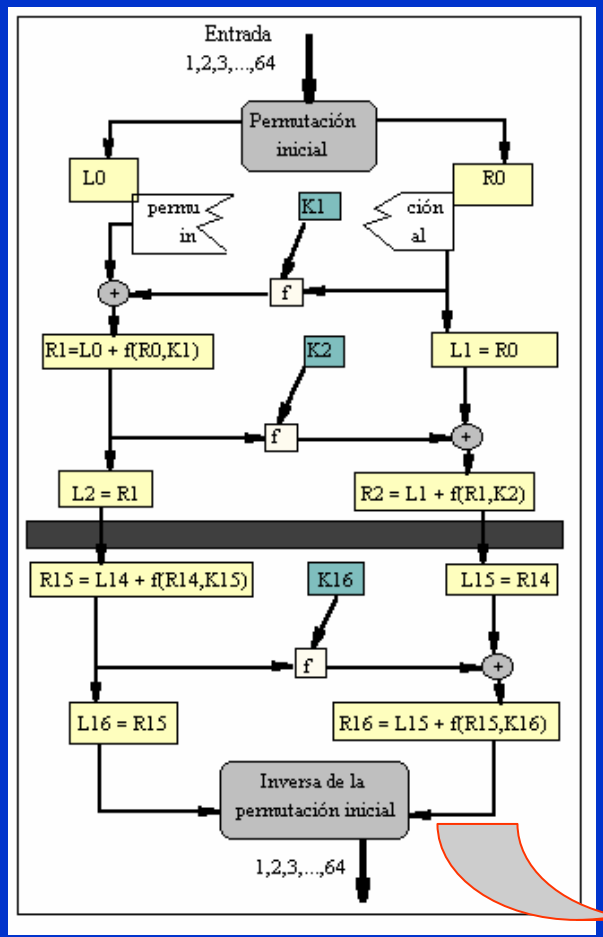


Tabla IP^{-1}

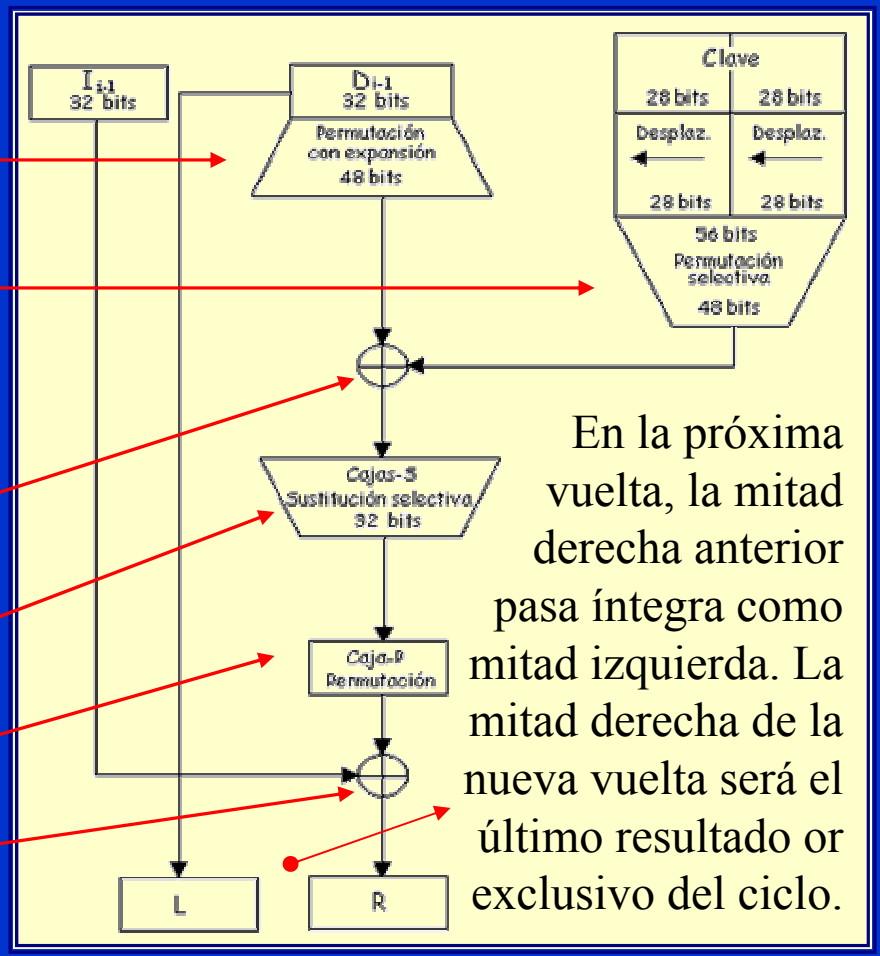
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

El bit 40 vuelve a la posición 1 y todos los demás bits a su posición inicial antes de IP.

Operaciones en cada ciclo del DES

EN CADA CICLO:

- Se permuta la mitad derecha R_i aplicando expansión a 48 bits
- La clave de 56 bits se desplaza, permuta y se seleccionan los 48 bits de K_i
- La nueva mitad derecha R_i y la clave K_i se suman XOR
- Se reducen los 48 bits de salida a 32 bits mediante las Cajas-S
- Se permuta el resultado
- El resultado se suma XOR con la mitad izquierda L_i

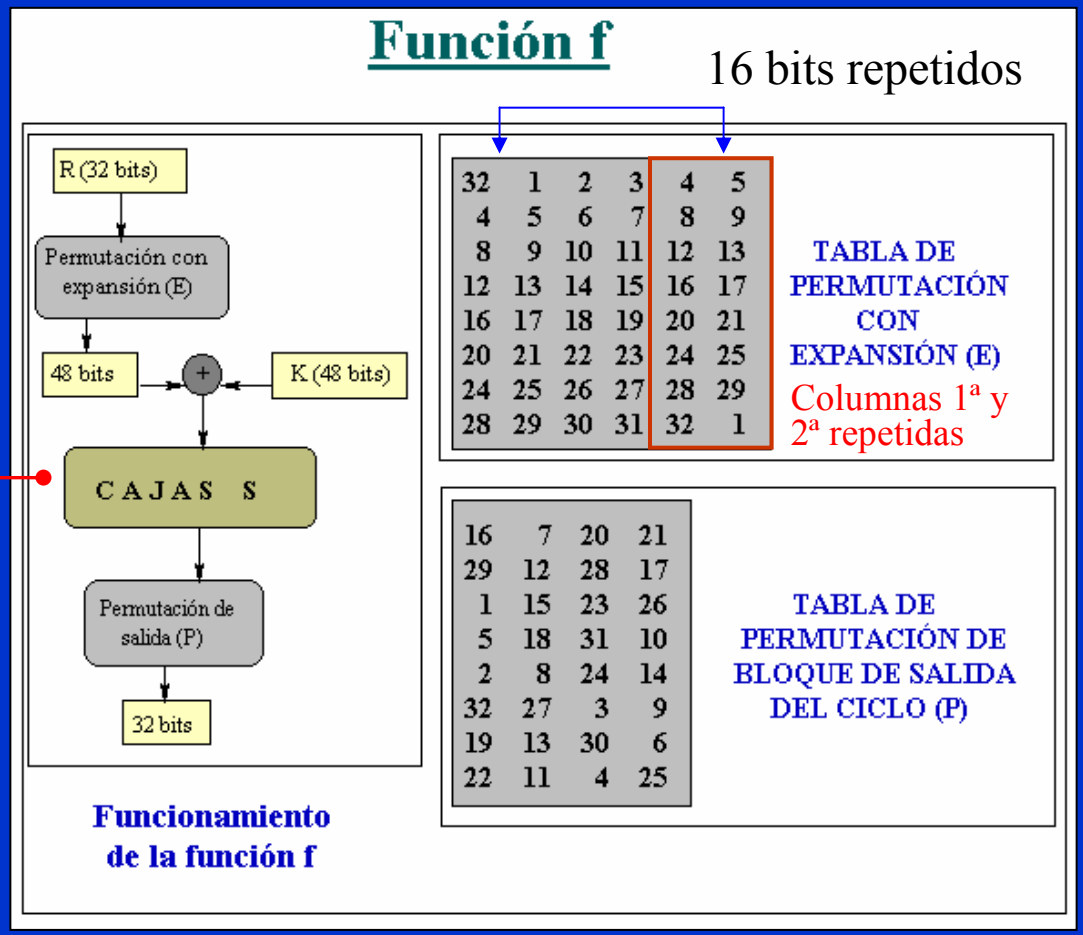


En la próxima vuelta, la mitad derecha anterior pasa íntegra como mitad izquierda. La mitad derecha de la nueva vuelta será el último resultado or exclusivo del ciclo.

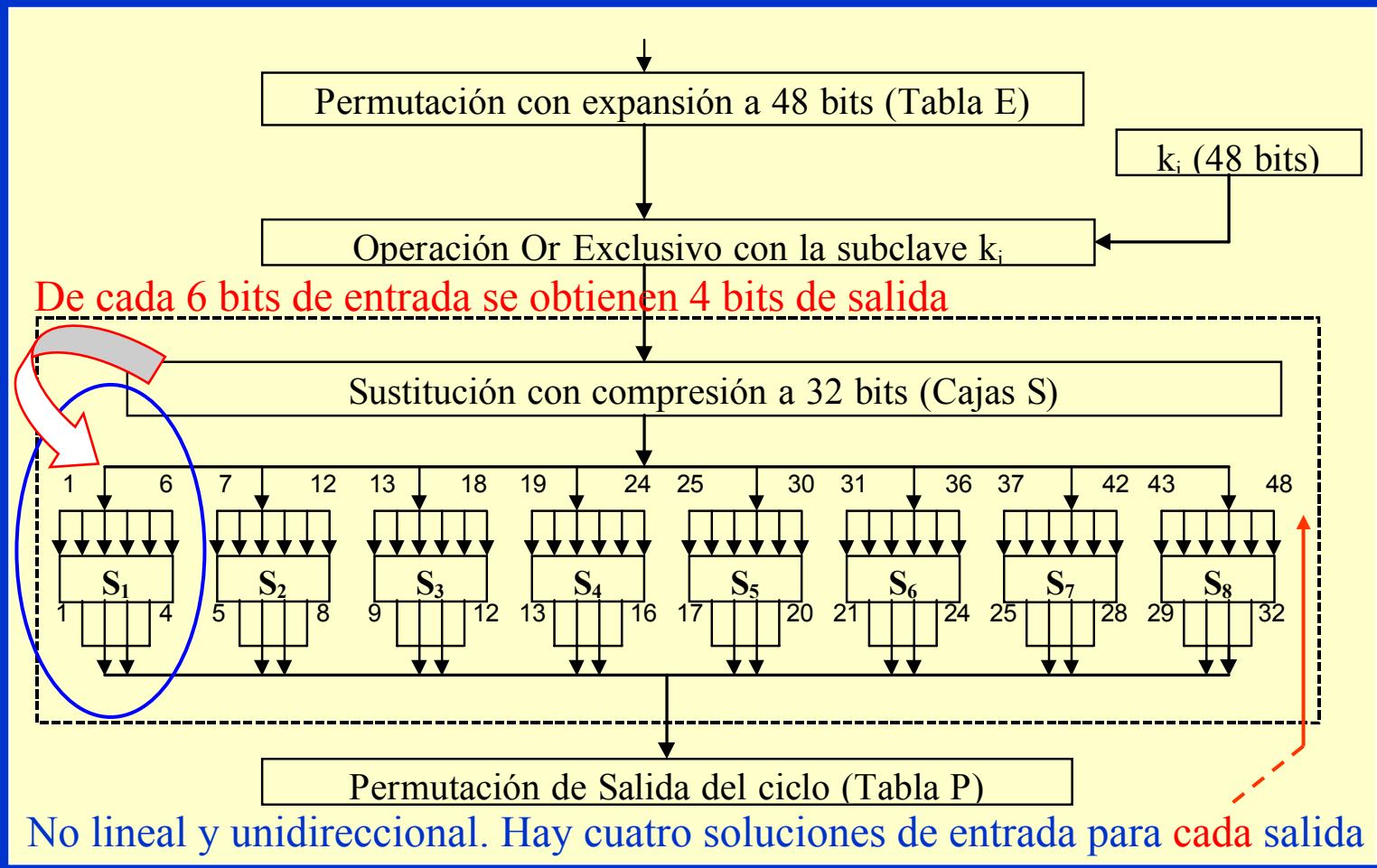
Módulo de cifra en DES

Esquema de la función de cifra f en cada ciclo

En las cajas S se logra la fortaleza del algoritmo. Es una función unidireccional y no lineal.



Operación de las cajas S en el DES



Valores de las cajas S_1 y S_2 del DES

COLUMNAS

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FILAS 0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

COLUMNAS

S_2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FILAS 0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Valores de las cajas S_3 y S_4 del DES

COLUMNAS

S_3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FILAS 0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

COLUMNAS

S_4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FILAS 0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Valores de las cajas S_5 y S_6 del DES

COLUMNAS

S_5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FILAS 0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

COLUMNAS

S_6	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FILAS 0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

Valores de las cajas S_7 y S_8 del DES

COLUMNAS

S_7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FILAS 0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

COLUMNAS

S_8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FILAS 0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Ejemplo de operación de cajas S del DES

Ejemplo:
Sean los bits 7 al 12 los siguientes: 101100
Los bits corresponderán entonces a la entrada de la caja S_2
Para seleccionar la fila tomamos los bits extremos: $10_2 = 2_{10} = 2$
Para seleccionar la columna tomamos los bits centrales: $0110_2 = 6_{10} = 6$
La caja S_2 indica una salida igual a $13_{10} = 1101_2$

explicación

S2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Entrada: 101100
Salida: 1101

Cálculo de subclaves en el DES (PC-1)

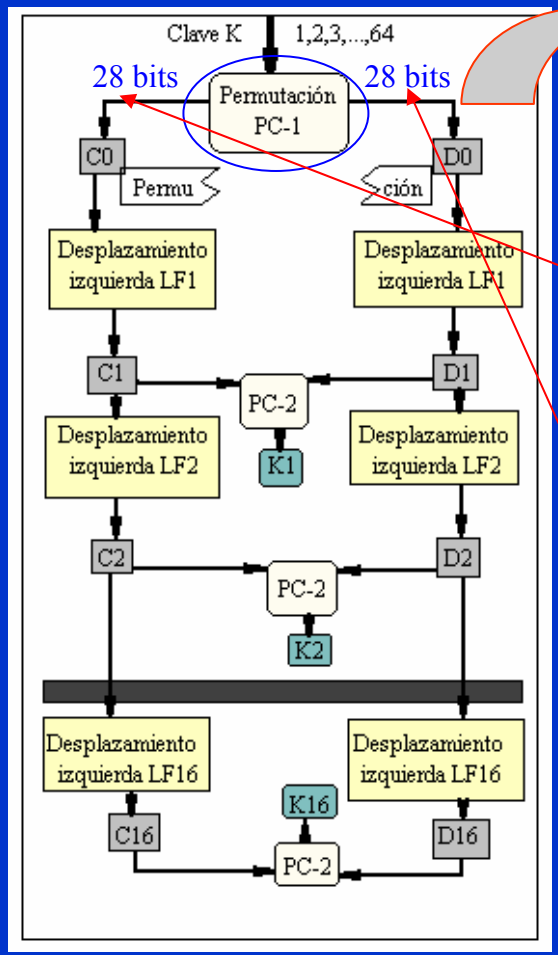


Tabla PC-1 (56 bits)

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Se han eliminado los bits de paridad:
8, 16, 24, 32, 40, 48, 56, 64.

Cálculo de subclaves en el DES (PC-2)

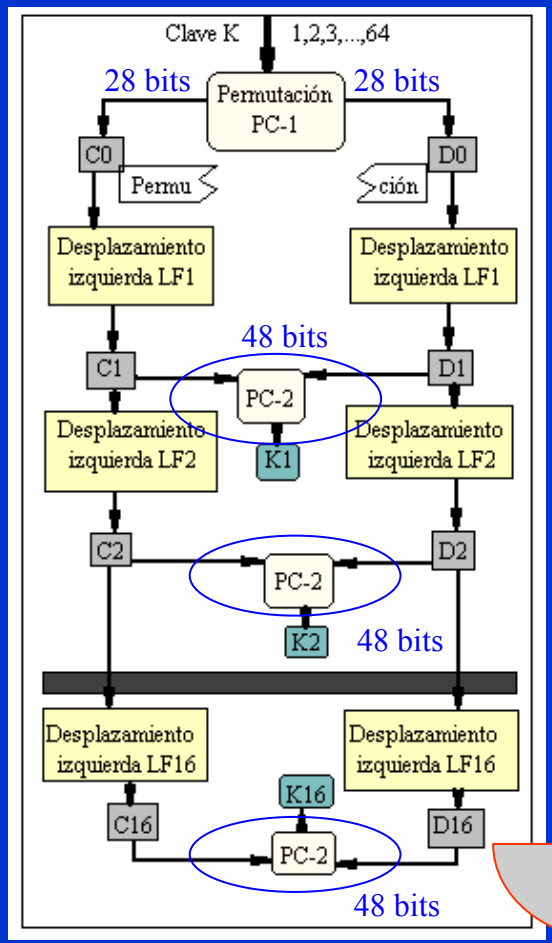
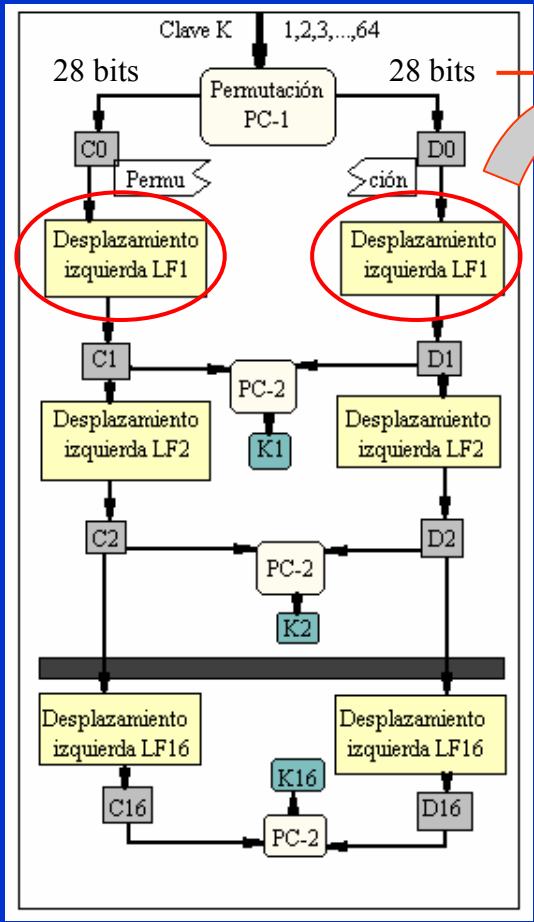


Tabla PC-2 (48 bits) $\Rightarrow k_1, k_2, \ldots, k_{16}$

4	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Se han eliminado los bits:
9, 18, 22, 25, 35, 38, 43, 54.

Desplazamiento de subclaves en el DES



Se produce un desplazamiento total igual a 28, todos los bits de cada bloque Ci y Di

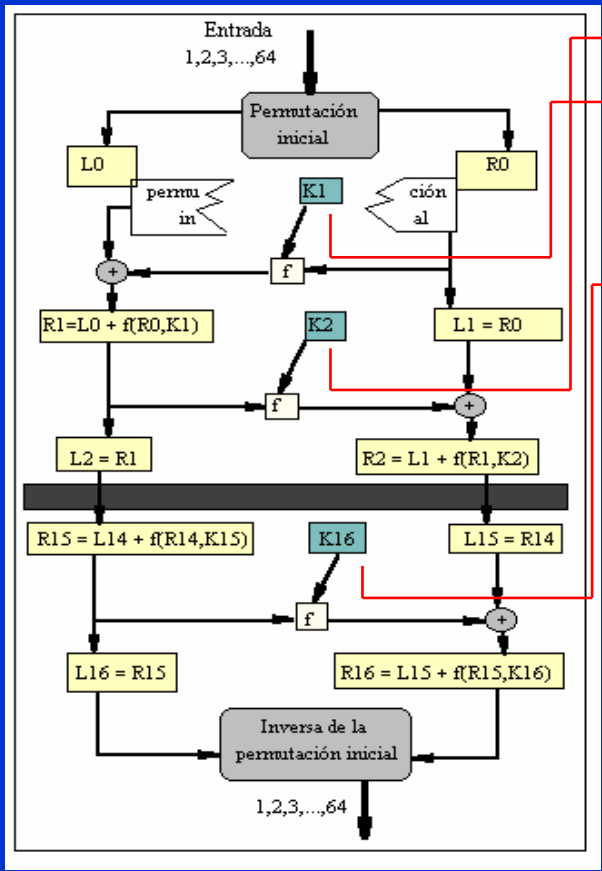
$LF_1, LF_2, \ldots, LF_{16}$

Vuelta i	Bits Desp. Izda.	Vuelta i	Bits Desp. Izda.
1	1	9	1
2	1	10	2
3	2	11	2
4	2	12	2
5	2	13	2
6	2	14	2
7	2	15	2
8	2	16	1

$\sum$

Operación de descifrado en el DES

64 bits de criptograma



Se toman en sentido contrario:

$k_{16}, k_{15}, k_{14}, k_{13}, k_{12}, k_{11}, k_{10},$
$k_9, k_8, k_7, k_6, k_5, k_4, k_3, k_2, k_1$

Como se aplica un desplazamiento de 28 bits en cada bloque de clave, entonces $D_{16} = D_0$ y $C_{16} = C_0$

Los desplazamientos para el cálculo de las subclaves de descifrado son los mismos de la tabla anterior pero ahora se toman hacia la derecha, puesto que los desplazamientos coinciden.

Claves débiles y semidébiles

Claves débiles:

Una clave es débil si se verifica que: $E_k^2(M) = M$

Son claves k débiles estas cuatro:

```
0101010101010101        FEFEFEFEFEFEFEFE
E0E0E0E0F1F1F1F1        1F1F1F1F0E0E0E0E
```

Nota: E0E0E0E0F1F1F1F1 = ààààñññññ (en ANSI).

Compruebe esta clave débil con el software CripMod de la asignatura.

Claves semidébiles:

Una clave es semidébil si se verifica que: $E_{k1}[E_{k2}(M)] = M$

Son claves k_1,k_2 semidébiles las siguientes seis parejas:

```
(01FE01FE01FE01FE, FE01FE01FE01FE01)
(1FE01FE00EF10EF1, E01FE01FF10EF10E)
(01E001E001F101F1, E001E001F101F101)
(1FFE1FFE0EFE0EFE, FE1FFE1FFE0EFE0E)
(011F011F010E010E, 1F011F010E010E01)
(E0FEE0FEF1FEF1FE, FEE0FEE0FEF1FEF1)
```

Ejemplo de cálculo de claves en DES

Caso 1: a partir de las tablas PC-1 y PC-2, encuentre la secuencia de bits de los registros C_1 y D_1 y la subclave k_1.

Solución:

```
i   = 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
C1  = 49 41 33 25 17 09 01 58 50 42 34 26 18 10 02 59 51 43 35 27 19 11 03 60 52 44 36 57
i   = 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
D1  = 55 47 39 31 23 15 07 62 54 46 38 30 22 14 06 61 53 45 37 29 21 13 05 28 20 12 04 63
k1  = 10 51 34 60 49 17 33 57 02 09 19 42 03 35 26 25 44 58 59 01 36 27 18 41
      22 28 39 54 37 04 47 30 05 53 23 29 61 21 38 63 15 20 45 14 13 62 55 31
```

Caso 2: si la clave es PRUEBALO, encuentre C_0 y D_0 y la subclave k_1. Para encontrar C_0 y D_0 escriba en ASCII la clave y elimine el último bit como si éste fuese el de paridad.

Solución:

```
C0 = 0000 0000 1111 1111 0000 0000 0000
C1 = 1001 0010 1100 1100 1100 0000 0111
k1 = 101000 001001 001001 000010 101101 010100 100111 100100
```

Modo de cifra ECB

Lo dicho, estos modos son válidos para todos los cifradores.

MODO ECB

Electronic CodeBook: cifra cada bloque con la clave k de forma independiente. Por lo tanto, el resultado es como si se codificase mediante un gran libro electrónico de códigos.

☞ Recuerde: codificar no es lo mismo que cifrar.

Debilidades:

☹ Se podría reconstruir ese libro electrónico sin necesidad de conocer la clave.

☹ Aparece el problema denominado de comienzos y finales fijos que permiten un tipo de ataque sencillo.

☹ Se ataca a través de la repetición de bloques similares.

Características del modo ECB en DES

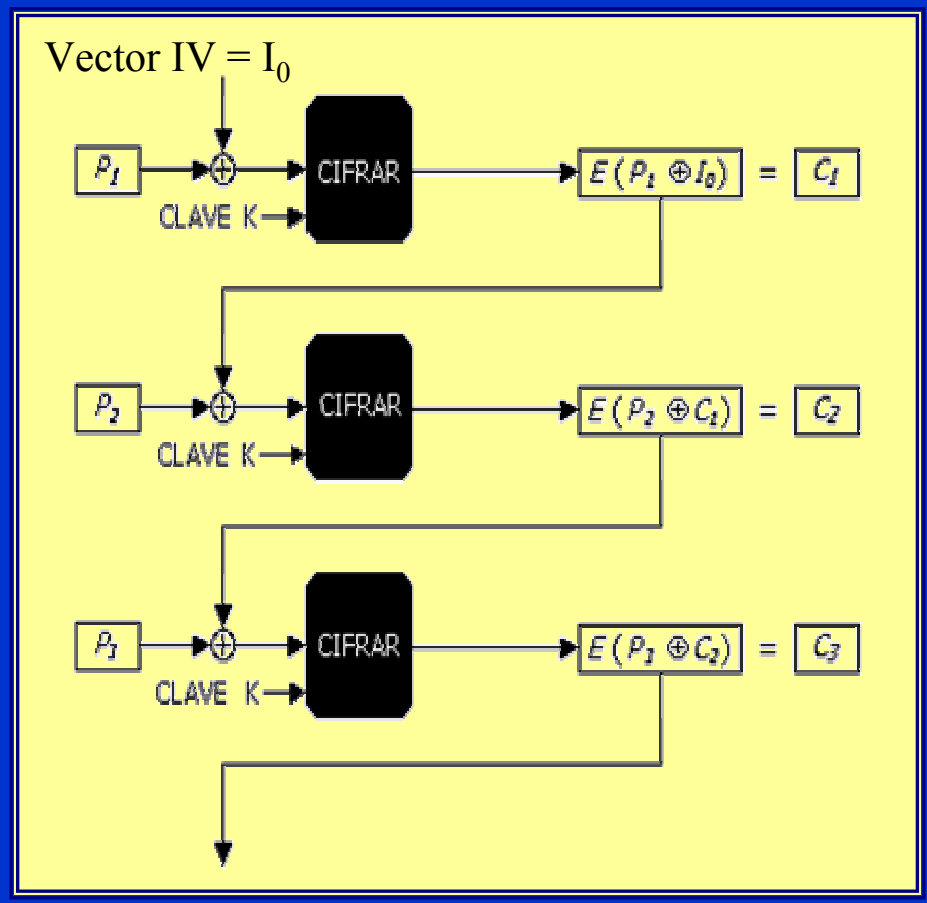
- Cada bloque de 64 bits del texto en claro se pasa por el cifrador, usando la misma clave de 64 bits.
- Para bloques de texto en claro iguales, se obtiene siempre el mismo criptograma
- Como a cada bloque de texto en claro le corresponde un único código o texto cifrado de salida y éste es constante, este modo de cifra lleva por nombre Libro Electrónico de Códigos. Es como si tuviésemos un gran libro de código con un código distinto para cada mensaje.

Modo de cifra CBC en DES

Cipher Block Chaining

Cifra por encadenamiento de bloques

- Se encadenan los bloques de texto en claro con el bloque del criptograma anterior.
- Usa un vector de inicialización IV de 64 bits que se guarda en secreto.



Operaciones de cifra modo CBC en DES

Cifrado

El vector IV se suma XOR a los 64 bits de texto en claro.

Se cifra con la clave K esa suma.

El resultado C_i se usa como vector IV para el nuevo bloque.

Descifrado

Se descifra el primer bloque con vector IV:

$P_1 = D(C_1) \oplus I_0$

$P_1 = D[E(P_1 \oplus I_0)] \oplus I_0$

Se guarda el bloque C_{i-1} en un registro. Se descifra el bloque C_i y luego XOR entre esos bloques:

$M_i = D(C_i) \oplus C_{i-1}$

CARACTERÍSTICAS:

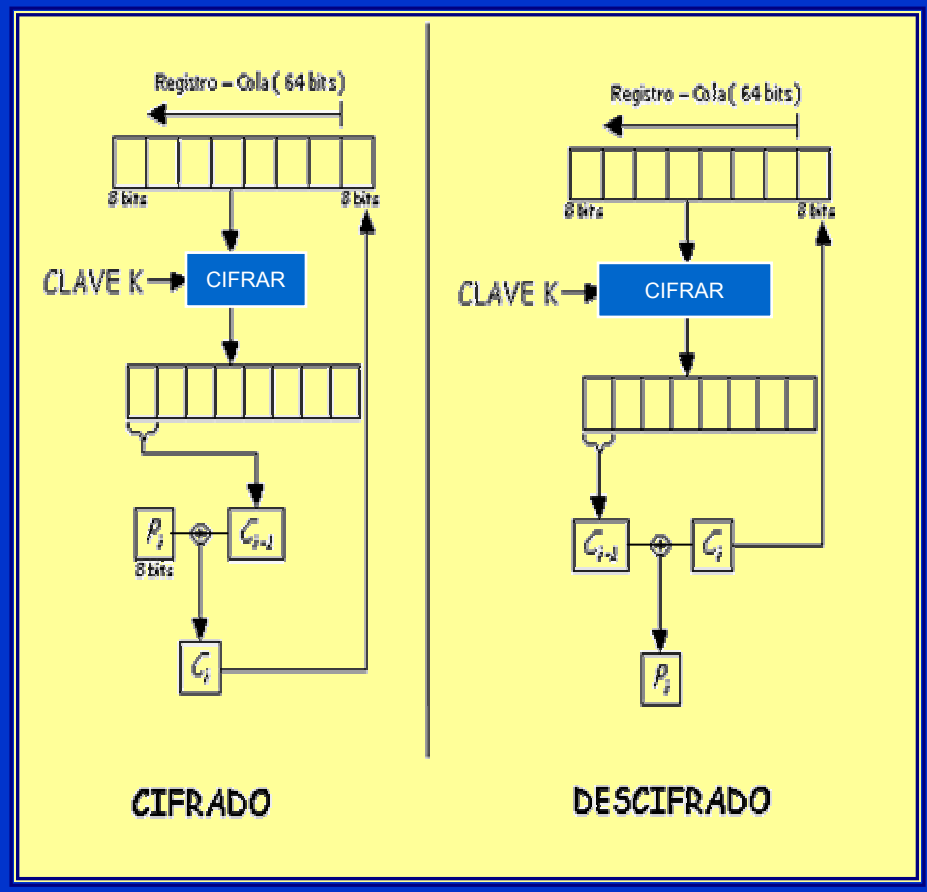
Evita el ataque por repetición de bloque; enmascara el mensaje lo mismo que la cifra en flujo; el espacio de claves es igual a 64 bits; la propagación de un error afecta a dos bloques contiguos.

Modo de cifra CFB en DES

Cipher FeedBack

Cifra por realimentación de bloques

- Se pueden cifrar unidades de datos más pequeñas que bloques, por lo general un byte.
- Se usa un registro de desplazamiento RD de 64 bits como vector inicial IV.



Operaciones de cifra modo CFB en DES

Cifrado

Se suma XOR cada byte del texto claro con bytes resultado de la cifra de RD y la clave K.

El byte C_i se envía al registro; se desplaza 8 bits a la izquierda hasta formar otro RD y se repite el proceso de cifra.

Descifrado

Se cifra el registro RD.

Se obtienen de esta forma los elementos de C_{i-d}.

Se suma XOR los C_{i-d} con los C_i del criptograma para obtener P_i.

Se realimenta C_i al registro RD y se repite el proceso.

CARACTERÍSTICAS:

Evita el ataque por repetición de bloque; enmascara el mensaje como en cifra en flujo, el espacio de claves es igual a 64 bits; la propagación de un error se limita a un bloque.

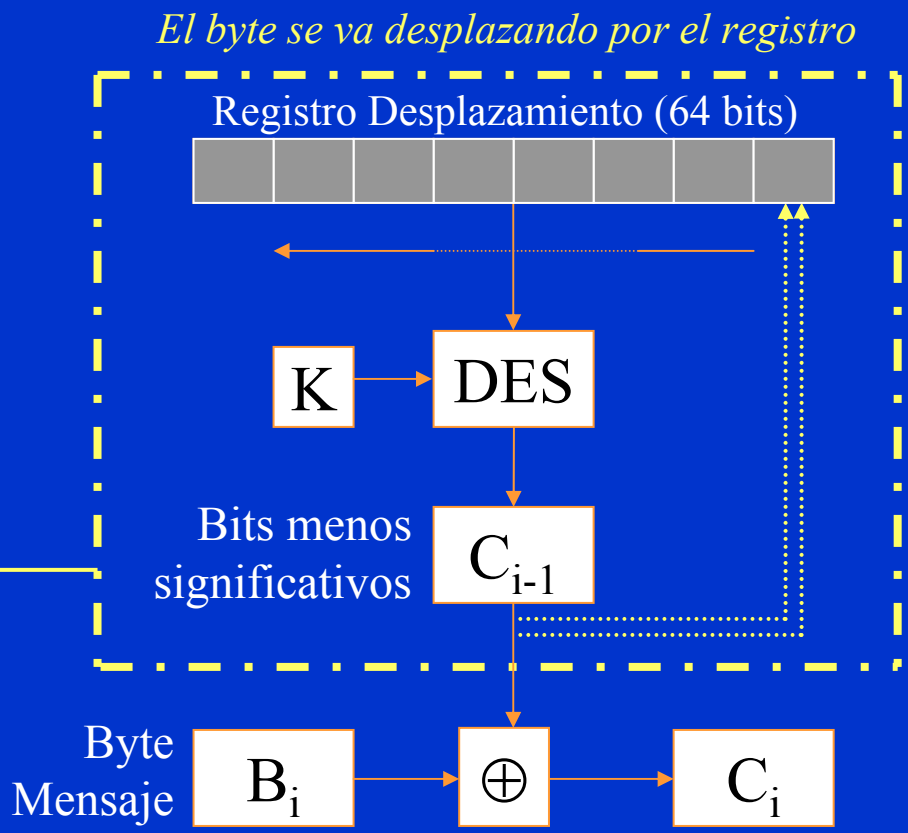
Modo de cifra OFB en DES

Output FeedBack

Cifra por realimentación de bloques de salida

La realimentación de la señal se realiza antes de la operación XOR.

El DES, la clave y el Registro RD actúan como un generador de secuencia cifrante.



Si la cifra se realiza bit a bit, OFB se convierte en cifrador de flujo.

Características del modo OFB en DES

- Evita el ataque por repetición de bloque.
- Produce un enmascaramiento del mensaje similar al de un cifrador de flujo.
- El espacio de claves es igual a 64 bits.
- La propagación de un error afecta sólo a un byte, el que se realimenta en el registro de desplazamiento.
- Las operaciones de cifrado y descifrado son iguales.

A pesar de las propiedades interesantes de los últimos modos, el más utilizado en los sistemas de cifra de diversos protocolos es el CBC.

Cifrado múltiple en un grupo

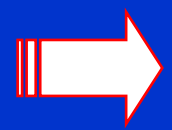
Si un sistema forma un grupo, entonces cifrar un mensaje M con una clave k_1 y luego el resultado con una clave k_2, es lo mismo que cifrar el mensaje con una única clave k_3.

Por ejemplo, el cifrador de Vigenère es un grupo como se demuestra a continuación. Sea k_1 = PACO y k_2 = CINE y el mensaje a cifrar M = ESTO ES UN GRUPO.

M_1	=	ESTO	ESUN	GRUP O
k_1	=	PACO	PACO	PACO P
C_1	=	TSVD	TSWB	VRWE E

M_2	=	TSVD	TSWB	VRWE E
k_2	=	CINE	CINE	CINE C
C_2	=	VAIH	VAJF	XZJI G

Obtendremos lo mismo si ciframos el mensaje M con la clave $k_3 = k_1 + k_2$ = PACO + CINE = RIOS.

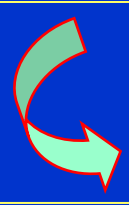


El DES no es un grupo

M_1 = ESTO ESUN GRUP O M_2 = TSVD TSWB VRWE E

k_1 = PACO PACO PACO P k_2 = CINE CINE CINE C

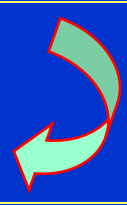
C_1 = TSVD TSWB VRWE E C_2 = VAIH VAJF XZJI G



M_3 = ESTO ESUN GRUP O

k_3 = RIOS RIOS RIOS R

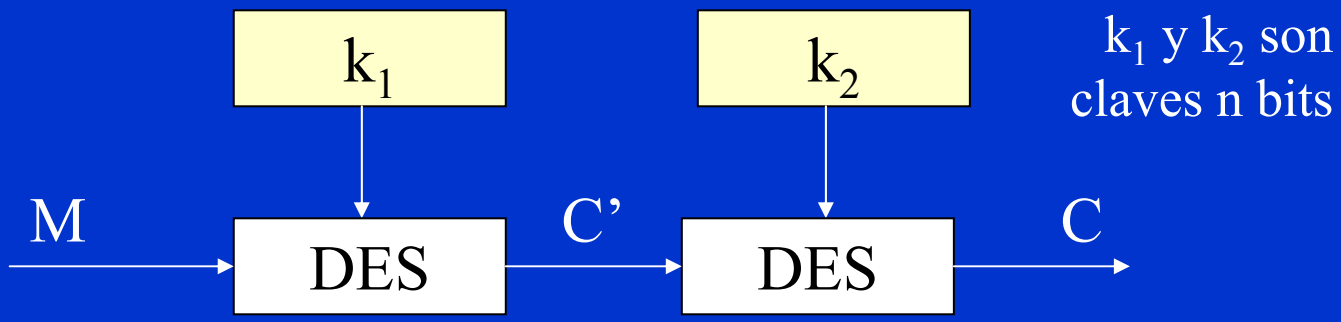
C_3 = VAIH VAJF XZJI G



Como ejercicio compruebe que a resultados similares llega si, por ejemplo, usa ahora los siguientes pares de claves: LAPALA y LANUCA; PASA y NADA; PAÑOS y TERMA. ¿Cuáles son las claves k_3 en cada caso? ☺

El DES no será un grupo y, por lo tanto, permitirá el cifrado múltiple. Esto aumentará el tamaño efectivo de la clave.

Cifrado DES múltiple

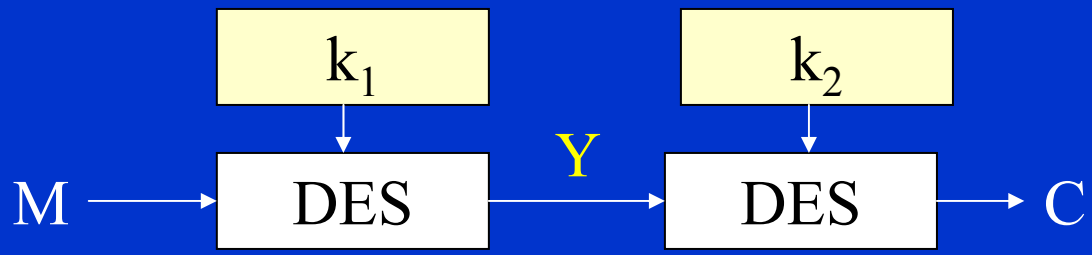


¿Se duplica la longitud de la clave?

En este modelo, cabe esperar que la longitud efectiva de la clave sea 2^{2n} donde n representa la longitud de bits de las claves k_1 y k_2. No obstante esto no es cierto.

En realidad el tamaño de la clave resultante en este caso es equivalente a 2^{n+1}, un aumento insignificante para n grande y por esta razón no se usa.

Ataque por encuentro a medio camino



a) Se descripta el criptograma C por fuerza bruta usando las 2^n claves posibles y realizando entonces 2^n cálculos. Se obtiene así Y.

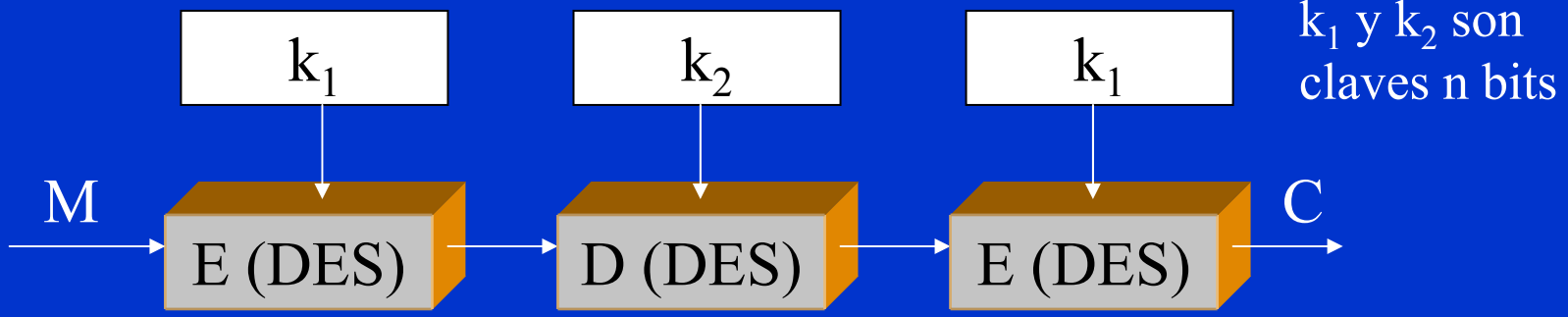
b) Con los "textos intermedios" Y se forma una tabla ordenada de textos cifrados con sus correspondientes valores k_2.

c) Se cifran los textos en claro M elegidos con todas las claves k_1 y se comparan con Y, realizando un máximo de 2^n cálculos.

d) Una de las claves será la verdadera y se ha realizado un número menor que $2^n + 2^n = 2^{n+1}$ cálculos. Luego la clave real es igual a 2^{n+1}.

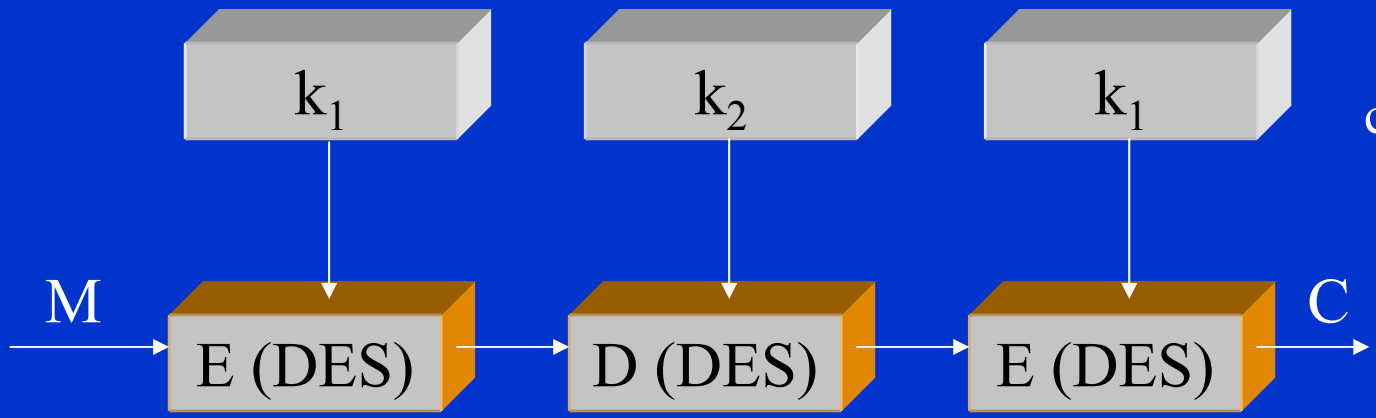
Este ataque se conoce con el nombre de *meet-in-the-middle*.

Triple DES



- En este caso se logra un valor efectivo de longitud de clave igual a 2^{2n} bits, es decir $2^{2 \cdot 56} = 2^{112}$ bits.
- El ejemplo anterior con sólo dos claves (equivalente al de tres claves) se usa por motivos de compatibilidad con el DES de clave única. Propuesto por Matyas y Meyer de IBM, se denomina EDE: Encrypt-Decrypt-Encrypt.
- Es inmune a ataques por encuentro a medio camino.

Usos de Triple DES



Aunque el algoritmo DES haya sufrido diversos ataques y no se haya vuelto a certificar por el NIST como estándar de cifrado, el Triple DES sí tiene una gran seguridad debido al tamaño de su clave de 112 bits efectivos y sigue siendo muy válido en el año 2002. De hecho, es el algoritmo propuesto en el protocolo SET y se encuentra, entre otras aplicaciones, en el programa PGP.

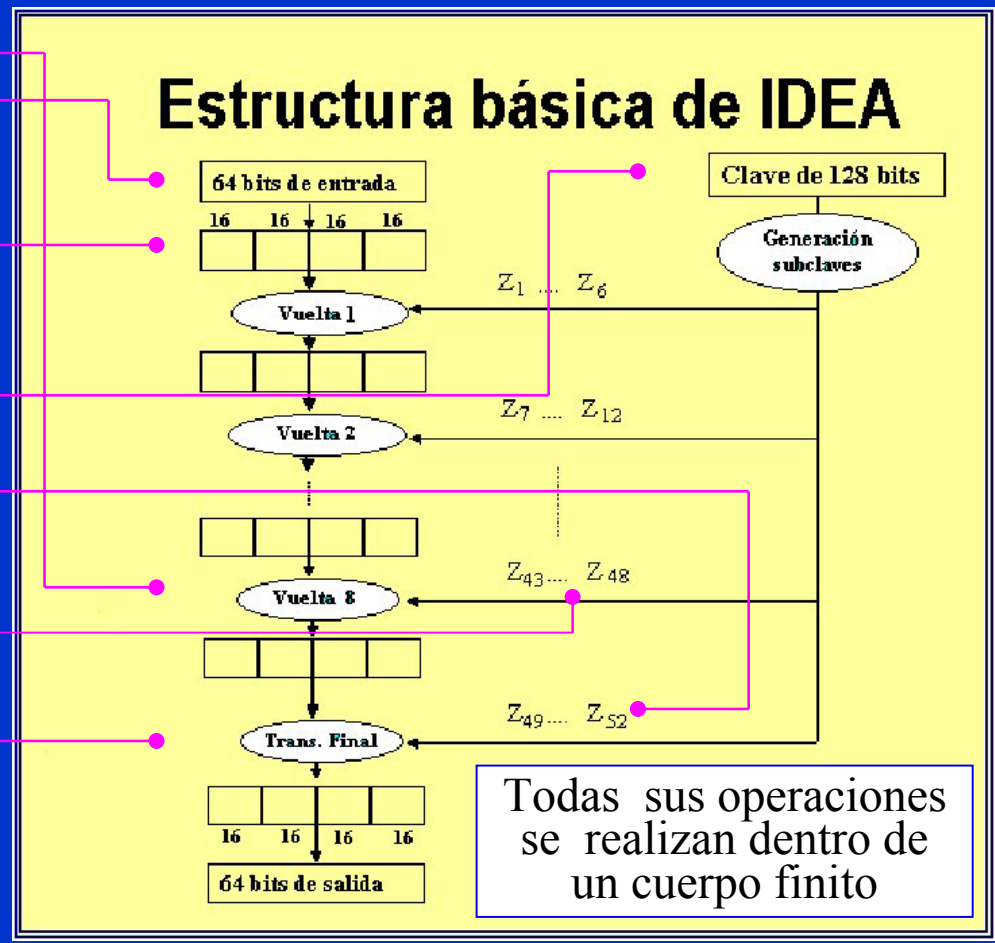
International Data Encryption Algorithm IDEA

Historia del IDEA

- En 1990 Xuejia Lai y James Massey proponen el PES, Proposed Encryption Standard.
- En 1991 -debido a los avances de Biham y Shamir en el criptoanálisis diferencial- los autores proponen el IPES, Improved Proposed Encryption Standard.
- En 1992 los autores proponen finalmente el algoritmo IDEA, International Data Encryption Algorithm.
- En 1999 el algoritmo IDEA, mucho más seguro que el DES y sus versiones, se comienza a usar ampliamente en el sistema de correo electrónico seguro PGP.

Estructura y esquema de IDEA

- Cifra bloques de 64 bits en 8 vueltas
- Divide la entrada M en cuatro bloques de 16 bits
- Se generan 52 subclaves de 16 bits a partir de la clave maestra de 128 bits
- Usa 6 claves por vuelta
- Hay una transformación final con 4 claves para invertir operación inicial



Operaciones matemáticas en IDEA

Operaciones básicas

- XOR
- Suma módulo 2^{16} (mod 65.536)
- Multiplicación módulo $2^{16}+1$ (65.537)

Es primo y se asegura el inverso multiplicativo

Todas las operaciones se realizan con bloques de 16 bits y el *truco* está en que los bloques cuyo valor sea 0 (16 bits) se cambiarán por la constante 2^{16} ... ¡de 17 bits! ☺. Comprobar aquí esto con tantos dígitos es complicado pero podemos ver un ejemplo con números pequeños que puede interpolarse.

Operaciones +, ⊗ y ⊕ en grupo pequeño

Ejemplo dentro de un grupo n pequeño

Como $2^n + 1$ debe ser primo, sea n = 2 ya que $2^2 = 4$ y $2^2 + 1 = 5$

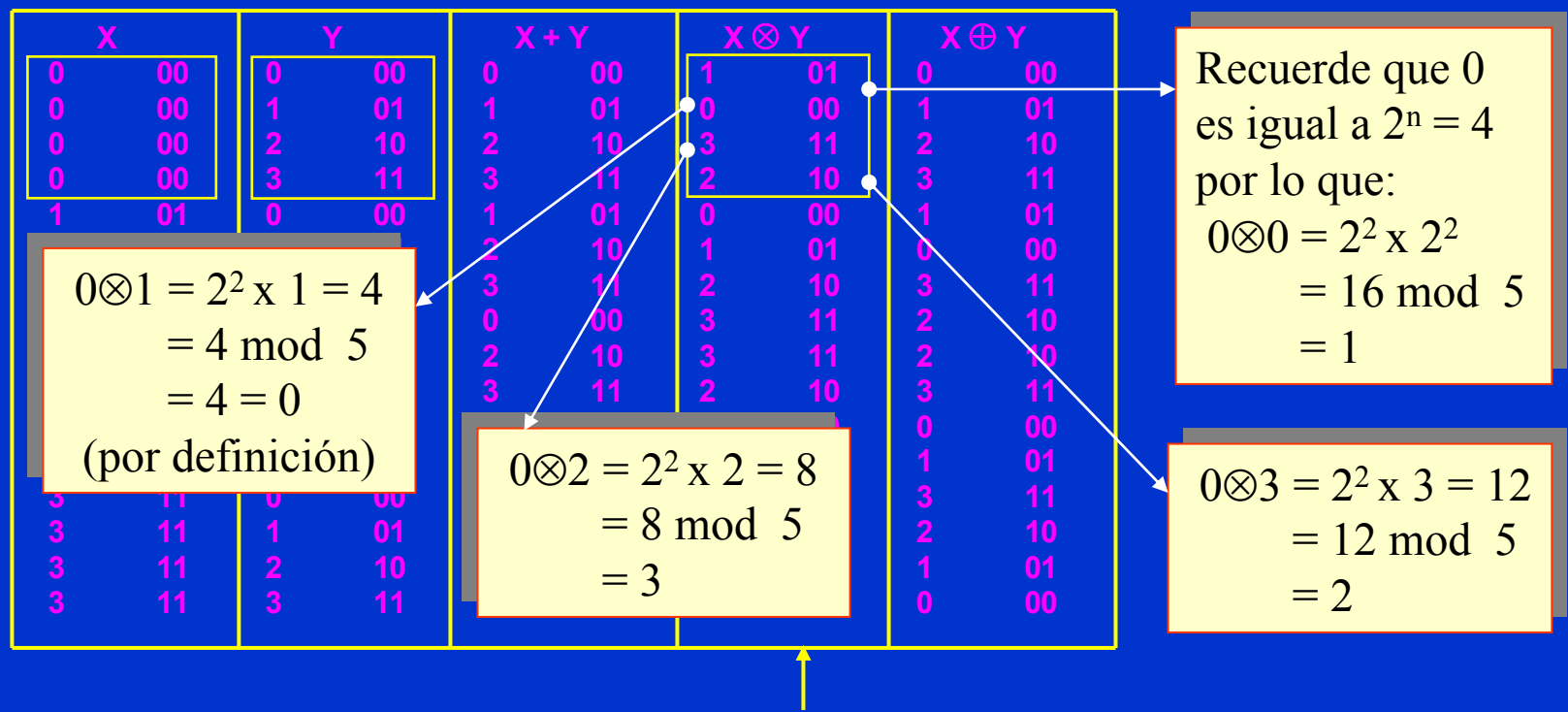
n = 2
dos bits

X		Y		X + Y		X ⊗ Y		X ⊕ Y	
0	00	0	00	0	00	1	01	0	00
0	00	1	01	1	01	0	00	1	01
0	00	2	10	2	10	3	11	2	10
0	00	3	11	3	11	2	10	3	11
1	01	0	00	1	01	0	00	1	01
1	01	1	01	2	10	1	01	0	00
1	01	2	10	3	11	2	10	3	11
1	01	3	11	0	00	3	11	2	10
2	10	0	00	2	10	3	11	2	10
2	10	1	01	3	11	2	10	3	11
2	10	2	10	0	00	0	00	0	00
2	10	3	11	1	01	1	01	1	01
3	11	0	00	3	11	2	10	3	11
3	11	1	01	0	00	3	11	2	10
3	11	2	10	1	01	1	01	1	01
3	11	3	11	2	10	0	00	0	00

Veremos cómo se opera con la multiplicación. La suma y el or exclusivo son operaciones similares.

Operaciones: + mod 2^n (mod 4), ⊗ mod 2^n+1 (mod 5), XOR (mod 2)

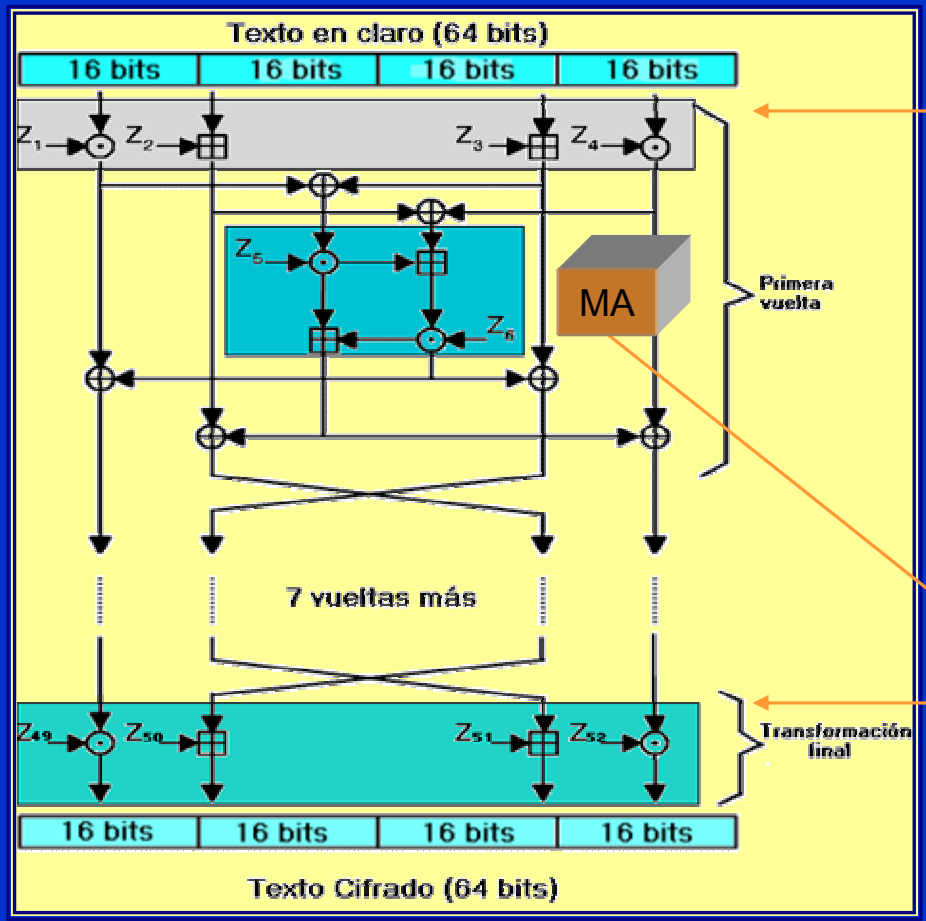
Ejemplo de operación ⊗ en IDEA



Operaciones: + mod 2^n (mod 4), ⊗ mod 2^n+1 (mod 5), XOR (mod 2)

Los demás cálculos con los diferentes valores de X e Y son todos similares

Detalles del algoritmo IDEA



Operación cifrado

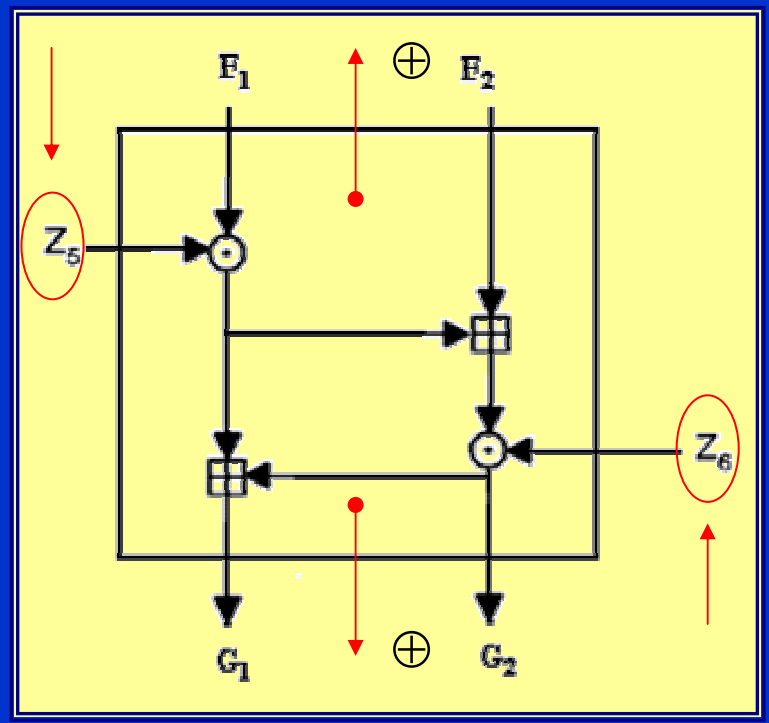
Operaciones inversas al comienzo y al final del algoritmo. Esto permite usar el mismo algoritmo para cifrar que para descifrar.

Bloque principal

Bloque principal de IDEA

Estas tres operaciones provocan *confusión* y no cumplen las leyes distributiva ni asociativa.

La estructura que crea la *difusión* es un bloque básico denominado Estructura MA *Multiplication / Addition*.
Usa sólo dos claves por cada vuelta y sus entradas F_1, F_2 así como sus salidas G_1, G_2 están conectadas por XOR.



Generación de claves en IDEA

A partir de una entrada de 128 bits, se generan las 52 subclaves de cifrado.

Se produce un desplazamiento de 25 bits a la izquierda en cada una de las 7 fases de generación de claves.

Los 64 últimos bits de la fase 7 no se usan.

Con los primeros 128 bits se generan 8 subclaves de 16 bits cada una.

Desplazamientos de la clave en IDEA

En cada operación sobre la clave de 128 bits, se obtienen 8 claves de 16 bits de las que sólo se usan 6 en cada vuelta. Las claves restantes se guardan para la siguiente vuelta.

Clave Principal k = 128 bits

001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032
033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064
065 066 067 068 069 070 071 072 073 074 075 076 077 078 079 080 081 082 083 084 085 086 087 088 089 090 091 092 093 094 095 096
097 098 099 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128

	Primeros 16 bits de clave	....	Ultimos 16 bits de clave
1ª	001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016	....	113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128
2ª	026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041	....	010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025
3ª	051 052 053 054 055 056 057 058 059 060 061 062 063 064 065 066	....	035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050
4ª	076 077 078 079 080 081 082 083 084 085 086 087 088 089 090 091	....	060 061 062 063 064 065 066 067 068 069 070 071 072 073 074 075
5ª	101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116	....	085 086 087 088 089 090 091 092 093 094 095 096 097 098 099 100
6ª	126 127 128 001 002 003 004 005 006 007 008 009 010 011 012 103	....	110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
7ª	023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038	....	007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022

Claves usadas por IDEA en cada en vuelta

La distribución de bits de subclaves en cada vuelta sigue una lógica

Primera vuelta:	$k_1k_2k_3k_4k_5k_6$	B[1…96]
Segunda vuelta:	$k_7k_8k_9k_{10}k_{11}k_{12}$	B[97…128; 26…89]
Tercera vuelta:	$k_{13}k_{14}k_{15}k_{16}k_{17}k_{18}$	B[90…128; 1…25; 51…82]
Cuarta vuelta:	$k_{19}k_{20}k_{21}k_{22}k_{23}k_{24}$	B[83…128; 1…50]
Quinta vuelta:	$k_{25}k_{26}k_{27}k_{28}k_{29}k_{30}$	B[76…128; 1…43]
Sexta vuelta:	$k_{31}k_{32}k_{33}k_{34}k_{35}k_{36}$	B[44…75; 101…128; 1…36]
Séptima vuelta:	$k_{37}k_{38}k_{39}k_{40}k_{41}k_{42}$	B[37…100; 126…128; 1…29]
Octava vuelta:	$k_{43}k_{44}k_{45}k_{46}k_{47}k_{48}$	B[30…125]
Transformación:	$k_{49}k_{50}k_{51}k_{52}$	B[23…86]

Primeras claves en cada vuelta en IDEA

Las primeras claves de cada vuelta k_1, k_7, k_{13}, k_{19}, k_{25}, k_{31}, k_{37} y k_{43} usan un conjunto diferente de bits. Excepto en las vueltas primera y octava, los 96 bits de subclave usados en cada vuelta, no son contiguos. Debido al desplazamiento en cada fase de 25 bits a la izquierda, se hace muy difícil el ataque a la clave.

K_1:	001	002	003	004	005	006	007	008	009	010	011	012	013	014	015	016
K_7:	097	098	099	100	101	102	103	104	105	106	107	108	109	110	111	112
K_{13}:	090	091	092	093	094	095	096	097	098	099	100	101	102	103	104	105
K_{19}:	083	084	085	086	087	088	089	090	091	092	093	094	095	096	097	098
K_{25}:	076	077	078	079	080	081	082	083	084	085	086	087	088	089	090	091
K_{31}:	044	045	046	047	048	049	050	051	052	053	054	055	056	057	058	059
k_{37}:	037	038	039	040	041	042	043	044	045	046	047	048	049	050	051	052
k_{43}:	030	031	032	033	034	035	036	037	038	039	040	041	042	043	044	045

Ejemplo de cálculo de claves en IDEA

Si la clave es “IDEA es la clave”, encuentre los 16 bits de la segunda clave de la cuarta vuelta.
Solución:

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
0	1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0

31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
0	1	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1	1	1	0	0	1	1	0	0	1	0

61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
0	0	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1

91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114
1	0	0	0	1	1	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	1	0	1

115	116	117	118	119	120	121	122	123	124	125	126	127	128
1	1	0	1	1	0	0	1	1	0	0	1	0	1

Como en cada vuelta se usan 6 subclaves, la segunda clave de la cuarta vuelta será la número 3∗6+2 = 20. Como la clave 19 termina en el bit 98, la clave 20 serán los 16 bits siguientes, es decir del 99 al 114: k_{20} = 10110001 10000101.

Descifrado con IDEA

El algoritmo IDEA, al igual que el DES, permite cifrar y descifrar con la misma estructura. Como las operaciones se hacen dentro de un cuerpo finito, en este caso las claves se toman como los inversos de las operaciones XOR, suma mod 2^{16} y producto mod $2^{16}+1$, dependiendo de las operaciones realizadas en la fase de cifrado.

INVERSOS

Inverso XOR: se aplica la misma función

Inverso aditivo: suma mod 2^{16}

$$Z_j \boxplus -Z_j = 0$$

Inverso multiplicativo: producto mod $2^{16}+1$

$$Z_j \odot Z_j^{-1} = 1$$

Claves de descifrado en IDEA

$d_1 = k_{49}^{-1}$	$d_2 = -k_{50}$	$d_3 = -k_{51}$	$d_4 = k_{52}^{-1}$	$d_5 = k_{47}$	$d_6 = k_{48}$
$d_7 = k_{43}^{-1}$	$d_8 = -k_{45}$	$d_9 = -k_{44}$	$d_{10} = k_{46}^{-1}$	$d_{11} = k_{41}$	$d_{12} = k_{42}$
$d_{13} = k_{37}^{-1}$	$d_{14} = -k_{39}$	$d_{15} = -k_{38}$	$d_{16} = k_{40}^{-1}$	$d_{17} = k_{35}$	$d_{18} = k_{36}$
$d_{19} = k_{31}^{-1}$	$d_{20} = -k_{33}$	$d_{21} = -k_{32}$	$d_{22} = k_{34}^{-1}$	$d_{23} = k_{29}$	$d_{24} = k_{30}$
$d_{25} = k_{25}^{-1}$	$d_{26} = -k_{27}$	$d_{27} = -k_{26}$	$d_{28} = k_{28}^{-1}$	$d_{29} = k_{23}$	$d_{30} = k_{24}$
$d_{31} = k_{19}^{-1}$	$d_{32} = -k_{21}$	$d_{33} = -k_{20}$	$d_{34} = k_{22}^{-1}$	$d_{35} = k_{17}$	$d_{36} = k_{18}$
$d_{37} = k_{13}^{-1}$	$d_{38} = -k_{15}$	$d_{39} = -k_{14}$	$d_{40} = k_{16}^{-1}$	$d_{41} = k_{11}$	$d_{42} = k_{12}$
$d_{43} = k_7^{-1}$	$d_{44} = -k_9$	$d_{45} = -k_8$	$d_{46} = k_{10}^{-1}$	$d_{47} = k_5$	$d_{48} = k_6$
$d_{49} = k_1^{-1}$	$d_{50} = -k_2$	$d_{51} = -k_3$	$d_{52} = k_4^{-1}$		

Inversos de la suma

Inversos del producto

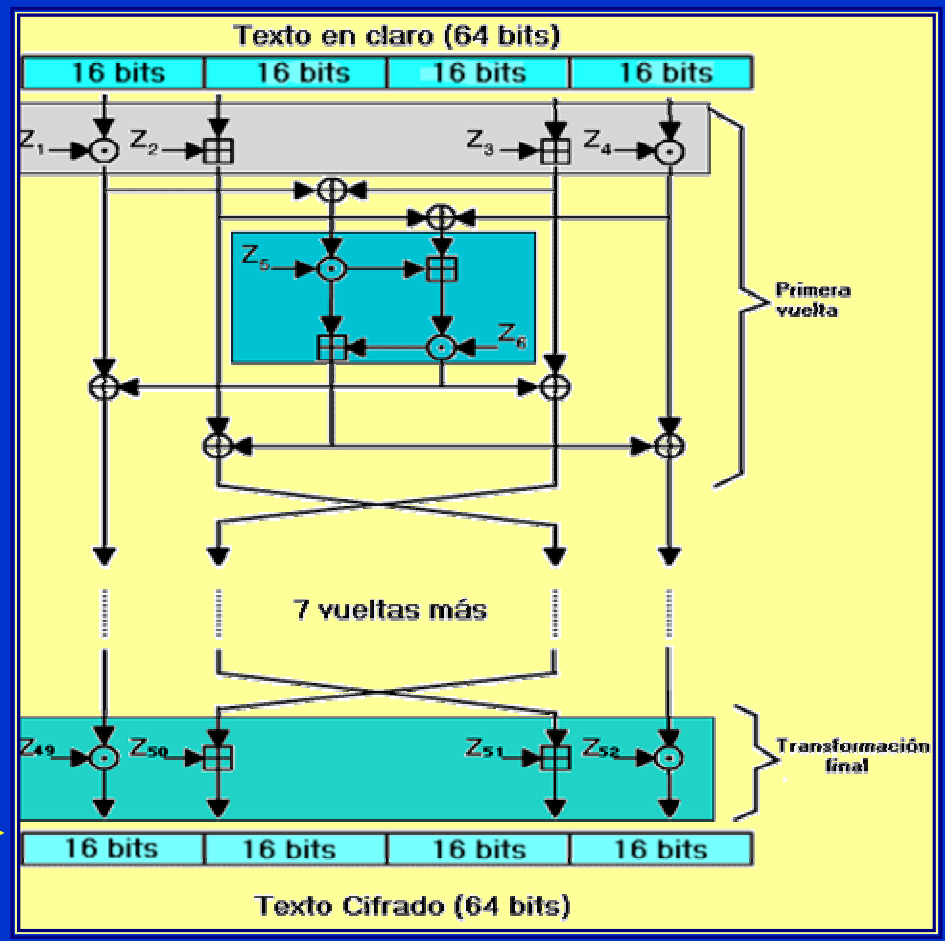
Inversos del XOR

Operación de descifrado con IDEA

Módulo IDEA

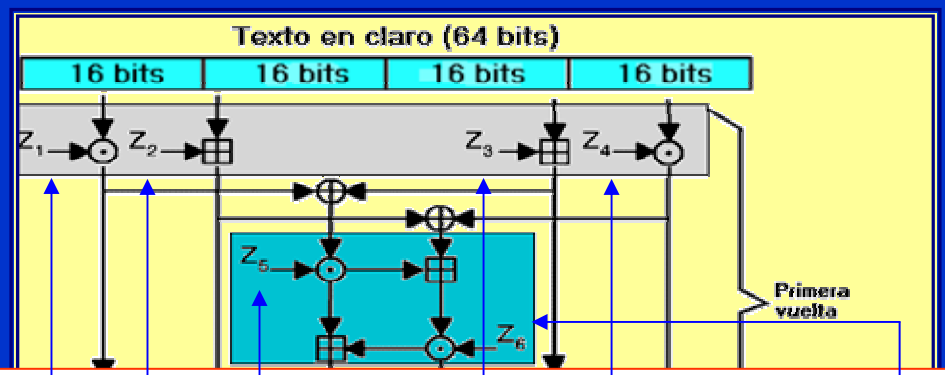
Para descifrar, cada bloque de criptograma se dividirá en cuatro subbloques de 16 bits

Las operaciones se hacen ahora hacia arriba



Uso de claves inversas en descifrado IDEA

Ultimas 6 claves de descifrado



$d_1 = k_{49}^{-1}$	$d_2 = -k_{50}$	$d_3 = -k_{51}$	$d_4 = k_{52}^{-1}$	$d_5 = k_{47}$	$d_6 = k_{48}$
$d_7 = k_{43}^{-1}$	$d_8 = -k_{45}$	$d_9 = -k_{44}$	$d_{10} = k_{46}^{-1}$	$d_{11} = k_{41}$	$d_{12} = k_{42}$
$d_{13} = k_{37}^{-1}$	$d_{14} = -k_{39}$	$d_{15} = -k_{38}$	$d_{16} = k_{40}^{-1}$	$d_{17} = k_{35}$	$d_{18} = k_{36}$
$d_{19} = k_{31}^{-1}$	$d_{20} = -k_{33}$	$d_{21} = -k_{32}$	$d_{22} = k_{34}^{-1}$	$d_{23} = k_{29}$	$d_{24} = k_{30}$
$d_{25} = k_{25}^{-1}$	$d_{26} = -k_{27}$	$d_{27} = -k_{26}$	$d_{28} = k_{28}^{-1}$	$d_{29} = k_{23}$	$d_{30} = k_{24}$
$d_{31} = k_{19}^{-1}$	$d_{32} = -k_{21}$	$d_{33} = -k_{20}$	$d_{34} = k_{22}^{-1}$	$d_{35} = k_{17}$	$d_{36} = k_{18}$
$d_{37} = k_{13}^{-1}$	$d_{38} = -k_{15}$	$d_{39} = -k_{14}$	$d_{40} = k_{16}^{-1}$	$d_{41} = k_{11}$	$d_{42} = k_{12}$
$d_{43} = k_7^{-1}$	$d_{44} = -k_9$	$d_{45} = -k_8$	$d_{46} = k_{10}^{-1}$	$d_{47} = k_5$	$d_{48} = k_6$
$d_{49} = k_1^{-1}$	$d_{50} = -k_2$	$d_{51} = -k_3$	$d_{52} = k_4^{-1}$		

Fortaleza del algoritmo IDEA

- IDEA se muestra inmune ante un criptoanálisis diferencial. Sus autores conocían esta debilidad del DES y lo hicieron resistente.
- Joan Daemen descubre en 1992 una clase de claves débiles. La siguiente clave k = 0000,0000,0x00,0000,0000,000x,xxxx,x000 en hexadecimal es débil, en el sentido de que un criptoanalista podría identificarla en un ataque con texto en claro elegido. Las posiciones x pueden ser cualquier número en hexadecimal.
- La probabilidad de que se use este tipo de claves es sólo de uno en 2^{96} y se puede, además, eliminar por diseño.
- A la fecha, año 2003, no se conoce todavía ningún sistema o algoritmo de ataque que haya criptoanalizado el IDEA.
- Joan Daemen y Vincent Rijmen crearán en 1997 el RIJNDAEL, nuevo estándar mundial del NIST desde finales de 2001.

Algoritmo RC2

- Cifrador en bloque de clave variable propuesto por Ron Rivest.
- El código es secreto industrial de RSA Data Security Inc.
- Tamaño del bloque de texto: 64 bits.
- Con una clave con tamaño variable (de 8 a 1.024 bits) forma una tabla de 128 bytes (1.024 bits) que depende de la clave inicial.
- No usa cajas S y es casi tres veces más rápido que DES.
- Se usa en SMIME con longitudes de clave de 40, 64 y 128 bits.
- Los algoritmos RC2 y RC4 se incluyen en productos para la exportación, como navegadores, limitando la clave a 40 bits.
- Operaciones primitivas de cifra: suma en módulo 2^{32}, operación or exclusivo, complemento de bits, operación AND y rotación circular a la izquierda.
- Realiza 18 vueltas conocidas como mixing y mashing.

Algoritmo RC5

- RC5 es un cifrador en bloque de tamaño variable de Ron Rivest.
- Cifra bloques de texto de 32, 64 ó 128 bits.
- Tamaño de clave hasta 2.048 bits, en función número de vueltas.
- Número de vueltas de 0 a 255.
- Versiones específicas: RC5 –w/r/b donde w es el tamaño de la palabra (16, 32 ó 64 bits) -RC5 cifra bloques de dos palabras-, r es el número de vueltas y b es el tamaño en octetos de la clave K. El valor propuesto por Rivest como mínimo es RC5 –32/12/16.
- Rutina expansión de clave: se expande K para llenar una tabla.
- Rutinas de cifrado y descifrado: usa primitivas de suma módulo 2^w, or exclusivo y rotación circular a la izquierda.
- Características: muy rápido, arquitectura simple, bajos requisitos de memoria y alta seguridad. Las rotaciones dependientes de los datos le fortalecen ante el criptoanálisis diferencial.

Algoritmos SAFER 64 y 128

- SAFER: Secure and Fast Encryption Routine (James Massey).
- Cifra bloques de texto de 64 bits. Cada bloque a cifrar de texto se divide en 8 bytes.
- Tamaño de clave: 64 ó 128 bits.
- Número de vueltas de 0 a 10; mínimo recomendable 6.
- Operaciones de cifrado y descifrado distintas basadas en bytes, que orientan su uso en aplicaciones de tarjetas inteligentes.
- En cada vuelta hay operaciones or y sumas normales, potencias y logaritmos discretos en p = 257, usando 45 como raíz primitiva.
- Al final del algoritmo hay tres niveles de operaciones lineales conocidas como Pseudo Transformaciones de Hadamard, PTH, cuyo objetivo es aumentar la difusión de los bits.
- Existen versiones SAFER SK-64 y SK-128 más seguras ante claves débiles que sus antecesoras.

Algoritmo Blowfish

- Cifrador tipo Feistel de clave variable (Bruce Schneier).
- Cifra bloques de texto de 64 bits.
- Tamaño de clave: de 32 hasta 448 bits. Se generan 18 subclaves de 32 bits y cuatro cajas S de 8x32 bits, en total 4.168 bytes.
- Número de vueltas: 16, en cada una de ellas se realiza una permutación función de la clave y una sustitución que es función de la clave y los datos.
- Operaciones básicas: or exclusivo y suma módulo 2^{32}.
- Cajas S: en cada vuelta hay cuatro con 256 entradas cada una.
- Características: compacto porque necesita sólo 5 K de memoria, es muy rápido (5 veces más veloz que DES), es conceptualmente simple y su fortaleza puede variarse según longitud de la clave. Usa una función F con las cuatro cajas S y operaciones básicas de suma y or exclusivo que provocan un efecto de avalancha.

Algoritmo CAST 128

- Cifrador Feistel propuesto por C. Adams y S. Tavares (Canadá).
- Cifra bloques de texto de 64 bits con claves de 40 hasta 128 bits en incrementos de octetos.
- Cifra en 16 vueltas.
- Usa ocho cajas S de 8 bits de entrada y 32 bits de salida con unas funciones no lineales óptimas (funciones bent), cuatro cajas en procesos de cifra y las otras cuatro para la generación de claves. Cada caja es un *array* de 32 columnas y 256 filas. Los 8 bits de entrada seleccionan una fila y los 32 bits de ésta es la salida.
- Operaciones básicas: suma y resta módulo 2^{32}, or exclusivo y rotaciones circulares hacia la izquierda.
- Características: inmune a ataques por criptoanálisis diferencial y lineal; algoritmo estándar de cifra en últimas versiones de PGP.

Algoritmo Skipjack

- Ha sido desarrollado por la NSA, National Security Agency, está contenido en los chip Clipper y Capstone y su implementación sólo está permitida en hardware.
- Cifra bloques de 64 bits con una clave de 80 bits.
- Los usuarios depositan sus claves secretas en diversas agencias de gobierno.
- Usa 32 vueltas en cada bloque de cifra.
- Los detalles del algoritmo no son públicos.
- Características: imposición de los EEUU para comunicaciones con la administración, tiene una puerta trasera que puede dejar en claro la cifra, nadie puede asegurar que el algoritmo tenga la suficiente fortaleza pero los Estados Unidos piensa usarlo en su DMS, Defense Messaging System. Ha sido duramente criticado.

El DES deja de ser un estándar

- El DES se adopta como estándar en 1976.
- El NIST certifica al DES en 1987 y luego en 1993.
- Durante esos años se estandariza como algoritmo de cifra en todo el mundo. Su uso principal lo encontramos en el cifrado de la información intercambiada en transacciones de dinero entre un cajero automático y el banco respectivo.
- En 1997 NIST no certifica al DES y llama a un concurso internacional para buscar un nuevo estándar mundial de cifra denominado AES *Advanced Encryption Standard*.
- Precisamente entre 1997 y 1999 el DES se enfrenta a tres ataques o desafíos conocidos como DES Challenge que impulsa y promociona la compañía RSA.

DES Challenge I y II

- 29 enero 1997: DES Challenge I. Se rompe la clave en 96 días con 80.000 de ordenadores en Internet que evalúan 7.000 millones de clave por segundo. Para encontrar la clave se debe recorrer el 25% del espacio de claves ☺.
- 13 enero 1998: DES Challenge II-1. Se rompe la clave en 39 días con un ataque tipo distribuido por distributed.net que llega a evaluar 34.000 millones de claves por segundo y debe recorrer el 88% del espacio de claves ☹.
- 13 julio de 1998: DES Challenge II-2. Electronic Frontier Foundation EFF crea el DES Cracker con una inversión de US $ 200.000 y en 56 horas (2½ días) rompe la clave evaluando 90.000 millones de claves por segundo.

DES Challenge III

- 18 enero 1999: DES Challenge III. Se unen la máquina DES Cracker y distributed.net con 100.000 ordenadores conectados en Internet para romper la clave en 22 horas, menos de 1 día, evaluando 245.000 millones de claves por segundo tras recorrer el 22% del espacio de claves.
- Se trata del último desafío propuesto por RSA que pone en evidencia la capacidad de ataque distribuido a través de los tiempos muertos de procesador de máquinas conectadas a Internet que, con un programa cliente, van resolviendo un pequeño trozo del espacio de claves, comunicándose para ello con un servidor.

El nuevo estándar en cifra AES

AES: Advanced Encryption Standard

- El DES, estándar desde 1976, pasa la certificación de la NBS National Bureaux of Standards en 1987 y en 1993.
- En 1997 el NIST National Institute of Standards and Technology (antigua NBS) no certifica al DES y llama a concurso público para un nuevo estándar: el AES.
- En octubre del año 2000 el NIST elige el algoritmo belga RIJNDAEL como nuevo estándar para algoritmo de cifra del siglo XXI. Es software de libre distribución y está disponible desde finales del año 2001.

Características del algoritmo RIJNDAEL

RIJNDAEL: autores Vincent Rijmen y Joan Daemen

- No es de tipo Feistel.
- Implementado para trabajar en los procesadores de 8 bits usados en tarjetas inteligentes y en CPUs de 32 bits.
- Tamaño de clave variable: 128, 192 y 256 bits (estándar) o bien múltiplo de 4 bytes.
- Tamaño del bloque de texto: 128 bits o múltiplo de 4 bytes.
- Operaciones modulares a nivel de byte (representación en forma de polinomios) y de palabra de 4 bytes: 32 bits.
- Número de etapas flexible según necesidades del usuario.
- Usa un conjunto de Cajas S similar al DES.

Transformaciones o capas de RIJNDAEL

- Hay tres transformaciones distintas llamadas capas en las que se tratan los bits. Estas constan de:
  - Capa de Mezcla Lineal: en ella se busca la difusión de los bits.
  - Capa No Lineal: se trata de una zona similar a las cajas S del DES.
  - Capa Clave: operaciones con una función Xor de la subclave y la información de esta etapa intermedia.
- Las transformaciones realizadas en cada paso del algoritmo se denominan estados y se representa por un array de 4 filas.
- Puede ver un esquema detallado del algoritmo en la página Web de su autor http://www.esat.kuleuven.ac.be/~rijmen/rijndael/ y en http://home.ecn.ab.ca/~jsavard/crypto/co040401.htm.

Operaciones con bytes en RIJNDAEL

Operaciones a nivel de byte en GF(2^8)

- Suma y multiplicación. Son cálculos en Campos de Galois GF(2^8) con 8 bits. Para la reducción de exponente se usará un polinomio primitivo p(x).
- Producto por x. Esta operación conocida como xtime(a) al igual que en el caso anterior usa la reducción de exponente. Puede implementarse fácilmente con desplazamientos y operaciones or exclusivo.

Ejemplos

Ejemplo de suma en GF(2^8)

Vamos a sumar los valores hexadecimales 57 y 83:

$A = 57_{16} = 0101\ 0111_2$ $\quad$ $B = 83_{16} = 1000\ 0011_2$

que expresados en polinomios dentro de GF(2^8) serán:

$A = 0101\ 0111_2 = x^6 + x^4 + x^2 + x + 1$

$B = 1000\ 0011_2 = x^7 + x + 1$

Sumando: $A+B = (x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) \bmod 2$

$A+B = (x^7 + x^6 + x^4 + x^2 + 2x + 2) \bmod 2$

$A+B = x^7 + x^6 + x^4 + x^2 = 1101\ 0100 = D4_{16}$

Y lo mismo se obtiene con la suma Or exclusivo:

$0101\ 0111 \oplus 1000\ 0011 = 1101\ 0100 = D4_{16}$

Ejemplo de producto en $GF(2^8)$ (1)

Vamos a multiplicar los valores hexadecimales 57 y 83:

$A = 57_{16} = 0101\ 0111_2$ $\quad$ $B = 83_{16} = 1000\ 0011_2$

que expresados en polinomios dentro de $GF(2^8)$ serán:

$A = 0101\ 0111_2 = x^6 + x^4 + x^2 + x + 1$

$B = 1000\ 0011_2 = x^7 + x + 1$

Sea $p(x) = x^8 + x^4 + x^3 + x + 1 \Rightarrow x^8 = x^4 + x^3 + x + 1$

$A*B = (x^6 + x^4 + x^2 + x + 1)*(x^7 + x + 1) \bmod 2$

$A*B = x^{13} + x^{11} + x^9 + x^8 + 2x^7 + x^6 + x^5 + x^4 + x^3 + 2x^2 + 2x + 1$

$A*B = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$

Este resultado hay que reducirlo por $p(x) = x^8 + x^4 + x^3 + x + 1$

Ejemplo de producto en GF(2^8) (2)

Están fuera del cuerpo de 8 bits

p(x): $x^8 = x^4 + x^3 + x + 1$

$$A*B = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$$

$$x^{13} = x^5 * x^8 = x^5 * (x^4 + x^3 + x + 1) = x^9 + x^8 + x^6 + x^5$$

$$x^{13} = x * (x^4 + x^3 + x + 1) + (x^4 + x^3 + x + 1) + x^6 + x^5$$

$$x^{13} = (x^5 + x^4 + x^2 + x) + (x^4 + x^3 + x + 1) + x^6 + x^5$$

$$x^{13} = x^6 + x^3 + x^2 + 1$$

$$x^{13} = x^6 + x^3 + x^2 + 1$$

Ejemplo de producto en GF(2^8) (3)

Están fuera del cuerpo de 8 bits

p(x): $x^8 = x^4 + x^3 + x + 1$

$$A*B = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$$

$$x^{11} = x^3 * x^8 = x^3 * (x^4 + x^3 + x + 1)$$

$$x^{11} = x^7 + x^6 + x^4 + x^3$$

$x^{11} = x^7 + x^6 + x^4 + x^3$

$$x^9 = x * x^8 = x * (x^4 + x^3 + x + 1)$$

$$x^9 = x^5 + x^4 + x^2 + x$$

$x^9 = x^5 + x^4 + x^2 + x$

$x^8 = x^4 + x^3 + x + 1$

Ejemplo de producto en $GF(2^8)$ (4)

$x^{13} = x^6 + x^3 + x^2 + 1$

$x^{11} = x^7 + x^6 + x^4 + x^3$

$x^9 = x^5 + x^4 + x^2 + x$

$x^8 = x^4 + x^3 + x + 1$

$$A*B = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$$

Reemplazando los cálculo anteriores en la expresión:

$$A*B = (x^6 + x^3 + x^2 + 1) + (x^7 + x^6 + x^4 + x^3) + (x^5 + x^4 + x^2 + x) + (x^4 + x^3 + x + 1) + x^6 + x^5 + x^4 + x^3 + 1 \mod 2$$

$$A*B = x^7 + x^6 + 1 = 1100\ 0001 = C1_{16}$$

Fin del Tema 10

Cuestiones y ejercicios (1 de 3)

1. ¿Qué particularidad tiene el cifrado tipo Feistel?
2. ¿Qué importante diferencia tiene el algoritmo Skipjack con respecto a todos los demás? Razone si eso es bueno o malo en criptografía.
3. ¿Cuál es la razón principal de la debilidad del algoritmo DES?
4. ¿Con cuál de las dos versiones respecto a la reducción de clave aplicada al DES por la NSA se queda Ud.? Razone las dos respuestas posibles.
5. ¿Qué tamaño de bloque de mensaje cifra el DES, con qué longitud de clave y con cuántas vueltas?
6. ¿Tiene algún interés criptográfico la tabla de permutación inicial IP que se repite en sentido contrario al final en el DES? ¿Por qué?
7. ¿Qué distribución especial observa en los dos bloques de texto a cifrar L_0 y R_0 en DES? ¿Qué separación en bits hay entre ellos?

Cuestiones y ejercicios (2 de 3)

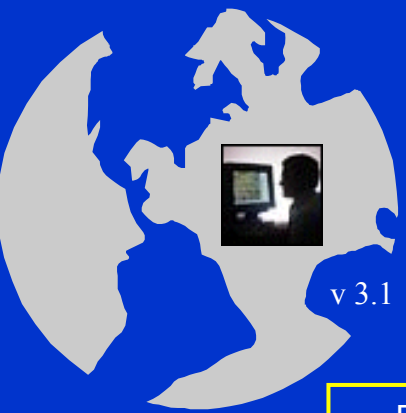
8. ¿Cómo se las arregla DES para realizar operaciones suma módulo dos con sub-bloques de texto de 32 bits y sub-claves de 56 bits?
9. ¿Qué dos importantes funciones cumplen las cajas S en el DES?
10. En la caja S_3 del DES entra la secuencia de bits 101101, ¿qué sale?
11. Si la clave DES en ASCII (no números) es HOLAPACO, ¿cuáles serán la primera y segunda sub-claves de cifrado?
12. ¿Por qué no debe usarse nunca el modo de cifra ECB?
13. ¿Podemos usar el DES como un generador de secuencia cifrante?
14. ¿Por qué decimos que el DES no es un grupo? ¿Qué significa eso?
15. ¿En qué consiste un ataque por encuentro a medio camino?
16. ¿Por qué se usa en el triple DES un cifrado con sólo dos claves y no con tres como su nombre indica?

Cuestiones y ejercicios (3 de 3)

17. ¿Por qué en IDEA se usa una palabra de 16 bits y no de 32 bits? ¿Se podría usar una palabra de 8 bits ó 24 bits? Justifique su respuesta.
18. Encuentre los resultados de las tres operaciones básicas para un sistema simulado IDEA que trabaja con 4 bits.
19. ¿Qué tamaño de bloque cifra IDEA, con qué longitud de clave y con cuántas vueltas?
20. ¿Cómo se generan las sub-claves en IDEA?
21. ¿Cuáles son las claves Z_9 y Z_{10} en un sistema IDEA en el que la clave maestra en ASCII es K = UnaClaveDePrueba.
22. Encuentre las claves de descifrado de las siguientes claves de cifra en IDEA: $k_{12} = 3.256$; $k_{13} = 34.517$; $k_{14} = 45.592$.
23. Sume y multiplique 31 y 18 en GF(2^8) según algoritmo RIJNDAEL.

Tema 11
Cifrado Asimétrico con Mochilas

Seguridad Informática y Criptografía

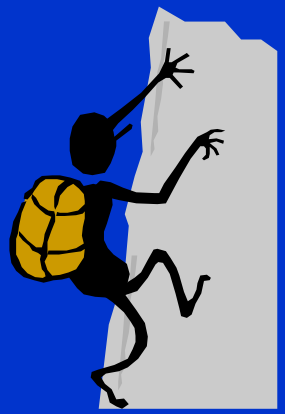


Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 28 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

El problema de la mochila



El problema matemático de la mochila referido ahora a números y no elementos que entran en ella puede plantearse como sigue:

Dada la siguiente secuencia de m números enteros positivos $S = \{S_1, S_2, S_3, ..., S_{m-2}, S_{m-1}, S_m\}$ y un valor u objetivo T, se pide encontrar un subconjunto de S $S_S = \{S_a, S_b, ..., S_j\}$ que cumpla con el objetivo:

$$T = \Sigma S_S = S_a + S_b + ... + S_j$$

Solución al problema de la mochila

Si los elementos de la mochila son números grandes, no están ordenados y no siguen una distribución supercreciente -en este tipo de distribución el elemento iésimo S_i de la mochila es mayor que la suma de todos sus antecesores-, la resolución de este problema es de tipo no polinomial.

Se trata de encontrar los vectores V_i de 0s y 1s de forma que:

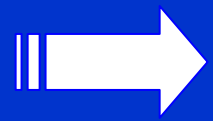
$$\Sigma S_i * V_i = T$$

Si se cumple esta relación, la mochila tiene solución. En caso contrario, no existirá solución.

Un ejemplo del problema de la mochila

Tenemos la mochila $S = \{20, 5, 7, 36, 13, 2\}$ con $m = 6$ y el valor $T = 35$. Se pide encontrar una solución, si es que ésta existe, en una única vuelta. En este momento no importa que los valores de la mochila no estén ordenados.

SOLUCIÓN: Sin hacer ningún cálculo mental, podemos recorrer todos los valores (se puede descartar el elemento S_4 pues es mayor que el objetivo T) de la mochila S, bien de izquierda a derecha o al revés (da igual el sentido elegido) y restaremos el elemento iésimo si es menor que el objetivo T en esa etapa del algoritmo, como se indica:



Solución al ejemplo de la mochila

$S = \{S_1, S_2, S_3, S_4, S_5, S_6\} = \{20, 5, 7, 36, 13, 2\}$ $T = 35$

$S_1 = 20$ ¿Es menor que objetivo $T = 35$? Sí $\Rightarrow$ $T = 35\text{-}20 = 15$
$S_2 = 5$ ¿Es menor que objetivo $T = 15$? Sí $\Rightarrow$ $T = 15\text{-}5 = 10$
$S_3 = 7$ ¿Es menor que objetivo $T = 10$? Sí $\Rightarrow$ $T = 10\text{-}7 = 3$
$S_4 = 36$ ¿Es menor que objetivo $T = 3$? No $\Rightarrow$ $T = 3$
$S_5 = 13$ ¿Es menor que objetivo $T = 3$? No $\Rightarrow$ $T = 3$
$S_6 = 2$ ¿Es menor que objetivo $T = 3$? Sí $\Rightarrow$ $T = 3\text{-}2 = 1 \neq 0$

Se ha recorrido toda la mochila y no se ha encontrado solución.

En cambio sí existe una solución:

$S_S = \{S_1+S_5+S_6\} = 20+13+2 = 35$ ⇒ $V_i = [1,0,0,0,1,1]$

¿Puede haber soluciones múltiples?

Si para la misma mochila S = {20, 5, 7, 36, 13, 2} buscamos ahora el valor T = 27, encontramos tres soluciones válidas:

$S_{S1} = \{S_1+S_3\} = 20+7$ $\quad$ $S_{S2} = \{S_1+S_2+S_6\} = 20+5+2$

$S_{S3} = \{S_2+S_3+S_5+S_6\} = 5+7+13+2$

Esto sería inadmisible en un sistema de cifra puesto que el resultado de una operación de descifrado debe ser única ya que proviene de un único mensaje. La solución será el uso de las denominadas mochilas simples en que la solución al problema de la mochila, si existe, es única.

Mochila simple o supercreciente

Una mochila es simple o supercreciente si el elemento S_k es mayor que la suma de los elementos que le anteceden:

$$S_k > \sum_{j=1}^{k-1} S_j$$

Por ejemplo, la mochila S = {2, 3, 7, 13, 28, 55, 110, 221} con m = 8 elementos es supercreciente y la solución para un objetivo T = 148 es única: $V_i = [S_2+S_3+S_5+S_7]$.

Para resolver cualquier valor T válido para esta mochila, ésta se recorre de derecha a izquierda una sola vez con el algoritmo ya visto.

Compruebe que para T = 289, 196 y 353 los vectores son $V_1 = 00010101$; $V_2 = 01001110$; $V_3 = 10110011$.

Operación de cifra con mochila simple

Se representa la información en binario y se pasan los bits por la mochila. Los bits 1s incluyen en la suma el elemento al que apuntan y los bits 0s no.

Con la mochila S = {2, 4, 10, 19, 40} de m = 5 elementos cifraremos el mensaje M = ADIOS.

SOLUCIÓN: Usando código ASCII/ANSI: A = 01000001; D = 01000100; I = 01001001; O = 01001111; S = 01010011

M = 01000 00101 00010 00100 10010 10011 11010 10011

C = (4), (10+40), (19), (10), (2+19), (2+19+40), (2+4+19), (2+19+40)

C = 4, 50, 19, 10, 21, 61, 25, 61

Descifrado con mochila simple

C = 4, 50, 19, 10, 21, 61, 25, 61

S = {2, 4, 10, 19, 40}

La operación de descifrado es elemental: pasamos por la mochila los valores de C, encontramos el vector V_i y por último agrupamos el resultado en grupos de 8 bits. En este caso $4 \Rightarrow V_i = 01000$, $50 \Rightarrow V_i = 00101$, etc.

PROBLEMA: Es muy fácil cifrar y descifrar pero también criptoanalizar el sistema de cifra porque se usa una mochila simple.

Una posible solución es usar mochilas de Merkle y Hellman.

Mochila de Merkle y Hellman MH

- En 1978 Ralph Merkle y Martin Hellman proponen un sistema de cifra de clave pública denominado Mochila con Trampa.
- El algoritmo se basa en crear una mochila difícil a partir de una mochila simple de forma que el cifrado se haga con la mochila difícil y el descifrado con la mochila simple o fácil. Se puede pasar fácilmente de la mochila simple a la difícil o viceversa usando una trampa.

La trampa será nuestra clave secreta.
La mochila difícil será nuestra clave pública.

- Hay otros sistemas de cifra basados en mochilas algo más complejos pero, básicamente, son iguales y no presentan una fortaleza significativamente mayor. No tiene sentido su estudio.

Diseño mochila de Merkle y Hellman (1)

1. Se selecciona una mochila supercreciente de m elementos $S' = \{S_1', S_2', ..., S_m'\}$.
2. Se elige un entero μ (módulo de trabajo) mayor que la suma de los elementos de la mochila.

$$\mu > \sum_{i=1}^{m} S_i'$$

más fácil: $\mu \geq 2 * S_m'$

3. Se elige un entero ω primo relativo con μ.

$$mcd(\omega, \mu) = 1$$

Se asegura el inverso

Se recomienda que ω no tenga factores con los elementos de S'

4. Se multiplica S' por ω mod μ.

$$S_i = \omega * S_i' \bmod \mu$$

Obteniendo una mochila difícil $S = \{S_1, S_2, ..., S_m\}$

Diseño mochila de Merkle y Hellman (2)

5. Se calcula el inverso de ω en el cuerpo μ. $\omega^{-1} = inv(\omega,\mu)$

Clave privada: μ, ω^{-1}

Clave pública: mochila S

Esto se interpreta como encontrar los vectores que cumplan con un valor de T.

CIFRADO:

$C = S * M$

como $S = \omega * S' \bmod \mu$

$C = \omega * S' * M \bmod \mu$

DESCIFRADO:

$M = \omega^{-1} * C \bmod \mu$

Entonces obtenemos:

$S' * M$

Cifrado mochila de Merkle y Hellman (1)

Vamos a cifrar el mensaje M = Sol usando la mochila simple y supercreciente S' = {3, 5, 12, 21}.

1. Elección de μ: $\mu \geq 2*S'_4 \geq 2*21$ $\mu = 45$
2. Elección de ω: $mcd(\omega, \mu) = 1$ $\omega = 32$ $(\omega^{-1} = 38)$
3. Mochila S: $S = \omega * S' \bmod \mu$

$S_1 = 32 * 3 \bmod 45 = 96 \bmod 45 = 6$

$S_2 = 32 * 5 \bmod 45 = 160 \bmod 45 = 25$

$S_3 = 32 * 12 \bmod 45 = 384 \bmod 45 = 24$

$S_4 = 32 * 21 \bmod 45 = 672 \bmod 45 = 42$

Clave pública: S = {6,25,24,42}

Clave privada: $\mu=45$, $\omega^{-1}=38$

Cifrado mochila de Merkle y Hellman (2)

Clave pública: S = {6,25,24,42}

Clave privada: $\mu = 45$, $\omega^{-1} = 38$

Como m = 4, cifraremos bloques de 4 bits, convirtiendo el mensaje a su equivalente en binario del código ASCII.

Cifrado: M = Sol = 0101 0011 0110 1111 0110 1100

C = (25+42), (24+42), (25+24), (6+25+24+42), (25+24), (6+25)

C = 67, 66, 49, 97, 49, 31

Descifrado mochila de Merkle y Hellman

Clave pública: S = {6,25,24,42}

Clave privada: $\mu = 45$, $\omega^{-1} = 38$

Cifrado: M = Sol = 0101 0011 0110 1111 0110 1100

C = (25+42), (24+42), (25+24), (6+25+24+42), (25+24), (6+25)

C = 67, 66, 49, 97, 49, 31

Como S' = {3,5,12,21}

Descifrado:

38*67 mod 45=2546 mod 45 = 26	38*97 mod 45=3686 mod 45 = 41
38*66 mod 45=2508 mod 45 = 33	38*49 mod 45=1862 mod 45 = 17
38*49 mod 45=1862 mod 45 = 17	38*31 mod 45=1178 mod 45 = 8

M = 0101 0011 0110 1111 0110 1100 = Sol

Valores de diseño de mochilas M-H (1)

Merkle y Hellman proponen los siguientes parámetros:

a) Tamaño de la mochila $m \geq 100$

b) Módulo μ uniforme en el siguiente intervalo:

Intervalo μ: $[2^{2m+1}+1, 2^{2m+2}-1] \Rightarrow 2m+2$ bits

Si m = 100: todos los elementos de S son de 202 bits.

c) Valores de S_i' elegidos uniformemente en el intervalo:

Intervalo S_i': $[(2^{i-1}-1)*2^m+1, 2^{i-1}*2^m]$

Si m = 100: $1 \leq S_1' \leq 2^{100} \leq S_2' \leq 2^{101} \leq S_3' \leq 2^{102}$ ...

d) Elegir un valor x en el intervalo $[2, \mu-2]$. El factor ω se calcula como: $\omega = mcd(\mu, x)$

Mochila con parámetros proporcionales (1)

a) Mochila con m = 6

Todos los elementos serán de (2m+2) = 14 bits

b) Intervalo μ: $[2^{2m+1}+1, 2^{2m+2}-1] = [2^{2*6+1}+1, 2^{2*6+2}-1]$
$= [2^{13}+1, 2^{14}+1] = [8.193, 16.385]$ $\mu = 13.515$

c) Elección de los valores S'_i:

$i=1 : [(2^{1-1}-1)*2^6+1, (2^{1-1})*2^6]$	$1 \le S_1' \le 64$
$i=2 : [(2^{2-1}-1)*2^6+1, (2^{2-1})*2^6]$	$65 \le S_2' \le 128$
$i=3 : [(2^{3-1}-1)*2^6+1, (2^{3-1})*2^6]$	$193 \le S_3' \le 256$
$i=4 : [(2^{4-1}-1)*2^6+1, (2^{4-1})*2^6]$	$449 \le S_4' \le 512$
$i=5 : [(2^{5-1}-1)*2^6+1, (2^{5-1})*2^6]$	$961 \le S_5' \le 1.024$
$i=6 : [(2^{6-1}-1)*2^6+1, (2^{6-1})*2^6]$	$1.985 \le S_6' \le 2.048$

UNA ELECCIÓN

$S_1' = 39$
$S_2' = 72$
$S_3' = 216$
$S_4' = 463$
$S_5' = 1.001$
$S_6' = 1.996$

Mochila con parámetros proporcionales (2)

d) Cálculo del factor ω. Buscamos un valor x en el intervalo [2, μ-2] = [2, 13.513], por ejemplo x = 9.805.

Como el máximo común divisor entre μ = 13.515 y x = 9.805 es 265, luego ω = 9.805/265 = 37.

Vamos a elegir:

ω = 37 de forma que $\omega^{-1} = 4.018$ inv (37,13.515) = 4.018

Luego, la mochila simple y la clave privada serán:

S' = {39, 72, 216, 463, 1.001, 1.996}

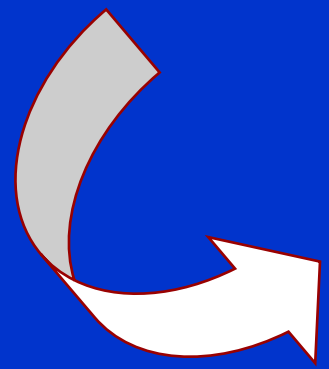
Módulo: μ = 13.515 $\omega^{-1} = 4.018$

Mochila con parámetros proporcionales (3)

Mochila simple:

S' = {39, 72, 216, 463, 1.001, 1.996} Módulo: $\mu = 13.515$

Factor multiplicador: $\omega = 37$; $\omega^{-1} = 4.018$ Clave privada



$S_1 = 39*37 \bmod 13.515 = 1.443$

$S_2 = 72*37 \bmod 13.515 = 2.664$

$S_3 = 216*37 \bmod 13.515 = 7.992$

$S_4 = 463*37 \bmod 13.515 = 3.616$

$S_5 = 1.001*37 \bmod 13.515 = 10.007$

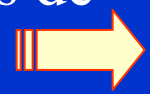
$S_6 = 1.996*37 \bmod 13.515 = 6.277$

Mochila difícil:

S = {1.443, 2.664, 7.992, 3.616, 10.007, 6.277} Clave pública

Fortaleza de las mochilas M-H

En el año 1982 Adi Shamir y Richard Zippel encuentran debilidades a las mochilas de Merkle-Hellman:

- Si se conoce el módulo μ (o puede deducirse) ...
- Y si los dos primeros elementos (S_1 y S_2) de la mochila difícil se corresponden con los dos primeros elementos (S_1' y S_2') de la mochila simple y son primos con μ ...
- Entonces podemos generar la mochila simple a partir de la difícil ya que encontraremos la clave secreta ω^{-1} ... 
- Aunque como veremos existen fuertes restricciones, esta debilidad no hace recomendable el uso de las mochilas de M-H al menos para el cifrado de la información ... ☹ 

Criptoanálisis de Shamir y Zippel

El ataque para m = 100 supone:

a) Que los dos primeros elementos de S' de 100 y 101 bits son mucho más pequeños que el módulo μ de 202 bits.

b) Que podemos identificar los elementos S_1 y S_2 en la mochila difícil y hacerlos corresponder con S_1' y S_2'.

c) Que conocemos el módulo μ o podemos deducirlo.

- Con estos datos se trata de encontrar los valores de S_1' y S_2' además del factor de multiplicación ω.
- Con estos valores podemos generar la mochila fácil S.

Pasos del ataque de Shamir y Zippel (1)

1. Se calcula $q = (S_1/S_2) \bmod \mu$

   Como $S_i = S_i' * \omega \bmod \mu$ entonces:

   $q = (S_1'/S_2') \bmod \mu = [S_1' * inv(S_2', \mu)] \bmod \mu$

   Esto implica una condición: $mcd\ (S_2', \mu) = 1$

2. Se calculan todos los múltiplos modulares del valor q con multiplicadores en el rango $[1, 2^{m+1}] = [1, 2^{101}]$

   $CM = \{1*q \bmod \mu, 2*q \bmod \mu, ..., 2^{m+1}*q \bmod \mu\}$

3. El candidato para S_1' será el valor más pequeño de CM puesto que ese elemento debe ser más pequeño de la mochila fácil S'.

Pasos del ataque de Shamir y Zippel (2)

4. Encontrado el candidato para S_1'se calcula:
   $\omega = (S_1/S_1') \bmod \mu = [S_1 * \text{inv}(S_1', \mu)] \bmod \mu$
   Esto implica una nueva condición: mcd $(S_1', \mu) = 1$
5. Conocido ω encontramos $\omega^{-1} = \text{inv}(\omega, \mu)$ y así calculamos todos los elementos de la mochila $S_i' = S_i * \omega^{-1} \bmod \mu$ que debería ser del tipo supercreciente.
6. Si no se genera una mochila supercreciente, se elige el siguiente valor más pequeño del conjunto CM y así hasta recorrer todos sus valores. Si con este conjunto CM no se obtiene una mochila simple, se repite el punto 2 tomando ahora valores en el rango 2^{m+i} con i = 2, 3, etc. Por lo general el ataque prospera con el primer conjunto CM.

Ejemplo de ataque de Shamir y Zippel (1)

La clave pública de un sistema de mochila Merkle-Hellman es:
$S = \{S_1, S_2, S_3, S_4, S_5\} = \{3.241, 572, 2.163, 1.256, 3.531\}$
Si de alguna forma hemos conseguido conocer que el módulo $\mu = 4.089$, se pide encontrar la mochila fácil $S' = \{S_1', S_2', S_3', S_4', S_5'\}$.

Solución:

- $q = S_1/S_2 \bmod \mu = S_1 * \text{inv}(S_2, \mu) \bmod \mu$. Calculamos ahora $\text{inv}(S_2, \mu)$ es decir inv (572, 4,089) = 309, luego $q = 3.241 * 309 \bmod 4.089 = 599$.
- Múltiplos CM = $\{1*q \bmod \mu, 2*q \bmod \mu, 3*q \bmod \mu, ..., 64*q \bmod \mu\}$ puesto que la mochila tiene m = 5 elementos y el intervalo será $[1, 2^{5+1}]$.
- Luego CM = [599, 1.198, 1.797, 2.396, 2.995, 3.594, 104, 703, 1.302, 1.901, 2.500, 3.099, 3.698, 208, 807, 1.406, 2.005, 2.604, 3.203, 3.802, 312, 911, 1.510, 2.109, 2.708, 3.307, 3.906, 416, 1.015, 1.614, 2.213, 2.812, 3.411, 4.010, 520, 1.119, 1.718, 2.317, 2.916, 3.515, 25, 624, 1.223, 1.822, 2.421, 3.020, 3.619, 129, 728, 1.327, 1.926, 2.525, 3.124, 3.723, 233, 832, 1.431, 2.030, 2.629, 3.228, 3.827, 337, 936, 1.535].

Ejemplo de ataque de Shamir y Zippel (2)

- Suponemos que el número más pequeño de CM es candidato a $S_1' = 25$.
- El factor de multiplicación sería $\omega = (S_1/S_1') = S_1 * inv(S_1', \mu) \bmod \mu$.
- Como $inv(S_1', \mu) = inv(25, 4{,}089) = 2.617$, el factor de multiplicación $\omega = 3.241 * 2.617 \bmod 4.089 = 1.111$.
- Por lo tanto su valor inverso será $\omega^{-1} = inv(\omega, \mu) = inv(1.111, 4.089)$. Luego $\omega^{-1} = 622$.
- Multiplicamos ahora los valores S de la mochila difícil por ω^{-1} a ver si obtenemos una mochila supercreciente S' ($S_i' = S_i * \omega^{-1} \bmod \mu$):
  - $S_1' = 25$ (valor elegido como candidato del conjunto CM)
  - $S_2' = S_2 * \omega^{-1} \bmod \mu = 572 * 622 \bmod 4.089 = 41$ 👍
  - $S_3' = S_3 * \omega^{-1} \bmod \mu = 2.163 * 622 \bmod 4.089 = 105$ 👍
  - $S_4' = S_4 * \omega^{-1} \bmod \mu = 1.256 * 622 \bmod 4.089 = 233$ 👍
  - $S_5' = S_5 * \omega^{-1} \bmod \mu = 3.531 * 622 \bmod 4.089 = 489$ 👍
- Como la mochila S = {25, 41, 105, 233, 489} es supercreciente, el ataque ha prosperado y hemos encontrado la clave privada. ☹

Usos de protección con mochilas

Existen varios algoritmos propuestos como sistemas de cifra usando el problema de la mochila: el de Graham-Shamir, Chor-Rivest, etc.

No obstante todos han sucumbido a los criptoanálisis y en la actualidad en el único entorno que se usan es en la protección de diversos programas de aplicación, en forma de hardware que se conecta en la salida paralela del computador para descifrar el código ejecutable de esa aplicación dejando, sin embargo, activa la salida a impresora. De esta manera sólo en aquel sistema con la mochila instalada se puede ejecutar el programa. No se usa en comunicaciones.

Fin del Tema 11

Cuestiones y ejercicios (1 de 2)

1. Recorra de izquierda a derecha y de derecha a izquierda la mochila S = {13, 6, 1, 3, 4, 9, 10} para T = 24. ¿Tiene solución rápida?
2. Para la mochila de la pregunta anterior, ¿hay una o más soluciones?
3. ¿Interesa usar en criptografía el problema de la mochila con una solución no única? ¿Por qué sí o no?
4. ¿Qué significa que una mochila sea supercreciente? ¿Es la mochila S = {3, 4, 9, 18, 32, 73} supercreciente? ¿Por qué?
5. A partir de la mochila S' = {3, 5, 10, 21, 43} obtenga la mochila MH difícil S. Para ω y μ use los valores mínimos posibles.
6. Si la mochila fácil es S' = {1, 2, 4, 8, 16, 32, 64, 128} con $\mu = 257$ y $\omega = 21$, cifre con una mochila de MH el mensaje en ASCII de 10 caracteres M = Hola amigo (recuerde que el espacio se cifra).
7. Descifre el criptograma obtenido en la pregunta anterior.

Cuestiones y ejercicios (2 de 2)

8. ¿Qué valores mínimos de diseño propusieron Merkle y Hellman para su sistema de cifrado con mochila? ¿Por qué?
9. Diseñe una mochila de MH con parámetros proporcionales si m = 5.
10. No es un buen criterio elegir m = 4, m = 8 o m = 16. ¿Por qué?
11. ¿En qué consiste el ataque de Shamir y Zippel a la mochila de MH?
12. En el ejemplo de los apuntes, ¿cuántas operaciones ha tenido que hacer nuestro algoritmo para romper la clave privada?
13. Con el software de mochilas de la asignatura genere una mochila difícil a partir de la mochila fácil S' = {122, 250, 506, 1.018, 2.042, 4.090, 8.186}, con $\mu = 59.369$ y $\omega = 59.361$. Cifre algo con ella y luego haga un ataque. ¿Cuántas mochilas S encuentra? Coméntelo.
14. ¿Usaría un sistema de mochila para cifrar información en un entorno como Internet? ¿Y en una intranet para respuestas a un examen?

Tema 12
Cifrado Asimétrico Exponencial

Seguridad Informática y Criptografía

Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 56 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Cifrado exponencial con clave del receptor

- En el cifrado del mensaje $E_e(M) = C$ y en el descifrado del criptograma $E_d(C) = M$, se usa una exponenciación.
- En la operación de cifrado, el subíndice e significa el uso de la clave pública del receptor (R) en el extremo emisor y el subíndice d el uso de la clave privada del receptor (R) en el extremo receptor.

$$C = E_{eR}(M) = M^{eR} \bmod n_R \quad \Rightarrow \quad M = E_{dR}(C) = C^{dR} \bmod n_R$$

- M debe ser un elemento del CCR de n_R.
- Esta operación se usará para realizar el intercambio de una clave de sesión entre un emisor y un receptor.

Cifrado exponencial con clave del emisor

- En la operación de cifrado el subíndice d significa el uso de la clave privada del emisor (E) en el extremo emisor y el subíndice e el uso de la clave pública del emisor (E) en el extremo receptor.

$$C = E_{dE}(M) = M^{dE} \bmod n_E \Rightarrow M = E_{eE}(C) = C^{eE} \bmod n_E$$

- M debe ser un elemento del CCR de n_E.
- Esta operación se usará para autenticar la identidad de un usuario mediante una firma digital al mismo tiempo que la integridad del mensaje.

Cifrado exponencial genérico tipo RSA

Sea el grupo de trabajo $n = p*q \Rightarrow \phi(n) = (p-1)(q-1)$

Se eligen una clave pública e y una privada d de forma que:
$e*d \bmod \phi(n) = 1 \Rightarrow e*d = k(p-1)(q-1) + 1.$

Si $e*d = k\phi(n) + 1$

Por el Teorema de Euler se tiene que:

$$M^{k\phi(n)} \bmod n = 1$$

para todo M primo con n

y ...

Por el Teorema del Resto Chino se tiene que:

$$M^{ed} = M \bmod n$$

ssi

$$M^{ed} = M \bmod p$$
$$M^{ed} = M \bmod q$$

Luego, el sistema de cifra es válido para cualquier valor de M

Operación de descifrado exponencial

Al cifrar el mensaje M con una clave pública e (en este caso para intercambio de clave, aunque es igual de válido con una clave d en caso de firma digital) tenemos:

Cifrado: $C = M^e \bmod n$

Descifrado: $C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$

$C^d \bmod n = M^{k\phi(n)+1} \bmod n = M * M^{k\phi(n)} \bmod n$

$C^d \bmod n = M * 1 \bmod n = M \bmod n$

Por lo tanto, la operación $C^d \bmod n$ recupera el mensaje M.

Comprobación de recuperación de texto

Sea $n = p*q = 5*11 = 55 \Rightarrow \phi(n) = (5-1)(11-1) = 40$

Mensaje $M = 50 = 2*5^2$ (*debe ser un elemento de n*)

Se elige $e = 3 \Rightarrow d = inv[e, \phi(n)] = inv(3,40) = 27$

$e*d \bmod \phi(n) = 3*27 \bmod 40 = 81 \bmod 40 = 1$

$C = M^e \bmod n = 50^3 \bmod 55 = (2*5^2)^3 \bmod 55$

$C = [(2)^3 \bmod 55 * (5^2)^3 \bmod 55] \bmod 55$ - *por reducibilidad* ↓ -

$M = C^d \bmod n = \{[(2)^3 \bmod 55 * (5^2)^3 \bmod 55] \bmod 55\}^{27} \bmod 55$

$M = [(2)^{3*27} \bmod 55 * (5^2)^{3*27} \bmod 55] \bmod 55$

$M = [2^{2\phi(n)+1} * 5^{2\phi(n)+1} * 5^{2\phi(n)+1}] \bmod 55$

Por el Teorema de Euler y del Resto Chino $= 2 * 5 * 5 \bmod 55 = 50$

Intercambio de clave de Diffie y Hellman

- El comienzo de los sistemas de clave pública se debe al estudio hecho por Whitfield Diffie y Martin Hellman (1976).

  Protocolo de Intercambio de Claves de Diffie y Hellman

  A y B seleccionan un grupo multiplicativo (con inverso) p y un generador α de dicho primo, ambos valores públicos

  - A genera un número aleatorio a y envía a B $\alpha^a \bmod p$
  - B genera un número aleatorio b y envía a A $\alpha^b \bmod p$
  - B calcula $(\alpha^a)^b \bmod p = \alpha^{ab} \bmod p$ y luego destruye b
  - A calcula $(\alpha^b)^a \bmod p = \alpha^{ba} \bmod p$ y luego destruye a
  - El secreto compartido por A y B es el valor $\alpha^{ab} \bmod p$

Ejemplo de intercambio de clave de DH

Adela (A) y Benito (B) van a intercambiar una clave de sesión dentro del cuerpo p = 1.999, con $\alpha = 33$. El usuario A elegirá a = 47 y el usuario B elegirá b = 117.

Algoritmo:

- A calcula $\alpha^a \bmod p = 33^{47} \bmod 1.999 = 1.343$ y se lo envía a B.
- B calcula $\alpha^b \bmod p = 33^{117} \bmod 1.999 = 1.991$ y se lo envía a A.
- B recibe 1.343 y calcula $1.343^{117} \bmod 1.999 = 1.506$.
- A recibe 1.991 y calcula $1.991^{47} \bmod 1.999 = 1.506$.

La clave secreta compartida por (A) y (B) será 1.506

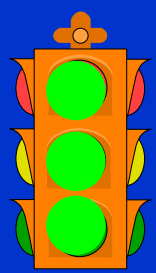
¿Puede un intruso atacar la clave DH?

Un intruso que conozca las claves públicas p y α e intercepte el valor $\alpha^a \bmod p$ que ha transmitido A y el valor $\alpha^b \bmod p$ transmitido por B no podrá descubrir los valores de a, b ni $\alpha^{ab} \bmod p$ ...

salvo que se enfrente al Problema del Logaritmo Discreto (PLD) que, como ya hemos visto, se vuelve computacionalmente intratable para valores del primo p grandes.

¿Es vulnerable el protocolo de DH?

- A elige un número a con $1 < a < p-1$ y envía a B $\alpha^a \bmod p$
- C intercepta este valor, elige un número c con $1 < c < p-1$ y envía a B $\alpha^c \bmod p$
- B elige un número b con $1 < b < p-1$ y envía a A $\alpha^b \bmod p$
- C intercepta este valor y envía a A $\alpha^c \bmod p$ (valor anterior)
- A y B calculan sus claves $k_A = (\alpha^c)^a \bmod p$, $k_B = (\alpha^c)^b \bmod p$
- C calcula también las claves:
  - $k_{CA} = (\alpha^a)^c \bmod p$
  - $k_{CB} = (\alpha^b)^c \bmod p$



La solución a este problema es el sellado de tiempo

Por lo tanto, a partir de ahora C tiene "luz verde" y puede interceptar todos los mensajes que se intercambian A y B.

Intercambio de clave DH entre n usuarios

El protocolo DH se puede generalizar para n usuarios: sea n = 3.

A, B y C seleccionan un grupo p y un generador α

- A genera un número aleatorio a y envía $\alpha^a \bmod p$ a B
- B genera un número aleatorio b y envía $\alpha^b \bmod p$ a C
- C genera un número aleatorio c y envía $\alpha^c \bmod p$ a A
- A recibe $\alpha^c \bmod p$ y calcula $(\alpha^c)^a \bmod p$ y se lo envía a B
- B recibe $\alpha^a \bmod p$ y calcula $(\alpha^a)^b \bmod p$ y se lo envía a C
- C recibe $\alpha^b \bmod p$ y calcula $(\alpha^b)^c \bmod p$ y se lo envía a A
- A recibe $\alpha^{bc} \bmod p$ y calcula $(\alpha^{bc})^a \bmod p = \alpha^{bca} \bmod p$
- B recibe $\alpha^{ca} \bmod p$ y calcula $(\alpha^{ca})^b \bmod p = \alpha^{cab} \bmod p$
- C recibe $\alpha^{ab} \bmod p$ y calcula $(\alpha^{ab})^c \bmod p = \alpha^{abc} \bmod p$

- El secreto compartido por A, B y C es el valor $\alpha^{abc} \bmod p$

Condiciones del intercambio de clave D-H

CONDICIONES DEL PROTOCOLO:

- El módulo p debe ser un primo grande, al menos 1.024 bits.
- Interesa que p-1 debe tener factores primos grandes.
- El generador α debe ser una raíz primitiva del módulo p.

☹ Si el módulo es un primo pequeño, se puede hacer un ataque por fuerza bruta dentro de un tiempo razonable.

☹ Si el generador no es una raíz primitiva del grupo p, entonces la operación $\alpha^i \bmod p$ $(1 \leq i \leq p-1)$ no genera todos los restos del grupo y esto facilita el ataque por fuerza bruta.

Veamos un ejemplo →

Raíz α incorrecta (falsa)

MALA ELECCIÓN DE LOS PARÁMETROS:

Sean el grupo de trabajo p = 13 y un valor α = 3 ... *entonces*

$3^1 \bmod 13 = 3$ $3^2 \bmod 13 = 9$ $3^3 \bmod 13 = 1$ $1 \Rightarrow$ *vamos mal*

$3^4 \bmod 13 = 3$ $3^5 \bmod 13 = 9$ $3^6 \bmod 13 = 1$

$3^7 \bmod 13 = 3$ $3^8 \bmod 13 = 9$ $3^9 \bmod 13 = 1$

$3^{10} \bmod 13 = 3$ $3^{11} \bmod 13 = 9$ $3^{12} \bmod 13 = 1$ $(\alpha^{p-1} \bmod p = 1)$

Sólo debería darse unidad en este caso.

Se repiten los restos 3, 9 y 1 porque 3 no es un generador de Z_{13}.
Observe que $3^4 \bmod 13 = (3^3)(3)^1 \bmod 13 = 1*(3)^1 \bmod 13 = 3$.

Un ataque por fuerza bruta deberá buscar sólo en una tercera parte del espacio de claves y, lo que es peor, la probabilidad de éxito de encontrar un valor verdadero b en $\alpha^b \bmod p$ aumenta de 1/12 a 1/3.

Raíz α correcta

¿Y si ahora α = 2 ?

Primero intente calcularlo... y luego para comprobar sus resultados, avance.

$2^1 \bmod 13 = 2$ $2^2 \bmod 13 = 4$ $2^3 \bmod 13 = 8$

$2^4 \bmod 13 = 3$ $2^5 \bmod 13 = 6$ $2^6 \bmod 13 = 12$

$2^7 \bmod 13 = 11$ $2^8 \bmod 13 = 9$ $2^9 \bmod 13 = 5$

$2^{10} \bmod 13 = 10$ $2^{11} \bmod 13 = 7$ $2^{12} \bmod 13 = 1$

Ahora sí están todos los restos multiplicativos del cuerpo Z_{13} porque el resto 2 es un generador dentro de este cuerpo.
Observe que el valor unidad sólo se obtiene para $\alpha^{p-1} \bmod p$.

Como vimos en el capítulo de Teoría de Números, en p = 13 serán generadores g = 2, 6, 7, 11.

Algoritmo de cifra asimétrica RSA

En febrero de 1978 Ron Rivest, Adi Shamir y Leonard Adleman proponen un algoritmo de cifra de clave pública: RSA

Algoritmo

1. Cada usuario elige un grupo $n = p*q$ (pueden ser distintos).
2. Los valores p y q no se hacen públicos.
3. Cada usuario calcula $\phi(n) = (p-1)(q-1)$.
4. Cada usuario elige una clave pública e que sea parte del cuerpo n y que cumpla: $mcd\ [e, \phi(n)] = 1$.
5. Cada usuario calcula la clave privada $d = inv\ [e,\phi(n)]$.
6. Se hace público el grupo n y la clave e.
7. Se guarda en secreto la clave d.

Podrían destruirse ahora p, q y $\phi(n)$.

Ejemplo de cifrado y descifrado con RSA

Grupo n = 91 = 7∗13; $\phi(n) = \phi(7*13) = (7-1)(13-1) = 72$ M = 48

Elegimos e = 5 pues mcd (5,72) = 1 ∴ d = inv(5,72) = 29

CIFRADO:

$C = M^e \bmod n = 48^5 \bmod 91 = 5245.803.968 \bmod 91 = 55$

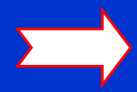
DESCIFRADO:

$M = C^d \bmod n = 55^{29} \bmod 91 = 48$ ... 55^{29} ya es "*número grande*"

55^{29} es un número con 51 dígitos...
$55^{29} = 295473131755644748809642476009391248226165771484375$
¿Cómo podemos acelerar esta operación?

1ª opción: usar reducibilidad 

2ª opción: algoritmo exp. rápida 

Opción óptima: usar el Teorema del Resto Chino  

Uso del Teorema del Resto Chino en RSA

- Normalmente la clave pública e de RSA es un valor bastante bajo, por ejemplo $2^{16} + 1$. Luego, en el proceso de cifra (no en la firma) no tendremos problemas con la velocidad de cifrado porque el exponente e es relativamente bajo.
- Como el cuerpo de trabajo n = p*q es mucho mayor, del orden de 2^{1024} si hablamos de claves de 1024 bits, entonces la clave privada d será por lo general mucho mayor que el valor de e. Por lo tanto, puede ser costoso para el receptor descifrar algo con su clave privada o bien firmar digitalmente un documento con dicha clave.
- La solución está en aplicar el Teorema del Resto Chino. En vez de trabajar en n, lo haremos en p y q. Entonces las exponenciaciones modulares se harán en p y q, mucho más rápido que hacerlo en n.

Descifrado RSA aplicando el TRC

$M = C^d \bmod n$ Aplicando el Teorema del Resto Chino en n:

$$M = \{A_p[C_p^{dp} \pmod p] + A_q[C_q^{dq} \pmod q]\} \bmod n$$

con $C_p = C \bmod p$ $\quad C_q = C \bmod q$

$d_p = d \bmod (p-1)$ $\quad d_q = d \bmod (q-1)$

$A_p = q\,[\text{inv}\,(q,p)] = q^{p-1} \bmod n$

$A_q = p\,[\text{inv}\,(p,q)] = p^{q-1} \bmod n$

Recuerde que p y q son la trampa que sólo conoce el dueño de la clave

Ejemplo de descifrado RSA usando el TRC

Sea: $p = 89$, $q = 31$, $n = p*q = 89*31 = 2.759$, $\phi(n) = 88*30 = 2.640$

Elegimos $e = 29 \Rightarrow d = inv\,[e, \phi(n)] = inv\,[29, 2.640] = 2.549$

Si el bloque de mensaje ya codificado es $M = 1.995$, entonces:

$C = M^e \bmod n = 1.995^{29} \bmod 2.759 = 141$

$M = C^d \bmod n = 141^{2.549} \bmod 2.759 = 1.995$

$A_p = q^{p-1} \bmod n = 31^{88} \bmod 2.759 = 713$	$A_q = p^{q-1} \bmod n = 89^{30} \bmod 2.759 = 2.047$
$C_p = C \bmod p = 141 \bmod 89 = 52$	$C_q = C \bmod q = 141 \bmod 31 = 17$
$d_p = d \bmod (p-1) = 2.549 \bmod 88 = 85$	$d_q = d \bmod (q-1) = 2.549 \bmod 30 = 29$

Reemplazando en: $M = \{A_p[C_p^{dp} (\bmod p)] + A_q[C_q^{dq} (\bmod q)]\} \bmod n$

$M = \{713[52^{85} \bmod 89] + 2.047[17^{29} \bmod 31]\} \bmod 2.759$

$M = \{713*37 + 2.047*11\} \bmod 2.759 = (26.381 + 22.517) \bmod 2.759 = 1.995$

El problema en la elección del valor de n

Si p y q son muy cercanos, es fácil factorizar n.

- Si $p \approx q$ y suponemos que $p > q$, entonces $(p-q)/2$ es un entero muy pequeño y por otra parte $(p+q)/2$ es ligeramente superior a $\sqrt{n}$.
- Además se cumplirá que: $n = (p+q)^2/4 - (p-q)^2/4$. Esto lo podemos escribir como $n = x^2 - y^2 \Rightarrow y^2 = x^2 - n$
- Elegimos enteros $x > \sqrt{n}$ hasta que $(x^2 - n)$ sea cuadrado perfecto. En este caso $x = (p+q)/2$; $y = (p-q)/2$. Por lo tanto rompemos el valor n: $p = (x+y)$; $q = (x-y)$.

Ejemplo de mala elección del valor de n

- Sea p = 181; q = 251 $\Rightarrow$ n = 181*251 = 45.431
- Como $\sqrt{45.431} = 213{,}14$ buscaremos valores enteros de x mayores que 213 de forma que $(x^2 - 45.431)$ sea un cuadrado perfecto ↓

1. $x = 214 \Rightarrow x^2 - 45.431 = 365$ $\therefore$ $\sqrt{365} = 19{,}10$ ☹
2. $x = 215 \Rightarrow x^2 - 45.431 = 794$ $\therefore$ $\sqrt{794} = 28{,}17$ ☹
3. $x = 216 \Rightarrow x^2 - 45.431 = 1.225$ $\therefore$ $\sqrt{1.225} = 35$ ☺

   Entonces: $p = x - y = 216 - 35 = 181$

   $q = x + y = 216 + 35 = 251$

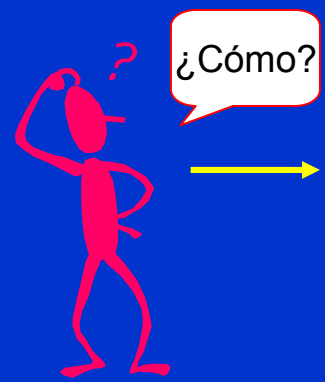
Para evitar otros problemas, es recomendable usar los denominados primos seguros.

Elección de los números primos

Los valores primos deben elegirse apropiadamente:

Sistema RSA

a) p y q deben diferir en unos pocos dígitos.
b) p y q no deben ser primos muy cercanos.
c) Longitud mínima de p y q: 500 bits.
d) Valores de (p-1) y (q-1) del Indicador de Euler deben tener factores primos grandes.
e) El mcd entre p-1 y q-1 debe ser pequeño.



Cálculo de números primos seguros

Primos seguros: se elige r un primo grande de modo que:

$p = 2*r + 1$; $q = 2*p + 1$ sean primos y mcd (p-1, q-1) pequeño.

EJEMPLO: Si r es el primo de 4 dígitos 1.019:

$p = 2*1.019 + 1 = 2.039$ Es primo

$q = 2*2.039 + 1 = 4.079$ Es primo

Luego: $p-1 = 2.038$; $q-1 = 4.078$

$p-1 = 2*1.019$; $q-1 = 2*2.039 \Rightarrow$ mcd (p-1, q-1) = 2

Luego los primos p y q cumplen la condición de primos seguros

El módulo será $n = p*q = 8.317.081$

El problema de las claves privadas parejas

Una clave pareja d', permite descifrar el criptograma C resultado de una cifra con la clave pública e sin que d' sea el inverso de la clave pública e. En el sistema RSA habrá como mínimo una clave d' pareja de la clave pública e.

Ejemplo:

Si p = 5; q = 13; n = 65, $\phi(n) = 48 = 2^4 x 3$, elegimos e = 5. Calculamos d = inv (5, 48) = 29 que es único. Si ciframos el mensaje M = 59, obtenemos $C = 59^5 \bmod 65 = 24$. Luego sabemos que $M = C^d \bmod n = 24^{29} \bmod 65 = 59$ ✌

¡Pero también lo desciframos con d' = 5, 17, 41 y 53! ☹

Número de claves privadas parejas

Si: $\gamma = mcm\ (p-1),(q-1)$ sea $d_\gamma = e^{-1} \bmod \gamma = inv\ (e, \gamma)$
La clave pública e tendrá λ claves parejas de forma que:

$$d_i = d_\gamma + i\,\gamma \qquad 1 < d_i < n$$
$$i = 0, 1, \ldots \lambda-1 \qquad \lambda = \lfloor (n - d_\gamma)/\gamma \rfloor$$

En el ejemplo anterior tenemos que $\gamma = mcm\ (4, 12) = 12$.
Luego: $d_\gamma = inv\ (5, 12) = 5$, así $d_i = d_\gamma + i\,\gamma = 5 + i*12$.
Es decir $d_i = 5, 17, 29, 41, 53$; el número de claves que corresponden al valor $\lambda = \lfloor (n - d_\gamma)/\gamma \rfloor = \lfloor (65 - 5)/12) = 5$.

En este caso hablamos de debilidad de la clave, al igual que sucedía con las claves débiles y semidébiles por ejemplo en DES e IDEA.

Minimizando el número de claves parejas

Para que λ sea lo más pequeño posible ($\lambda = 1$) deberemos elegir los primos p y q como primos seguros.

Ejemplo:

Sea $r = 5 \Rightarrow p = 2*r + 1 = 11$ (es primo 👍)

$q = 2*p + 1 = 23$ (es primo 👍)

En estas condiciones con $n = 253$ y $\phi(n) = 220$, sea $e = 7$

Luego $\gamma = mcm\ (10, 22) = 110$ y $d_\gamma = inv\ (7, 110) = 63$

Entonces $\lambda = \lfloor (n - d_\gamma)/\gamma \rfloor = \lfloor (253 - 63)/110 \rfloor = 1$

Así: $d_i = d_\gamma + i\,\gamma = 63 + i*110 = 63, 173$ (con $i = 0, 1$)

En efecto $63 = inv\ [e, \phi(n)] = inv\ (7, 220)$ y lo cifrado con $e = 7$ también se descifra con $d' = 173$. Sólo dos claves.

Mensajes no cifrables

- Si $M^e \bmod n = M$ se dice que el mensaje es no cifrable. Aunque la clave e sea válida, éste se envía en claro ☹.
- En RSA habrá como mínimo 9 mensajes no cifrables.

Ejemplo:

Sea el cuerpo n = 35 (p = 5, q = 7), con $\phi(n) = 24$ y e = 11. Dentro de los mensajes posibles {0, 34} serán no cifrables: {6, 14, 15, 20, 21, 29, 34} además de los obvios {0, 1}.

- En el caso más crítico, todos los mensajes del cuerpo n pueden ser no cifrables como veremos más adelante.

¿Cómo se obtienen estos mensajes no cifrables? 

Número de mensajes no cifrables

El número de mensajes no cifrables dentro de un cuerpo n viene dado por:

$$\sigma_n = [1 + mcd\ (e-1, p-1)][1 + mcd\ (e-1, q-1)]$$

Los mensajes no cifrables serán:

$$M = [q\{inv\ (q, p)\}M_p + p\{inv\ (p, q)\}M_q] \bmod n$$

con: M_p las soluciones de $M^e \bmod p = M$

M_q las soluciones de $M^e \bmod q = M$

Esto último debido al TRC puesto que $M^e \bmod n = M$

En el ejemplo anterior se da el caso mínimo: ↓ ↓

$\sigma_n = [1 + mcd\ (10, 4)][1 + mcd\ (10, 6)] = (1+2)(1+2) = 9$

$M^{11} \bmod 5 = M \Rightarrow M_5 = \{0, 1, 4\}$ $M^{11} \bmod 7 = M \Rightarrow M_7 = \{0, 1, 6\}$

$M = [7\{inv\ (7, 5)\}M_p + 5\ \{inv\ (5,7)\}M_q] \bmod 35$

$M = [7*3\ M_p + 5*3\ M_q] \bmod 35 = [21\{0, 1, 4\} + 15\ \{0, 1, 6\}] \bmod 35$

$M = \{0, 15, 90, 21, 36, 111, 84, 99, 175\} \bmod 35$ ordenando...

$M = \{0, 1, 6, 14, 15, 20, 21, 29, 34\}$

Ejemplo de mensajes no cifrables (1)

Sea $p = 13$; $q = 17$; $n = p*q = 221$

Elegimos $e = 7$ por lo que $d = \text{inv}(7, 192) = 55$, luego:

$\sigma_n = [1 + \text{mcd}(e-1, p-1)][1 + \text{mcd}(e-1, q-1)]$

$\sigma_{221} = [1 + \text{mcd}(6, 12)][1 + \text{mcd}(6, 16)] = (1+6)(1+2) = 21$

Soluciones de $M^7 \bmod 13 = M \Rightarrow M_p = \{0, 1, 3, 4, 9, 10, 12\}$

Soluciones de $M^7 \bmod 17 = M \Rightarrow M_q = \{0, 1, 16\}$

Los mensajes no cifrables serán:

$M = [q\{\text{inv}(q, p)\}M_p + p\{\text{inv}(p, q)\}M_q] \bmod n$

$M = [17\{\text{inv}(17, 13)\}M_p + 13\{\text{inv}(13, 17)\}M_q] \bmod 221$

$M = [\{17*10\}M_p + \{13*4\}M_q] \bmod 221$

$M = [170*M_p + 52*M_q] \bmod 221$

Ejemplo de mensajes no cifrables (2)

Teníamos $M_p = \{0, 1, 3, 4, 9, 10, 12\}$

$M_q = \{0, 1, 16\}$

$M = [170*M_p + 52*M_q] \bmod 221$ luego:

$M = [170*\{0, 1, 3, 4, 9, 10, 12\} + 52*\{0, 1, 16\}] \bmod 221$

Como $52*\{0, 1, 16\} = \{0, 52, 832\}$ entonces:

$M = [0+0, 0+52, 0+832, 170+0, 170+52, 170+832, ...] \bmod 221$

$M = [0, 52, 832, 170, 222, 1.002, 510, 562, 1.342, 680, 732, 1.512, 1.530, 1.582, 2.362, 1.700, 1.752, 2.531, 2.040, 2.092, 2.872] \bmod 221$

$M = [0, 52, 169, 170, 1, 118, 68, 120, 16, 17, 69, 186, 204, 35, 152, 153, 205, 101, 51, 103, 220]$ ordenando:

$M = [0, 1, 16, 17, 35, 51, 52, 68, 69, 101, 103, 118, 120, 152, 153, 169, 170, 186, 204, 205, 220]$ los 21 mensajes de σ_{221}

Número mínimo de mensajes no cifrables

Para que el número de mensajes no cifrables sea el mínimo posible, es decir 9, deberemos elegir la clave pública e de forma que:

$$\text{mcd}(e-1, p-1) = 2 \quad \text{y} \quad \text{mcd}(e-1, q-1) = 2$$

Entonces: $\sigma_n = [1 + 2][1 + 2] = 9$

Esto se logra usando primos seguros:

$p = 2r + 1$ y $q = 2p + 1$ con r, p y q primos grandes

ya que: $\text{mcd}(e-1, p-1) = \text{mcd}(e-1, (2r+1)-1) \Rightarrow \text{mcd} = 2$ o bien r

$\text{mcd}(e-1, q-1) = \text{mcd}(e-1, (2p+1)-1) \Rightarrow \text{mcd} = 2$ o bien p

Luego: $\sigma_n = \{9, 3(r+1), 3(p+1), (r+1)(p+1)\}$

Hay que comprobar en diseño que no se den valores del mcd igual a r o p pues tendríamos un número muy alto de este tipo de mensajes.

Número máximo de mensajes no cifrables

En el peor de los casos, mcd (e-1, p-1) = p-1 y mcd (e-1, q-1) = q-1

Entonces: $\sigma_n = [1 + mcd(e-1, p-1)][1 + mcd(e-1, q-1)]$

$\sigma_n = p*q = n$ ... ¡todas las cifras irán en claro!

Si en el ejemplo anterior con p = 13, q = 17, hubiésemos elegido como clave e = 49, con d = inv (49, 192) = 145, observamos que:

mcd (e-1, p-1) = mcd (48, 12) = 12

mcd (e-1, q-1) = mcd (48, 16) = 16

$\sigma_n = [1 + 12][1 + 16] = 13*17 = 221 = p*q = n$

Por lo tanto, cualquier mensaje en el cuerpo n = 221 será no cifrable para la clave pública e = 49. Compruebe que en este caso esto se cumple si $e = \phi(n)/k + 1$ (k = 1, 2 y 4), es decir e = 193, 97 y 49.

Nunca podrá usarse $e = \phi(n) + 1$ ya que la clave de descifrado es 1 ni $e = \phi(n)/2 + 1$ pues tampoco se cifrarán todos los mensajes de n.

Ataque a la clave por factorización de n

¿Qué fortaleza tendrá este algoritmo ante ataques?

- El intruso que desee conocer la clave secreta d a partir de los valores públicos n y e se enfrentará al Problema de la Factorización de Números Grandes (PFNG) puesto que la solución para conocer esa clave privada pasa por deducir el valor del Indicador de Euler $\phi(n) = (p-1)(q-1)$ para así poder encontrar el inverso de la clave pública $d = inv\ [e,\phi(n)]$.
- La complejidad asociada al PFNG viene dada por la ecuación $e^{\sqrt{\ln(n)}\ \ln\ln(n)}$, donde ln es logaritmo natural.

Tiempo necesario para afrontar el PFNG

Nº de bits (n)	Nº de dígitos	Días	Años
60	18	$1,7 \times 10^{-8}$	-
120	36	$1,5 \times 10^{-5}$	-
256	77	1,0	-
363	109	$9,0 \times 10^{2}$	2,5
442	133	$9,4 \times 10^{4}$	$2,5 \times 10^{2}$
665	200	$3,8 \times 10^{8}$	$1,0 \times 10^{6}$

Para un procesador de 2×10^{8} instrucciones por segundo.
Fuente: Criptografía Digital, José Pastor. Prensas Univ. de Zaragoza, 1998.

Existen, no obstante, otros tipos de ataques a este sistema que no pasan por la factorización de n.

Ataque al secreto de M por cifrado cíclico

Se puede encontrar el mensaje en claro M sin necesidad de conocer d, la clave privada del receptor.

Como $C = M^e \bmod n$, realizaremos cifrados sucesivos de los criptogramas C_i resultantes con la misma clave pública hasta obtener nuevamente el cifrado C original.

$$C_i = C^e_{i-1} \bmod n \quad (i = 1, 2, ...) \text{ y } C_0 = C$$

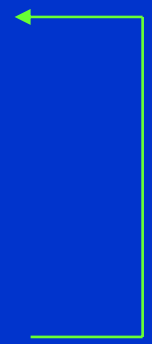
Si en el cifrado iésimo se encuentra el criptograma C inicial, entonces es obvio que el cifrado (i-1) será el mensaje buscado.
Esto se debe a que RSA es un grupo mutiplicativo. Para evitarlo hay que usar primos seguros de forma que los subgrupos de trabajo sean bastante altos.

Ejemplo de ataque por cifrado cíclico

Sea p = 13, q = 19, n = 247, $\phi(n) = 216$, e = 29 (d = 149, no conocido)
El mensaje a cifrar será M = 123 $\Rightarrow$ $C = 123^{29} \bmod 247 = 119$

Tabla de cifra

i	C_i
i = 0	$C_0 = 119$
i = 1	$C_1 = 119^{29} \bmod 247 = 6$
i = 2	$C_2 = 6^{29} \bmod 247 = 93$
i = 3	$C_3 = 93^{29} \bmod 247 = 175$
i = 4	$C_4 = 175^{29} \bmod 247 = 54$
i = 5	$C_5 = 54^{29} \bmod 247 = 123$
i = 6	$C_6 = 123^{29} \bmod 247 = 119$



El ataque ha prosperado muy rápidamente: como hemos obtenido otra vez el criptograma C = 119, es obvio que el paso anterior con C = 123 se correspondía con el texto en claro. ¿Y si usamos primos seguros?

Ataque cifrado cíclico: caso primos seguros

Sea p = 11, q = 23, n = 253, $\phi(n) = 220$, e = 17 (d = 134, no conocido)

Sea el mensaje secreto M = 123 $\Rightarrow$ $C = 123^{17} \bmod 253 = 128$

i	C_i
i = 0	$C_0 = 128$
i = 1	$C_1 = 128^{17} \bmod 253 = 6$
i = 2	$C_2 = 6^{17} \bmod 253 = 173$
i = 3	$C_3 = 173^{17} \bmod 253 = 101$
i = 4	$C_4 = 101^{17} \bmod 253 = 95$
i = 5	$C_5 = 95^{17} \bmod 253 = 39$
i = 6	$C_6 = 39^{17} \bmod 253 = 96$
i = 7	$C_7 = 96^{17} \bmod 253 = 2$
i = 8	$C_8 = 2^{17} \bmod 253 = 18$
i = 9	$C_9 = 18^{17} \bmod 253 = 215$
i = 10	$C_{10} = 215^{17} \bmod 253 = 151$
i = 11	$C_{11} = 151^{17} \bmod 253 = 167$
i = 12	$C_{12} = 167^{17} \bmod 253 = 150$
i = 13	$C_{13} = 150^{17} \bmod 253 = 193$
i = 14	$C_{14} = 193^{17} \bmod 253 = 118$
i = 15	$C_{15} = 118^{17} \bmod 253 = 200$
i = 16	$C_{16} = 200^{17} \bmod 253 = 73$
i = 17	$C_{17} = 73^{17} \bmod 253 = 94$
i = 18	$C_{18} = 94^{17} \bmod 253 = 41$
i = 19	$C_{19} = 41^{17} \bmod 253 = 123$ ✓
i = 20	$C_{20} = 123^{17} \bmod 253 = 128$

Hemos tenido que recorrer un espacio mucho mayor dentro de un cuerpo de cifra similar.

Ataque a la clave por paradoja cumpleaños

La paradoja del cumpleaños se verá en el próximo capítulo.

Algoritmo propuesto por Merkle y Hellman en 1981:

- El atacante elige dos números aleatorios distintos i, j dentro del cuerpo de cifra n. Elige además un mensaje M cualquiera.
- Para i = i+1 y para j = j+1 calcula M^i mod n y M^j mod n.
- Cuando encuentra una coincidencia de igual resultado para una pareja (i, j), puede encontrar d.

Ejemplo: sea p = 7; q = 13, n = p*q = 81, e = 11, d = 59. El atacante elige los valores i = 10, j = 50 y parte con el mensaje M = 20. Sólo conoce n = 81 y e = 11. ⟶

Ejemplo de ataque paradoja cumpleaños

i	C_i
i = 10	$C_{10} = 20^{10} \bmod 81 = 34$
i = 11	$C_{11} = 20^{11} \bmod 81 = 32$
i = 12	$C_{12} = 20^{12} \bmod 81 = 73$
i = 13	$C_{13} = 20^{13} \bmod 81 = 2$
i = 14	$C_{14} = 20^{14} \bmod 81 = 40$
i = 15	$C_{15} = 20^{15} \bmod 81 = 71$
i = 16	$C_{16} = 20^{16} \bmod 81 = 43$
i = 17	$C_{17} = 20^{17} \bmod 81 = 50$
i = 18	$C_{18} = 20^{18} \bmod 81 = 28$
i = 19	$C_{19} = 20^{19} \bmod 81 = 74$
i = 20	$C_{20} = 20^{20} \bmod 81 = 22$
i = 21	$C_{21} = 20^{21} \bmod 81 = 35$
i = 22	$C_{22} = 20^{22} \bmod 81 = 52$
i = 23	$C_{23} = 20^{23} \bmod 81 = 68$
i = 24	$C_{24} = 20^{24} \bmod 81 = 64$

j	C_i
i = 50	$C_{50} = 20^{50} \bmod 81 = 13$
i = 51	$C_{51} = 20^{51} \bmod 81 = 17$
i = 52	$C_{52} = 20^{52} \bmod 81 = 16$
i = 53	$C_{53} = 20^{53} \bmod 81 = 77$
i = 54	$C_{54} = 20^{54} \bmod 81 = 1$
i = 55	$C_{55} = 20^{55} \bmod 81 = 20$
i = 56	$C_{56} = 20^{56} \bmod 81 = 76$
i = 57	$C_{57} = 20^{57} \bmod 81 = 62$
i = 58	$C_{58} = 20^{58} \bmod 81 = 25$
i = 59	$C_{59} = 20^{59} \bmod 81 = 14$
i = 60	$C_{60} = 20^{60} \bmod 81 = 37$
i = 61	$C_{61} = 20^{61} \bmod 81 = 11$
i = 62	$C_{62} = 20^{62} \bmod 81 = 58$
i = 63	$C_{63} = 20^{63} \bmod 81 = 26$
i = 64	$C_{64} = 20^{64} \bmod 81 = 34$

Resultado del ataque paradoja cumpleaños

NOTA: Puesto que la explicación matemática de este algoritmo podría resultar algo compleja y larga para exponerla en estos apuntes, se dará sólo la solución numérica al problema que, como verá, es muy sencilla.

La primera coincidencia se encuentra para i = 10; j = 64. Así, el atacante conociendo la clave pública e = 11, calcula:

$$w = (10\text{-}64) / \text{mcd}\ (11, 10\text{-}64) = -54 / \text{mcd}\ (11, -54) = -54.$$

Entonces deberán existir valores s,t de forma que se cumpla lo siguiente:

$$w*s + e*t = 1 \quad \Rightarrow \quad -54*s + 11*t = 1$$

Las posibles soluciones a la ecuación son: $w*s \bmod e = 1$; $e*t \bmod w = 1$

$$-54*s = 1 \bmod 11 \quad \Rightarrow \quad s = \text{inv}\ (-54, 11) = 1$$

$$11*t = 1 \bmod 54 \quad \Rightarrow \quad t = \text{inv}\ (11, 54) = 5 \leftarrow$$

El valor t = 5 será una clave secreta d' pareja de d = 59. Compruebe que se verifica $w*s + e*t = 1$. Observe que $5 \bmod 54 = 59 \bmod 54 = 5$.

La otra historia del algoritmo RSA

- Rivest, Shamir y Adleman son los autores de RSA pero el desarrollo de un algoritmo de cifra asimétrico que usara el problema de factorizar números grandes como función unidireccional fue descubierto mucho antes.
- En el año 1969 el GCHQ (Government Communications Headquarters) en Gran Bretaña comienza a trabajar en la idea de poder distribuir claves a través de una cifra no simétrica. En 1973, el matemático Clifford Cocks llegará a la misma conclusión que los creadores de RSA.
- Desgraciadamente este trabajo fue considerado como alto secreto por el gobierno británico por lo que no se hace público ni se patenta como invento, algo que sí hacen Diffie y Hellman en 1976 con su intercambio de claves y en 1978 otro tanto los creadores de RSA.

Cifrado Pohlig y Hellman con clave secreta

- Stephen Pohlig y Martin Hellman proponen en enero de 1978 un algoritmo de cifra de clave secreta y que basa su seguridad en el problema del logaritmo discreto. Hablamos de sólo un mes antes que el algoritmo RSA... algo que también llama la atención.

- Se elige un grupo multiplicativo Z_p^*, p es un primo grande.
- Cada usuario elige una clave e, que sea primo relativo con el grupo $\phi(p) = p-1$ y calcula $d = inv(e, \phi(p))$.
- La clave secreta serán los valores e y d.

Operaciones de cifra

CIFRADO	DESCIFRADO
$C = M^e \bmod p$	$M = C^d \bmod p$

El sistema carece de firma digital y no puede competir en velocidad con los algoritmos de clave secreta por lo que no tiene uso.

Ejemplo de cifrado Pohlig y Hellman

A cifra un mensaje M que envía a B

$p = 263 \Rightarrow \phi(p) = 262$; $e = 15 \Rightarrow d = inv(15,262) = 35$

Sea M = Adiós = 65 100 105 243 115

Como se usa el código ANSI, podremos cifrar en bloques de un carácter pues el módulo p es algo mayor que 256.

Operación Cifrado:

$C = M^e \bmod p = 65^{15} \bmod 263,\ 100^{15} \bmod 263,$

$105^{15} \bmod 263,\ 243^{15} \bmod 263,\ 115^{15} \bmod 263$

C = 245, 143, 179, 86, 101

Ejemplo de descifrado Pohlig y Hellman

B descifra el criptograma C enviado por A

p = 263; d = inv (15, 262) = 35

C = 245, 143, 179, 86, 101

Operación Descifrado:

$M = C^d \bmod p = 245^{35} \bmod 263,\ 143^{35} \bmod 263,$

$179^{35} \bmod 263,\ 86^{35} \bmod 263,\ 101^{35} \bmod 263$

M = 065, 100, 105, 243, 115

Convirtiéndolo al código ANSI: M = Adiós

Algoritmo de cifra asimétrica de ElGamal

Taher ElGamal propone en 1985 un algoritmo de cifra que hace uso del Problema del Logaritmo Discreto PLD.

Características

- Se elige un grupo multiplicativo Z_p^*, p es primo grande.
- Del grupo p se elige una raíz α, generador del grupo.
- Cada usuario elige un número aleatorio λ dentro de p.
  - Esta será la clave privada.
- Cada usuario calcula $\alpha^{\lambda} \bmod p$.
  - Junto con p es la clave pública.

Para descubrir la clave privada el atacante deberá enfrentarse al Problema del Logaritmo Discreto para un valor p alto.

Operación de cifra con ElGamal

Operación Cifrado: A cifra un mensaje M que envía a B

- El usuario B ha elegido su clave privada *b* dentro del cuerpo del número primo p que es público.
- El usuario B ha hecho pública su clave $\alpha^b \bmod p$.
- El emisor A genera un número aleatorio ν de sesión y calcula $\alpha^\nu \bmod p$.
- Con la clave pública de B (α^b) el emisor A calcula:
  - $(\alpha^b)^\nu \bmod p$ y $M*(\alpha^b)^\nu \bmod p$
- A envía a B el par: $C = [\alpha^\nu \bmod p, M * (\alpha^b)^\nu \bmod p]$

Operación de descifrado con ElGamal

Operación Descifrado: B descifra el criptograma C que envía A

- El usuario B recibe $C = [\alpha^v \bmod p, M * (\alpha^b)^v \bmod p]$.
- B toma el valor $\alpha^v \bmod p$ y calcula $(\alpha^v)^b \bmod p$.
- B descifra el criptograma C haciendo la siguiente división: $[M * (\alpha^b)^v \bmod p\ ] / [(\alpha^v)^b \bmod p]$ $\qquad (\alpha^b)^v = (\alpha^v)^b$
- El paso anterior es posible hacerlo porque existirá el inverso de $(\alpha^v)^b$ en el grupo p al ser p un primo. Luego:

$$[M * (\alpha^b)^v * \{inv\ (\alpha^v)^b, p\}] \bmod p = M$$

Ejemplo de cifrado con ElGamal

Adela (A) enviará a Benito (B) el mensaje M = 10 cifrado dentro del cuerpo p = 13 que usa Benito.

CIFRADO

Claves públicas de Benito: p = 13, $\alpha = 6$, $(\alpha^b) \bmod p = 2$

Adela A elige por ejemplo $v = 4$ y calcula:

$(\alpha^v) \bmod p = 6^4 \bmod 13 = 9$

$(\alpha^b)^v \bmod p = 2^4 \bmod 13 = 3$

$M*(\alpha^b)^v \bmod p = 10*3 \bmod 13 = 4$

Envía a B $(\alpha^v) \bmod p$, $M*(\alpha^b)^v \bmod p = [9, 4]$

Ejemplo de descifrado con ElGamal

DESCIFRADO

La clave privada de Benito es b = 5

Benito recibe: $[(\alpha^v) \bmod p, M*(\alpha^b)^v \bmod p] = [9, 4]$

Benito calcula:

$(\alpha^v)^b \bmod p = 9^5 \bmod 13 = 3$

$[M*(\alpha^b)^v] * inv[(\alpha^v)^b, p] = 4 * inv\ (3, 13) = 4 * 9$

$M = 4 * 9 \bmod 13 = 10$ (se recupera el mensaje)

Recuerde que α debe ser una raíz de p. Como ya hemos visto, si α no es una raíz, aunque sí puede hacerse la cifra, se facilitará el ataque al Problema del Logaritmo Discreto.

Consideraciones sobre el bloque

Si el mensaje M fuese mayor que el módulo de trabajo (n = p*q para RSA y p para ElGamal)

¿cómo se generarían los bloques del mensaje a cifrar?

El mensaje M se transforma en números y éstos se dividen en bloques de de g-1 dígitos, siendo g el número de dígitos del módulo de trabajo: el valor n para RSA y p para ElGamal.

Nota: en la práctica esto no ocurrirá ya que el cuerpo de cifra es como mínimo de 512 bits y el "mensaje" a cifrar tendrá sólo una centena de bits.

Ejemplo

Ejemplo de elección del bloque con RSA

Se representará el mensaje en su valor ANSI decimal.

$n = p*q = 89*127 = 11.303 \Rightarrow$ bloques de cuatro dígitos

$\phi(n) = 11.088;\ e = 25;\ d = inv\ (25, 11.088) = 10.201$

M = Olé = 079 108 233 $\Rightarrow$ M = 0791 0823 3

Se recupera el mensaje agrupando en bloques de 4 dígitos excepto el último

CIFRADO

$C_1 = 791^{25} \bmod 11.303 = 7.853$

$C_2 = 823^{25} \bmod 11.303 = 2.460$

$C_3 = 3^{25} \bmod 11.303 = 6.970$

DESCIFRADO

$M_1 = 7.853^{10201} \bmod 11.303 = 0791$

$M_2 = 2.460^{10201} \bmod 11.303 = 0823$

$M_3 = 6.970^{10201} \bmod 11.303 = 3$

Fortaleza de la cifra exponencial

El problema de la Factorización de Números Grandes PFNG tiene una complejidad similar al del Logaritmo Discreto PLD: ambos suponen un tiempo de ejecución no polinomial.

Número de pasos que debe realizar el algoritmo PFNG:

$$e^{\sqrt{\{\ln(n)*\ln[\ln(n)]\}}}$$

Sistema que consuma 1 μseg por paso:

n = 60 dígitos	$\Rightarrow$ $2{,}7*10^{11}$ pasos	3 días
n = 100 dígitos	$\Rightarrow$ $2{,}3*10^{15}$ pasos	74 años
n = 200 dígitos	$\Rightarrow$ $1{,}2*10^{23}$ pasos	$3{,}8*10^{9}$ años

El PLD es matemáticamente similar:
número de pasos $\approx e^{\sqrt{\{\ln(p)*\ln[\ln(p)]\}}}$

Fin del Tema 12

Cuestiones y ejercicios (1 de 4)

1. A partir de la ecuación $e*d = k\phi(n) + 1$, compruebe que las claves RSA e = 133 y d = 38.797 son inversas en el cuerpo n = 40.501.
2. En el ejemplo de intercambio de clave DH del libro con p = 1.999, ¿podrían haber elegido Adela y Benito α = 34, 35, 36 ó 37?
3. Carmela (C) intercepta la clave de sesión DH que se intercambian Adela (A) y Benito (B) dentro del cuerpo p = 127. Si se usa como generador α = 19 y a = 3, b = 12 y c = 7, desarrolle el algoritmo que permite a C interceptar la comunicación y engañar a A y B.
4. Los usuarios A, B, C y D desean intercambiar una clave usando el método DH. Proponga un protocolo genérico que solucione este problema y presente un ejemplo en el cuerpo p = 23 y elija α.
5. Diseñe un sistema RSA en el que p = 53, q = 113. Elija como clave pública el mínimo valor posible de 2 dígitos.

Cuestiones y ejercicios (2 de 4)

6. Para los datos de diseño del ejercicio 5, cifre el mensaje M = 121 y luego descífrelo. Use el algoritmo de exponenciación rápida.
7. Vuelva a descifrar el criptograma usando ahora el Teorema del Resto Chino. Aunque en este caso haya hecho más cálculos ¿por qué es interesante usar aquí el Teorema del Resto Chino?
8. Si p = 353 y q = 1.103, cifre el mensaje ASCII de cuatro caracteres M = HOLA en bloques de tamaño eficiente (mínimo) para e = 17.
9. Para los datos del ejercicio anterior, encuentre la clave privada d con el algoritmo extendido de Euclides.
10. ¿Por qué se usa una clave pública e de un valor relativamente bajo y, por contrapartida, la clave privada d es alta? ¿Qué utilidad tiene?
11. ¿Cómo atacaría un sistema RSA mal diseñado con primos cercanos y cuyo módulo es n = 205.027? Encuentre los valores de p y q.

Cuestiones y ejercicios (3 de 4)

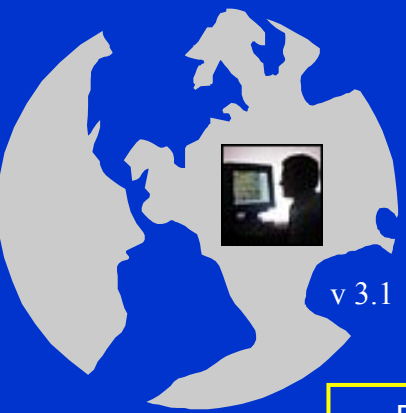
12. En el sistema RSA con p = 11 y q = 19, se elige como clave pública e = 31. ¿Cuántas y cuáles son las claves privadas parejas?
13. Para los mismos datos del ejercicio anterior, ¿cuántos y cuáles son los mensajes no cifrables? ¿Y si ahora e = 33?
14. ¿Cómo puede minimizarse el número de claves privadas parejas y el de mensajes no cifrables? Dé un ejemplo con primos de dos dígitos.
15. Atacamos un sistema RSA con un cifrado cíclico. ¿Qué es lo que vulneramos, el secreto de la clave privada o secreto del mensaje?
16. Se ataca por cifrado cíclico el sistema RSA en el que p = 23, q = 41 y e = 17. ¿Cuántos cifrados hay que hacer hasta encontrar M?
17. Cifre M = "La Puerta de Alcalá" (de 19 caracteres ANSI) con un sistema de Pohlig y Hellman. Use el primer primo que le permita cifrar bloques de 2 bytes. Descifre ahora el criptograma C.

Cuestiones y ejercicios (4 de 4)

18. Vamos a cifrar con ElGamal el mensaje en ASCII decimal de la letra A. ¿Cuáles son los valores mínimos del cuerpo de cifra, del generador α, de la clave pública y del valor local v?
19. Descifre el criptograma obtenido en el ejercicio anterior.
20. Se va a cifrar con RSA el mensaje M = Hola. Si p = 53 y q = 97, ¿de cuántos dígitos será el bloque óptimo de cifra?
21. En un sistema real, ¿tiene sentido hablar de bloques con un número de dígitos óptimos? ¿Por qué? Justifique su respuesta.
22. ¿Cuántos pasos debe realizar una factorización de un número n de 120 dígitos? Si tenemos un sistema que consume 1 µseg por paso, ¿cuántos años tardaríamos más o menos en factorizar ese número?
23. El valor encontrado en el ejercicio anterior, ¿es siempre fijo o es sólo una aproximación? Justifique su respuesta.

Tema 13
Funciones Hash en Criptografía

Seguridad Informática y Criptografía



Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 30 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Uso de las funciones hash en criptografía

Una de las aplicaciones más interesantes de la criptografía es la posibilidad real de incluir en un mensaje una firma digital. Todo esto comienza en el año 1976 cuando Diffie y Hellman presentan un modelo de cifrado asimétrico con clave pública. Con los sistemas de clave simétrica esto era inviable.

No obstante, dado que los sistemas de clave pública son muy lentos, en vez de firmar digitalmente el mensaje completo, en un sistema criptográfico se incluirá como firma digital una operación con la clave privada sobre un resumen o hash de dicho mensaje representado por sólo una centena de bits.

Funciones hash

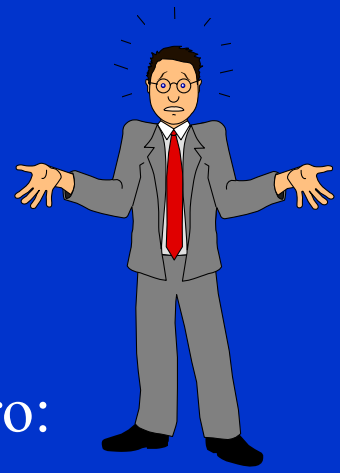
Mensaje = M $\Rightarrow$ Función Resumen = H(M)

Firma (rúbrica): $r = E_{dE}\{H(M)\}$

dE es la clave privada del emisor que firmará H(M)

¿Cómo se comprueba la identidad en destino?

Se descifra la rúbrica r con la clave pública del emisor eE. Al mensaje en claro recibido M' (se descifra si viene cifrado) se le aplica la misma función hash que en emisión. Si los valores son iguales, la firma es auténtica y el mensaje íntegro:



Calcula: $E_{eE}(r) = H(M)$

Compara: ¿$H(M') = H(M)$?

¿Qué seguridad nos da un resumen de k bits?

Seguridad asociada a una función hash

Suponga que hemos creado una función hash de forma que el resumen es sólo de 4 bits, independientemente del tamaño del mensaje de entrada.

La pregunta es: ¿cuál es la probabilidad de que dos mensajes distintos tengan igual función hash? Si esta probabilidad fuese muy baja (en este caso 1/16: hash desde 0000 hasta 1111) podría darse el siguiente caso: alguien modifica nuestro mensaje firmado, y envía ese mensaje falso con la firma del primero ya que en ambos casos son los mismos 4 bits...

Mensaje 1: "Rechazamos el contrato por no interesarnos nada" hash: 1101
Mensaje 2: "Firma todo lo que te pongan porque nos interesa" hash: 1101

Observe que ambos mensajes tienen 47 caracteres. Así, podríamos crear una gran cantidad de mensajes diferentes que digan casi lo mismo, incluso con igual número de caracteres... ¡hasta que los dos hash coincidan!

Por este motivo las funciones hash para que sean interesantes en criptografía, deben cumplir un conjunto de propiedades. 

Propiedades de las funciones hash

H(M) será segura si tiene las siguientes características:

1. Unidireccionalidad. Conocido un resumen H(M), debe ser computacionalmente imposible encontrar M a partir de dicho resumen.
2. Compresión. A partir de un mensaje de cualquier longitud, el resumen H(M) debe tener una longitud fija. Lo normal es que la longitud de H(M) sea menor.
3. Facilidad de cálculo. Debe ser fácil calcular H(M) a partir de un mensaje M.
4. Difusión. El resumen H(M) debe ser una función compleja de todos los bits del mensaje M. Si se modifica un bit del mensaje M, el hash H(M) debería cambiar aproximadamente la mitad de sus bits.

Colisiones: resistencia débil y fuerte

5. Colisión simple. Conocido M, será computacionalmente imposible encontrar otro M' tal que H(M) = H(M'). Se conoce como *resistencia débil a las colisiones*.
6. Colisión fuerte. Será computacionalmente difícil encontrar un par (M, M') de forma que H(M) = H(M'). Se conoce como *resistencia fuerte a las colisiones*.

Ataque por la paradoja del cumpleaños

Para tener *confianza* en encontrar dos mensajes con el mismo resumen H(M) -probabilidad ≥ 50%- no habrá que buscar en 2^n (p.e. 2^{128}), bastará una búsqueda en el espacio $2^{n/2}$ (2^{64}). ¡La complejidad algorítmica se reduce de forma drástica!

La paradoja del cumpleaños

Problema: Cuál será la confianza (probabilidad $> 50\%$) de que en un aula con 365 personas -no se tiene en cuenta el día 29/02 de los años bisiestos- dos de ellas al azar cumplan años en la misma fecha.

Solución: Se escribe en la pizarra los 365 días del año y entran al aula de uno en uno, borrando el día de su cumpleaños de la pizarra. Para alcanzar esa confianza, basta que entren 23 personas al aula, un valor muy bajo, en principio inimaginable y de allí el nombre de paradoja.

Explicación: El primero en entrar tendrá una probabilidad de que su número no esté borrado igual a $n/n = 1$, el segundo de $(n-1)/n$, etc. La probabilidad de no coincidencia será $p_{NC} = n!/(n-k)n^k$. Para $k = 23$ se tiene $p_{NC} = 0{,}493$ y así la probabilidad de coincidencia es $p_C = 0{,}507$.

☞ En nuestro caso, esto es equivalente a sacar dos mensajes de dos conjunto disjuntos y comparar sus hash; si los hash no son iguales sacamos otros dos mensajes, ...etc., hasta que haya una colisión.

Algoritmos de resumen en criptografía

- MD5: Ron Rivest 1992. Mejoras al MD4 y MD2 (1990), es más lento pero con mayor nivel de seguridad. Resumen de 128 bits.
- SHA-1: Del NIST, National Institute of Standards and Technology, 1994. Similar a MD5 pero con resumen de 160 bits. Existen otras nuevas propuestas conocidas como SHA-256 y SHA-512.
- RIPEMD: Comunidad Europea, RACE, 1992. Resumen de 160 bits.
- N-Hash: Nippon Telephone and Telegraph, 1990. Resumen: 128 bits.
- Snefru: Ralph Merkle, 1990. Resúmenes entre 128 y 256 bits. Ha sido criptoanalizado y es lento.
- Tiger: Ross Anderson, Eli Biham, 1996. Resúmenes de hasta 192 bits. Optimizado para máquinas de 64 bits (Alpha).
- Panama: John Daemen, Craig Clapp, 1998. Resúmenes de 256 bits de longitud. Trabaja en modo función hash o como cifrador de flujo.
- Haval: Yuliang Zheng, Josef Pieprzyk y Jennifer Seberry, 1992. Admite 15 configuraciones diferentes. Hasta 256 bits.

Message Digest 5, MD5

Algoritmo básico MD5

a) Un mensaje M se convierte en un bloque múltiplo de 512 bits, añadiendo bits si es necesario al final del mismo.

b) Con los 128 bits de cuatro vectores iniciales ABCD de 32 bits cada uno y el primer bloque del mensaje de 512 bits, se realizan diversas operaciones lógicas entre ambos bloques.

c) La salida de esta operación (128 bits) se convierte en el nuevo conjunto de 4 vectores ABCD y se realiza la misma función con el segundo bloque de 512 bits del mensaje y así hasta el último bloque del mensaje.

d) Al terminar, el algoritmo entrega un resumen que corresponde a los últimos 128 bits de estas operaciones.

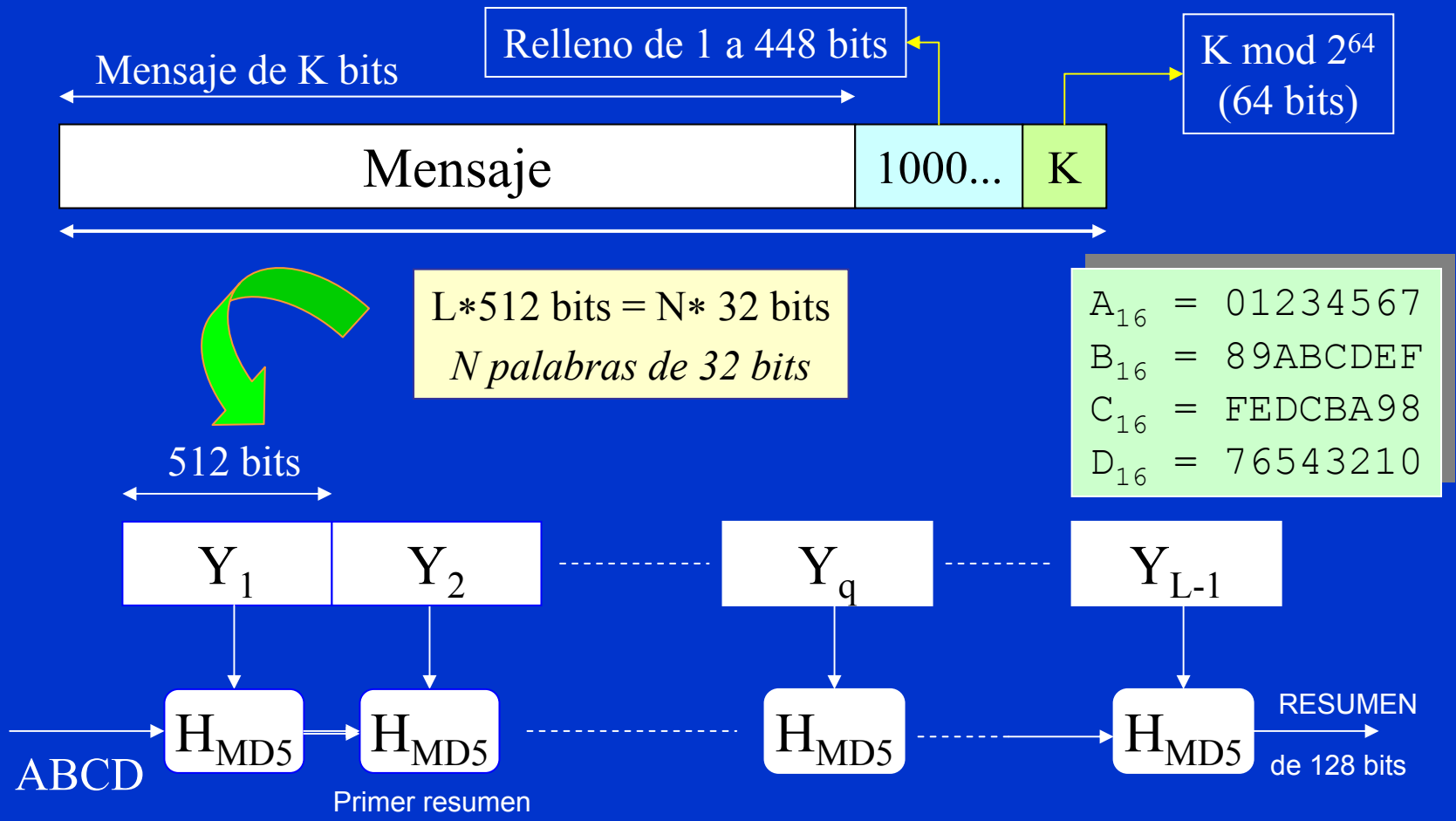
Etapas de MD5

Bloques funcionales de MD5

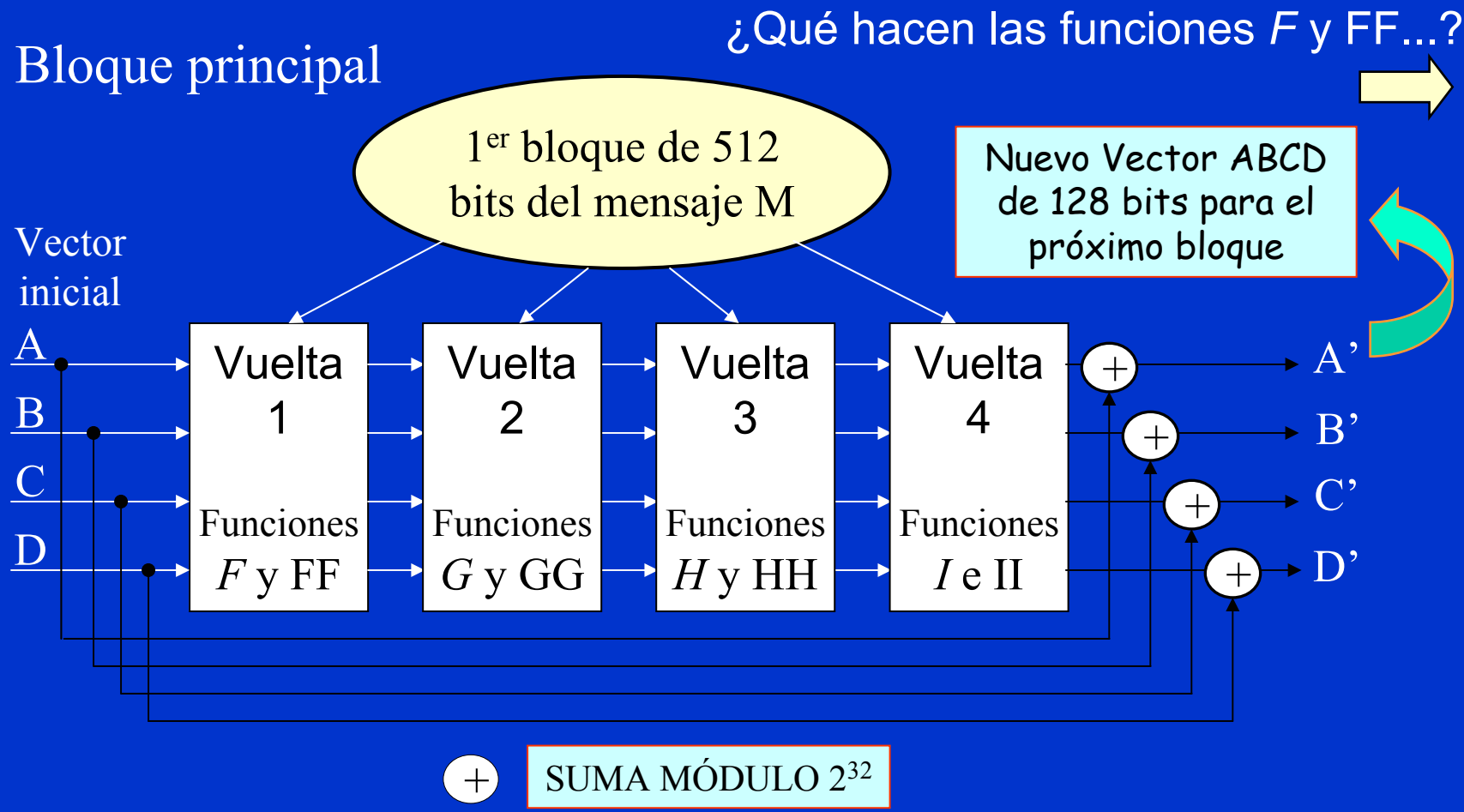
Esquema

a) Añadir bits para congruencia módulo 512, reservando los últimos 64 bits para un indicador de longitud.
b) Añadir indicación de la longitud del mensaje en los 64 bits reservados.
c) Inicializar el vector ABCD de claves.
d) Procesar bloques de 512 bits, entregando una salida de 128 bits que formarán nuevamente el vector ABCD.
e) Obtener el resumen de los últimos 128 bits.

Esquema de la función MD5



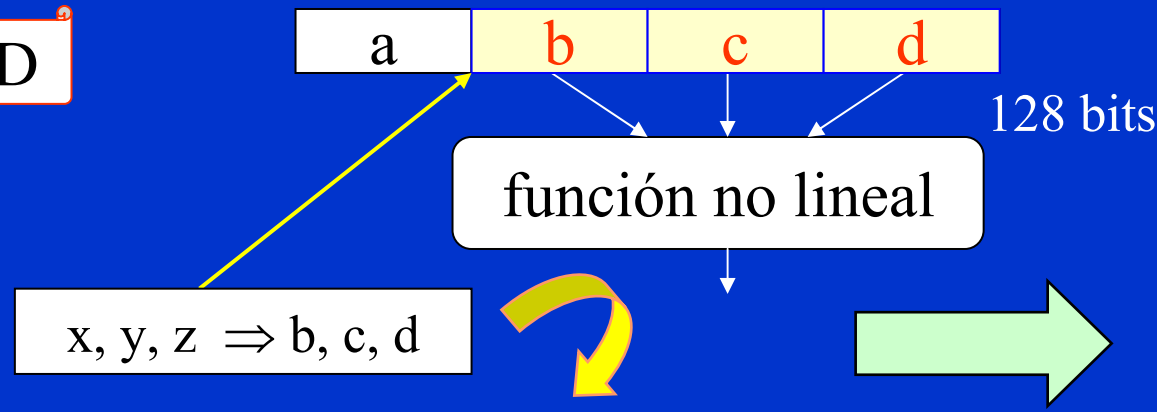
Bloque principal de MD5



Esquema funciones F, G, H, I en MD5

Vector inicial ABCD

A_{16} = 01234567
B_{16} = 89ABCDEF
C_{16} = FEDCBA98
D_{16} = 76543210



F (x, y, z)
(x AND y) OR (NOT x AND z)
G (x, y, z)
(x AND z) OR (y AND NOT z)
H (x, y, z)
x XOR y XOR z
I (x, y, z)
y XOR (x OR NOT z)

F (b, c, d)
(b AND c) OR (NOT b AND d)
G (b, c, d)
(b AND d) OR (c AND NOT d)
H (b, c, d)
b XOR c XOR d
I (b, c, d)
c XOR (b OR NOT d)

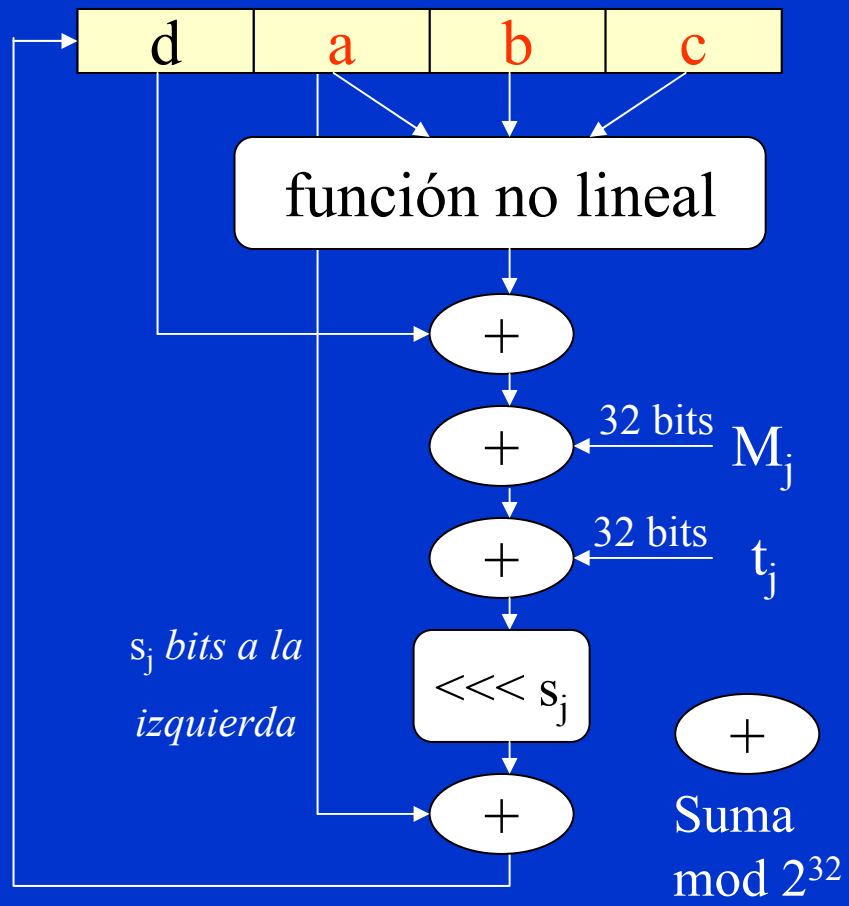
Algoritmo de las funciones en MD5

Desplazamiento del registro

Situación luego del desplazamiento

Se repite el proceso para M_{j+1} hasta 16 bloques del texto. En las vueltas 2, 3 y 4 se repite el proceso ahora con funciones G, H e I.

El algoritmo realiza 4∗16 = 64 vueltas para cada uno de los bloques de 512 bits



Funciones no lineales en MD5

Funciones no lineales en cada vuelta

Vector de 128 bits

a	b	c	d

1ª Vuelta:

$FF(a,b,c,d,M_j,t_j,s) \Rightarrow a = b + ((a + F(b,c,d) + M_j + t_j) <<< s)$

2ª Vuelta:

$GG(a,b,c,d,M_j,t_j,s) \Rightarrow a = b + ((a + G(b,c,d) + M_j + t_j) <<< s)$

3ª Vuelta:

$HH(a,b,c,d,M_j,t_j,s) \Rightarrow a = b + ((a + H(b,c,d) + M_j + t_j) <<< s)$

4ª Vuelta:

$II(a,b,c,d,M_j,t_j,s) \Rightarrow a = b + ((a + I(b,c,d) + M_j + t_j) <<< s)$

Algoritmo y desplazamiento en MD5

Vector de 128 bits

a	b	c	d

Sea f la función F, G, H o I según la vuelta.

El algoritmo será:

Para j = 0 hasta 15 hacer:

TEMP = $[(a + f(b,c,d) + M_j + t_j) <<< s_j]$

a = d

d = c

c = b

b = a

a = TEMP

Operaciones en 1ª y 2ª vueltas en MD5

Primera vuelta

FF (a, b, c, d, M_j, t_j, s)

FF(a, b, c, d, M_0, D76AA478, 7)
FF(d, a, b, c, M_1, E8C7B756, 12)
FF(c, d, a, b, M_2, 242070DB, 17)
FF(b, c, d, a, M_3, C1BDCEEE, 22)
FF(a, b, c, d, M_4, F57C0FAF, 7)
FF(d, a, b, c, M_5, 4787C62A, 12)
FF(c, d, a, b, M_6, A8304613, 17)
FF(b, c, d, a, M_7, FD469501, 22)
FF(a, b, c, d, M_8, 698098D8, 7)
FF(d, a, b, c, M_9, 8B44F7AF, 12)
FF(c, d, a, b, M_{10}, FFFF5BB1, 17)
FF(b, c, d, a, M_{11}, 895CD7BE, 22)
FF(a, b, c, d, M_{12}, 6B901122, 7)
FF(d, a, b, c, M_{13}, FD987193, 12)
FF(c, d, a, b, M_{14}, A679438E, 17)
FF(b, c, d, a, M_{15}, 49B40821, 22)

Segunda vuelta

M_j, t_j, s)

$_1$, F61E2562, 5)
$_6$, C040B340, 9)
$_{11}$, 265E5A51, 14)
$_0$, E9B6C7AA, 20)
$_5$, D62F105D, 5)
$_{10}$, 02441453, 9)
$_{15}$, D8A1E681, 14)
$_4$, E7D3FBC8, 20)
$_9$, 21E1CDE6, 5)
$_{14}$, C33707D6, 9)
$_3$, F4D50D87, 14)
$_8$, 455A14ED, 20)
$_{13}$, A9E3E905, 5)
$_2$, FCEFA3F8, 9)
$_7$, 676F02D9, 14)
$_{12}$, 8D2A4C8A, 20)

Operaciones en 3ª y 4ª vueltas en MD5

Tercera vuelta

HH (a, b, c, d, M_j, t_j, s)

HH(a, b, c, d, M_5, FFFA3942, 4)
HH(d, a, b, c, M_8, 8771F681, 11)
HH(c, d, a, b, M_{11}, 6D9D6122, 16)
HH(b, c, d, a, M_{14}, FDE5380C, 23)
HH(a, b, c, d, M_1, A4BEEA44, 4)
HH(d, a, b, c, M_4, 4BDECFA9, 11)
HH(c, d, a, b, M_7, F6BB4B60, 16)
HH(b, c, d, a, M_{10}, BEBFBC70, 23)
HH(a, b, c, d, M_{13}, 289B7EC6, 4)
HH(d, a, b, c, M_0, EAA127FA, 11)
HH(c, d, a, b, M_3, D4EF3085, 16)
HH(b, c, d, a, M_6, 04881D05, 23)
HH(a, b, c, d, M_9, D9D4D039, 4)
HH(d, a, b, c, M_{12}, E6DB99E5, 11)
HH(c, d, a, b, M_{15}, 1FA27CF8, 16)
HH(b, c, d, a, M_2, C4AC5665, 23)

Cuarta vuelta

M_j, t_j, s)

$_0$, F4292244, 6)
$_7$, 411AFF97, 10)
$_{14}$, AB9423A7, 15)
$_5$, FC93A039, 21)
$_{12}$, 655B59C3, 6)
$_3$, 8F0CCC92, 10)
$_{10}$, FFEFF47D, 15)
$_1$, 85845DD1, 21)
$_8$, 6FA87E4F, 6)
$_{15}$, FE2CE6E0, 10)
$_6$, A3014314, 15)
$_{13}$, 4E0811A1, 21)
$_4$, F7537E82, 6)
$_{11}$, BD3AF235, 10)
$_2$, 2AD7D2BB, 15)
$_9$, EB86D391, 21)

Función de resumen SHA-1

Un resumen de 128 bits tiene una complejidad algorítmica de sólo 2^{64}, un valor en la actualidad muy comprometido...

La función SHA-1, Secure Hash Algorithm, entregará un resumen de 160 bits $\Rightarrow$ una complejidad algorítmica de 2^{80}.

SHA-1

Vector Inicial : A = 67452301 B = EFCDAB89
C = 98BADCFE D = 10325476 E = C3D2E1F0

Esta forma de tomar los bits se verá más adelante

Algoritmo:

Es muy similar a MD5. El vector inicial tiene una palabra más de 32 bits (E) por lo que el resumen será de 160 bits. A cada bloque de 512 bits del mensaje se le aplicarán 80 vueltas.

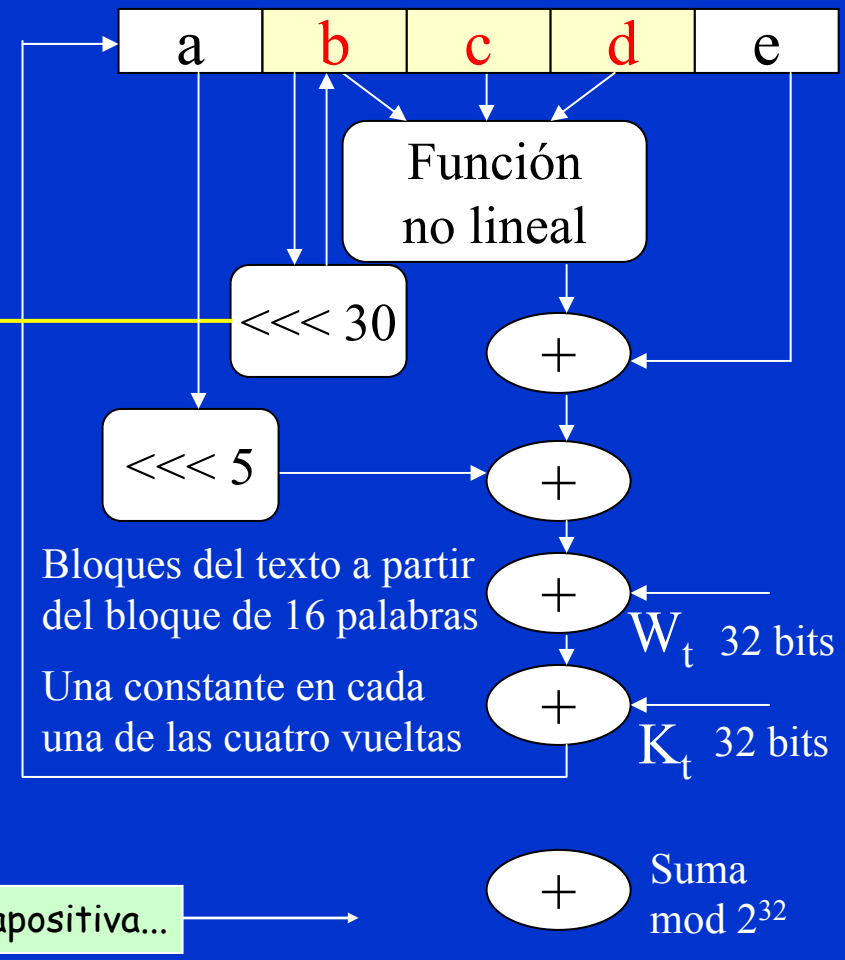
Esquema del resumen SHA-1

Vector inicial ABCDE

Registro de 160 bits

A_{16} = 67452301
B_{16} = EFCDAB89
C_{16} = 98BADCFE
D_{16} = 10325476
E_{16} = C3D2E1F0

Después de esta última operación, se produce el desplazamiento del registro hacia la derecha



Véase la próxima diapositiva...

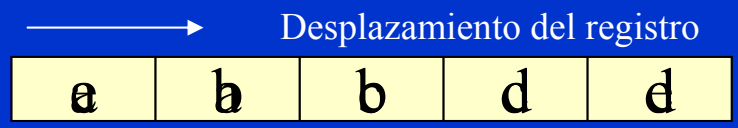
Vueltas funciones F, G, H, I en SHA-1

F (b, c, d) → vueltas t = 0 a 19
(b AND c) OR ((NOT b) AND d)

G (b, c, d) → vueltas t = 20 a 39
b XOR c XOR d

H (b, c, d) → vueltas t = 40 a 59
(b AND c) OR (b AND d) OR (c AND d)

I (b, c, d) → vueltas t = 60 a 79
b XOR c XOR d



Se repite el proceso con la función F para las restantes 15 palabras de 32 bits del bloque actual hasta llegar a 20. En vueltas 2, 3 y 4 se repite el proceso con funciones G, H e I.

Tenemos 4*20 = 80 pasos por cada bloque de 512 bits.
Pero ... ¿cómo es posible repetir 80 veces un bloque que sólo cuenta con 16 bloques de texto de 32 bits cada uno?

Las 80 vueltas en SHA-1

Vector de 160 bits

a	b	c	d	e

Cada bloque de 16 palabras del mensaje (M_0 ... M_{15}) se expandirá en 80 palabras (W_0 ... W_{79}) según el algoritmo:

$W_t = M_t$ (para t = 0, ..., 15)

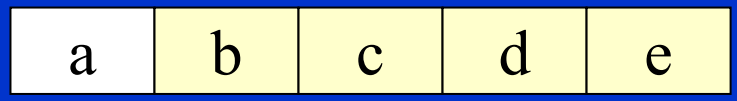
$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <<<1$ (para t = 16, ..., 79)

y además:

K_t = 5A827999	para t = 0, ..., 19
K_t = 6ED9EBA1	para t = 20, ..., 39
K_t = 8F1BBCDC	para t = 40, ..., 59
K_t = CA62C1D6	para t = 60, ..., 79

Algoritmo y desplazamiento en SHA-1

Vector de 160 bits



El algoritmo para cada bloque de 512 bits será:

Para t = 0 hasta 79 hacer:

TEMP = (a <<<5) + f_t(b,c,d) + e + W_t + K_t

a = e

e = d

d = c

c = b <<<30

b = a

a = TEMP

Comparativa entre MD5 y SHA-1

- SHA-1 genera una salida de 160 bits de longitud mientras que MD5 genera sólo 128 bits.
  - La dificultad de generar un mensaje que tenga un resumen dado es del orden de 2^{128} operaciones para MD5 y 2^{160} para SHA-1.
  - La dificultad de generar dos mensajes aleatorios distintos y que tengan el mismo resumen (ataques basados en paradoja del cumpleaños) es del orden de 2^{64} operaciones para MD5 y 2^{80} para SHA-1.
- Esta diferencia de 16 bits a favor de SHA-1 lo convierte en más seguro y resistente a ataques por fuerza bruta que el algoritmo MD5. Aunque es más lento que MD5, hoy es el estándar como función hash.

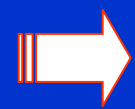
Pasos y tasas de cifra en MD5 y SHA-1

- Ambos algoritmos procesan bloques de 512 bits y emplean 4 funciones primitivas para generar el resumen del mensaje, pero...

- SHA-1 realiza un mayor número de pasos que MD5 (80 frente a los 64 que realiza MD5).
- SHA-1 debe procesar 160 bits de buffer en comparación con los 128 bits de MD5.
- Por estos motivos la ejecución del algoritmo SHA-1 es más lenta que la de MD5 usando un mismo hardware. Por ejemplo en un Pentium a 266 MHz un programa realizado en C entrega para SHA-1 una tasa del orden de 20 Mbits/seg y para MD5 esta tasa llega a los 60 Mbits/seg.

Más diferencias entre MD5 y SHA-1

- La longitud máxima del mensaje para SHA-1 debe ser menor de 2^{64} bits, mientras que MD5 no tiene limitaciones de longitud.
- MD5 emplea 64 constantes (una por cada paso), mientras que SHA-1 sólo emplea 4 (una para cada 20 pasos).
- MD5 se basa en la arquitectura little-endian, mientras que SHA-1 se basa en la arquitectura big-endian. Por ello el vector ABCD inicial en MD5 y SHA-1 son iguales:
  - A = 01234567 (MD5) ⇒ 67452301 (SHA-1)
  - B = 89ABCDEF (MD5) ⇒ EFCDAB89 (SHA-1)
  - C = FEDCBA98 (MD5) ⇒ 98BADCFE (SHA-1)
  - D = 76543210 (MD5) ⇒ 10325476 (SHA-1)



Arquitecturas little-endian v/s big-endian

Arquitectura little-endian:

- Esta es la arquitectura empleada en procesadores Intel de la familia 80xxx y Pentium.
- Para almacenar una palabra en memoria, el byte menos significativo de los que forman dicha palabra se guarda en la posición más baja de la memoria.

Ejemplo

Arquitectura big-endian:

- Empleada por otras arquitecturas como SUN.
- Para almacenar una palabra en memoria, el byte más significativo de los que forman dicha palabra se guarda en la posición más baja de memoria.

Ejemplo little-endian v/s big-endian

Supongamos que queremos almacenar en memoria la siguiente palabra de 32 bits (4 bytes) representada en hexadecimal:

76543210 $\Rightarrow$

76	54	32	10

Si consideramos que las posiciones de memoria más bajas se encuentran a la izquierda y las más altas a la derecha:

En formato *little-endian* se representa:

01	23	45	67

En formato *big-endian* se representa:

67	45	23	01

Funciones hash para la autenticación

- Las funciones hash vistas (MD5, SHA-1, etc.) pueden usarse además para autenticar a dos usuarios.
- Como carecen de una clave privada no pueden usarse de forma directa para estos propósitos. No obstante, existen algoritmos que permiten incluirles esta función.
- Entre ellos está HMAC, una función que usando los hash vistos y una clave secreta, autentica a dos usuarios mediante sistemas de clave secreta. Las funciones MAC, Message Authentication Code, y HMAC se tratarán en el próximo capítulo dedicado a la autenticación y firma digital.
- HMAC se usa en plataformas IP seguras como por ejemplo en Secure Socket Layer, SSL.

Fin del Tema 13

Cuestiones y ejercicios

1. ¿Qué propiedades debe tener una función hash para que su uso en criptografía sea de interés? ¿Vale cualquier función reductora?
2. ¿Por qué prospera un ataque basado en la paradoja del cumpleaños con un tiempo significativamente menor que un ataque elemental?
3. Se va a aplicar la función MD5 a un mensaje de longitud 250 bytes. ¿Cómo se rellena y cómo queda el último bloque?
4. ¿Por qué razón decimos que la función SHA-1 es actualmente un estándar más seguro que la función MD5?
5. ¿Cómo puede la función SHA-1 hacer 80 vueltas con bloques de 32 bits partiendo de un bloque de texto o mensaje de sólo 512 bits?
6. ¿Qué función hash es más rápida, MD5 o SHA-1? ¿Por qué?
7. Si en un mensaje cambiamos un bit, ¿en cuánto debería cambiar su hash aproximadamente con respecto a su resumen anterior?

Tema 14
Autenticación y Firma Digital

Seguridad Informática y Criptografía

Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 50 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Confidencialidad v/s integridad

- Confidencialidad
  - Para lograrla se cifra el mensaje M obteniendo un criptograma C.
- Integridad
  - Para lograrla se firma un hash del mensaje H(M), añadiendo una marca al mensaje o criptograma.

Si bien en ciertos escenarios es muy importante mantener el secreto de la información, si ésta lo requiere, en muchos casos tiene quizás más trascendencia el poder certificar la autenticidad entre cliente y servidor como ocurre en Internet.

Algunos problemas de integridad

a) Autenticidad del emisor

¿Cómo comprueba Benito (B) que el mensaje recibido del emisor que dice ser Adela (A) es efectivamente de esa persona?

b) Integridad del mensaje

¿Cómo comprueba Benito (B) que el mensaje recibido del emisor Adela (A) es el auténtico y no un mensaje falso?

c) Actualidad del mensaje

¿Cómo comprueba Benito (B) que el mensaje recibido del emisor Adela (A) es actual, no un mensaje de fecha anterior reenviado?

Más problemas de integridad

d) No repudio del emisor

¿Cómo comprueba Benito (B) que el mensaje enviado por el emisor Adela (A) -y que niega haberlo enviado- efectivamente ha llegado?

e) No repudio del receptor

¿Cómo comprueba Benito (B) que el mensaje enviado al receptor Adela (A) -y que niega haberlo recibido- efectivamente se envió?

d) Usurpación de identidad del emisor/receptor

¿Cómo comprueba Benito (B) que Adela (A), Carmela (C) u otros usuarios están enviando mensajes firmados como Benito (B)?

Primer escenario de integridad

1er escenario de desconfianza

1ª Solución. Uso de un Juez. El Juez tendrá una clave K_A con la que se comunica con A y una clave K_B con la que se comunica con B.

Usa criptografía simétrica

A envía un mensaje M a B:

A cifra M con la clave $K_A \Rightarrow E_{KA}(M)$ y lo envía al Juez. Este comprueba la integridad de A, lo descifra y envía a B, cifrado con K_B, el mensaje M, la identidad de A y la firma $E_{KA}(M)$: $E_{KB}\{M, A, E_{KA}(M)\}$. Ambos confían en el Juez y ante cualquier duda éste puede desvelar la identidad de A descifrando $E_{KA}(M)$.

Segundo escenario de integridad

2º escenario de desconfianza

No está claro que pueda convertirse en un estándar. Permanece la figura de la Autoridad de Certificación

... es decir

2ª Solución. Prescindir de la figura del Juez y aceptar la autenticidad e integridad por convencimiento propio y la confianza en los algoritmos.

Ambos confiarán en un protocolo seguro y se autenticarán a través de una Autoridad de Certificación AC.

Funciones de autenticación

- Autenticación mediante el cifrado de mensajes con criptografía simétrica
  - La cifra de datos puede servir como autenticación.
- Autenticación con MAC Message Code Authentication o checksum
  - Una función pública y una clave secreta producen un valor de longitud fija válida como autenticador.
- Autenticación mediante funciones hash
  - Una función pública reduce el mensaje a una longitud de valor hash que sirve como autenticador.
- Autenticación mediante firma digital del mensaje con criptografía asimétrica
  - Una función pública y un par de claves, pública y privada inversas en un cuerpo, permiten la autenticación completa.

Autenticación con sistemas simétricos

Si la clave de un sistema simétrico es segura, es decir no está en entredicho, podemos afirmar que, además de la confidencialidad que nos entrega dicha cifra, se obtienen también simultáneamente la integridad del emisor y autenticidad del mensaje, en tanto que sólo el usuario emisor (en quien se confía por el modelo de cifra) puede generar ese mensaje.

Con los sistemas de cifra simétricos no podremos realizar una autenticación completa (emisor y mensaje). Ahora bien, podremos hacer algo similar con algunos procedimientos más o menos complejos que usen sistemas de cifra simétrica como es el caso de Kerberos.

Los problemas de la autenticación simétrica

No obstante, subyacen los problemas característicos de un criptosistema: ¿cómo asegurar que la clave simétrica entre emisor y receptor es segura? o lo que es lo mismo, ¿cómo intercambiar claves de forma segura?

El intercambio de claves de forma segura ya hemos visto que se logra eficientemente sólo a través de sistemas asimétricos. Las herramientas más usuales para autenticación serán los códigos de autenticación de mensajes MACs y el sistema Kerberos con cifras simétricas. Kerberos también permite el intercambio seguro de una clave de sesión aunque es más complejo y largo que el algoritmo de Diffie Hellman.

Autenticación con MAC o Checksum

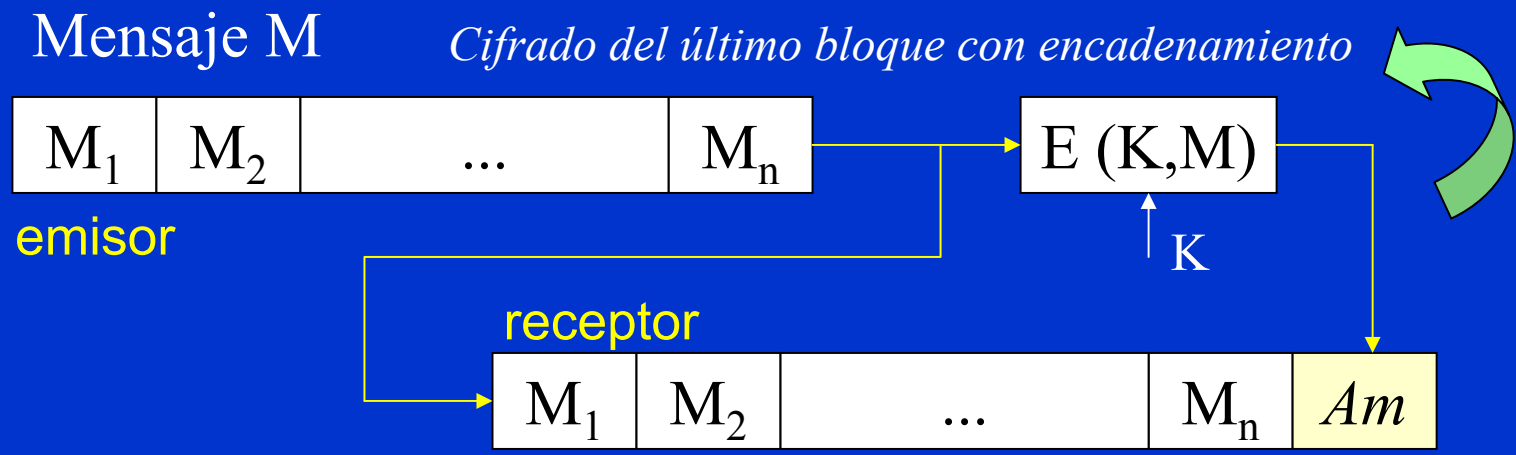
- Los dos extremos de la comunicación A y B comparten una única clave secreta que no está en entredicho.
- El MAC o checksum será una función que se aplica al mensaje M junto a una clave secreta K $\Rightarrow C_K(M)$.
- A envía el mensaje en claro y el Messsage Authentication Code (MAC) o Checksum $C_K(M)$ a B.
- Integridad: el receptor B puede estar seguro de que nadie ha modificado el mensaje durante la transmisión pues el valor $C_K(M)$ en destino coincide con el enviado por A.
- Autenticación: como solamente el emisor A y el receptor B comparten la clave secreta K, se asegura la autenticación de ambos usuarios $\Rightarrow$ la clave K debe ser muy segura.

Message Authentication Code con DES

- Se inserta un código al final del mensaje M transmitido en claro, consistente en la cifra con clave secreta de los últimos bytes del texto, por ejemplo 64 bits de DES.
- En destino, con el mismo algoritmo y clave secreta, se realiza la cifra y se comparan estos últimos bloques.
- Como la cifra es por encadenamiento o realimentación, esos bytes cifrados dependen de todo el mensaje por lo que cualquier modificación será detectada al no coincidir los resultados del cifrado de M en emisor y receptor.

Esquema

Esquema de autenticación MAC con DES



$E(K,M) = Am \in M_R$

M_R: espacio de marcas de autenticación

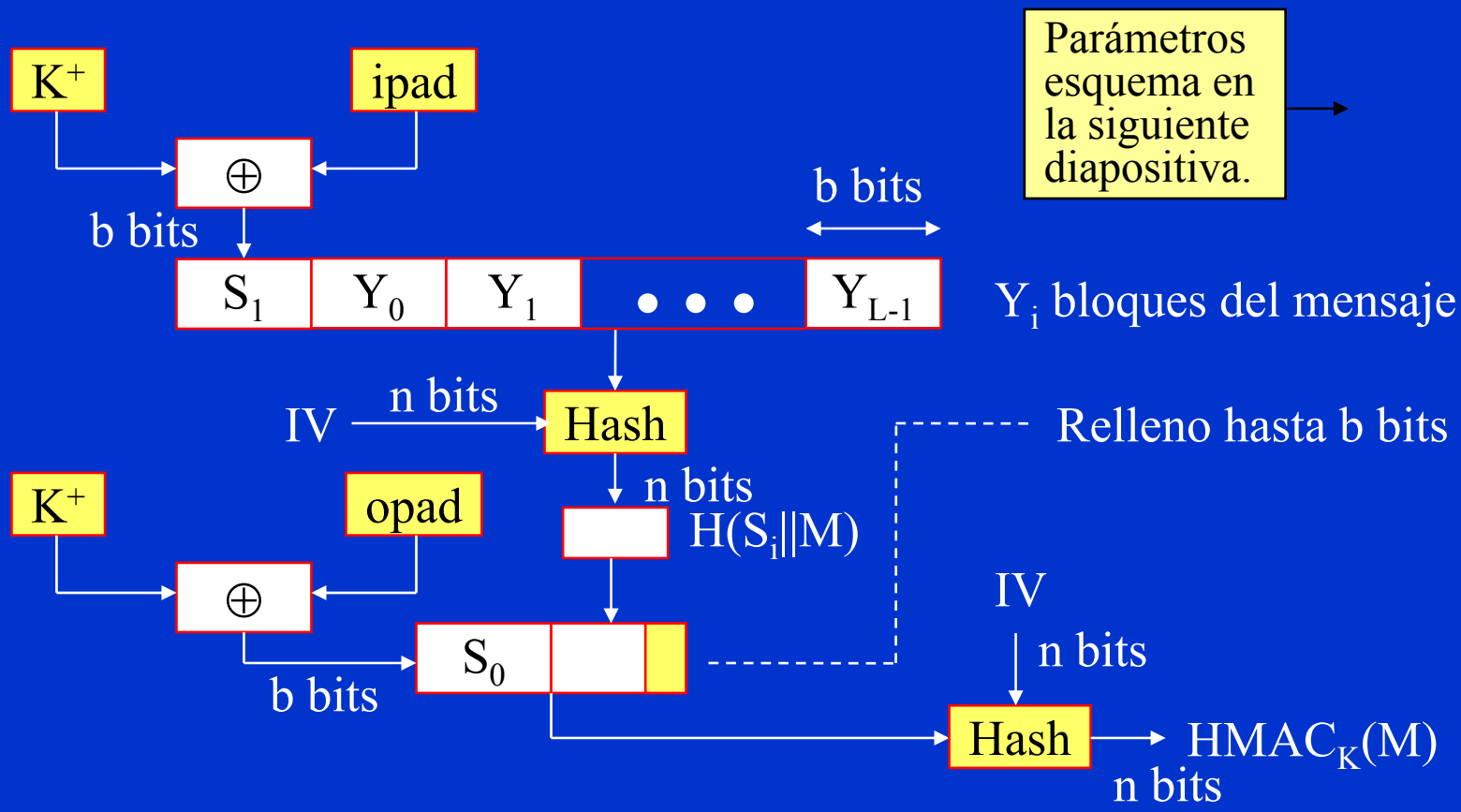
¿Cuáles son las debilidades de este modelo?

- La Marca Am son 16, 32 ó 64 bits.
- Es un tamaño muy pequeño y podría dar lugar a colisiones: mensajes distintos con iguales MACs.

Autenticación con funciones hash HMAC

- Las funciones hash vistas (MD5, SHA-1, etc.) no han sido diseñadas para la autenticación al carecer de clave secreta.
- No obstante, son interesantes puesto que su velocidad es mayor que muchos cifradores de bloque, su código fuente es abierto y sus propiedades y funcionamiento son muy conocidos.
- La RFC 2104 propone el uso de una autenticación en entornos seguros como SSL mediante una operación MAC en la que intervenga una función hash: su nombre es HMAC.
- HMAC usa entonces una función hash (MD5, SHA-1, etc.) como una caja negra, una clave K en lo posible mayor que el tamaño del hash en bits, una función xor y operaciones de relleno de bits, tal como se muestra en el siguiente esquema.

Esquema de HMAC



Parámetros, valores típicos y salida HMAC

M = mensaje de entrada incluyendo el relleno.
H = alguna función hash como MD5 (128 bits) o SHA-1 (160 bits).
Y_i = bloque iésimo de M.
L = número de bloques en M.
b = número de bits en cada bloque (512).
n = longitud del resumen del hash ocupado en el sistema (128 / 160 bits).
K = clave secreta (160 bits) aunque se recomienda sea mayor que n. Si la clave K es mayor que b, esta clave se hace pasar antes por la función hash para reducirla a una clave de n bits.
K^+ = clave K con ceros añadidos a la izquierda para alcanzar b bits.
ipad = 00110110 octeto repetido b/8 (64) veces.
opad = 01011010 octeto repetido b/8 (64) veces.

Salida HMAC: $HMAC_K = H\{(K^+ \oplus opad) \,||\, H[(K^+ \oplus ipad) \,||\, M\,]\}$

Operaciones y seguridad de HMAC

- Añadir ceros a la izquierda de K hasta obtener K^+, una cadena de b bits.
- Sumar or exclusivo K^+ con la cadena ipad para obtener S_1 de b bits.
- Añadir a S_1 el mensaje M como bloques Y_i de b bits cada uno.
- Aplicar la función hash elegida a la cadena de bits antes formada, con el vector inicial IV de n bits para generar un hash de n bits.
- Sumar or exclusivo K^+ con la cadena opad para obtener S_0 de b bits.
- Añadir a S_0 el hash anterior, rellenando este último hasta b bits.
- Aplicar la función hash elegida a los dos bloques de b bits generados en el paso anterior, con vector IV de n bits para generar un hash de n bits.
- El resultado de este último hash es el HMAC del mensaje M con la clave K, es decir $HMAC_K(M)$.

Aunque la seguridad de HMAC está directamente relacionada con el hash utilizado, el modelo presentado no es seguro. Hay otras configuraciones más desarrolladas que mejoran esta característica.

Autenticación con cifra simétrica y un KDC

Características

- Utilización de un KDC (Key Distribution Centre) o Distribuidor de Claves de Confianza, que comparte una clave llamada maestra con los clientes.
- Cada parte, usuario o entidad de la red comparte una clave secreta y única o clave maestra con el KDC.
- El KDC se ocupa de distribuir una clave de sesión que va a ser utilizada en la conexión entre dos partes.
- La clave de sesión se protege con la clave maestra de los participantes de una comunicación.

Propuesta de Needham y Schroeder (1978)

1. $A \rightarrow KDC$: $ID_A||ID_B||N_1$
2. $KDC \rightarrow A$: $E_{KA}[K_S||ID_B||N_1||E_{KB}[K_S||ID_A]]$
3. $A \rightarrow B$: $E_{KB}[K_S||ID_A]$
4. $B \rightarrow A$: $E_{KS}[N_2]$
5. $A \rightarrow B$: $E_{KS}[f(N_2)]$

f(N2) es una función conocida por A y B.

ID_A = Nombre de A o identificador de A
ID_B = Nombre de B o identificador de B
N_X = Valor nonce
K_S = Clave de sesión
E_{KX} = Cifrado con clave secreta de X
K_X = Clave secreta de X (A ó B)

¿Qué sucede si no se realizan los pasos 4 y 5?

Un intruso podría capturar el mensaje en el paso 3 y repetirlo, interfiriendo así las operaciones de B.

Más debilidades de la propuesta N-S

- Algo más improbable es que un intruso X tenga una clave de sesión antigua y repita el mensaje del paso 3 enviado a B, quien no tiene porqué recordar todas las claves de sesión usadas previamente con A.
- Si X captura el mensaje del paso 4, podrá suplantar la respuesta de A del paso 5, enviando a B mensajes que parecen venir de A con clave de sesión autenticada.
- Denning propone la inclusión de un valor timestamp en los pasos 2 y 3. La utilización de timestamps requiere la sincronización de relojes, pero la utilización de valores nonce es también vulnerable. Se propone como solución óptima dar una duración a la clave de sesión (tickets).

Solución propuesta por Denning y tickets

Denning

1. $A \rightarrow KDC: \quad ID_A||ID_B$
2. $KDC \rightarrow A: \quad E_{KA}[K_S||ID_B||T||E_{KB}[K_S||ID_A||T]]$
3. $A \rightarrow B: \quad E_{KB}[K_S||ID_A||T]$
4. $B \rightarrow A: \quad E_{KS}[N_1]$
5. $A \rightarrow B: \quad E_{KS}[f(N_1)]$

T = Timestamp

$|\ Clock - t\ | < \Delta t_1 + \Delta t_2$

Tickets

1. $A \rightarrow B: \quad ID_A||N_A$
2. $B \rightarrow KDC: \quad ID_B||N_B||E_{KB}[ID_A||N_A||T_B]$
3. $KDC \rightarrow A: \quad E_{KA}[ID_B||N_A||K_S||T_B]||E_{KB}[ID_A||K_S||T_B]||N_B$
4. $A \rightarrow B: \quad E_{KB}[ID_A||K_S||T_B]||E_{KS}[N_B]$

Tickets

Solución al problema del timestamp

Autenticación en un sentido

- El correo electrónico es un ejemplo de aplicación que requiere este tipo de autenticación:
  - No es necesario que el emisor y el receptor estén conectados al mismo tiempo.
  - El receptor necesita confirmar de alguna forma que el emisor del mensaje es quien dice ser.

1.	$A \rightarrow KDC$:	$ID_A \| ID_B \| N_1$
2.	$KDC \rightarrow A$:	$E_{KA}[K_S \| ID_B \| N_1 \| E_{KB}[K_S \| ID_A]]$
3.	$A \rightarrow B$:	$E_{KB}[K_S \| ID_A] \| E_{KS}[M]$

Se garantiza así que sólo el receptor puede leer el mensaje y autentica además al emisor. Es vulnerable a repeticiones.

Autenticación con Kerberos (1988)

Servicio de autenticación desarrollado en el Massachusetts Institute of Technology (MIT)

- Perro de tres cabezas y cola de serpiente según mitología griega, guardián de la entrada del Templo de Hades.
- Tres componentes guardarán la puerta: Autenticación, Contabilidad y Auditoría. Las dos últimas cabezas nunca han sido implementadas.

Características y fases de Kerberos

- Está basado en el protocolo de distribución de claves de Needham & Schroeder.
- Usa un intercambio de claves con una tercera parte de confianza.
- Fases de autenticación:
  - Obtención de credenciales
    - Tickets
    - Autenticadores
  - Petición de autenticación frente un servicio.
  - Presentación de las credenciales al servidor final.

Elementos del diálogo de autenticación

- Usuario, cliente y servidor: persona o máquina que necesita autenticarse en la red.
- Principal: entidad o cliente que usa Kerberos y que ha sido previamente autenticado por un servidor de autenticación.
- Servicio y servidor: servicio es una entidad abstracta (por ejemplo correo) y servidor el proceso que lo ejecuta.
- Clave y password: función aplicada a la clave de usuario.
- Credenciales: lo que se utiliza para autenticarse.
- Maestros y esclavos: la máquina que hospeda la base de datos es el master y esclavos son las máquinas que poseen copias de ésta.
- Dominio: nombre de entidad administrativa que mantiene los datos de autenticación.

El ticket en la credencial de Kerberos

- Existen dos tipos de credenciales utilizadas en el modelo de Kerberos: tickets y autenticadores.
- Es necesario un ticket para pasar de una forma segura la identidad del cliente entre el servidor de autenticación y el servidor final.

$$[s, c, addr, timestamp, life, K_{S,C}]K_S$$

s = nombre del servidor
c = nombre del cliente
addr = dirección Internet del cliente
timestamp = fecha de iniciación del ticket
life = duración
K_S = clave de sesión aleatoria

El autenticador en la credencial de Kerberos

- El autenticador contiene información adicional que, comparada con el ticket, prueba que el cliente que presenta el ticket es el mismo a quien se le entregó,

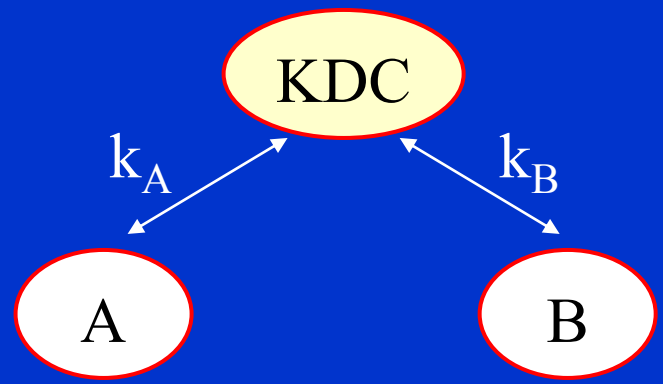
$$[c, addr, timestamp]K_{S,C}$$

c = nombre del cliente
addr = dirección Internet de la estación de trabajo
timestamp = fecha de iniciación del ticket
$K_{S,C}$ = clave de sesión que es parte del ticket

Los objetivos de las credenciales son minimizar el número de veces que un usuario tiene que introducir la password así como eliminar el problema de enviar la password en texto plano por la red.

Ejemplo de autenticación con Kerberos (1)

Se autenticará el usuario A ante el usuario B teniendo como tercera parte de confianza al KDC.
Usarán el algoritmo DES.



Paso 1. A pide una credencial a KDC.
$A \rightarrow KDC$ ID(A), ID(B), NA
Envía el número aleatorio NA para que nadie pueda suplantar su identidad.

Paso 2. KDC genera una clave de sesión K_S con período de validez L y una credencial c para el usuario B.
$m = E_{kA}[K_S, L, NA, ID(B)]$; $c = E_{kB}[K_S, L, ID(A)]$
$KDC \rightarrow A$ (m, c)

Ejemplo de autenticación con Kerberos (2)

Paso 3. A descifra con su clave k_A el valor m:
$D_{kA}[K_S, L, NA, ID(B)] = K_S, L, NA, ID(B)$
Y luego con la clave de sesión entregada por KDC generará un autenticador a para el usuario B junto con un sello temporal T_A.
$a = E_{KS}[ID(A), T_A]$
$A \rightarrow B \quad (c, a)$

Paso 4. B descifra con su clave k_B c:
$D_{kB}[K_S, L, ID(A)] = K_S, L, ID(A)$
Ahora que tiene K_S descifra el valor a:
$D_{KS}[ID(A), T_A] = ID(A), T_A$
Comprueba que coinciden los identificadores ID(A) del usuario A y que T_A está dentro del rango de validez L dado por KDC.

Ejemplo de autenticación con Kerberos (3)

Paso 5. B calcula una función acordada con A por ejemplo ($T_A + 1$) y con la clave de sesión K_S se lo envía cifrado.

$B \rightarrow A \quad h = E_{KS}[T_A + 1]$

Paso 6. A descifra h con la clave de sesión K_S:

$D_{KS}[T_A + 1] = T_A + 1$

Comprueba que coincide con lo acordado lo que le demuestra que B ha recibido correctamente la clave de sesión K_S.

El valor de T_A permite más autenticaciones dentro del período de validez L sin la intervención de la tercera parte de confianza KDC.

Resumen del ejemplo de autenticación

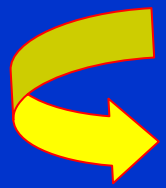
- Los valores de c y m aseguran la confidencialidad en la transmisión de la clave de sesión K_S.
- Los valores de a y h aseguran la confirmación de la recepción correcta de la clave de sesión K_S por B.
- En este protocolo sólo se autentica A ante B.
- Existen extensiones del algoritmo que permiten una autenticación doble entre A y B.
- Otra opción es que los usuarios A y B compartan una clave K_S sin que su valor sea conocido por KDC.

El hash en la autenticación asimétrica

- H(M) es el resultado de un algoritmo que con una entrada M de cualquier tamaño, produce salida de tamaño fijo.
- El emisor A envía el mensaje M y la función hash H(M) a B, quien aplica la misma función al mensaje M' recibido. Si los mensajes son iguales entonces se asegura que los hash también son iguales. No obstante, el hecho de que los hash coincidan no significa que los mensajes M y M' sean iguales (ya visto en el capítulo correspondiente).
- Integridad: el receptor B puede confiar en que nadie ha modificado el mensaje durante la transmisión pues el valor H(M) en destino coincide con el enviado por A.
- Autenticación: Es imposible la autenticación de usuario, salvo que se use un modelo con clave tipo HMAC ya visto.

Autenticación con sistemas asimétricos

Al existir una clave pública y otra privada que son inversas, se autentica el mensaje y al emisor.



Permite la firma digital, única para cada mensaje

Problema:

Los sistemas de cifra asimétricos son muy lentos y el mensaje podría tener miles o millones de bytes ...

Solución:

Se genera un resumen del mensaje, representativo del mismo, con una función hash imposible de invertir. La función hash comprime un mensaje de longitud variable a uno de longitud fija y pequeña.

Características de una firma digital

Requisitos de la Firma Digital:

a) Debe ser fácil de generar.
b) Será irrevocable, no rechazable por su propietario.
c) Será única, sólo posible de generar por su propietario.
d) Será fácil de autenticar o reconocer por su propietario y los usuarios receptores.
e) Debe depender del mensaje y del autor.

Son condiciones más fuertes que la de una firma manuscrita.

¡Esta última propiedad es muy importante!

Firma digital RSA de A hacia B

Clave Pública (n_A, e_A) Clave Privada (d_A)

Adela

Algoritmo:

Rúbrica: $r_A H(M) = H(M)^{dA} \bmod n_A$

A envía el mensaje M en claro (o cifrado) al destinatario B junto a la rúbrica: $\{M, r_A H(M)\}$

Benito

El destinatario B tiene la clave pública e_A, n_A de A y descifra $r_A H(M) \Rightarrow \{(H(M)^{dA})^{eA} \bmod n_A\}$ obteniendo así H(M). Como recibe el mensaje M', calcula la función hash H(M') y compara:

Si H(M') = H(M) se acepta la firma.

Valores y tamaños típicos de firmas

- En los siguientes ejemplos, por limitación del tamaño de los primos elegidos, se firmarán sólo bloques de una cantidad determinada de bits en función del cuerpo de trabajo.

- No obstante, en los sistemas reales esto no es así puesto que las funciones hash ya vistas entregarán -por lo general- resúmenes comprendidos entre 128 y 160 bits y, por otra parte, el cuerpo de trabajo de la cifra asimétrica para la firma digital será como mínimo de 512 bits (si bien en la actualidad se recomienda al menos 1.024). Por lo tanto el resumen que se firma es menor que el cuerpo de cifra o, lo que es lo mismo, es parte del conjunto de restos del grupo.

Ejemplo de firma digital RSA (B → A)

Benito

Hola. Te envío el documento. Saludos, Beni.

Adela

Sea H(M) = F3A9 (16 bits)

Claves Benito
$n_B = 65.669$
$e_B = 35$, $d_B = 53.771$

$2^{16} < 65.669 < 2^{17}$
Forzaremos firmar bloques de 16 bits

Claves Adela
$n_A = 66.331$
$e_A = 25$, $d_A = 18.377$

Firma

$H(M) = F3A9_{16} = 62.377_{10}$

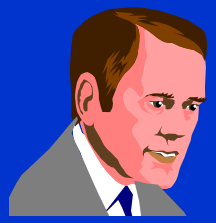
$r_{H(M)} = H(M)^{dB} \bmod n_B$

$r_{H(M)} = 62.377^{53.771} \bmod 65.669 = 24.622$

Benito envía el par (M, r) = (M, 24.622)

Nota: los primos que usa Benito son 97, 677 y Adela 113, 587

Comprobación la firma RSA por A



Benito

Claves Benito
$n_B = 65.669$
$e_B = 35$, $d_B = 53.771$

Claves Adela
$n_A = 66.331$
$e_A = 25$, $d_A = 18.377$

Adela

Teníamos que: $H(M) = F3A9_{16} = 62.377_{10}$

$r_{H(M)} = H(M)^{dB} \bmod n_B \; r_{H(M)} = 62.377^{53.771} \bmod 65.669 = 24.622$

Benito había enviado el par (M, r) = (M, 24.622)

Adela recibe un mensaje M' junto con una rúbrica r = 24.622:

- Calcula $r^{eB} \bmod n_B = 24.622^{35} \bmod 65.669 = 62.377$.
- Calcula el resumen de M' es decir H(M') y lo compara con H(M).
- Si los mensajes M y M' son iguales, entonces H(M) = H(M') y se acepta la firma como válida.
- NOTA: No obstante, H(M) = H(M') no implica que M = M'.

Firma digital ElGamal de A hacia B

Adela

ElGamal: El usuario A generaba un número aleatorio a (clave privada) del cuerpo p. La clave pública es $\alpha^a \bmod p$, con α generador.

Firma: (r, s)

Algoritmo de firma:

1º El usuario A genera un número aleatorio h, que será primo relativo con $\phi(p)$: $h / mcd\ \{h, \phi(p)\} = 1$

2º Calcula $h^{-1} = inv\ \{h, \phi(p)\}$

3º Calcula $r = \alpha^h \bmod p$

4º Resuelve la siguiente congruencia:

$$M = a*r + h*s \bmod \phi(p) \quad \therefore$$
$$s = (M - a*r)*inv[h,\phi(p)] \bmod \phi(p)$$

Comprobación de firma ElGamal por B

Benito

Conoce: p y $(\alpha^a) \bmod p$

Algoritmo comprobación de firma:

1º El usuario B recibe el par (r, s) y calcula:

$r^s \bmod p$ y $(\alpha^a)^r \bmod p$

2º Calcula $k = [(\alpha^a)^r * r^s] \bmod p$

Como r era igual a $\alpha^h \bmod p$ entonces:

$k = [(\alpha^{ar} * \alpha^{hs}] \bmod p = \alpha^{(ar + hs)} \bmod p = \alpha^{\beta} \bmod p$

3º Como $M = (a*r + h*s) \bmod \phi(p)$ y α es una raíz primitiva de p se cumple que:

$\alpha^{\beta} = \alpha^{\gamma}$ ssi $\beta = \gamma \bmod (p-1)$

4º Comprueba que $k = \alpha^M \bmod p$ ⟶

Si $k = [(\alpha^a)^r * r^s] \bmod p$ es igual a $\alpha^M \bmod p$ ...

Se acepta la firma

Ejemplo de firma ElGamal (B → A)

Benito

¡Hola otra vez! Soy Benito de nuevo. Saludos.

Adela

Sea H(M) = A69B (16 bits)

Claves Benito

$p_B = 79.903 \quad \alpha = 10$

$\alpha^b \bmod p = 3.631$

b = 20 h = 31

$2^{16} < 79.903 < 2^{17}$

Forzaremos firmar bloques de 16 bits

Firma

1) $h^{-1} = inv[h, \phi(p)] = inv\ (31, 79.902) = 5.155$
2) $r = \alpha^h \bmod p = 10^{31} \bmod 79.903 = 11.755$
3) $s = [H(M) - b*r]*[inv(h,\phi(p)] \bmod \phi(p)$ $\quad H(M) = A69B_{16} = 42.651_{10}$
4) $s = [42.651-20*11.755]*5.155 \bmod 79.902$
5) $s = 68.539$ Luego, la firma será ⟶ (r, s) = (11.755, 68.539)

Comprobación de firma ElGamal por A

Benito

Claves Benito

$p_B = 79.903 \quad \alpha = 10$

$\alpha^b \bmod p = 3.631$

$b = 20 \qquad h = 31$

$H(M) = A69B = 42.651$

Adela

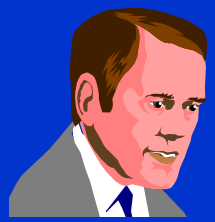
Adela recibe el par (r, s) = (11.755, 68.539)

Comprobación de la firma:

1) $r^s \bmod p = 11.755^{68.539} \bmod 79.903 = 66.404$
2) $(\alpha^b)^r \bmod p = 3.631^{11.755} \bmod 79.903 = 12.023$
3) $(\alpha^b)^r * r^s \bmod p = (12.023 * 66.404) \bmod 79.903 = 64.419 = k$
4) $\alpha^{H(M)} \bmod p = 10^{42.651} \bmod 79.903 = 64.419$

Como hay igualdad se acepta la firma

Importancia de α en la firma de ElGamal



Claves Benito

$p_B = 79.903$ $\alpha = 10$

$\alpha^b \bmod p = 3.631$

$b = 20$ $h = 31$

$\alpha = 10$ es un generador del cuerpo $p = 79.903$ puesto que:

$p-1 = 79.902 = 2*3^2*23*193$

$q_1 = 2; q_2 = 3; q_3 = 23; q_4 = 193$

y se cumple $10^{(p-1)qi} \bmod p \neq 1$

$10^{39.951} \bmod 79.903 = 79.902$

$10^{26.634} \bmod 79.903 = 71.324$

$10^{3.474} \bmod 79.903 = 2.631$

$10^{414} \bmod 79.903 = 41.829$

Si se elige $\alpha = 11$ que no es raíz, para el exponente 39.951 se obtiene el valor 1. No nos servirá para la firma porque será imposible comprobarla mediante la ecuación $k = \alpha^M \bmod p$.

Estándares de firma digital

1991: National Institute of Standards and Technology (NIST) propone el DSA, Digital Signature Algorithm, una variante de los algoritmos de ElGamal y Schnoor.

1994: Se establece como estándar el DSA y se conoce como DSS, Digital Signature Standard.

1996: La administración de los Estados Unidos permite la exportación de Clipper 3.11 en donde viene inmerso el DSS, que usa una función hash de tipo SHS, Secure Hash Standard.

El peor inconveniente de la firma propuesta por ElGamal es que duplica el tamaño del mensaje M al enviar un par (r, s) en Z_p y $\phi(p)$. No obstante, se solucionará con el algoritmo denominado DSS. →

Digital Signature Standard DSS

Parámetros públicos de la firma:

- Un número primo grande p (512 bits)
- Un número primo q (160 bits) divisor de p-1
- Un generador α "de orden q" del grupo p

Generador de orden q es aquella raíz α en el cuerpo Z_p de forma que q es el entero más pequeño que verifica:

$$\alpha^q \bmod p = 1$$

En este caso se cumple para todo t que:

$$\alpha^t = \alpha^{t \,(\bmod q)} \bmod p$$

Generación de firma DSS (A → B)

GENERACIÓN DE LA FIRMA POR PARTE DE A

- Claves públicas de A: primos p, q y el generador α
- Clave secreta de la firma: a $(1 < a < q)$ aleatorio
- Clave pública de la firma: $y = \alpha^a \bmod p$
- Para firmar un mensaje $1 < M < p$, el firmante A elige un valor aleatorio $1 < h < q$ y calcula:
  - $r = (\alpha^h \bmod p) \bmod q$
  - $s = [(M + a*r) * inv(h,q)] \bmod q$
- La firma digital de M será el par (r, s)

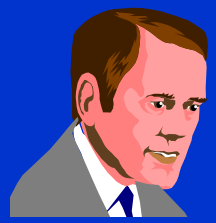
Comprobación de firma DSS por B

COMPROBACIÓN DE LA FIRMA DE A POR B

- B recibe el par (r, s)
- Luego calcula:
  - $w = inv(s, q)$
  - $u = M * w \bmod q$
  - $v = r * w \bmod q$
- Comprueba que se cumple la relación:
  - $r = (\alpha^u y^v \bmod p) \bmod q$
- Si se cumple, se acepta la firma como válida.

La firma tendrá en este caso un tamaño menor que q, es decir, menos bits que los del módulo de firma p ya que se elige por diseño p >> q

Ejemplo de firma DSS (B → A)



Benito

Hola Adela, soy Benito y lo firmo con DSS.

Sea H(M) = 1101000 = 104 (un elemento de p_B)

Adela

Claves Benito

$p_B = 223$ $q_B = 37$ $\alpha = 17$

$y = \alpha^b \bmod p = 30$

$b = 25$, $h = 12$

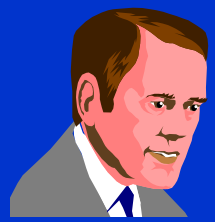
$2^8 < p_B = 223 < 2^7$

Forzaremos firmar bloques de 7 bits

Firma

1) inv (h,q) = inv (12, 37) = 34
2) $r = (\alpha^h \bmod p) \bmod q = (17^{12} \bmod 223) \bmod 37 = 171 \bmod 37 = 23$
3) $s = [H(M)+b*r]*[inv\ (h,q)] \bmod q = [104+25*23]*34 \bmod 37 = 35$
4) La firma digital de H(M) = 104 será: (r, s) = (23, 35)
5) Benito transmite a Adela el bloque (M, r, s) = (M, 23, 35)

Comprobación de la firma DSS por A



Benito

Claves Benito

$p_B = 223$ $q_B = 37$ $\alpha = 17$

$y = \alpha^b \bmod p = 30$

$b = 25$, $h = 12$

Adela recibe:

$(M, r, s) = (M, 23, 35)$

Adela

Comprobación de firma

1) $w = inv\ (s, q) = inv\ (35, 37) = 18$
2) $u = M*w \bmod q = 104*18 \bmod 37 = 22$
3) $v = r*w \bmod q = 23*18 \bmod 37 = 7$
4) $¿(\alpha^u\ y^v \bmod p) \bmod q = r\ ?$
5) $[(17^{22}30^7) \bmod 223] \bmod 37 = 23$

¿Igualdad?

En caso afirmativo, se acepta la firma

Y el tamaño será menor que $q_B = 37$ es decir << $p_B = 223$ que era precisamente el punto débil del sistema ElGamal.

Fin del Tema 14

Cuestiones y ejercicios (1 de 2)

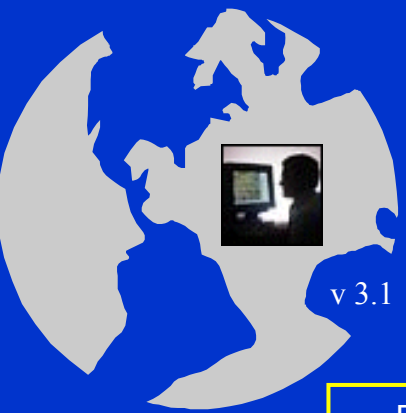
1. ¿Por qué cree que un escenario de integridad en la que hay siempre una tercera parte de confianza activa no sería adecuado en Internet?
2. Si una Autoridad de Certificación es la tercera parte de confianza, ¿actúa de forma activa o pasiva en la comunicación? Explíquelo.
3. ¿Qué importancia tiene la redundancia del lenguaje en un sistema de autenticación? ¿Qué sucedería si no hubiese esa redundancia?
4. Hacemos una autenticación de mensaje mediante un MAC en el que el algoritmo es DES. ¿Sería acertado usar modo ECB? ¿Por qué?
5. En un sistema de autenticación mediante Kerberos, ¿cómo sabe el que comienza el protocolo (A) que no hay un impostor que intenta suplantarle? ¿Cómo sabe el que comienza el protocolo (A) que el segundo usuario (B) o elemento del sistema ha recibido bien la clave de sesión entregada por el centro de distribución de claves KDC?

Cuestiones y ejercicios (2 de 2)

6. ¿Cómo podríamos usar una función hash para comprobar que un archivo no se ha modificado o que está libre de virus?
7. Nos afirman que podemos autenticar un mensaje usando para ello sólo una función hash. ¿Es esto verdadero? ¿Por qué?
8. ¿Por qué se dice que una firma digital tiene condiciones más fuertes que una firma manuscrita?
9. ¿Por qué es importante que la firma digital dependa del mensaje?
10. Firme digitalmente con RSA el mensaje M = 121 si los datos del firmante son p = 19; q = 31, d = 149. Compruebe la firma.
11. Con p = 331 y clave privada 15, vamos a firmar digitalmente con ElGamal el mensaje M = 250 . Para ello, elija los valores más pequeños posibles de los parámetros α y h. Compruebe la firma.
12. Repita el ejercicio 10 con DSS y valores propuestos por Ud.

Tema 15
Certificados Digitales

Seguridad Informática y Criptografía



Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 11 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Nota del autor

El contenido de este archivo corresponde a una primera introducción básica sobre certificados digitales, en particular X.509, y el intercambio de éstos entre cliente y servidor para una completa autenticación. En este año 2003 se irá ampliando este capítulo de forma que en la próxima versión de este libro se pueda contar con ya con un tema de certificación digital y Autoridades de Certificación como debe corresponder a un curso de criptografía y seguridad informática.

¿Qué son los certificados digitales?

Un certificado digital es un documento que contiene diversos datos, entre ellos el nombre de un usuario y su clave pública, y que es firmado por una Autoridad de Certificación (AC).

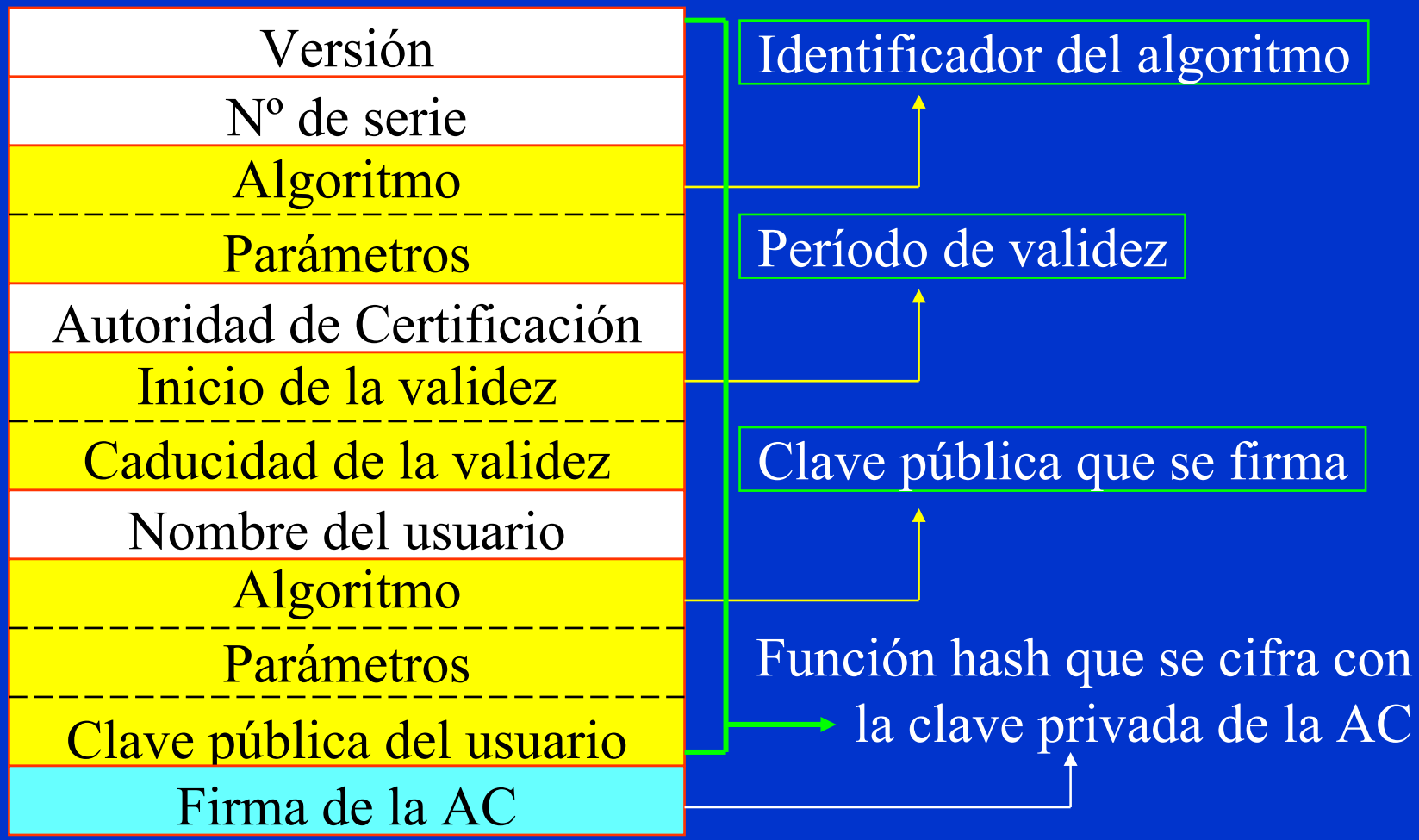
Como emisor y receptor confiarán en esa AC, el usuario que tenga un certificado expedido por ella se autenticará ante el otro, en tanto que su clave pública está firmada por dicha autoridad.

Certificado digital X.509

- X.509 está basado en criptografía asimétrica y firma digital.
- En X.509 se define un framework (una capa de abstracción) para suministrar servicios de autenticación a los usuarios del directorio X.500.
- La autenticación se realiza mediante el uso de certificados.

  - Un certificado contiene: el nombre de la AC, el nombre del usuario, la clave pública del usuario y cualquier otra información como puede ser un un indicador de tiempo o *timestamp*.
  - El certificado se cifra con la clave privada de la AC.
  - Todos los usuarios poseen la clave pública del AC.

Formato del certificado digital X.509



Campos del certificado digital X.509

- V: Versión del certificado (actualmente V3).
- SN: Número de serie.
- AI: identificador del algoritmo de firma que sirve para identificar el algoritmo usado para firmar el paquete X.509.
- CA: Autoridad certificadora.
- T_A: Periodo de validez.
- A: Propietario de la clave pública que se está firmando.
- P: Clave pública más identificador de algoritmo utilizado y más parámetros si son necesarios.
- Y{I}:Firma digital de Y por I usando la clave privada de la unidad certificadora.

CA<<A>> = CA { V, SN, AI, CA, T_A, A, AP }

Y<<X>> es el certificado del usuario X expedido por la autoridad certificadora Y.

Autoridades de Certificación

Autoridad de Certificación es un ente u organismo que, de acuerdo con unas políticas y algoritmos, certificará -por ejemplo- claves públicas de usuarios o servidores.

El usuario A enviará al usuario B su certificado (la clave pública firmada por AC) y éste comprobará con esa autoridad su autenticidad. Lo mismo en sentido contrario.

Elementos de una AC

El sistema de autenticación debe tener:

- Una política de certificación
- Un certificado de la CA
- Los certificados de los usuarios (X.509)
- Los protocolos de autenticación, gestión y obtención de certificados:
  - Se obtienen de bases de datos (directorio X.500)
  - O bien directamente del usuario en tiempo de conexión (WWW con SSL)

Algunas características de diseño de la AC

- Deberá definirse una política de certificación
  - Ambito de actuación y estructura
  - Relaciones con otras ACs
- Deberá definirse el procedimiento de certificación para la emisión de certificados:
  - Verificación on-line
  - Verificación presencial
- Deberá generarse una Lista de Certificados Revocados

Funcionamiento de una AC

- Puesta en marcha de la AC:
  - Generará su par de claves
  - Protegerá la clave privada con una *passphrase*
  - Generará el certificado de la propia AC
- Distribución del certificado de la AC:
  - A través del directorio X.500
  - Por medio de páginas Web
- Podrá certificar a servidores y a clientes

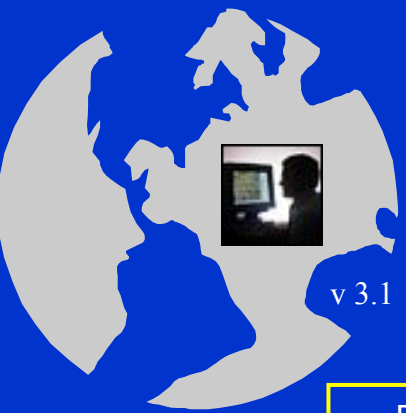
Fin del Tema 15

Cuestiones y ejercicios

1. ¿Qué es lo que certifica un certificado digital?
2. Solicitamos un certificado digital a una empresa. ¿Está nuestra clave privada entre los datos que nos solicita para el alta?
3. Si un certificado digital pierde la validez, ¿qué debemos hacer?
4. ¿Por qué una Autoridad de Certificación firma una función hash?
5. Si Ud. fuese una Autoridad de Certificación, ¿qué condición pondría para expedir con seguridad un certificado a un usuario?
6. Una Autoridad de Certificación emite certificados digitales de hasta 1.024 bits y duración un año. Si ella tiene un certificado digital raíz de 2.048 bits, ¿sería lógico plantear una validez de éste por 10 años?
7. ¿Qué es y cuándo se solicita la revocación de un certificado digital?
8. ¿Qué significa que la Autoridad de Certificación deba gestionar una lista de certificados revocados?

Tema 16
Aplicaciones Criptográficas

Seguridad Informática y Criptografía



Material Docente de
Libre Distribución

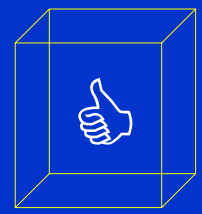
Ultima actualización: 03/03/03
Archivo con 83 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Resumen de los sistemas de clave secreta

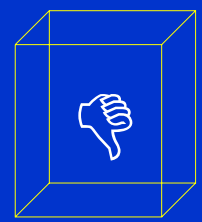
Pros y contras de los Sistemas de Clave Secreta

- El emisor y el receptor comparten una misma clave.
- La seguridad depende sólo del secreto de la clave.
- La velocidad de cifra es muy alta y los sistemas con un espacio de clave grande son muy seguros.
- Permiten autenticar los mensajes con MACs.



... pero

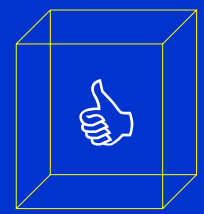
- Es imposible establecer un sistema de distribución y gestión de claves eficiente entre emisor y receptor.
- Carecen de una firma digital en sentido amplio.



Resumen de los sistemas de clave pública

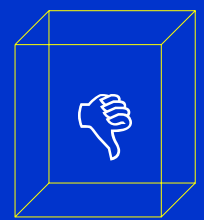
Pros y contras de los Sistemas de Clave Pública

- Emisor y receptor generan un par de claves, pública y privada, relacionadas por una función con trampa.
- Emisor y receptor de un mensaje usan claves diferentes para las operaciones de cifrado, descifrado y firma.
- La seguridad del sistema va asociada a la resolución de un problema matemático difícil.
- Tiene firma digital: autenticación de mensaje y del emisor.



... pero

- Es necesario contar con mecanismos de certificación para asegurar la veracidad de las claves públicas: ACs.
- Son sistemas de cifra muy lentos.



El correo electrónico seguro

A comienzos de los años 90 hacen su aparición dos sistemas de correo electrónico seguro:

- PEM (Private Enhanced Mail)
- PGP (Pretty Good Privacy)

De los dos, ha sido PGP quien se ha convertido en un estándar de hecho en clientes del e-mail seguro. Por lo tanto veremos sólo algunos aspectos genéricos de PEM y analizaremos más en profundidad PGP.

Su estudio nos permitirá ver una aplicación real de los sistemas de cifra y firma analizados en el curso.

Private Enhanced Mail PEM

- Es una propuesta de la IETF Internet Engineering Task Force en 1985. El documento técnico se publica en 1993.
- Las especificaciones técnicas están en las RFCs Request For Comments números 1421, 1422, 1423 y 1424.
- Se usa conjuntamente con el protocolo SMTP Simple Mail Internet Protocol.
- Cifrado de la información: DES modo CBC.
- Generación y gestión de claves: RSA de 508 a 1024 bits. Estructura de certificados según la norma X.509.
- Clave de sesión: DES modo ECB, TripleDES-EDE.
- Firma digital: RSA, MD2, MD5.

Implementación de PEM

- Es compatible con otros modelos de mensajería como, por ejemplo, X.400.
- PEM se implementa en el nivel de aplicación:
  - es independiente de los protocolos de los niveles OSI o TCP/IP inferiores.
  - es independiente de los sistemas operativos o del ordenador.
- Se puede implementar como un módulo independiente que trabaje con el cliente de correo habitual para el usuario.

Servicios de seguridad en PEM

- Servicios de seguridad contemplados:
  - Autenticación del origen.
  - Confidencialidad.
  - Integridad del mensaje.
  - No repudio del origen cuando se utiliza gestión de clave con algoritmo de clave asimétrica.
- Servicios de seguridad no contemplados:
  - Control de acceso.
  - Confidencialidad del tráfico de mensajes.
  - No repudio del mensaje por parte del receptor.

Formato e implementación de PEM

TIS/PEM

Plataformas UNIX. Trusted Information System. Código fuente disponible para los ciudadanos o empresas de Estados Unidos y Canadá. Usa una jerarquía de certificación múltiple.

RIPEM

Implementa parte de los protocolos PEM sin certificados para autenticación de claves. Gratuito para aplicaciones no comerciales. Exportación prohibida fuera de Estados Unidos. Existen versiones utilizadas en todo el mundo.

Pretty Good Privacy PGP

- Philip Zimmermann publica la versión 1.0 de PGP en 1991 con mínimos requisitos de hardware y software.
- En 1992 aparece la versión 2.0 en la que ya participan programadores de todo el mundo. Su código se escribe fuera de USA para evitar las leyes restrictivas respecto al software criptográfico y sus problemas legales.
- En 1993 aparece la versión 2.3a muy popular en sitios FTP y válida para varias plataformas de sistemas operativos.
- En 1994 participa en el proyecto el Massachusetts Institute of Technology MIT y aparecen las versiones 2.4, 2.5 y 2.6.
- La versión 2.6.3i se populariza a nivel mundial.

Nota aclaratoria sobre versiones de PGP

Aunque hay más de una oferta de software para correo seguro que el programa PGP, éste se ha convertido en un estándar de hecho. Si bien las últimas versiones del programa orientadas a entornos Windows presentan altas prestaciones, las operaciones básicas siguen siendo las mismas que en la conocida versión 2.6.3i.

Las nuevas versiones de PGP en entorno Windows cambian muy rápidamente por lo que resulta muy difícil tener unos apuntes permanentemente actualizados. Por ello, se usará la versión 2.6.3i como versión simple para la explicación de las operaciones de cifra y firma con PGP y, posteriormente, haremos un repaso de las características de la versiones 6.5.1 y 8.0, la última a la fecha.

La filosofía de las nuevas versiones es exactamente la misma

Tutorial de PGP 2.6.3i

Si no conoce PGP o no ha trabajado nunca con este entorno, le recomiendo que descargue desde la página Web de la Red Temática CriptoRed el archivo del Tutorial de PGP 2.6.3i en formato HTML.

http://www.criptored.upm.es/paginas/software.htm

Esta aplicación, obra de D. David Liñán Zayas, alumno que realizó este proyecto como su Tesis Fin de Carrera tutorizado por el autor de estos apuntes, le servirá para aprender rápidamente los comandos y prestaciones de esta versión de PGP, muy similar a las actuales. Además PGP 2.6.3i ocupa menos que un disquete de 1,4 MB.

Características de PGP 2.6.3i

- PGP, en su versión 2.6.3i (internacional) se convirtió a mediados de la década de los 90 en un estándar de hecho. De hecho, muchos usuarios "siguen fieles" a esta versión.
- Cifra todo tipo de datos en entornos MS-DOS y UNIX. Su orientación principal es el cifrado de los datos y la firma digital en correo electrónico.
- Los algoritmos básicos que usa son:
  - IDEA para cifrar con sistema de clave secreta.
  - RSA para intercambio de claves y firma digital.
  - MD5 para obtener la función hash de la firma digital.

Algoritmos usados en PGP 2.6.3i

- Operación -	- Algoritmo -	- Descripción de su función -
Compresión	**ZIP**	• Se comprime el mensaje en claro y la firma para almacenarlo o transmitirlo.
Generación de claves	**RSA, MD5**	• Genera una clave pública y otra privada, encontrando dos números primos muy grandes. El valor privado se guarda cifrado con IDEA usando como clave un resumen MD5 de la frase de paso secreta.
Cifrado Convencional	**IDEA**	• Cifra el mensaje con una clave de sesión de 128 bits (única) generada en el emisor de forma aleatoria.
Intercambio de claves	**IDEA, RSA**	• Cifra la clave de sesión IDEA con la clave pública del destinatario con RSA y la añade en el criptograma.
Firma Digital	**MD5, RSA**	• La función hash MD5 genera un resumen de 128 bits, que representa al mensaje en claro completo, y que se cifra en RSA con la clave privada del emisor. Se añade al mensaje enviado.
Compatibilidad e-mail	**Base-64**	• Permite transmitir el mensaje a todo tipo de aplicaciones e-mail. Convierte los octetos en caracteres imprimibles.
Segmentación		• Divide el criptograma final en bloques de menos de 50.000 bytes para su correcta transmisión en Internet y su recuperación.

Características del cifrado local

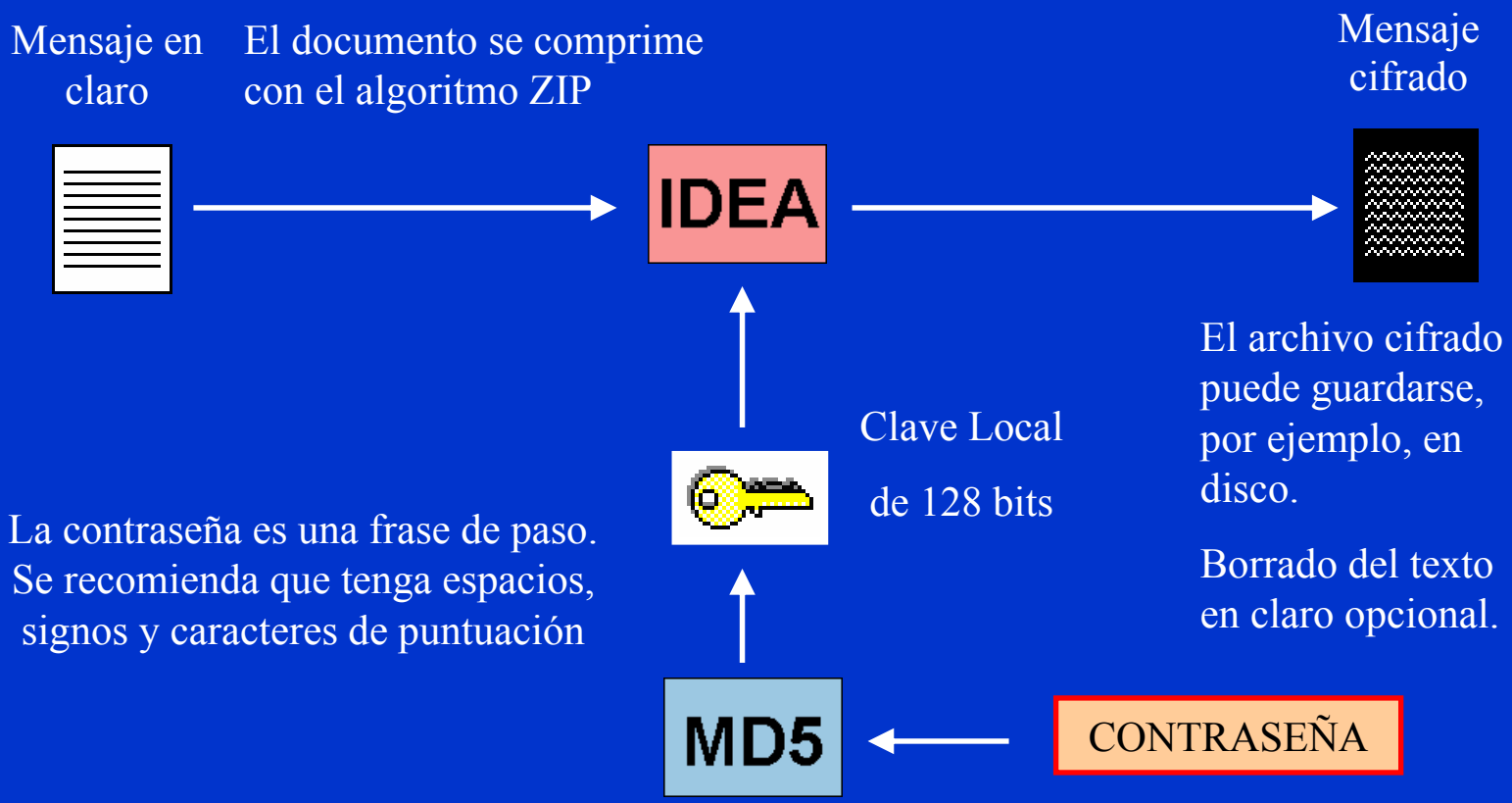
- Esta operación sirve para mantener los archivos protegidos, por ejemplo en el disco duro.
- El acceso al texto en claro sólo será posible si se conoce una clave o contraseña que es la frase de paso usada al cifrar.
- Recuerde que si después de cifrar el archivo borra físicamente el texto en claro -operación que realiza una grabación de unos y ceros aleatorios en la zona de almacenamiento del disco- le será imposible recuperarlo si olvida la contraseña.

Pasos del cifrado local con IDEA

Pasos:

1. PGP solicita una frase de paso: ésta debe ser lo suficientemente larga como para evitar ataques por combinaciones.
2. Se aplica el algoritmo de resumen MD5 a esa contraseña, generando así una clave de 128 bits.
3. PGP cifra el documento con el algoritmo IDEA y le pone como extensión .pgp.
4. Permite luego hacer un borrado físico del archivo en claro.

Esquema de cifrado local con IDEA



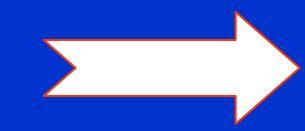
Cada nuevo cifrado requiere una contraseña. Esta puede ser igual o distinta.

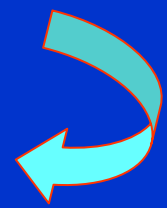
Operaciones con claves asimétricas

- Las operaciones de PGP para cifrar, descifrar, firmar y la comprobación posterior de la firma digital, usan los algoritmos de funciones hash, de clave pública y de clave secreta ya vistos en capítulos anteriores.
- Para poder enviar y recibir correo seguro, es necesario contar al menos con las siguientes claves:

  Clave pública del destinatario.

  Par de claves asimétricas del emisor.





Generación de claves con RSA

Generación de claves asimétricas tipo RSA

- Una vez instalado PGP, se procede a la generación de claves asimétricas del usuario propietario.
- Se elige el tamaño del módulo n, por ejemplo 1.024 bits.
- PGP generará un par de números primos e (clave pública) y d (clave privada) de forma que $e*d \bmod \phi(n) = 1$.
- Para una mayor facilidad en el descifrado en destino, el valor de la clave pública e será pequeño (un valor típico es el número primo $65.537 = 2^{16}+1$).
- PGP pedirá una contraseña o *passphrase* y con ella y MD5 generará una clave de 128 bits con la que cifrará la clave privada antes de almacenarla en el disco.

Anillos de claves asimétricas

- Con las claves pública y privada generadas y otras claves públicas que podrá importar de otros usuarios, se crean dos anillos de claves:
  - Anillo de claves públicas: archivo pubring.pgp en el que se guardan las claves públicas del usuario propietario (puede tener más de una identidad) y las claves públicas de importadas.
  - Anillo de claves privadas: archivo secring.pgp en el que se guarda la o las claves privadas del usuario propietario.
  - Nota: estos anillos cambiarán su extensión en las nuevas versiones.

Estructura del anillo de claves privadas

Sellado de tiempo	Clave ID*	Clave pública	Clave privada cifrada	ID usuario
T_1	$e_1 \bmod 2^{64}$	Clave púb. 1	Clave priv. 1	Usuario 1
---	---	---	---	---
T_i	$e_i \bmod 2^{64}$	e_i	$E_{H(FPi)}(d_i)$	Usuario i
---	---	---	---	---
T_n	$e_n \bmod 2^{64}$	Clave púb. n	Clave priv. n	Usuario n

Descripción de los campos

(*) Se usa este campo para la indexación de la tabla en ambos anillos

Campos de los anillos de claves

Sellado de tiempo:

Fecha y hora de la generación del par de claves.

Clave ID:

Identificador de clave (últimos 64 bits de la clave pública e).

Clave pública:

Número primo e, inverso del primo d en el cuerpo $\phi(n)$.

Clave privada cifrada:

Cifra $E_{H(FPi)}$ de la clave privada d con IDEA y la función hash de la frase de paso del propietario como clave secreta.

ID usuario:

Identificación del usuario, normalmente dirección de email.

Estructura del anillo de claves públicas (1)

Sellado de tiempo	Clave ID*	Clave pública	Confianza propietario	ID usuario
T_1	e_1 mod 2^{64}	Clave púb. 1	flag_confianza 1	Usuario 1
---	---	---	---	---
T_i	e_i mod 2^{64}	e_i	flag_confianza i	Usuario i
---	---	---	---	---
T_n	e_n mod 2^{64}	Clave púb. n	Clave priv. n	Usuario n

continúa en próxima diapositiva

(*) Se usa este campo para la indexación de la tabla en ambos anillos

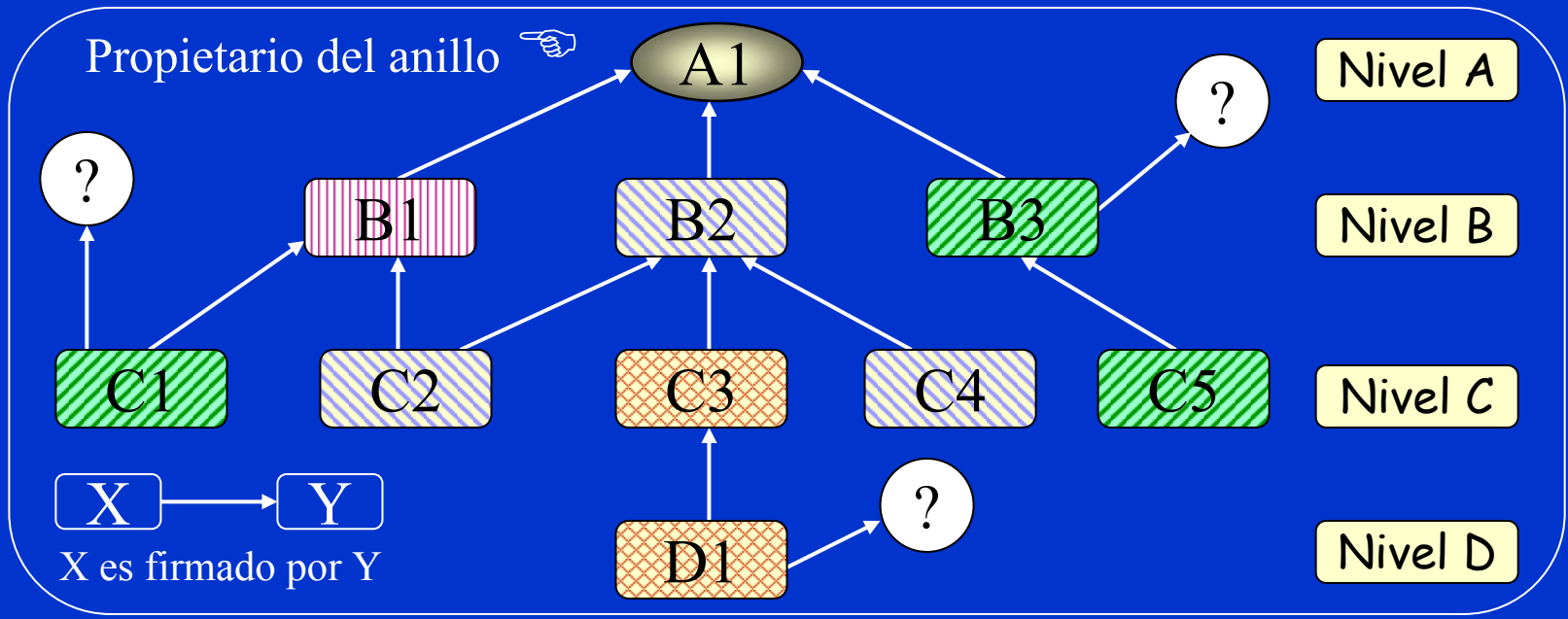
Estructura del anillo de claves públicas (2)

	Legitimación de clave	Firma(s)	Confianza de Firmas
...	flag_confianza 1	...	...
...	---	---	---
...	flag_confianza i	...	...
...	---	---	---
...	flag_confianza n	...	...

viene de la diapositiva anterior

Con la clave pública del destinatario ya podemos enviar el correo cifrado y/o firmado. Pero ... ¿cómo se gestionan las claves en PGP?

Gestión del anillo de claves públicas



A1 cree en el propietario de la clave para firmar otra clave

A1 cree parcialmente en el propietario de la clave para firmar otra clave

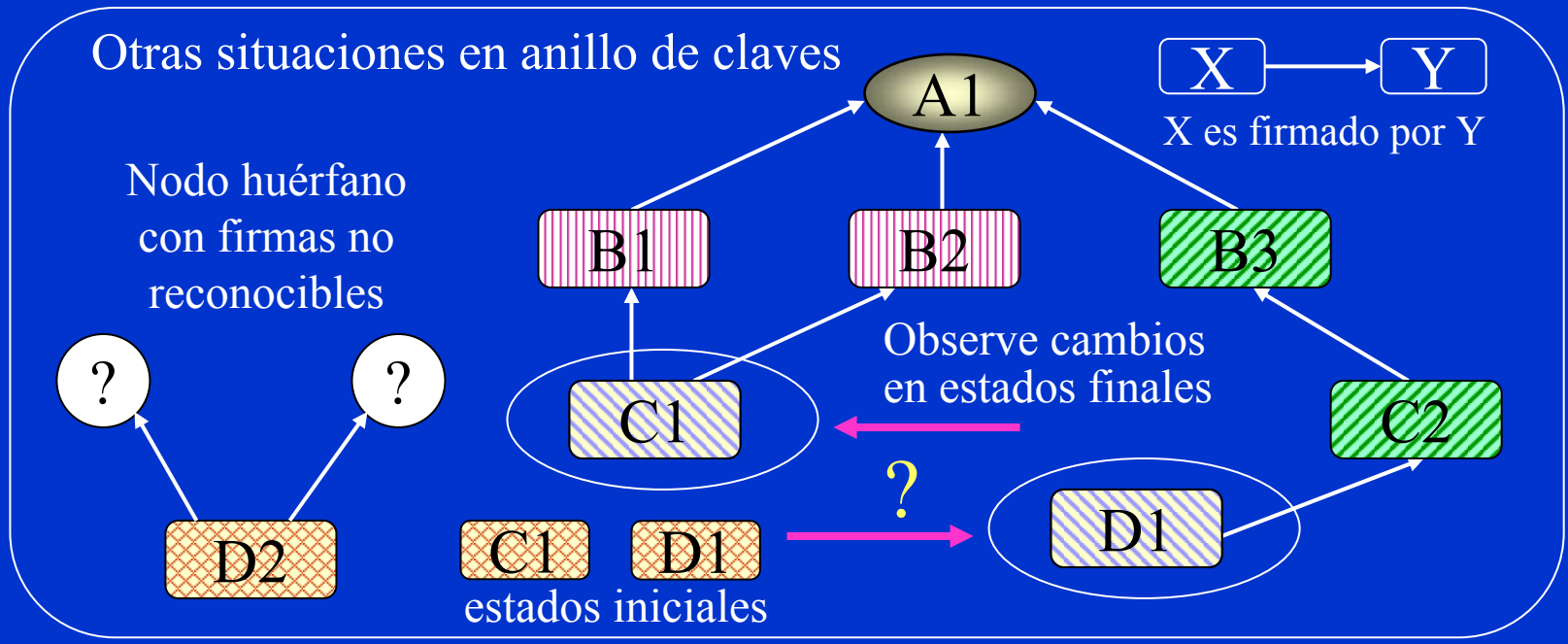
A1 cree en legitimidad de clave

A1 no cree que la clave sea legítima

más

? La clave está firmada por un usuario o Autoridad que no está en anillo de claves de A1

Otros escenarios de confianza en PGP



A1 cree en el propietario de la clave para firmar otra clave

A1 cree parcialmente en el propietario de la clave para firmar otra clave

PGP hace que A1 crea en la legitimidad de las claves pues tienen al menos dos firmas parciales (B1-B2)o una completa (C2) pero no da confianza para firmar

A1 no cree que la clave sea legítima

Problema en estos escenarios de confianza

La gestión de claves en PGP se basa en la confianza mutua: ¡los amigos de tus amigos son mis amigos!

- ✓ En un sistema abierto en Internet como puede ser el comercio electrónico, esta situación y otras más que pueden darse en este sistema de gestión de claves de confianza mutua, resulta inaceptable.
- ✓ La solución, que PGP ya contempla en sus últimas versiones, es la aceptación de las Autoridades de Certificación como certificadores de claves públicas.

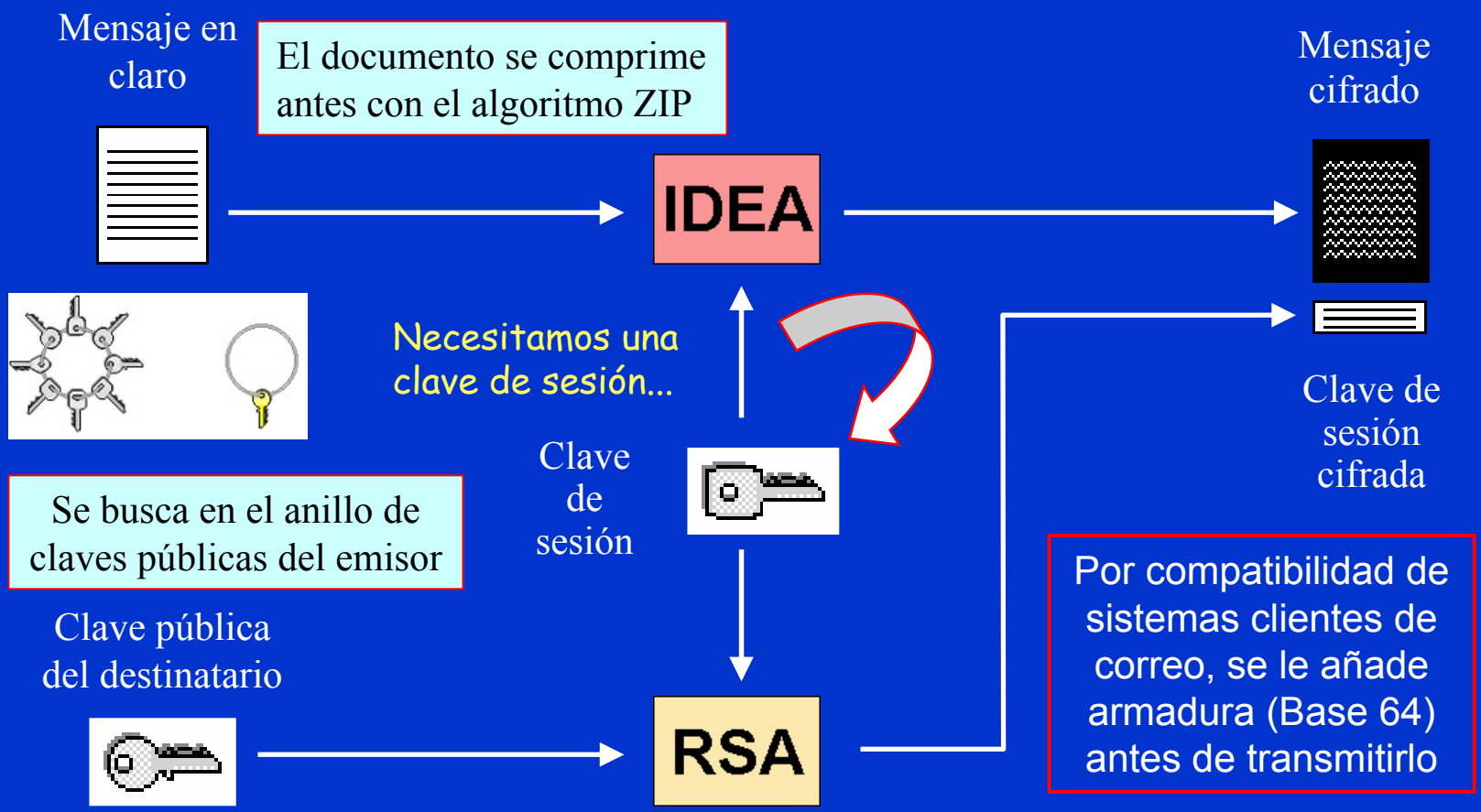
Pasos cifrado con clave pública de destino

Pasos:

1. PGP genera un número aleatorio de 128 bits que será la clave de sesión.
2. Se cifra el mensaje con dicha clave usando IDEA.
3. Se cifra la clave de sesión con la clave pública RSA del destinatario y se añade al criptograma.
4. Se añade el identificador ID de la clave pública del destinatario a la clave de sesión cifrada en el paso 3 como indicativo de la identidad del receptor.

Recuerde que el correo electrónico no es en general una comunicación en tiempo real por lo que, aunque se envía una clave para descifrar el criptograma en recepción, no se trata de una clave de sesión en los mismos términos que se usa, por ejemplo, en una comunicación SSL.

Cifrado con clave pública de destino

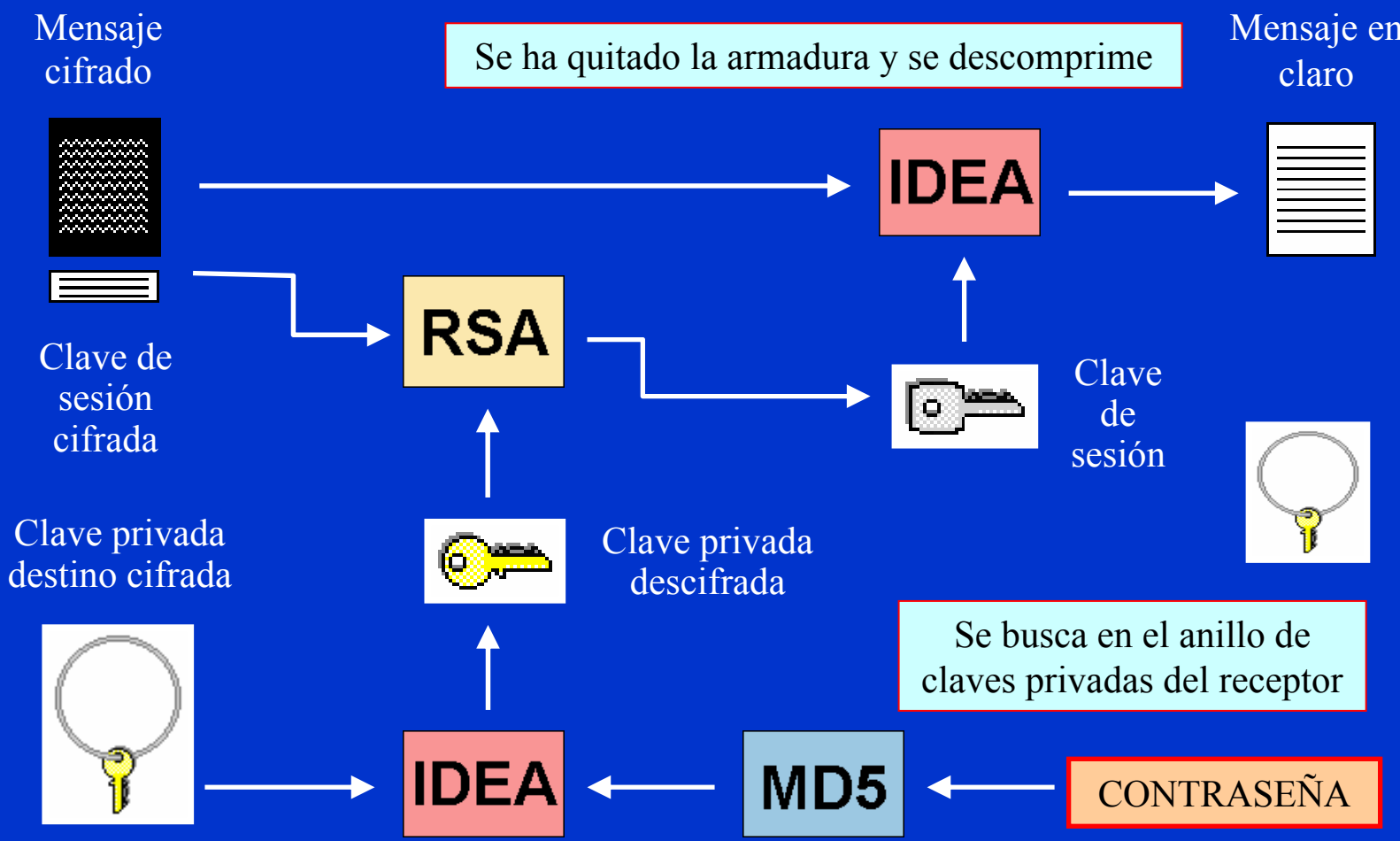


Pasos descifrado con clave privada destino

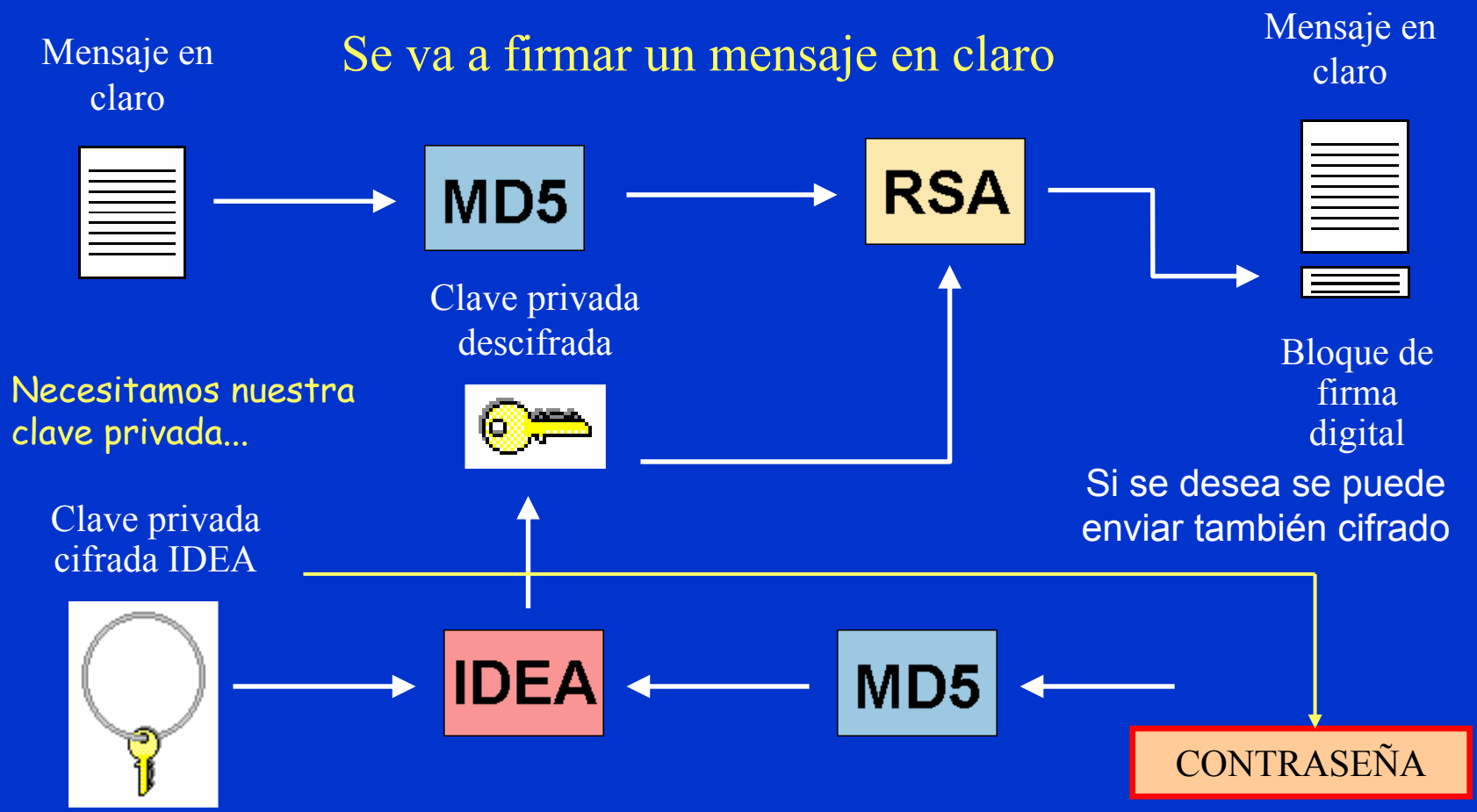
Pasos:

1. PGP busca en la cabecera del criptograma el identificador de usuario ID (receptor) que se ha añadido en la clave de sesión cifrada.
2. Se busca la clave privada del identificador ID en el anillo de claves privadas del receptor.
3. Se accede a la clave privada en claro, descifrándola con IDEA al introducir el propietario ID su frase de paso. Sólo en ese momento está en claro.
4. Con la clave privada se descifra la clave de sesión.
5. Con la clave de sesión se descifra el criptograma.

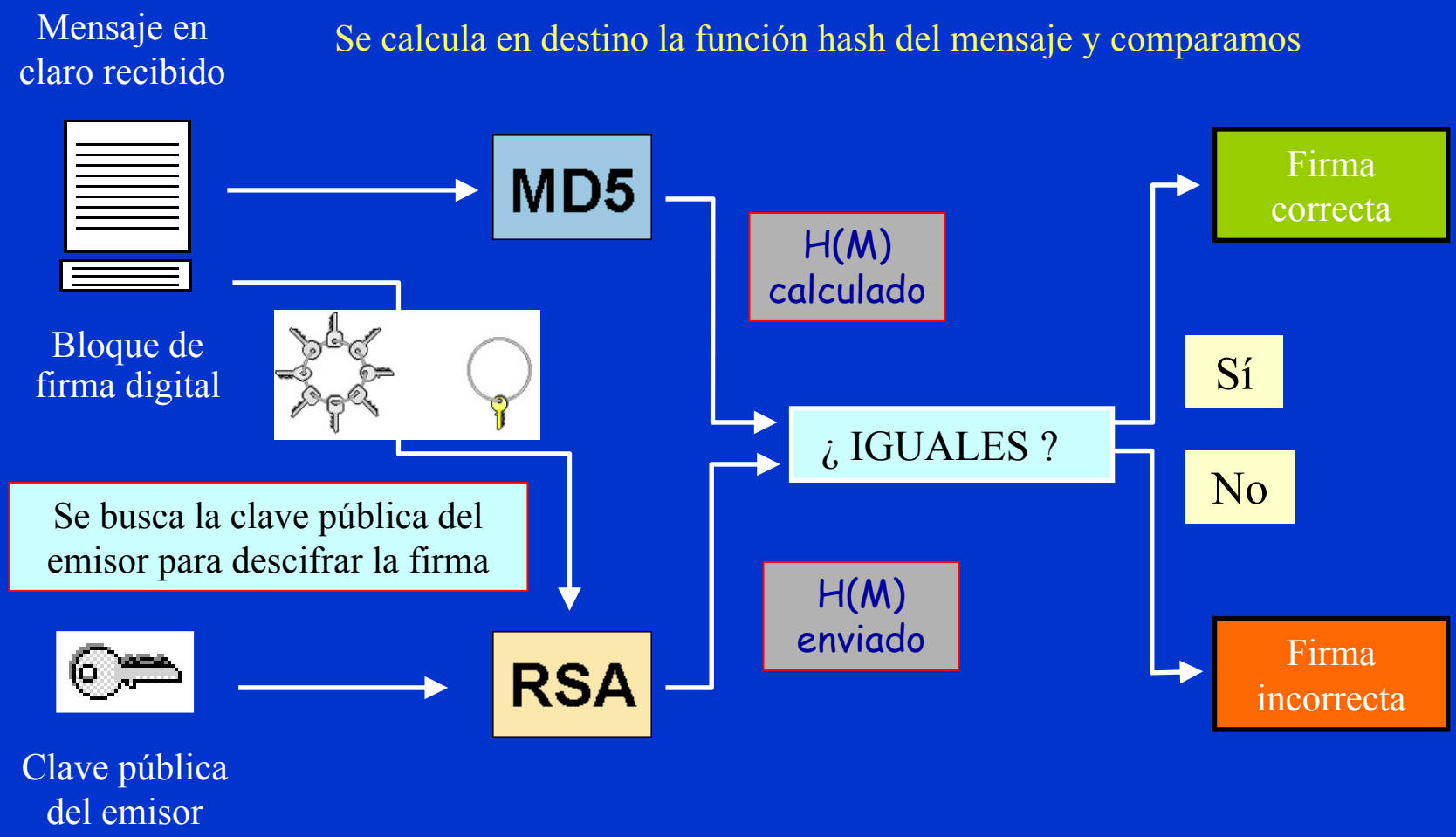
Descifrado con la clave privada de destino



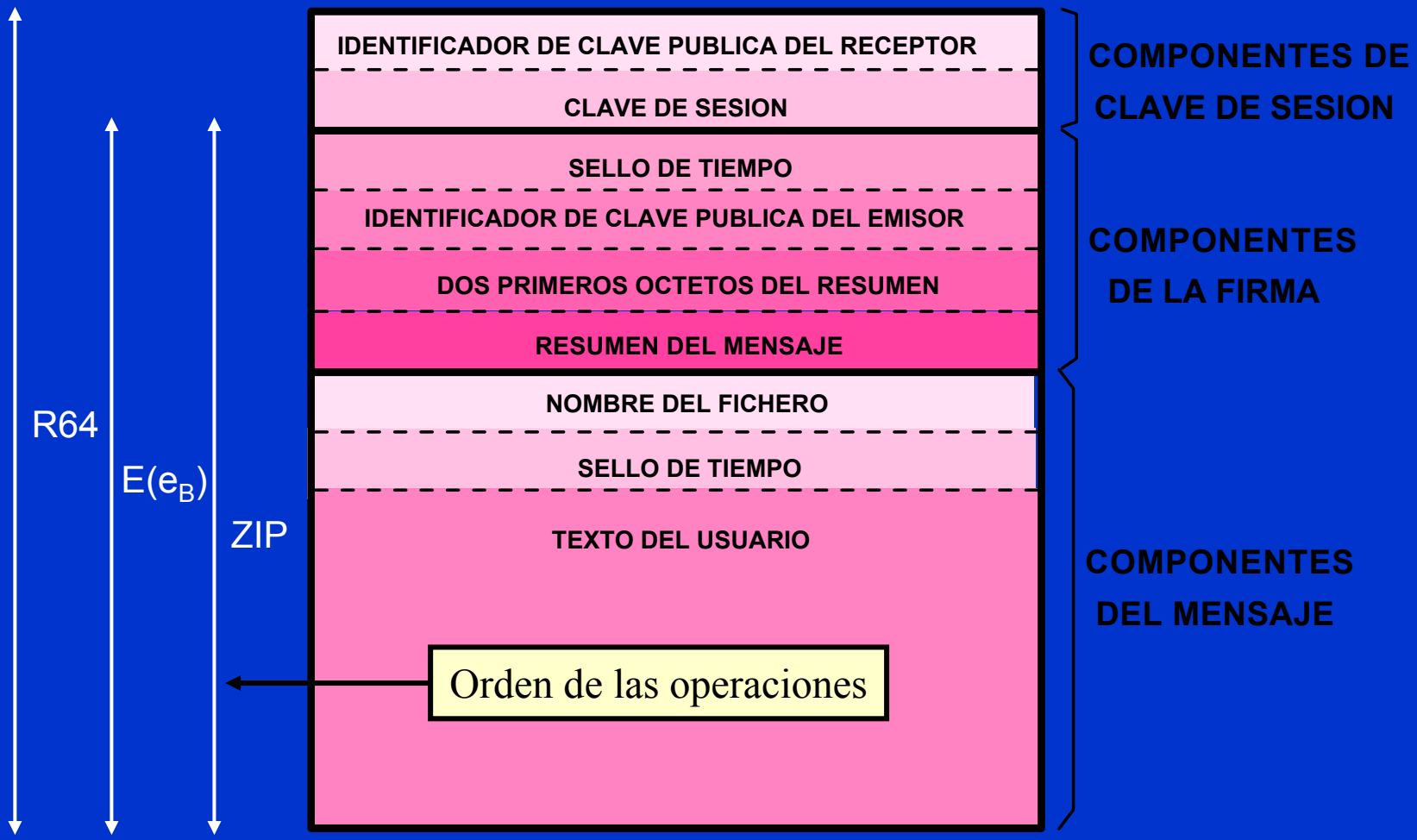
Firma digital RSA



Comprobación de la firma digital RSA



Formato de un mensaje PGP dirigido a B



Los tiempos cambian, pero ...

La mítica versión de PGP 2.6.3 del MIT se convierte rápidamente en el software de libre distribución freeware más popular en el mundo de los PCs y especialmente en entornos de correo electrónico: usa cifra y firma con criptografía calificada como fuerte.

Las versiones en entorno Windows a través de Network Associates presentan opciones avanzadas, servicios de red para seguimiento de paquetes y autenticación mediante Autoridades de Certificación.

Existe una versión freeware para usos no comerciales ☺.

Las versiones 5 y 6 tuvieron su código fuente abierto, en la 7 el código deja de ser público ☹ y la nueva versión 8.0 (diciembre 2002) ahora con PGP Corporation ha liberado otra vez el código.

¿Ha vuelto otra vez la cordura?

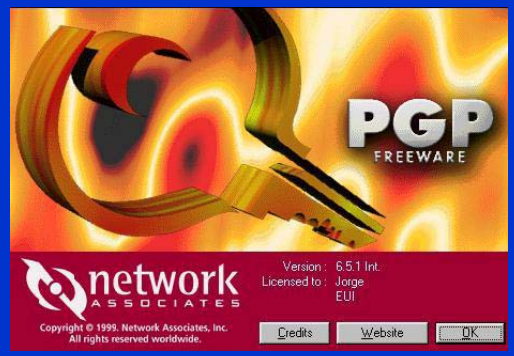
Aquí puede haber varias opiniones....

Algoritmos en nuevas versiones de PGP

- Generación de claves
  - RSA: 1.024, 1.536, 2.048 bits
  - Diffie y Hellman: 1.024, 1.536, 2.048, 3.072, 4.096 bits
- Firma digital
  - DSS Digital Signature Standard 1.024 bits
- Cifrado
  - CAST, IDEA, TripleDES, AES, Twofish
- Resumen
  - SHA-1 (160 bits) y MD5 (128 bits)

Algunas versiones de PGP en Windows

Desde la versión 5.0 hasta la actual 8.0 en el momento de escribir este libro (febrero de 2003) los esquemas de cifra y firma digital han cambiado muy poco aunque presentan mayores prestaciones. No obstante, recuerde que algunas prestaciones sólo estarán activadas en versiones comerciales.



PGP 6.5.1

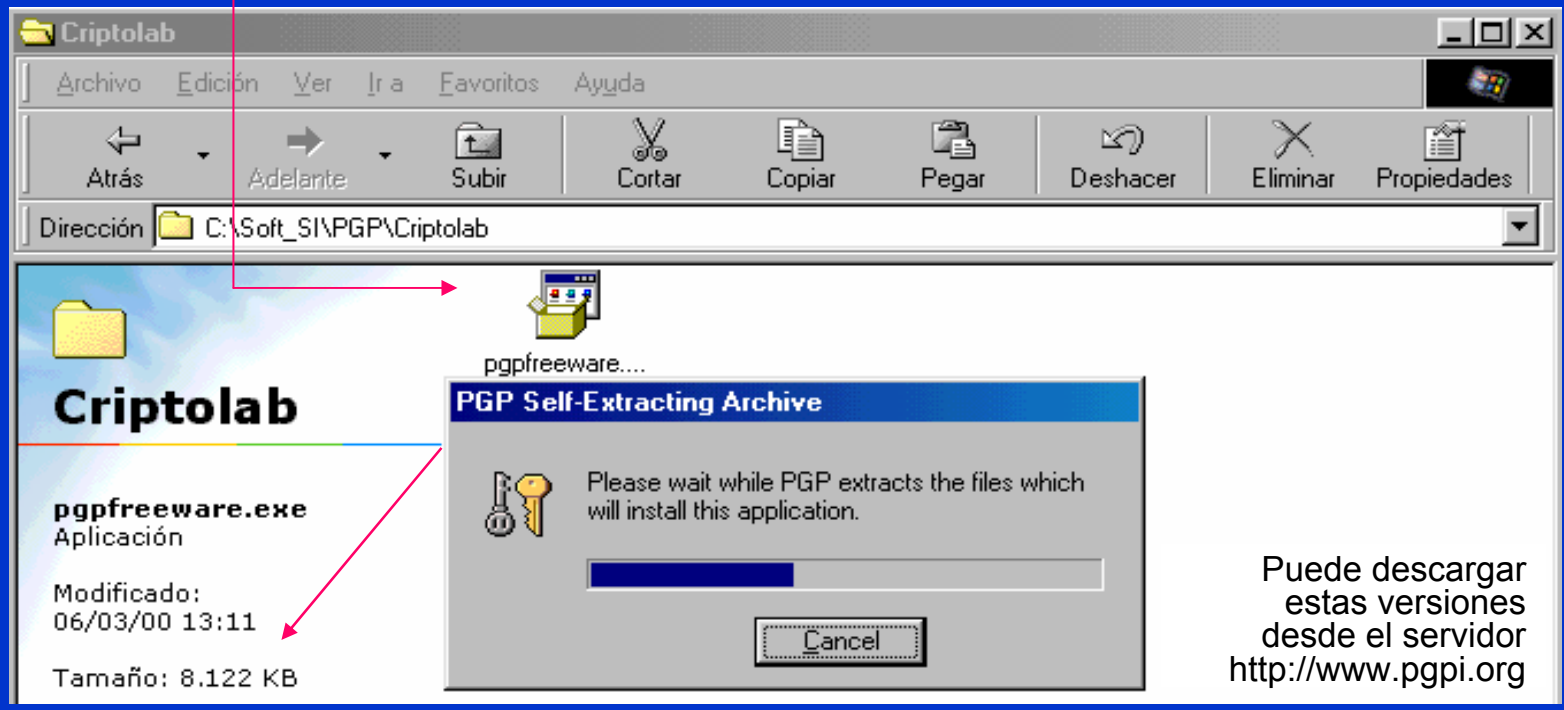
PGP 7.0.3

PGP 8.0

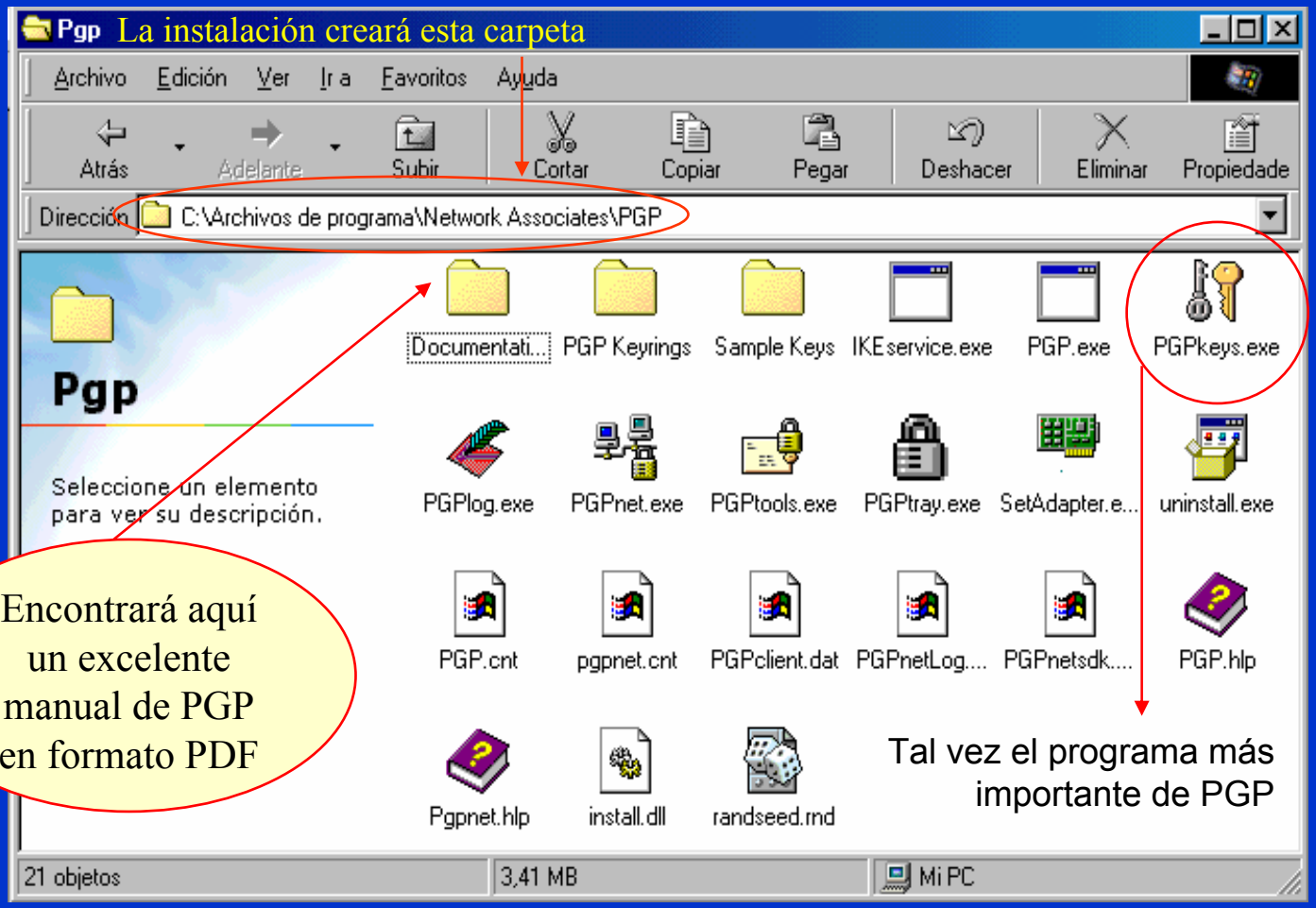
Veremos algunas operaciones de estas tres versiones con mayor detalle. Recuerde, eso sí, que la versión 7.0.3 no tiene su código fuente abierto.

Instalación de la versión PGP 6.5.1

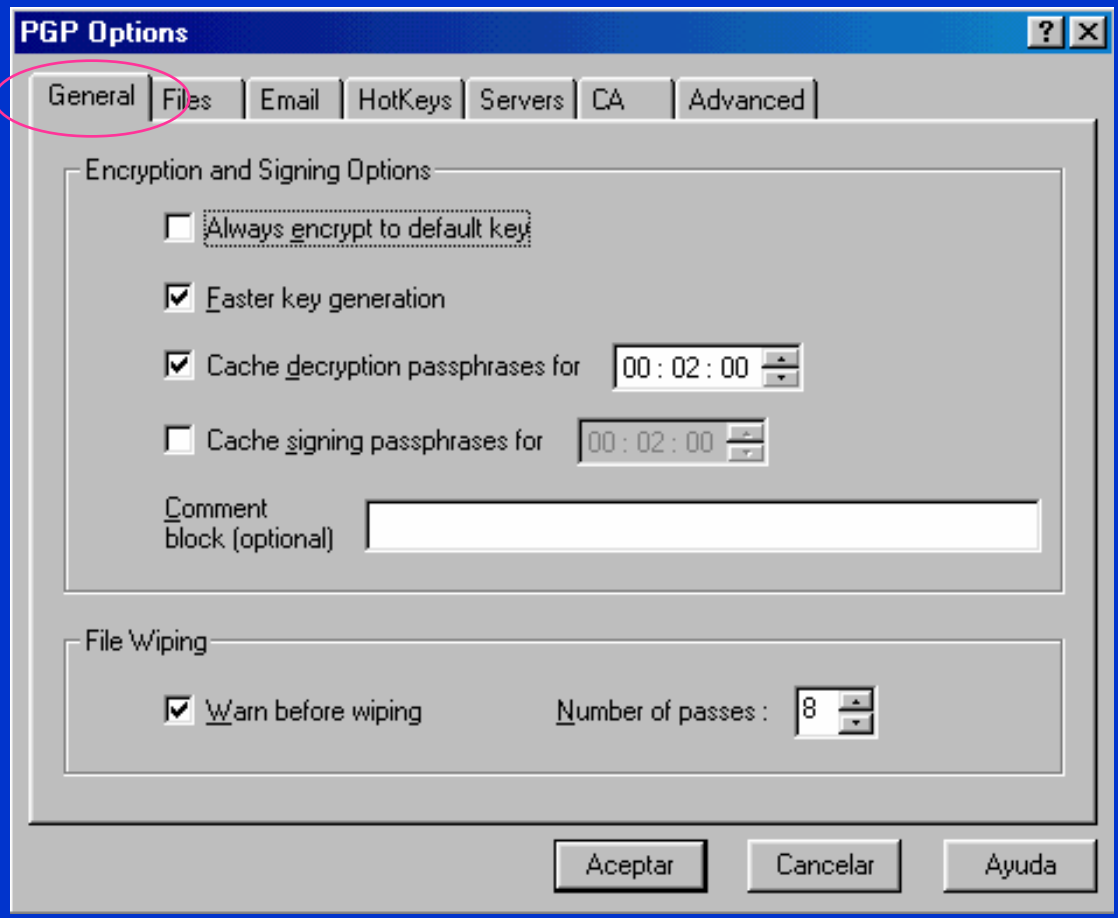
PGP 6.5.1 internacional aparece en el año 1999. Puede considerarse como una de las versiones seguras mejor optimizadas desde la primera en entorno Windows.



Carpetas y programas de PGP 6.5.1

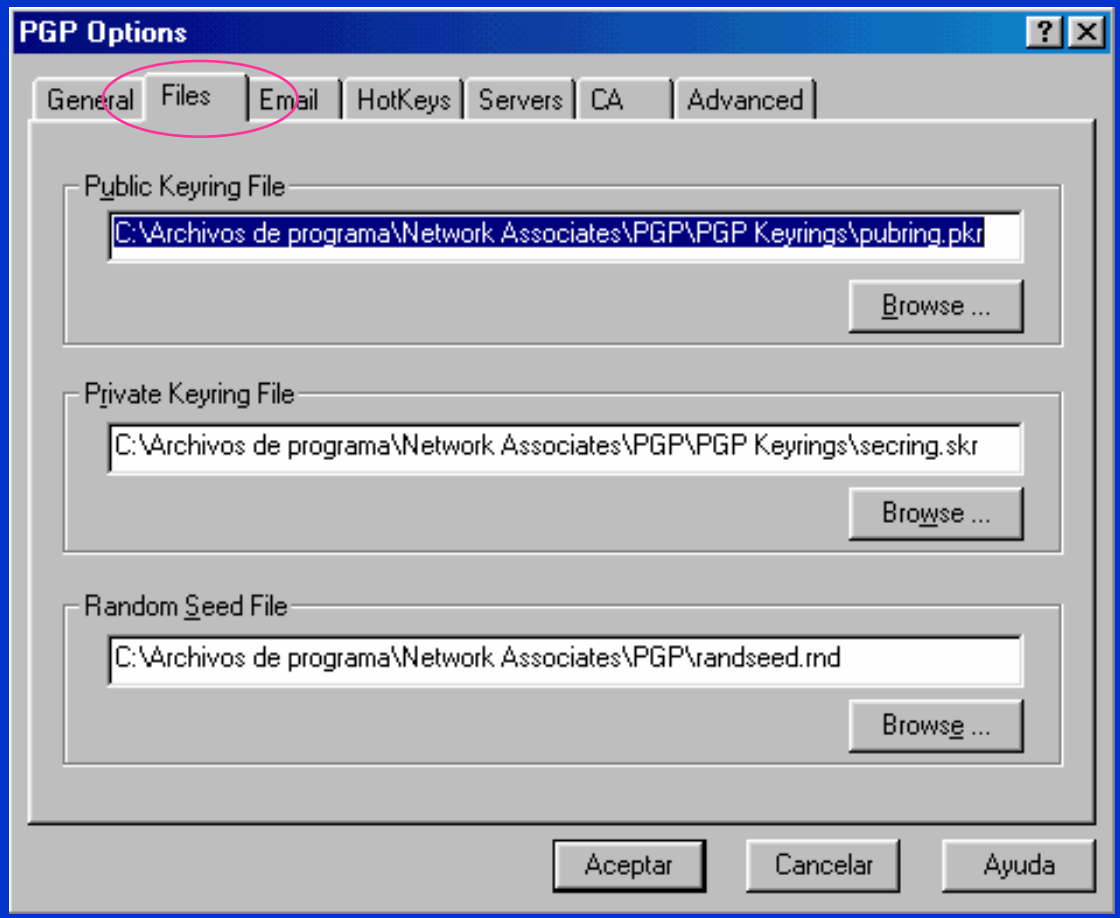


Opciones generales de PGP 6.5.1



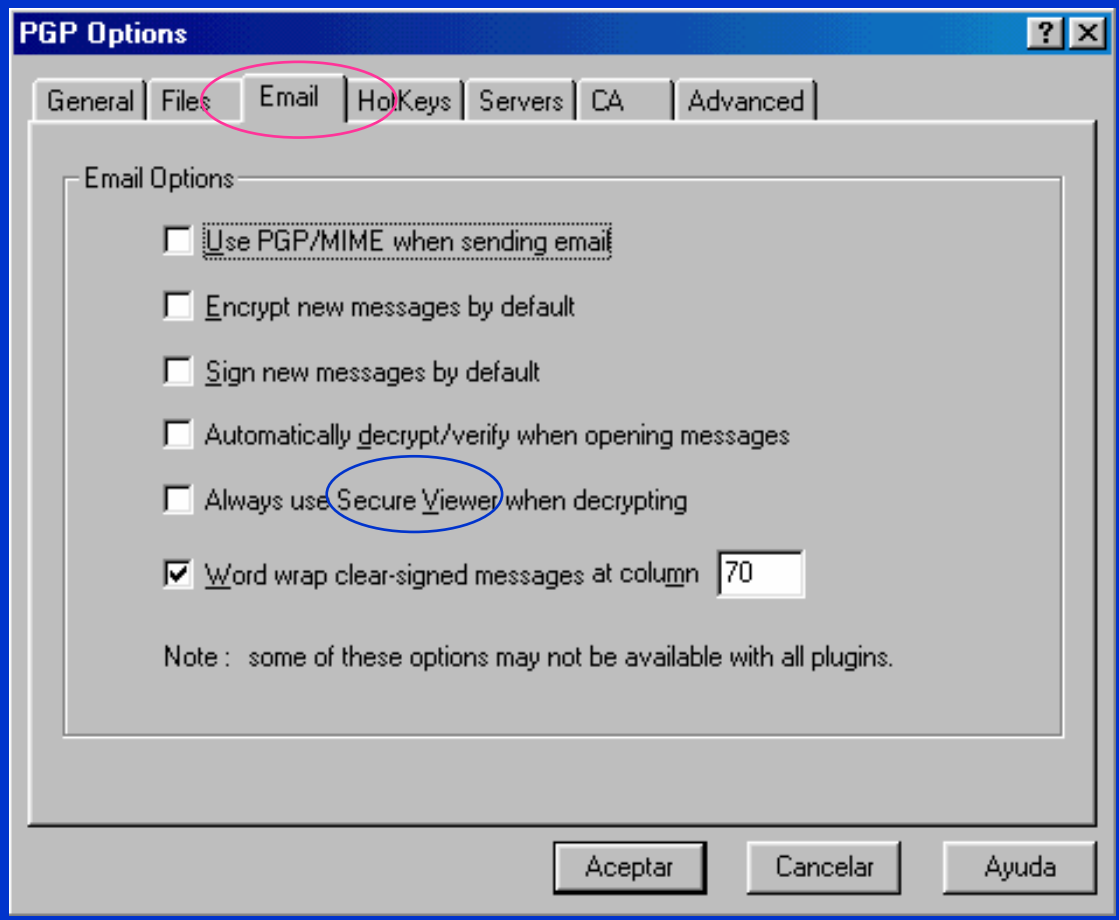
- ✓ La generación rápida de claves sólo puede hacerse para valores DH de una longitud predeterminada.
- ✓ Se puede limitar el tiempo de descifrado de la frase de paso en memoria caché.
- ✓ El borrado físico de datos y ficheros se hace escribiendo 1s y 0s aleatorios en los cluster, desde 8 hasta 32 veces.

Opciones de ficheros de PGP 6.5.1



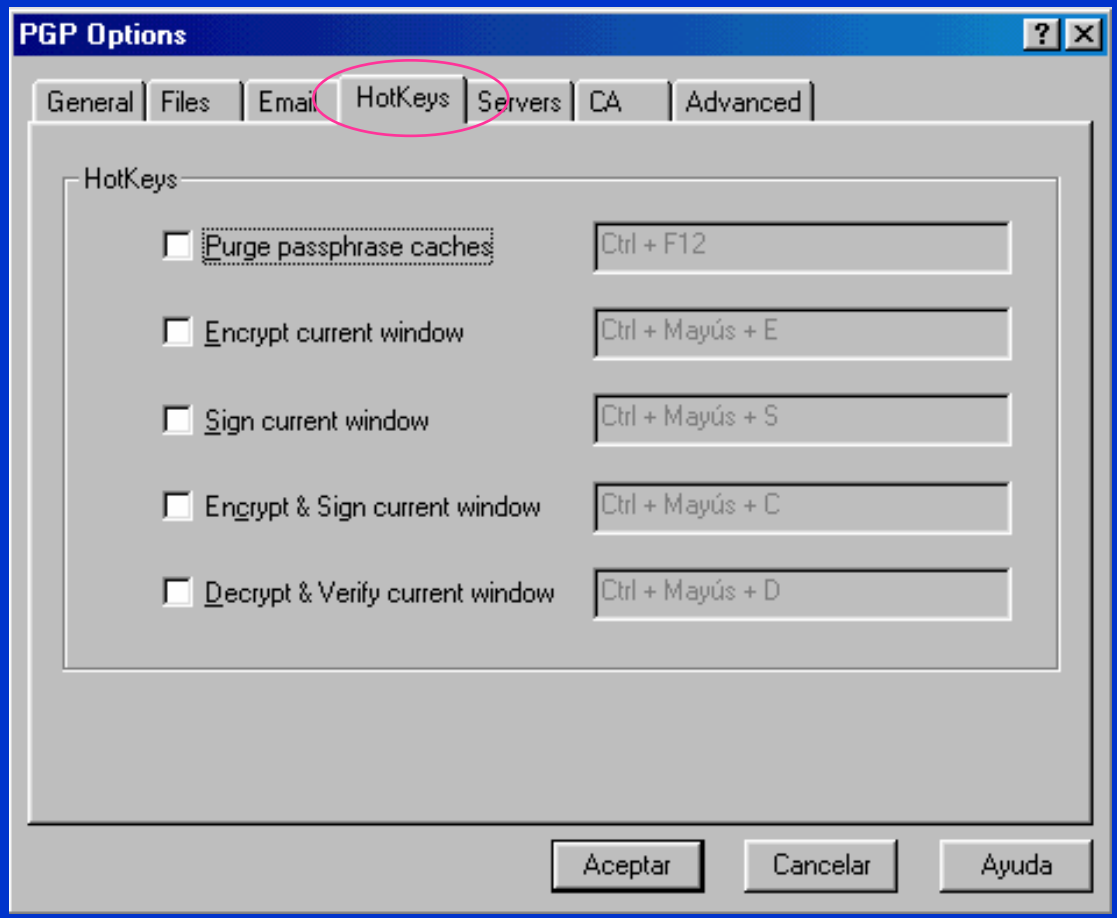
- ✓ Los archivos donde guarda las claves públicas y claves privadas siguen llamándose pubring y secring pero ahora, a diferencia de versiones anteriores, usa como extensiones pkr.
- ✓ El archivo de semilla permite generar números aleatorios para crear claves.

Opciones de e-mail de PGP 6.5.1



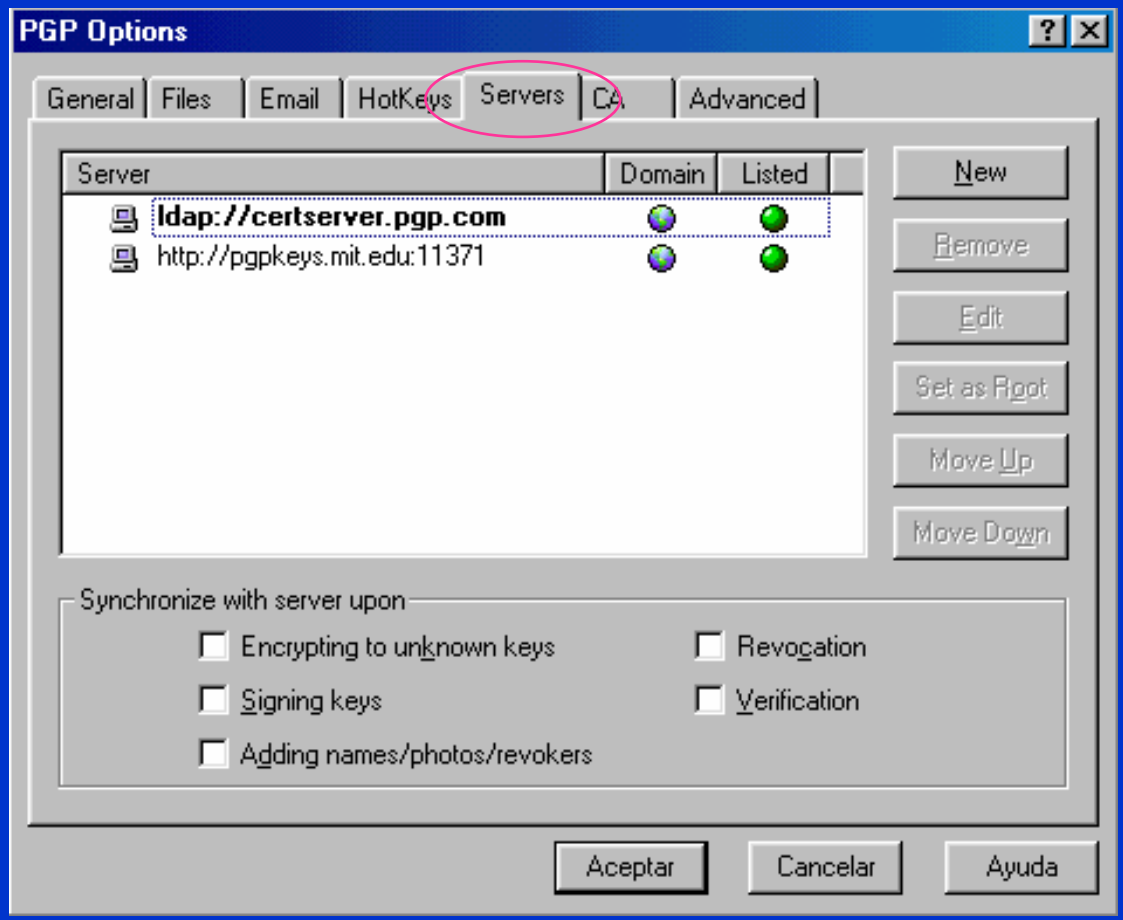
- ✓ El PGP/MIME sólo funciona con plugins.
- ✓ Se puede pedir que cifre, firme o descifre y compruebe firma por defecto al enviar o abrir mensajes.
- ✓ Si usa Secure Viewer, al descifrar un archivo éste sólo se muestra en la pantalla usando para ello una técnica de enmascarado que evita los ataques por captura de radiofrecuencias del teclado, TEMPEST.

Opciones de atajos de PGP 6.5.1



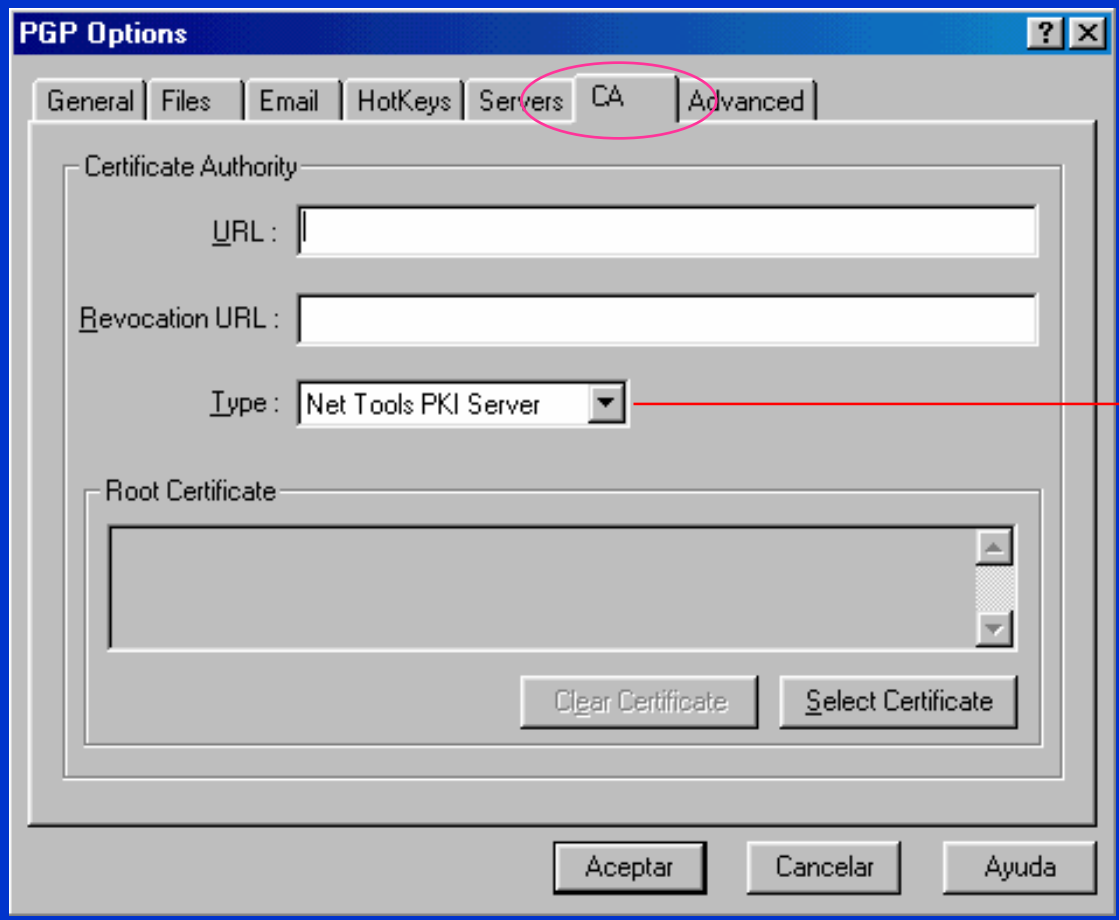
- ✓ La opción de usar teclas para atajos es poco interesante pero puede activarse si se desea.

Opciones de servidores de PGP 6.5.1



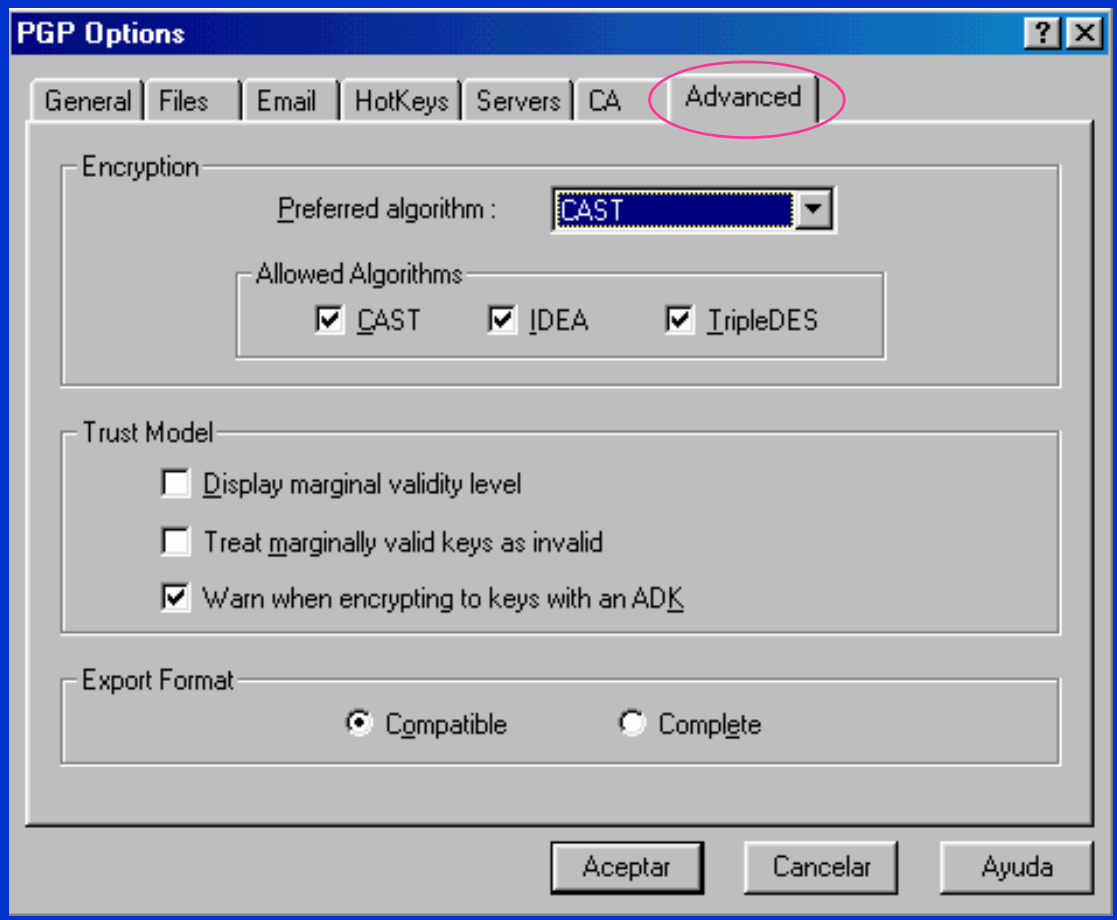
- ✓ A estos servidores se puede enviar nuestra clave pública para que los usuarios accedan más fácilmente a ella.
- ✓ Es interesante estar sincronizado con el servidor por el tema de las claves revocadas.

Opciones de ACs de PGP 6.5.1



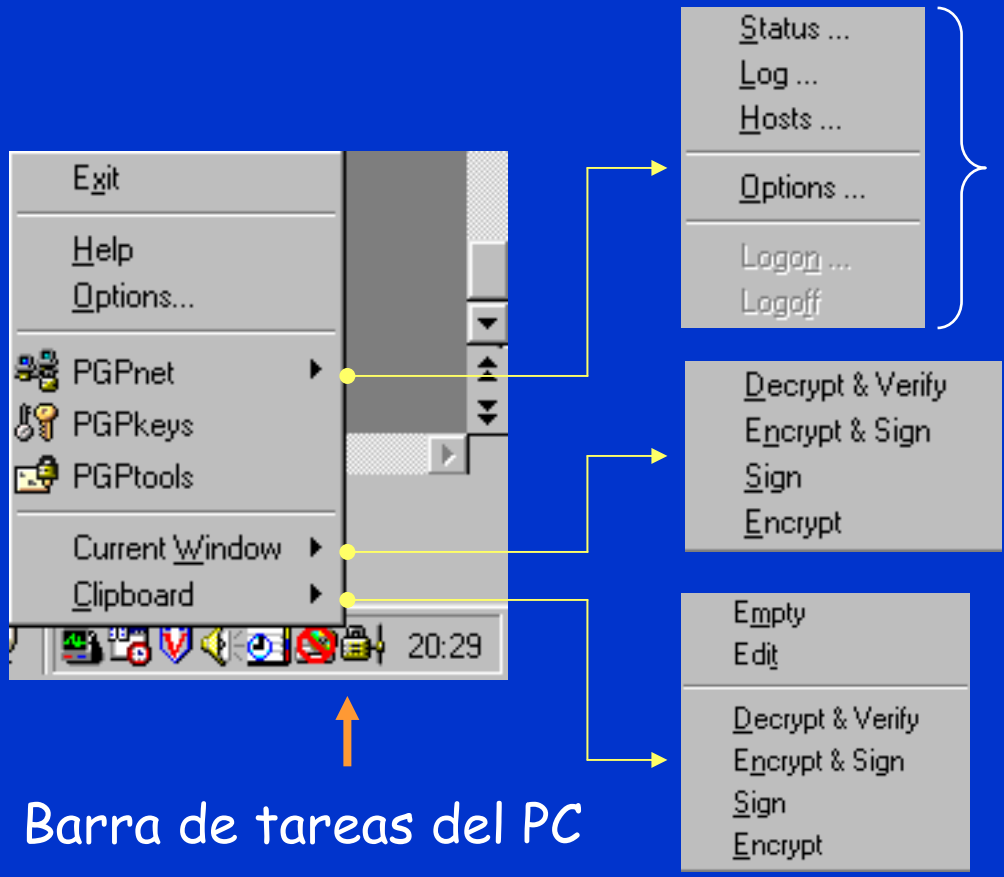
También VeriSign OnSite y Entrust

Opciones avanzadas de PGP 6.5.1



- ✓ IDEA era en versiones anteriores el algoritmo de cifra por defecto.
- ✓ El aviso sobre uso de Additional Decryption Key (ADK) significa que el administrador del sistema puede usar una clave extra que le permita descifrar lo cifrado, en caso de necesidad o por un requerimiento judicial.
- ✓ El formato compatible es el código base 64.

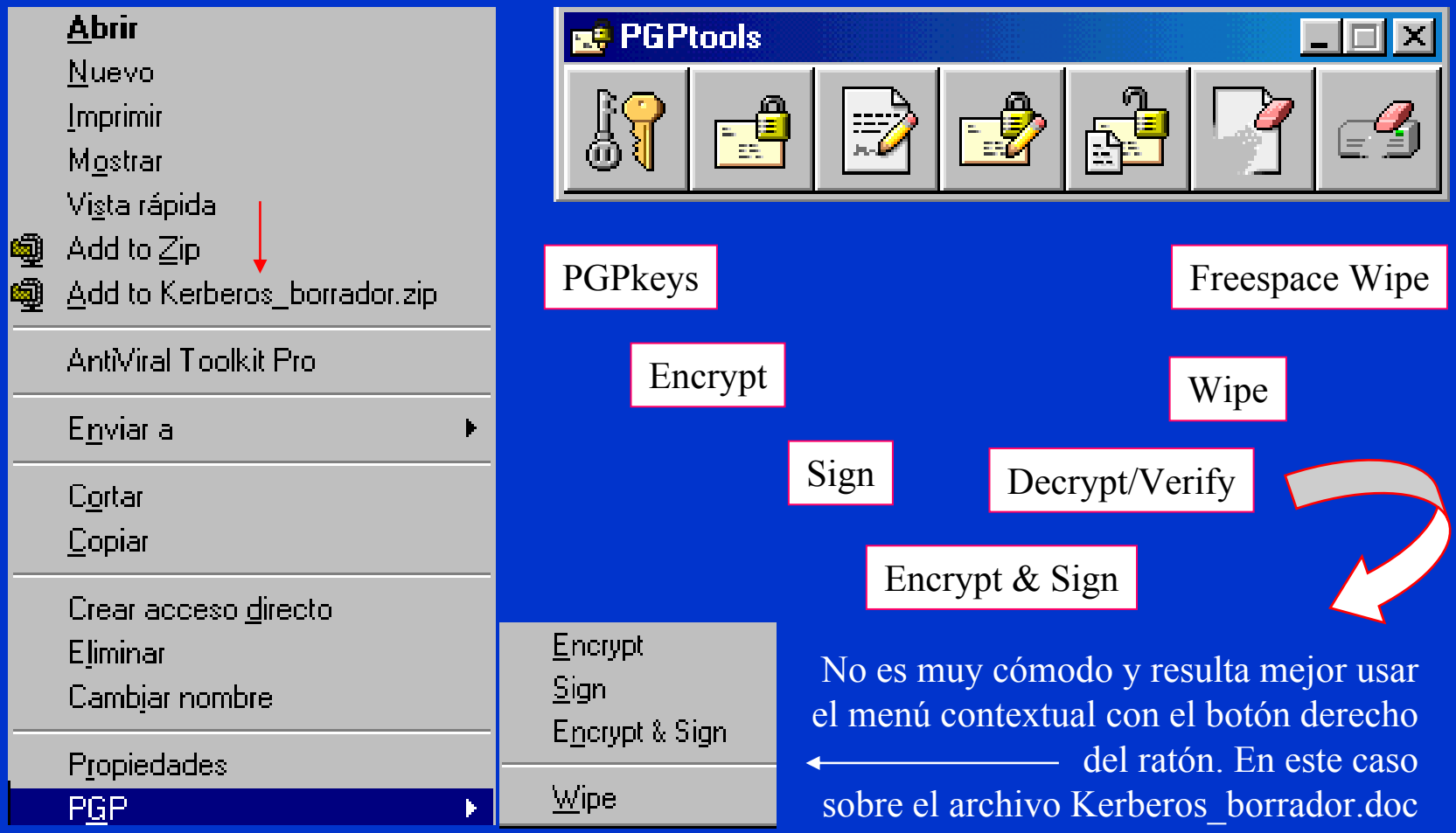
El programa PGPtray de acceso directo



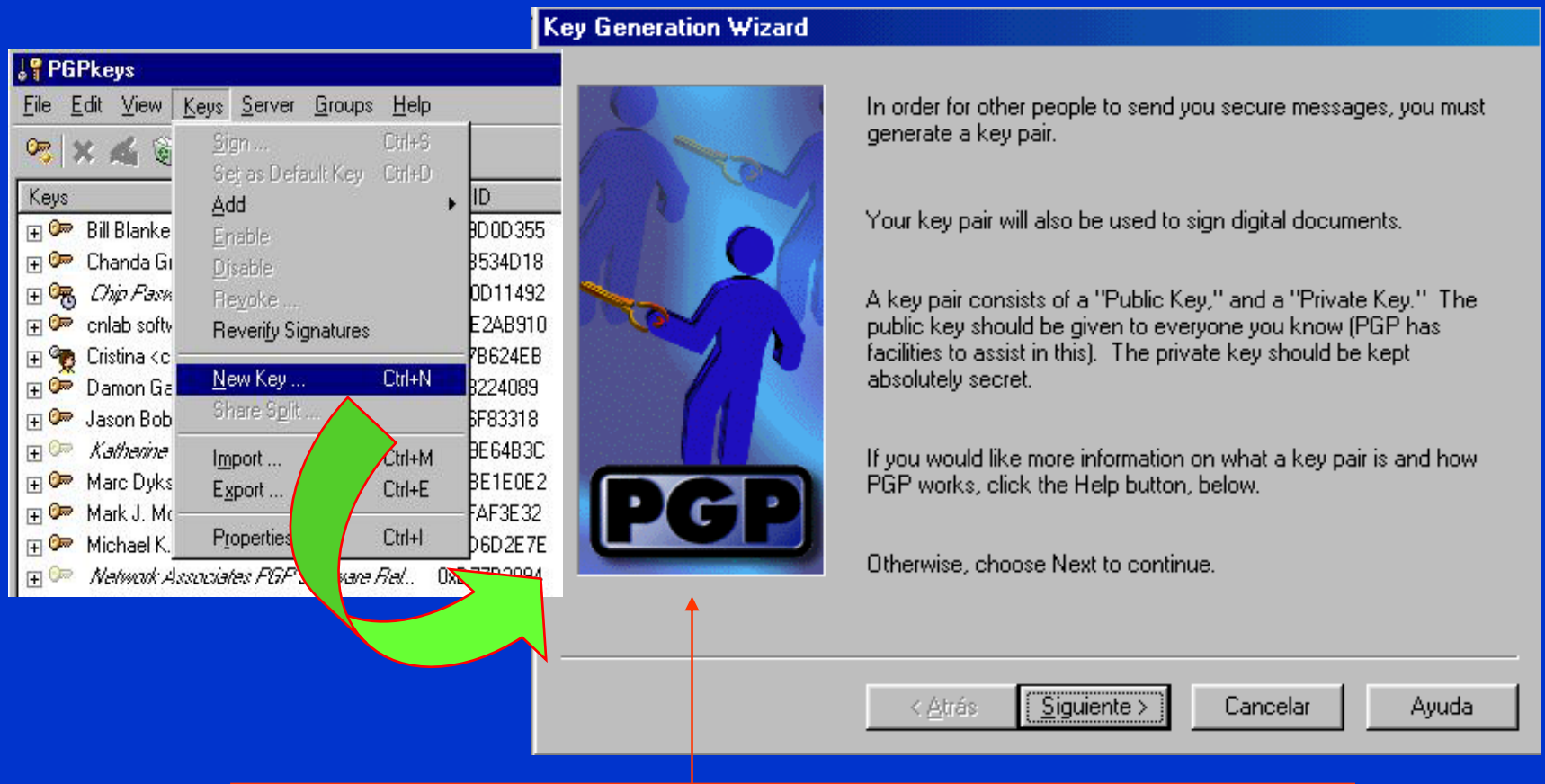
Si PGPnet está instalado, en la barra de tareas, al lado del candado de PGPtray, nos aparecerá un icono vertical como se ve en la figura.

- ✓ La ventana actual cifra texto y no documentos con formato.
- ✓ El uso de portapapeles es la solución si no tenemos el plugin para correo electrónico.

Barra flotante de PGPtools

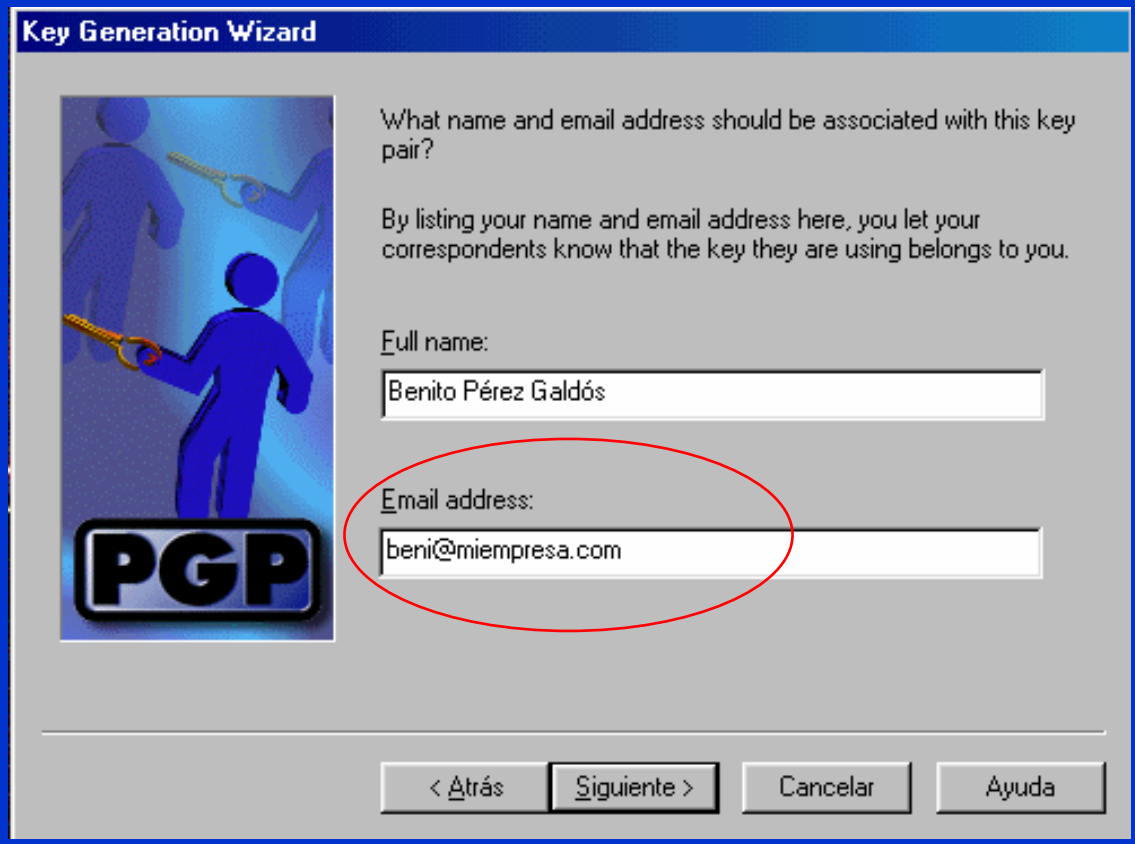


Generación de claves con PGPkeys 6.5.1



Esta es la primera pantalla que aparece una vez instalado PGP.

Nombre del usuario y su correo



- ✓ No es necesario que la dirección de correo sea la real. No obstante sirve para los que se comunican con nosotros sepan que esa clave pertenece a esa dirección de email.

Elección del tipo de clave asimétrica

- ✓ El estándar para la generación de las claves asimétricas es en la actualidad Diffie y Hellman junto con la Digital Signature Standard, y se representará como DH/DSS.
- ✓ También puede usar claves RSA que son compatibles con las versiones anteriores de PGP.

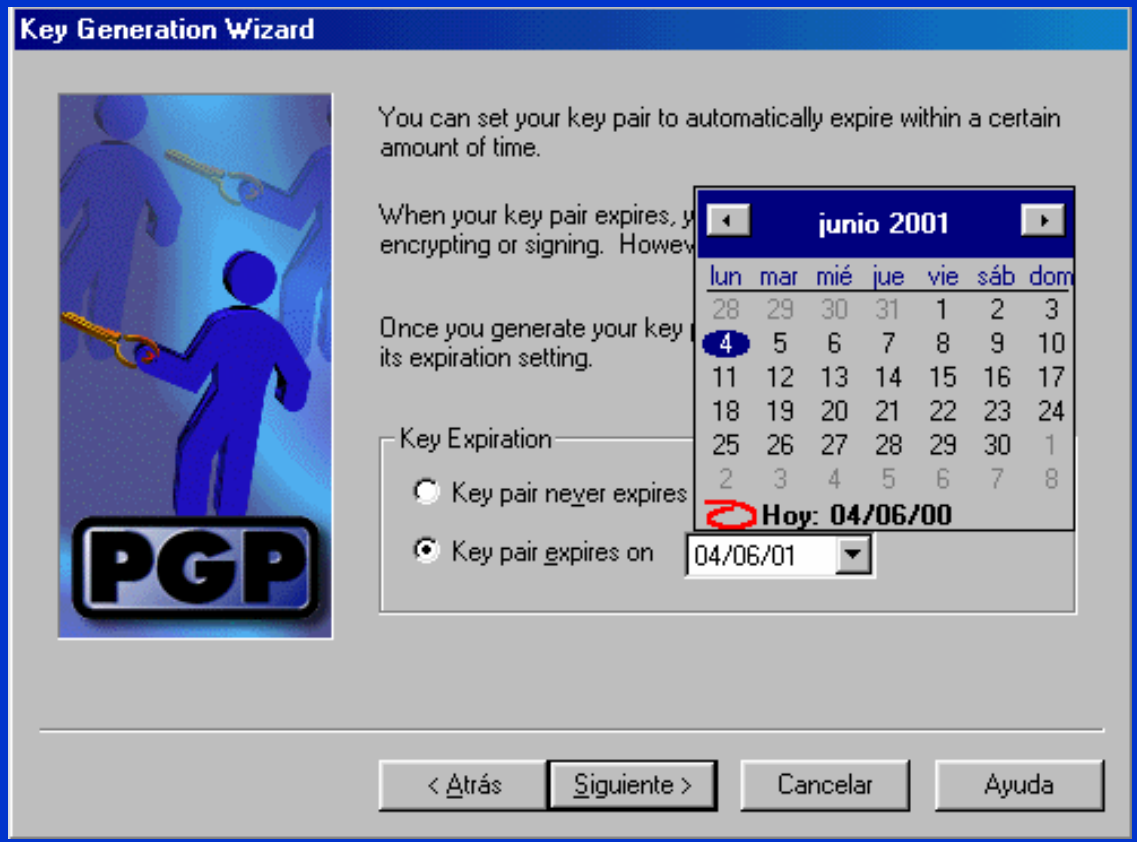
Elección de la longitud de la clave

- ✓ Si genera claves de longitud distinta a los valores que se proponen, puede tardar bastante tiempo generarlas. Algo similar sucede si no usa la opción generación rápida de claves.
- ✓ Es recomendable que use una clave de 2.048 bits. Esta se generará sólo en esta fase.

Clave sin caducidad

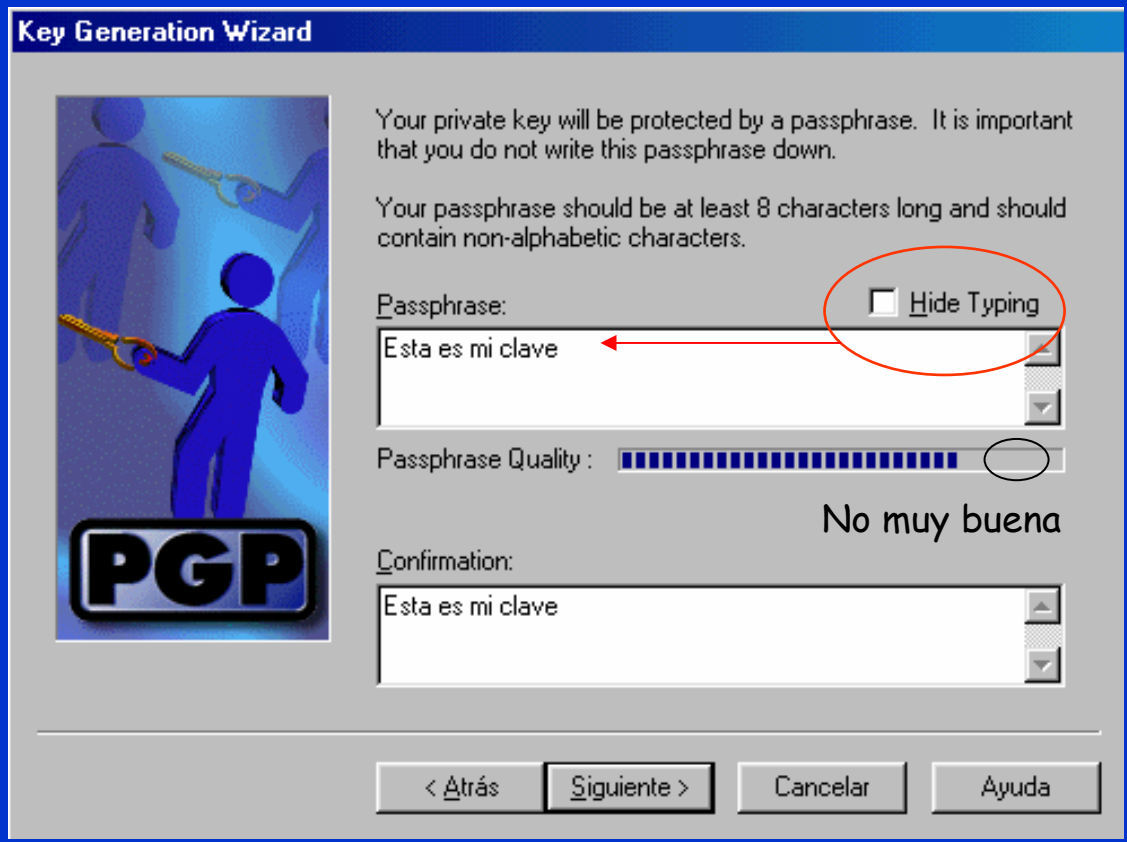
- ✓ Puede optar porque su clave no caduque nunca eligiendo esa opción.

Clave con caducidad



- ✓ Puede optar que la clave caduque de acuerdo con un calendario que nos muestra PGP. En el ejemplo la clave era válida desde el 4 de junio de 2000 al 4 de junio de 2001.

Frase de paso para cifrar la clave privada



- ✓ La frase de paso debe tener varias palabras para que sea difícil un ataque por diccionario.
- ✓ Si quita la opción Hide Typing podrá ver lo que escribe.
- ✓ Por seguridad, no está permitido usar el portapapeles para copiar la frase de arriba en la casilla de confirmación.

Generación de los números primos

- ✓ PGP genera dos primos tanto para las claves RSA (los valores p y q) como para DH/DSS, en este caso el primo p para el intercambio de clave y el primo q para la firma DSS.
- ✓ Normalmente tarda pocos segundos si se eligen los valores estándar que nos propone PGP.

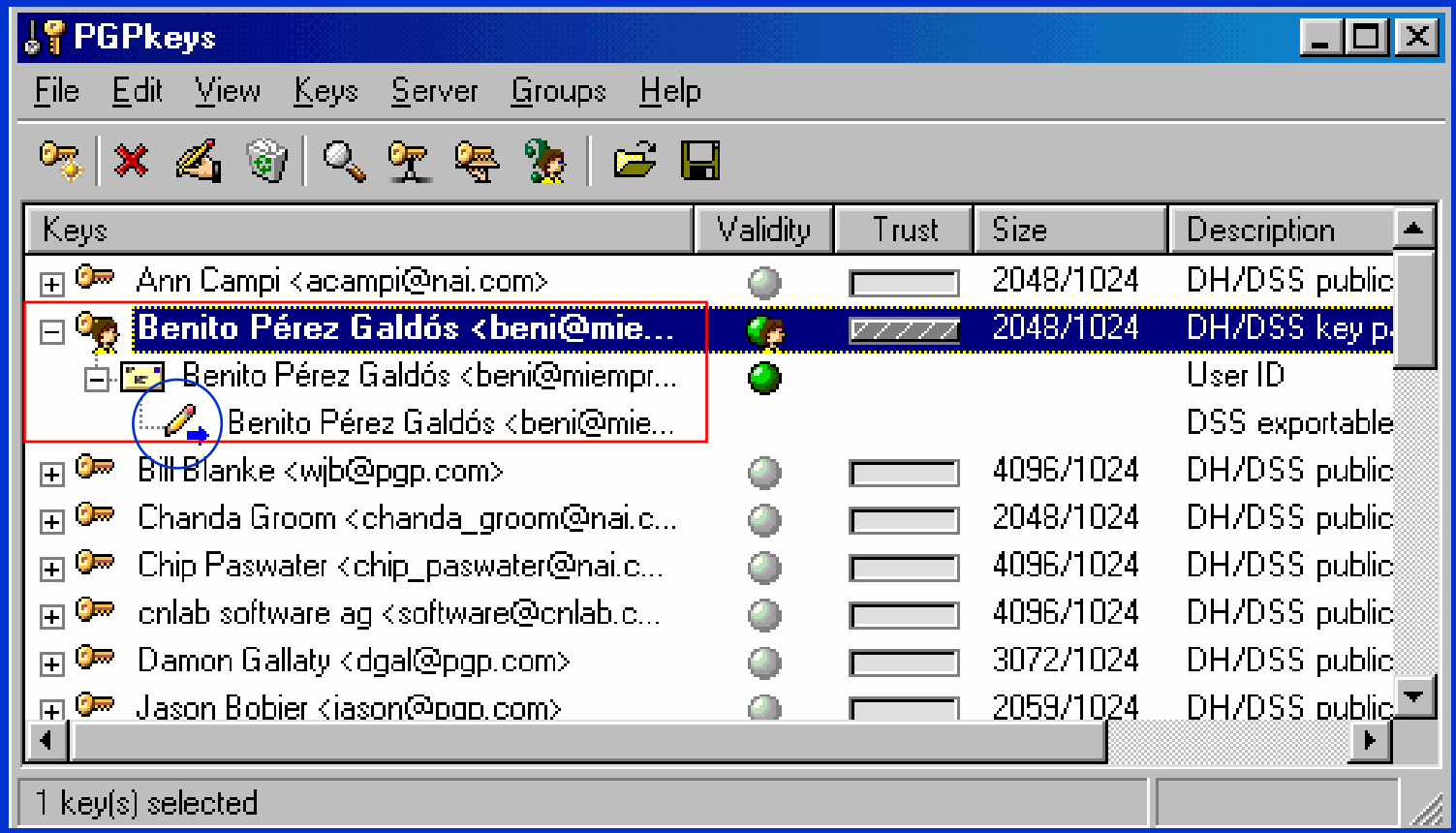
Envío de la clave al servidor de claves

- ✓ Se puede enviar la clave a un servidor.

Generación de clave concluida

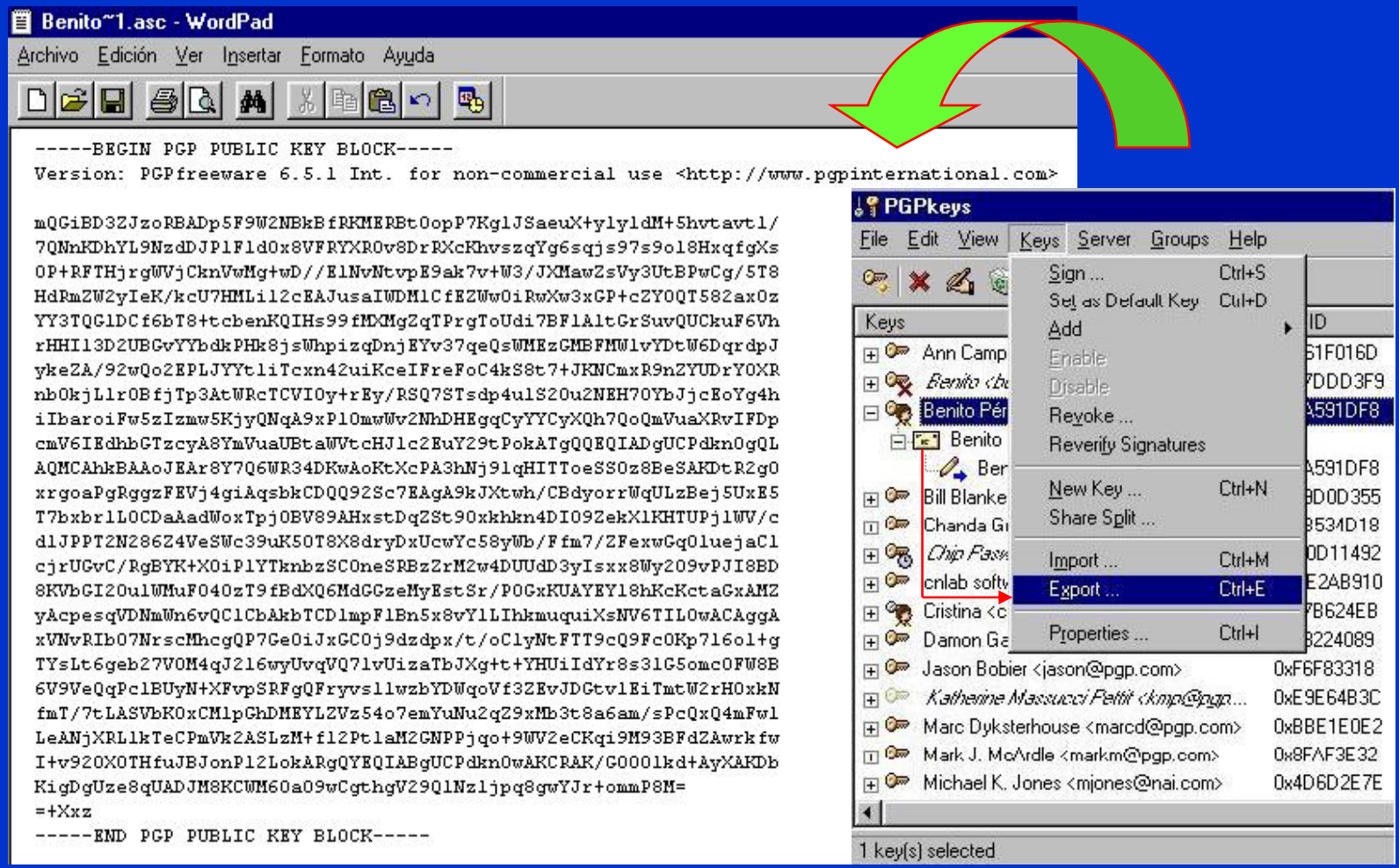
- ✓ Se ha concluido satisfactoriamente la generación del par de claves pública y privada que se guardarán en los anillos pubring.pkr y secring.pkr.
- ✓ La clave privada se guarda cifrada con una clave de sesión que se genera al aplicar una función hash a la frase de paso del propietario.

Visualización de la clave de Benito

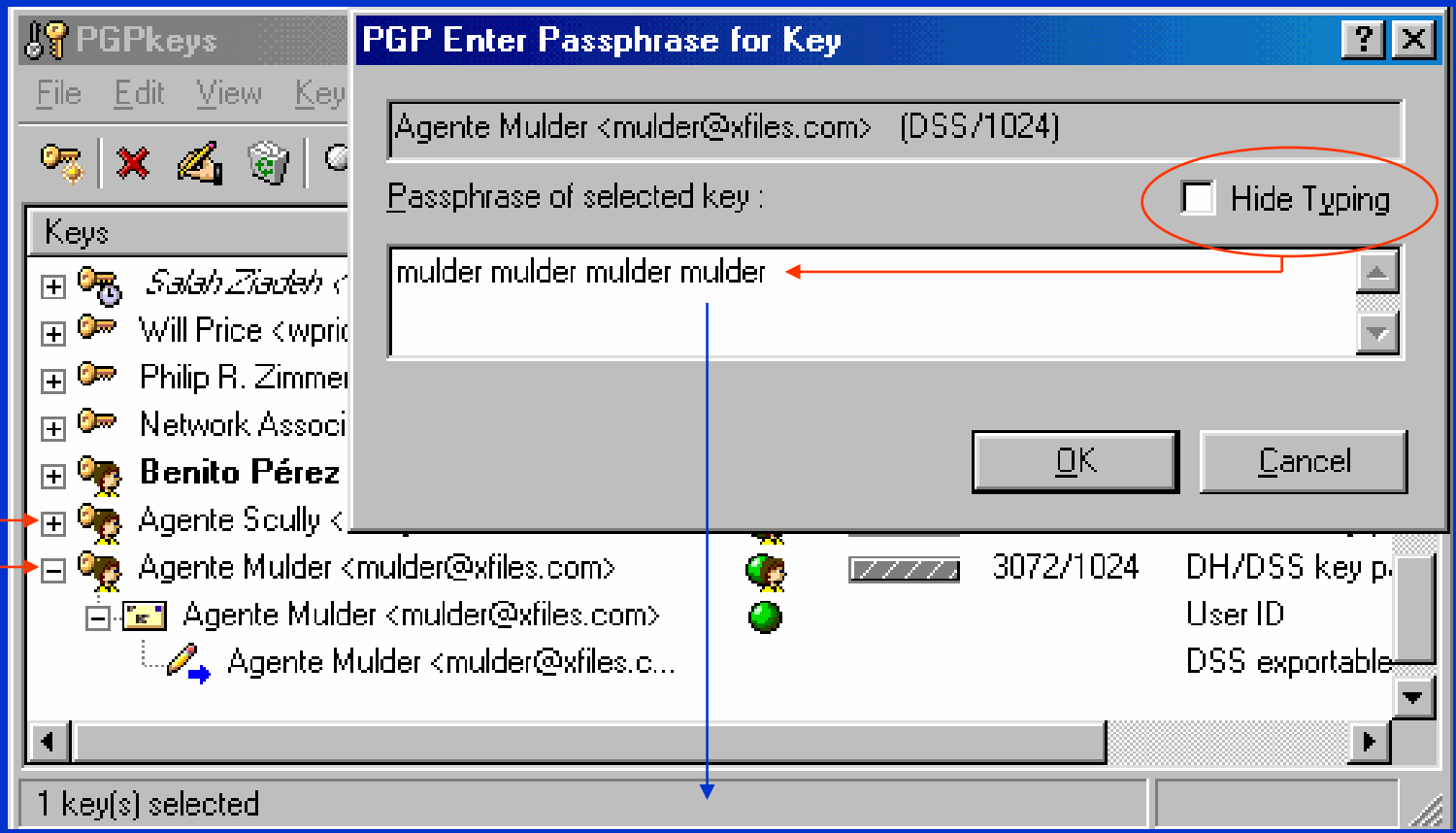


Por defecto, el propietario se firma la clave pública con su clave privada.

Exportación de la clave pública de Benito

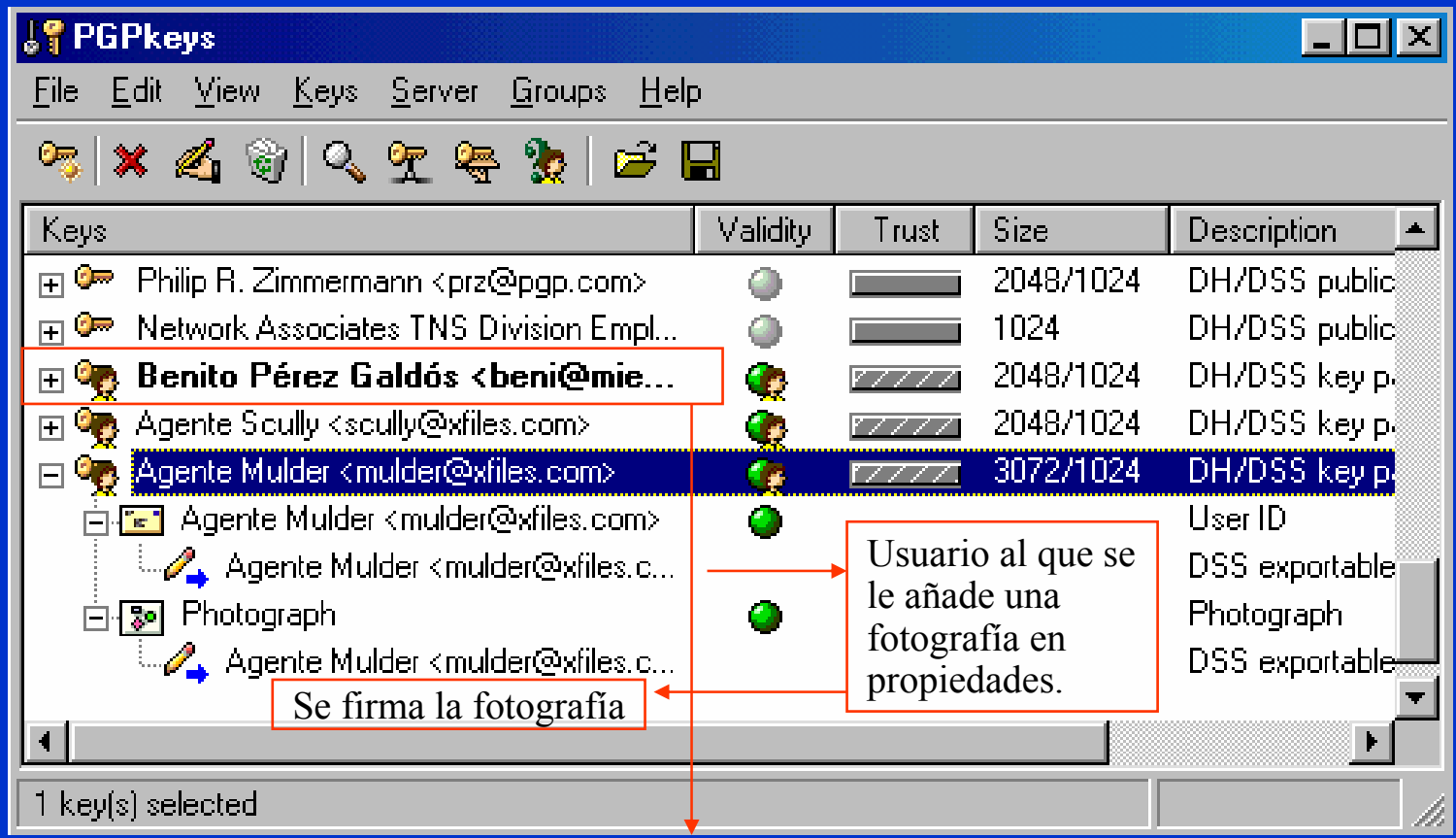


Claves de dos usuarios para un ejemplo



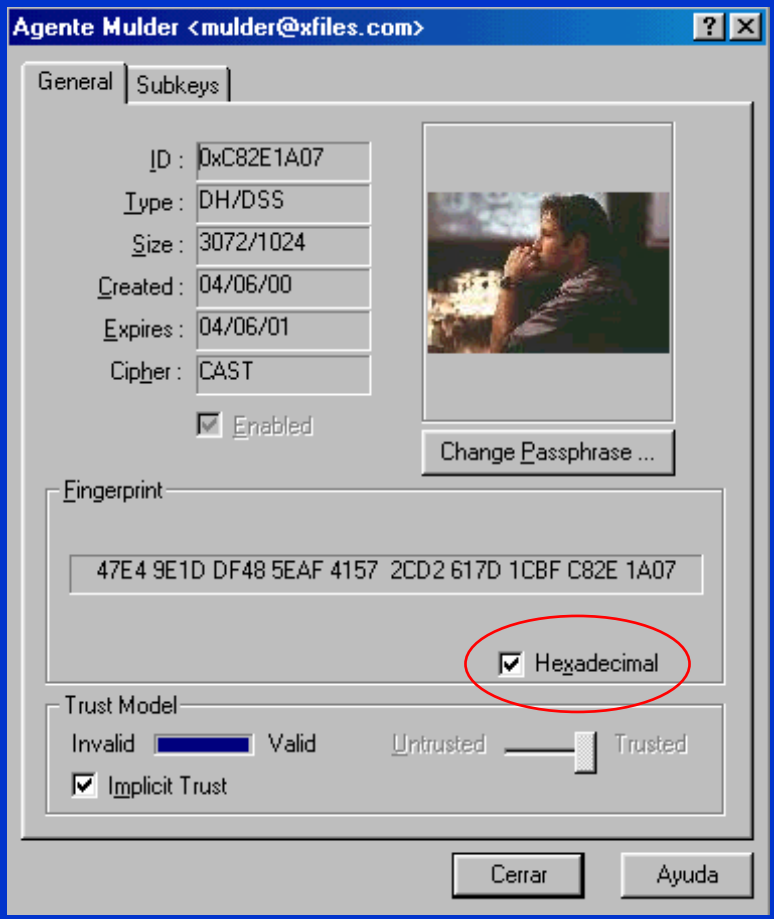
Frase de paso para descifrar la clave privada de este usuario

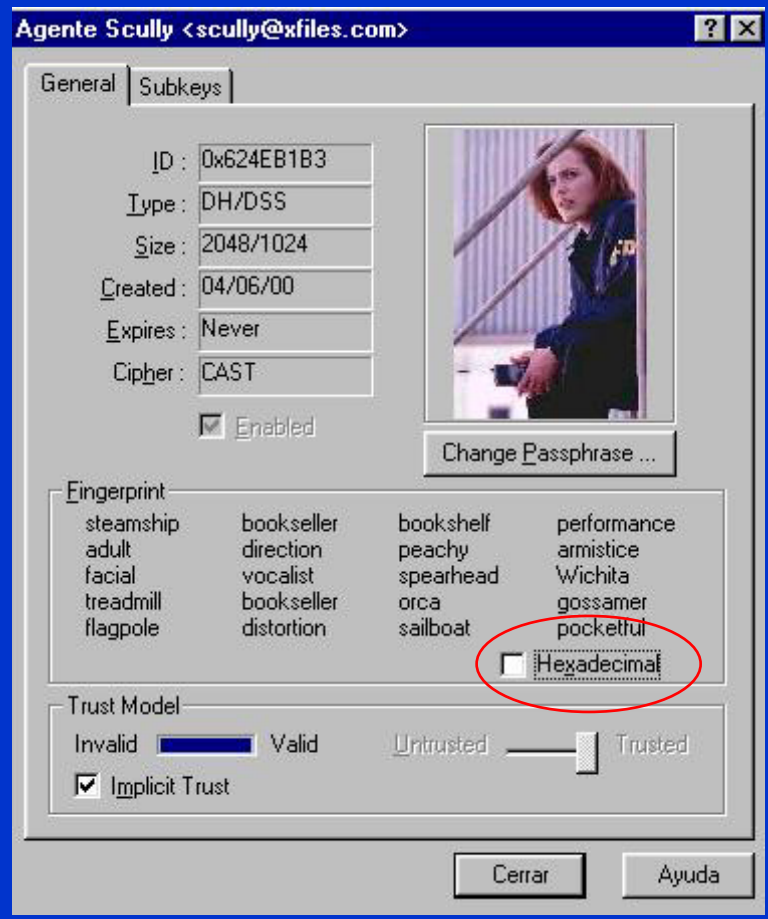
Inclusión de fotografía en clave pública



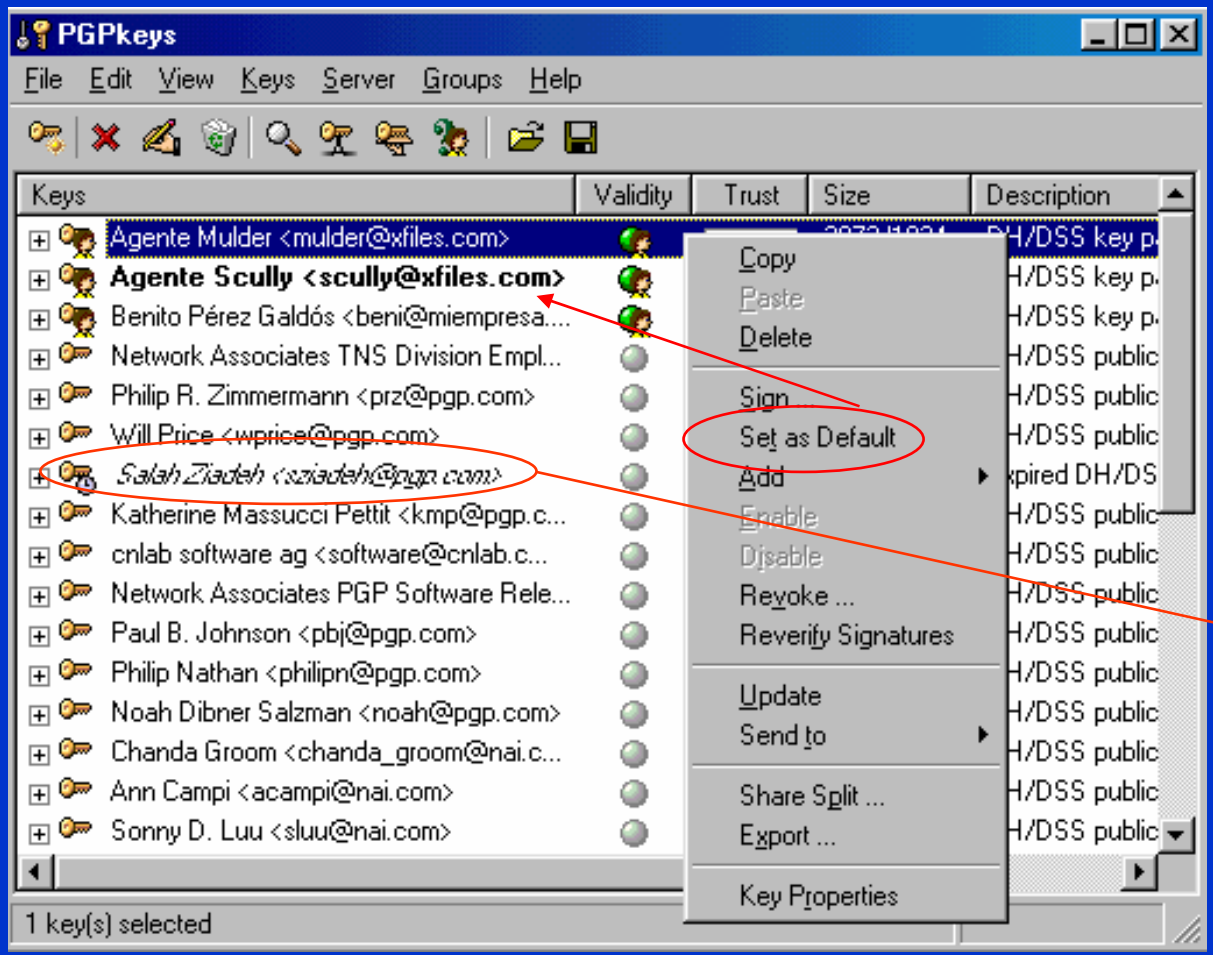
En negrita el usuario por defecto

Las claves públicas del ejemplo





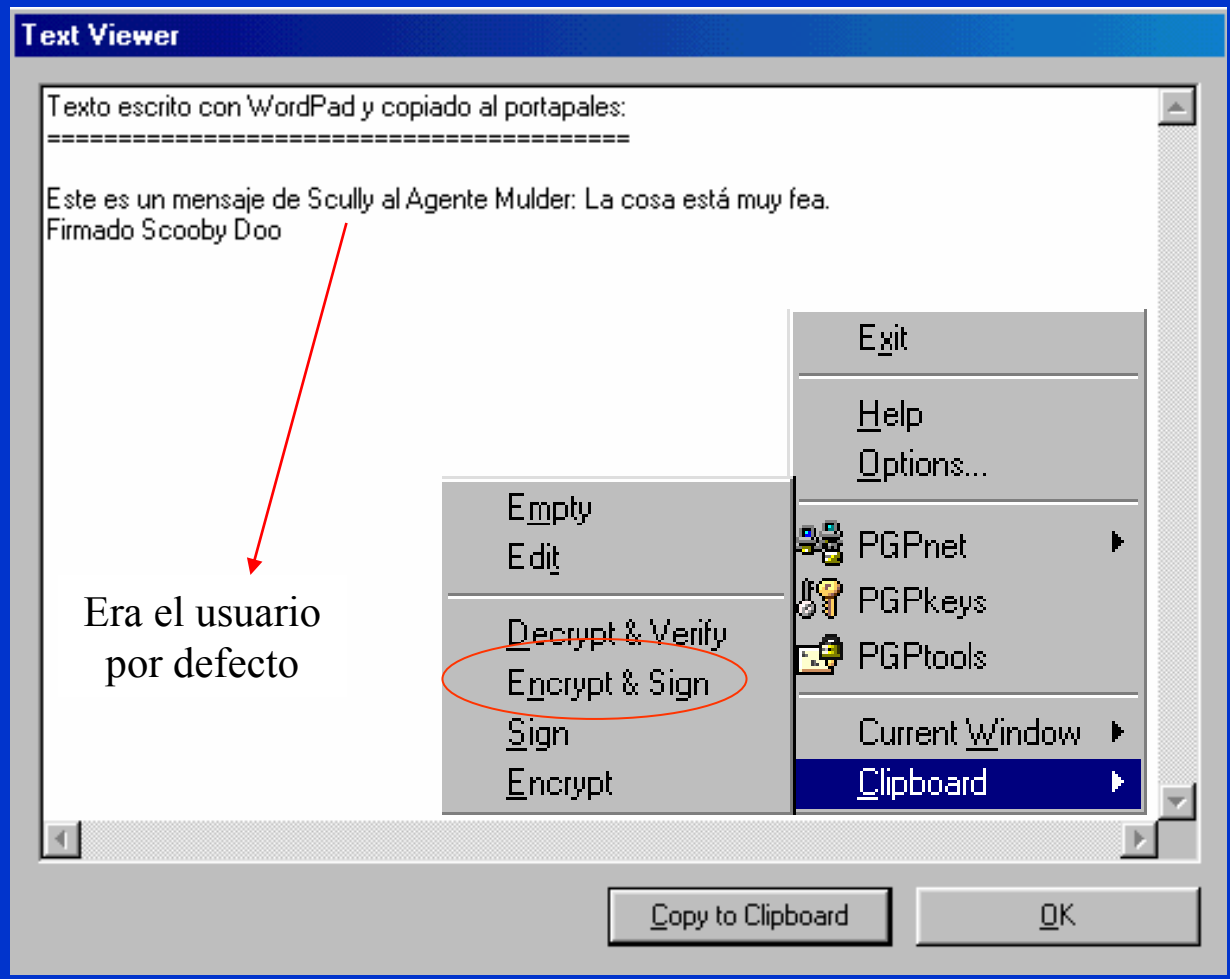
Definir un usuario por defecto



El usuario en negrita es aquel que se ha definido por defecto.

En cursiva: usuario cuya clave ha sido revocada o bien ha caducado su validez.

Ejemplo de cifra y firma con portapapeles

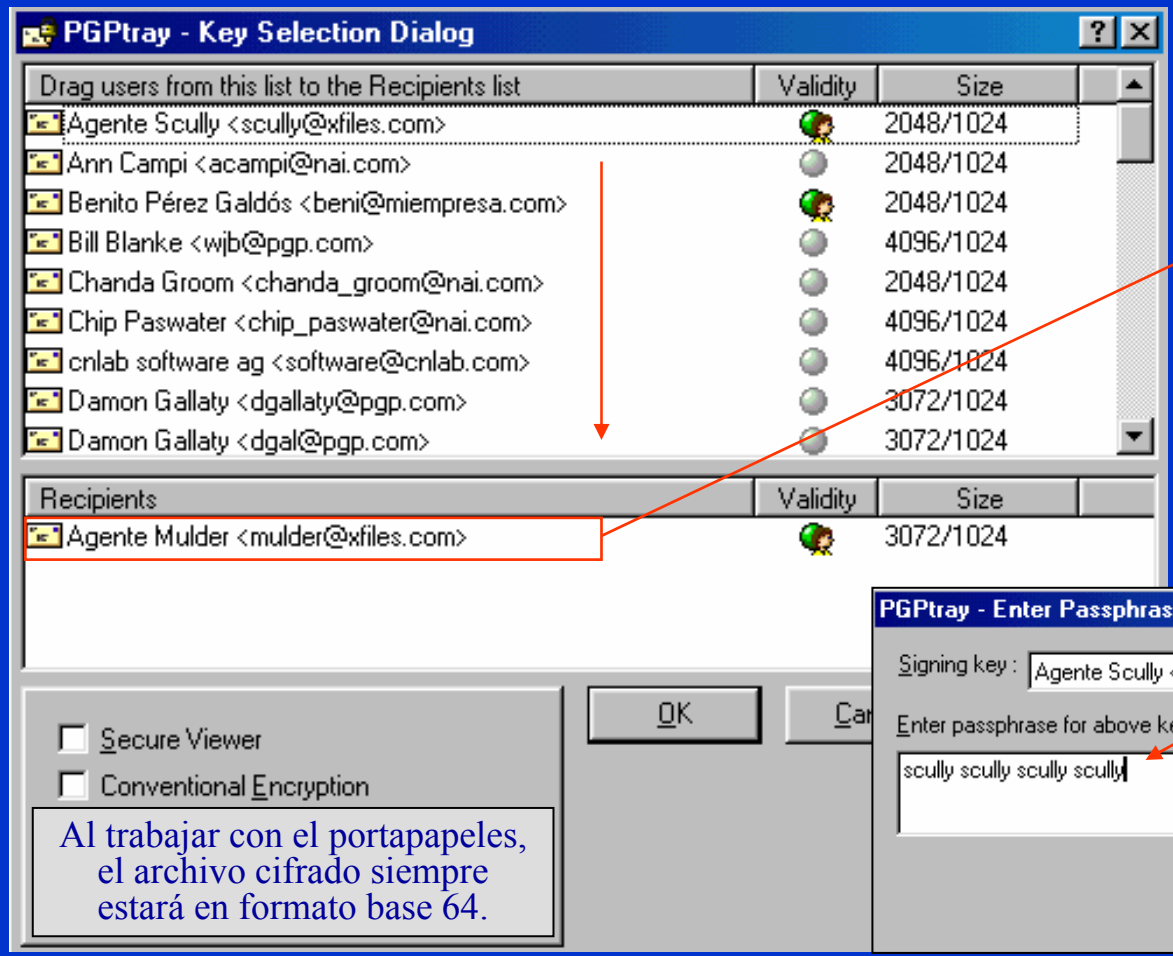


El texto se escribe con WordPad o con el Bloc de Notas y luego se copia al portapapeles.

Si lo desea también puede crear el texto con la opción Edit y copiarlo luego al portapapeles.

Las operaciones (sólo sobre textos) se realizarán en el portapapeles por lo que en este entorno no creará archivos.

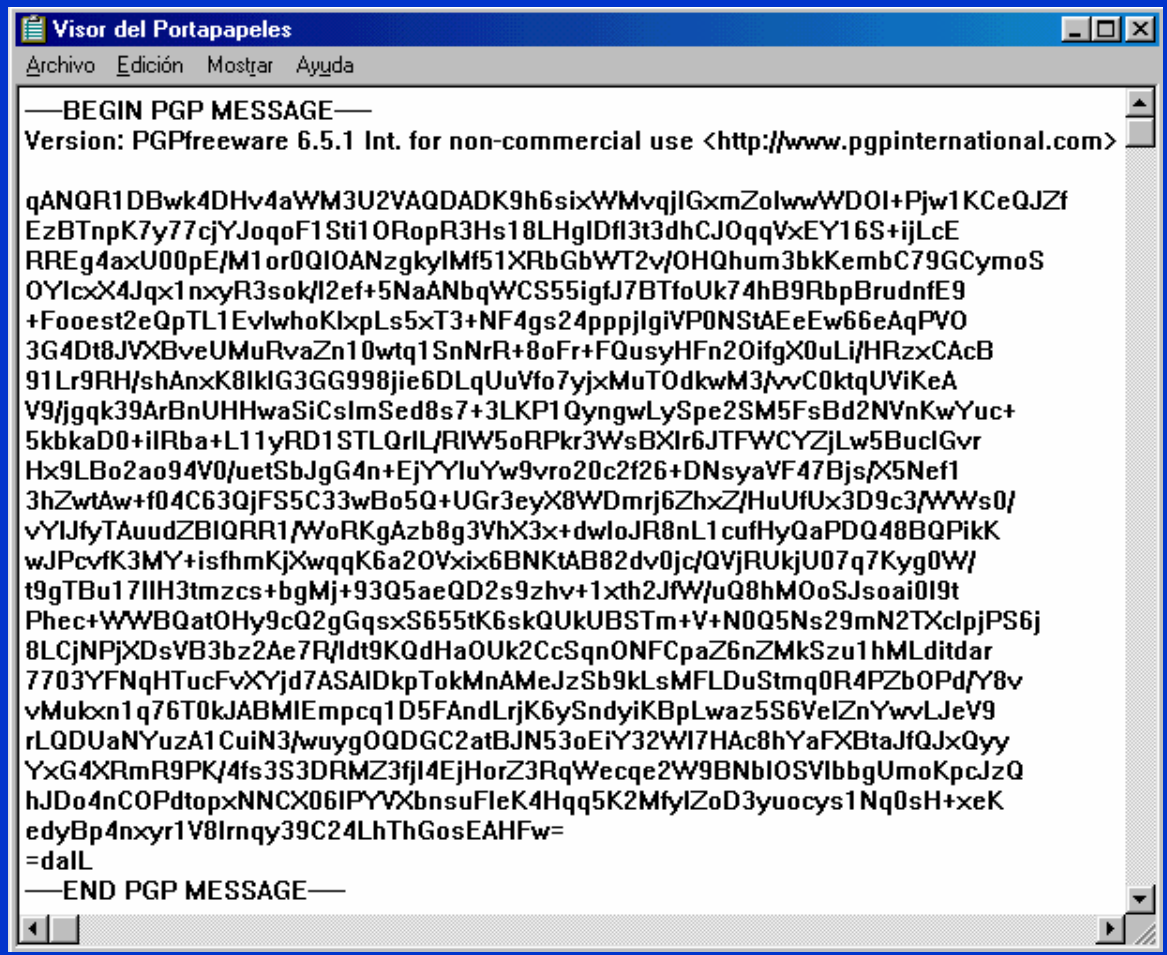
Destinatario y frase de paso para la firma



Se elige el destino y se arrastra hacia la zona de destinatarios. Puede ser más de uno y hacerlo con doble clic. Para quitar a un receptor se hace doble clic sobre él.

Como se va a firmar, PGP pedirá la frase de paso del emisor, la agente Scully.

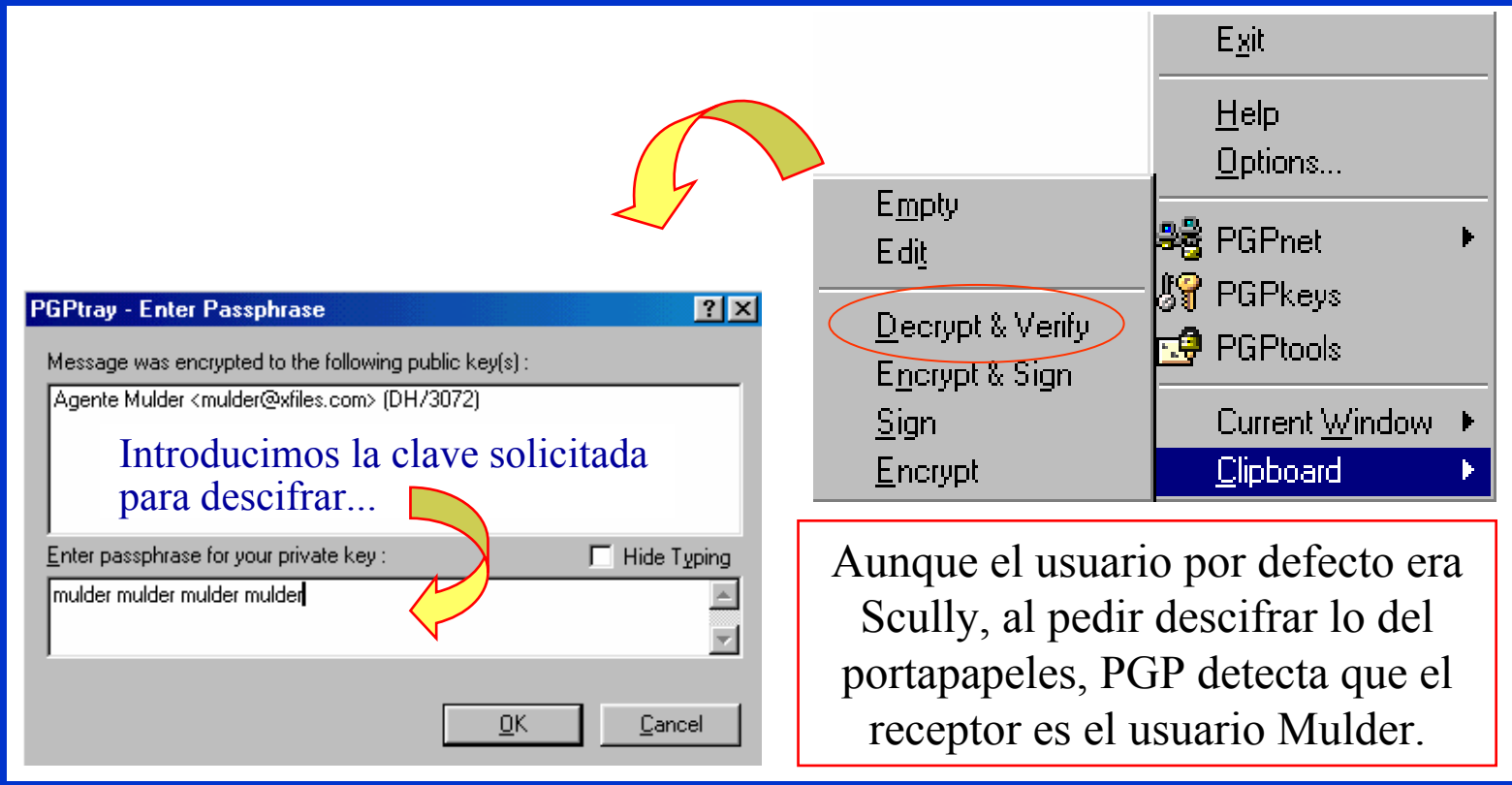
Documento portapapeles formato base 64



Todo el texto que hay en el portapapeles, incluido BEGIN PGP y END PGP, puede ahora copiarse en el cuerpo del cliente de correo electrónico.

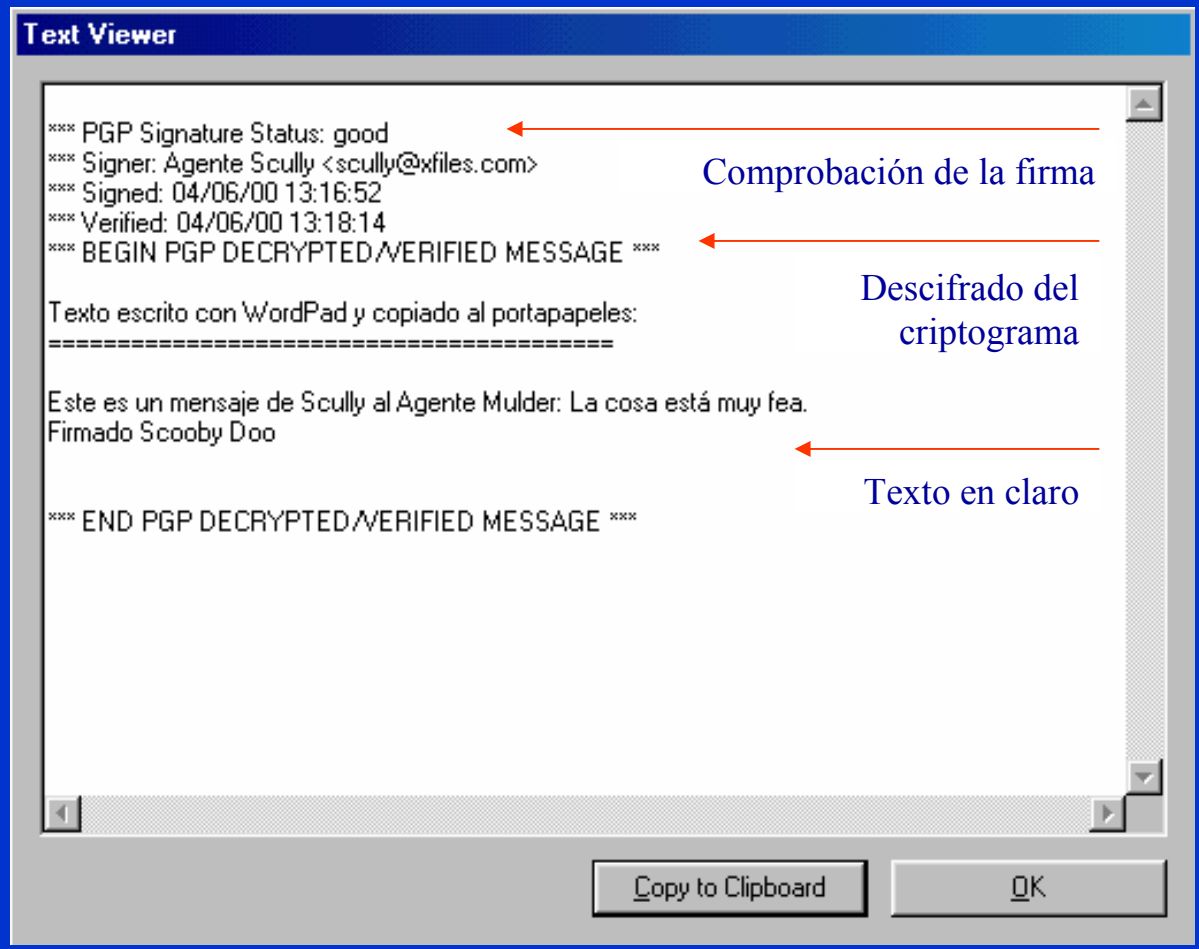
Si desea enviar por el cliente de correo un archivo adjunto cifrado y/o firmado, deberá crear primero ese documento con cualquier programa como Word, Excel, etc. y aplicar sobre el archivo el menú contextual del botón derecho del ratón.

Descifrado del criptograma por destinatario



Aunque el usuario por defecto era Scully, al pedir descifrar lo del portapapeles, PGP detecta que el receptor es el usuario Mulder.

Mensaje en claro y firma comprobada



Otras operaciones con PGP

PGP permite hacer otras operaciones interesantes:

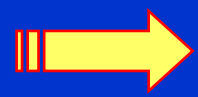
- Dividir (split) una clave privada en varias subclaves, de forma que para deshacer una operación de cifra o firmar un documento se requiere un umbral de estas subclaves dadas por diseño. Está basado en el esquema de Blakely-Shamir.
- Firmar las claves públicas de otros usuarios con distintos niveles de confianza.
- Revocar una clave, habilitar o deshabilitar una clave.
- Enviar, buscar y actualizar claves desde servidores.
- Cifrar con la opción sólo para tus ojos, crear grupos, etc.

Le recomiendo que éstas y otras operaciones las realice a modo de ejercicio, instalando en su PC el programa.

PGP versión 7.0.3

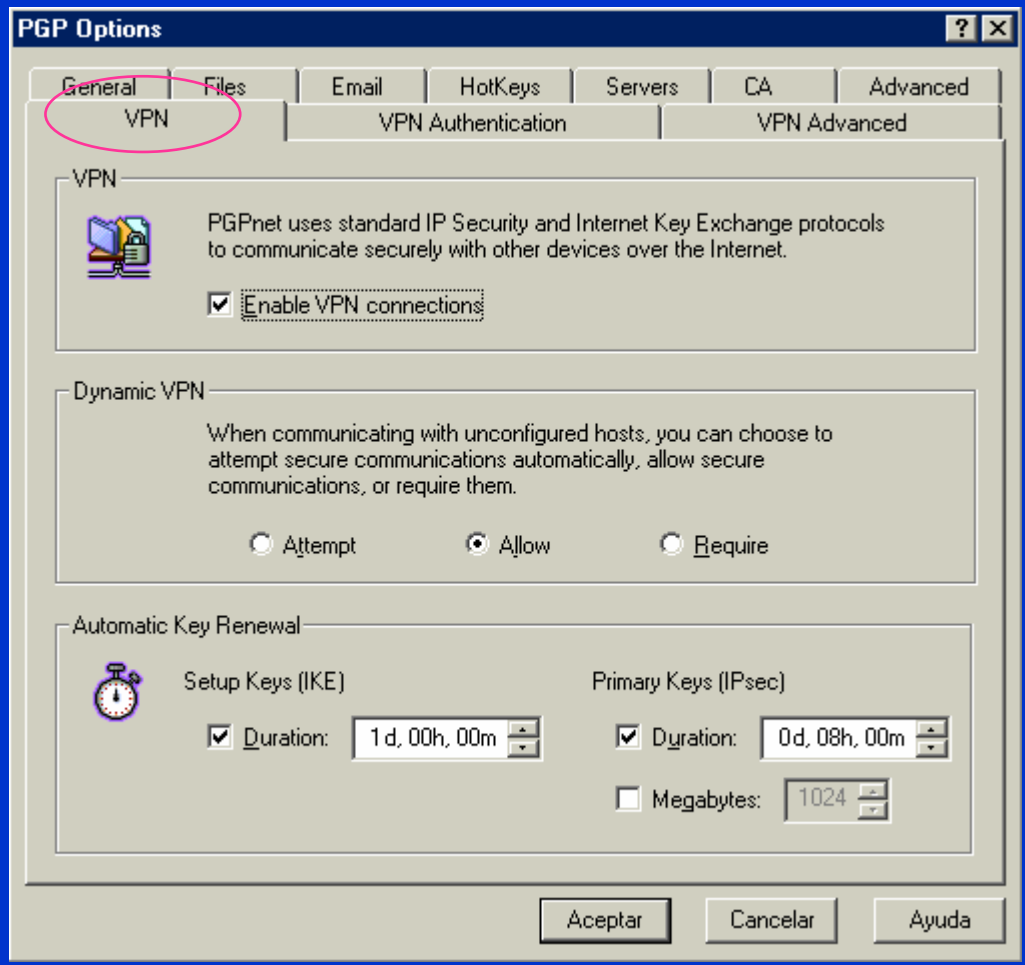
Es básicamente el mismo programa de la versión 6.5.1 pero con ligeras diferencias:

- Los mayores cambios se observan en menú PGP Options. Incluye opción de descifrado automático muy interesante.
- Incluye dos nuevos algoritmos: AES y Twofish
- La creación de claves es de forma automática DH/DSS con 1.024 bits. Si queremos crear claves RSA, debemos entrar obligatoriamente en la opción experto.
- Añade opciones de configuración de VPNs.

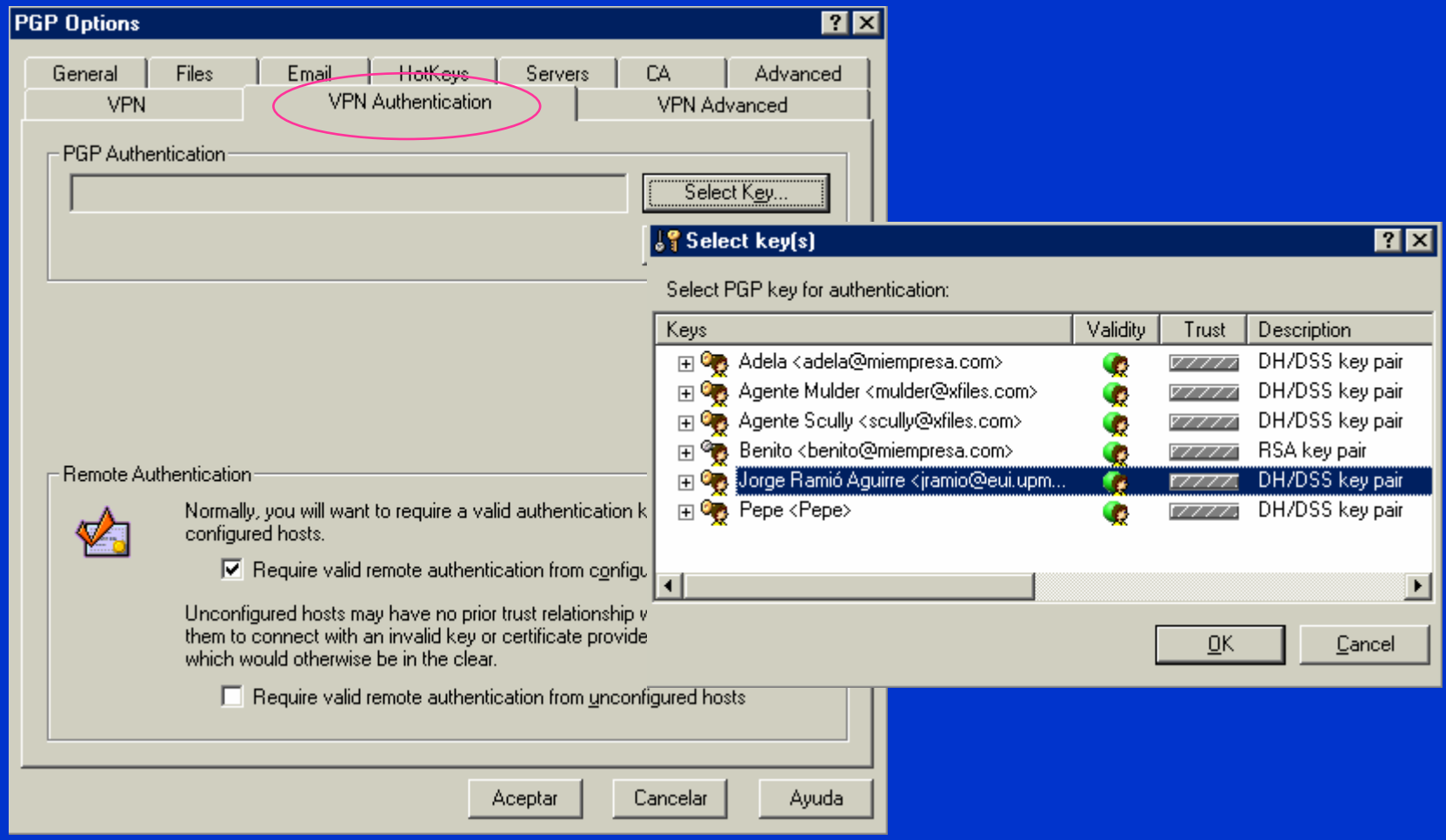


El peor inconveniente es que su código no es público por lo que nadie puede asegurar que el programa haga exactamente lo que dice que hace, por ejemplo la fortaleza de la cifra, protección ante ataques tempest, etc.

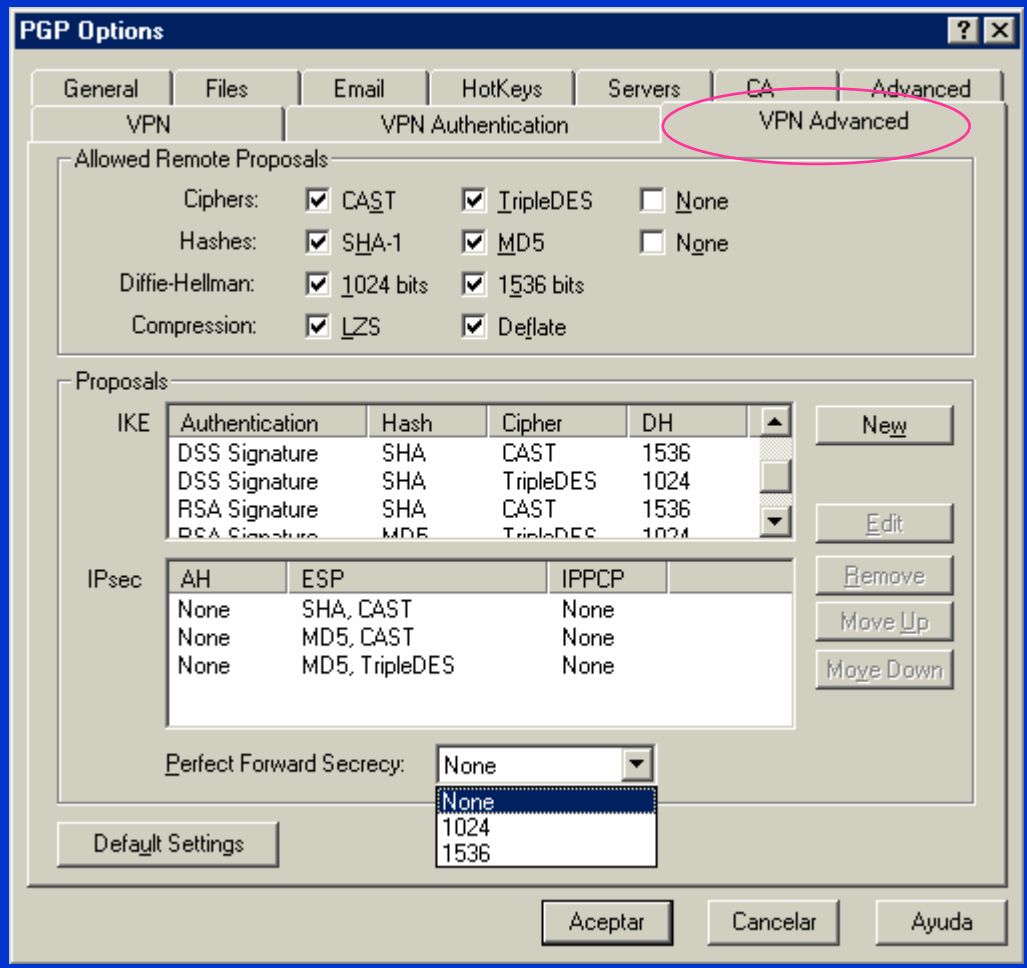
Opciones de VPN en PGP 7.0.3



Autenticación VPN en PGP 7.0.3

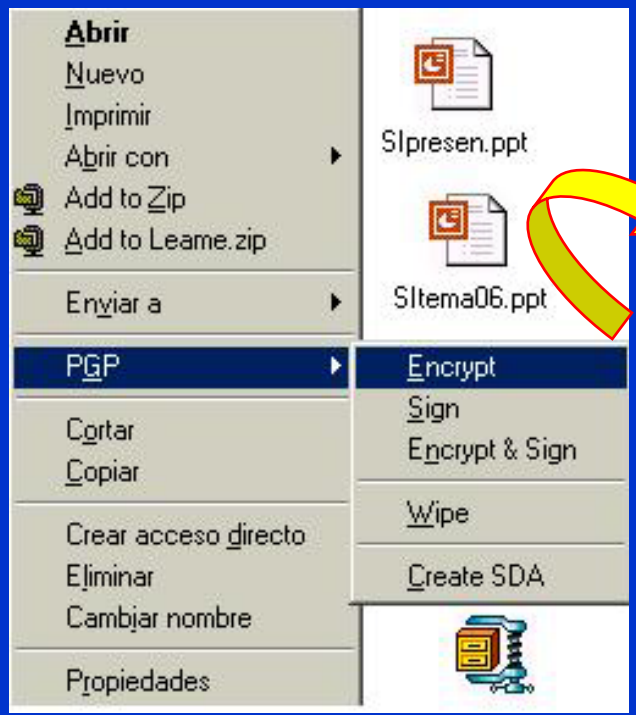


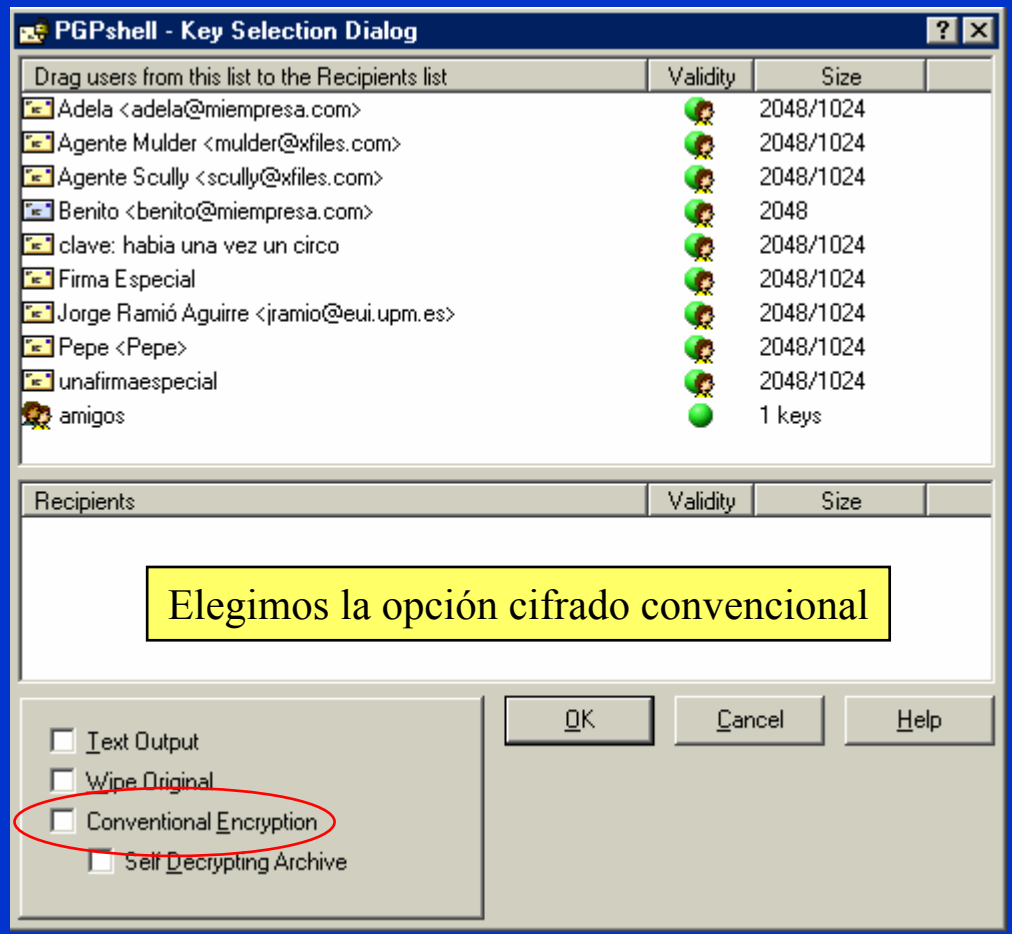
Opciones VPN avanzadas en PGP 7.0.3



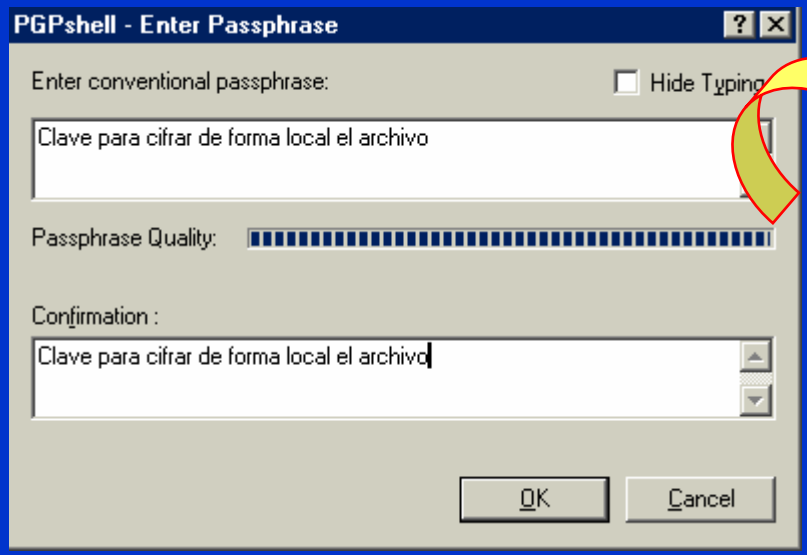
Cifrado local de ficheros con PGP 7.0.3

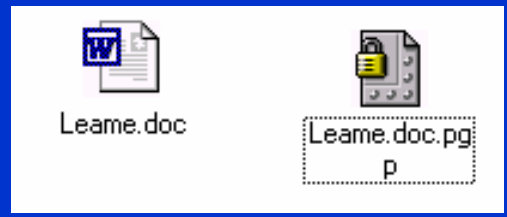
Podemos usar el botón derecho del ratón sobre archivo Leame.doc...





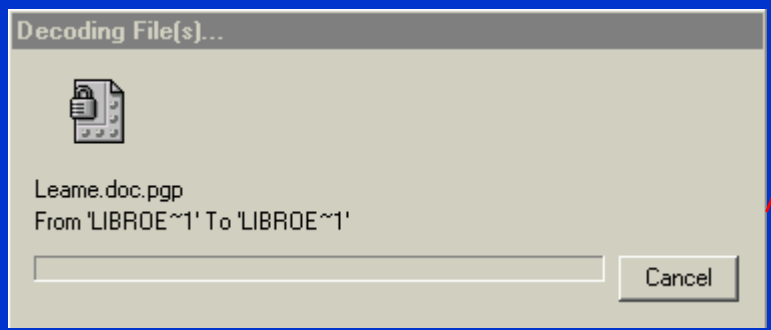
Descifrado de ficheros con PGP 7.0.3





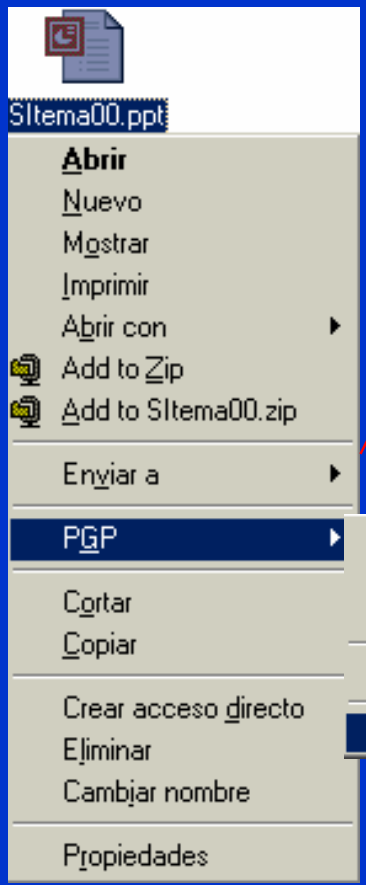
El archivo queda cifrado, con icono de PGP y extensión pgp. Observe que el archivo original permanece porque no hemos activado la opción wipe original.

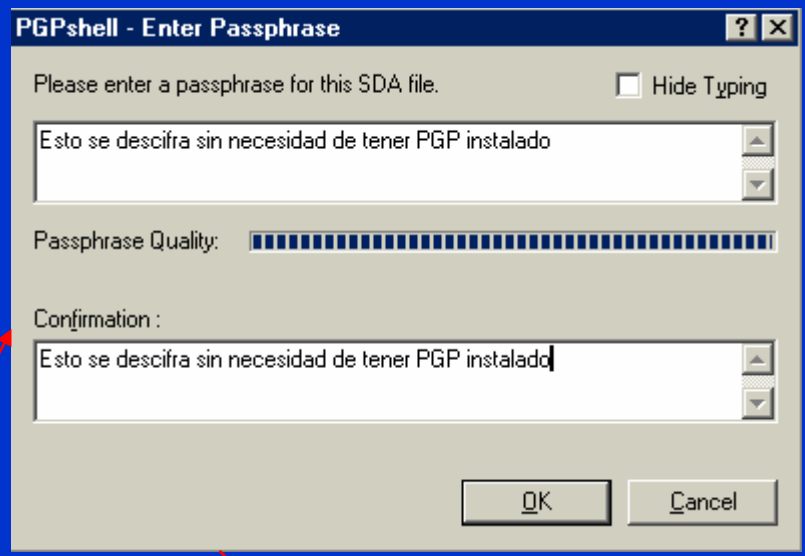
Pinchando dos veces sobre el icono...

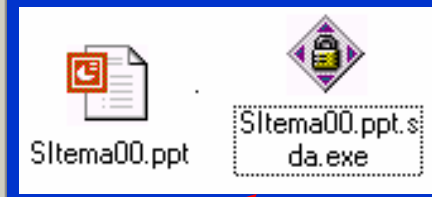


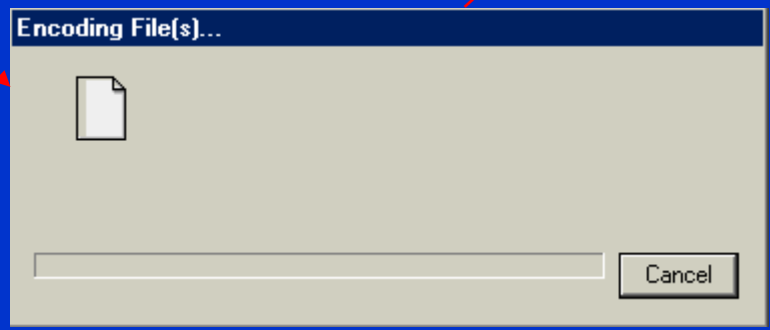
¡Si olvidamos la clave y usamos wipe, nunca podremos recuperar el archivo!

Cifrado en modo SDA con PGP 7.0.3



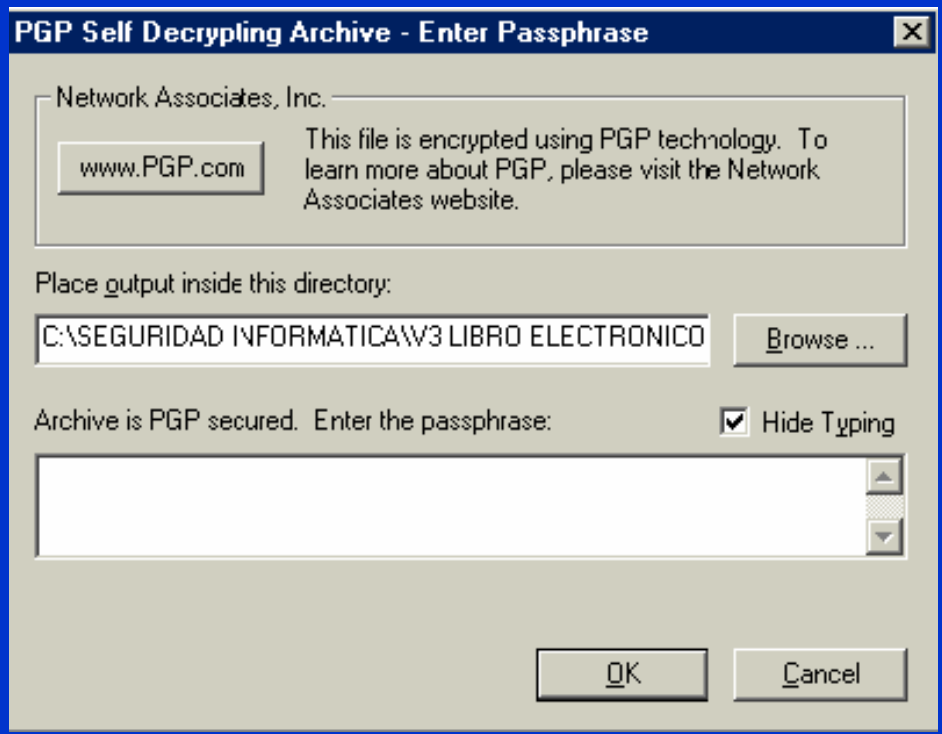






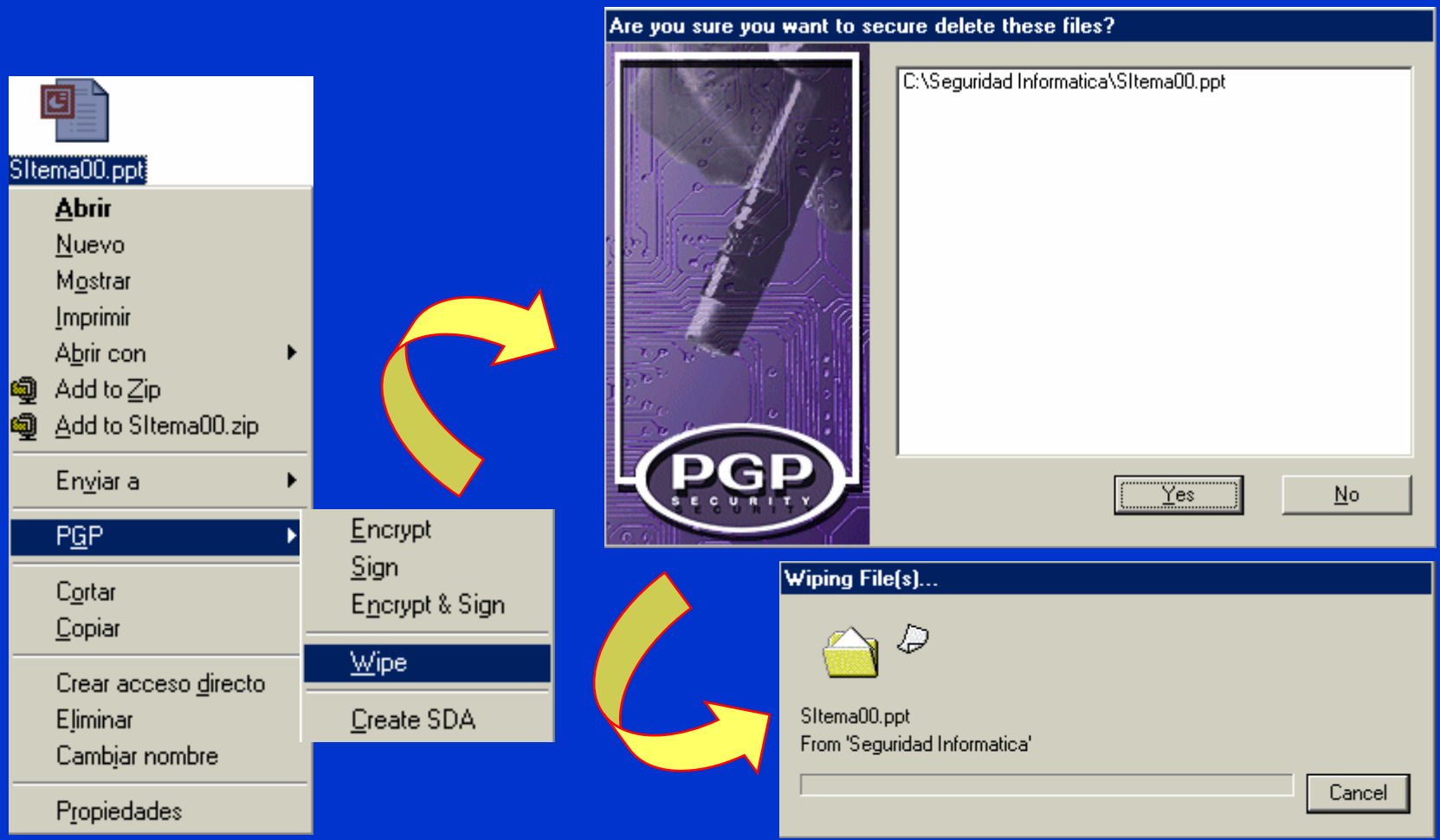
Opción SDA: Self Decrypting Archive

Descifrado SDA con PGP 7.0.3



Este archivo ejecutable se descifra de forma automática en recepción sin necesidad de que el usuario tenga instalado PGP. ☺

Borrado físico de archivos con PGP 7.0.3

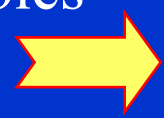


PGP versión 8.0

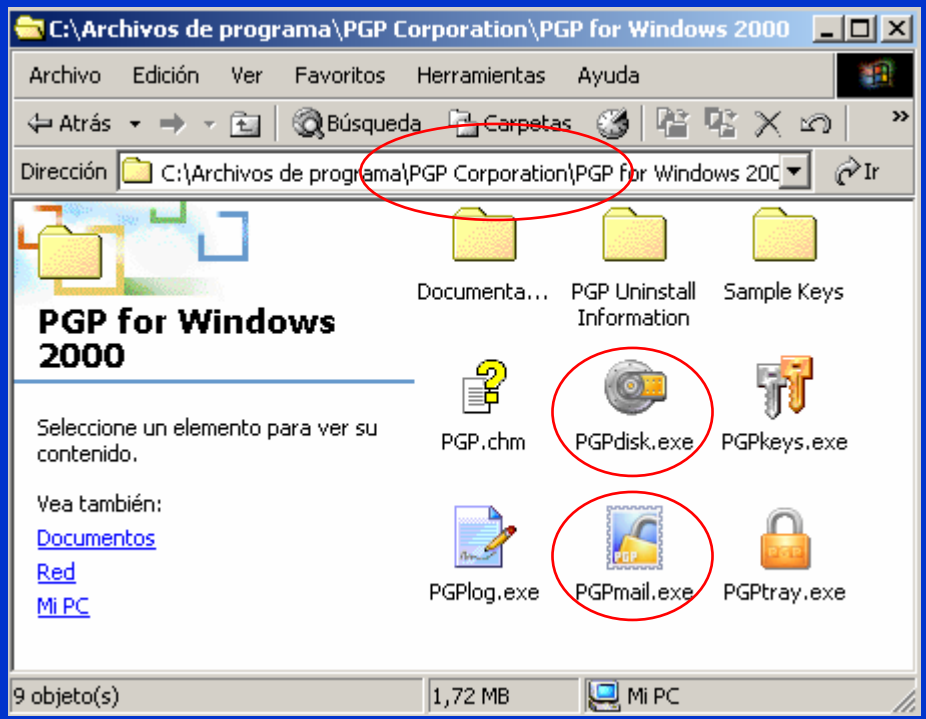
Las operaciones sobre archivos para cifra y firma digital siguen siendo muy similares a las versiones anteriores. La oferta de producto, además de la versión freeware contempla:

- PGP Desktop
- PGP Desktop Upgrade to Enterprise
- PGP Enterprise
- PGP Mobile
- PGP Personal

Además de las carpetas de instalación, veremos algunas de las opciones de configuración con diferencias notables respecto a las versiones anteriores.



Carpeta e iconos de PGP versión 8.0

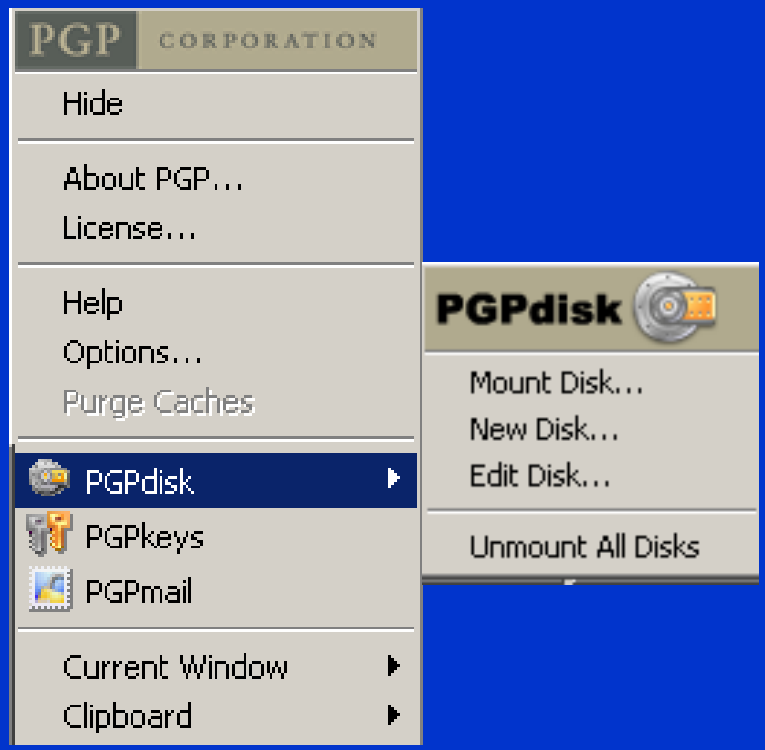


La versión 8.0 se instala en la carpeta PGP Corporation que cuelga de la carpeta de C:/Archivos de Programa.

Incluye un nuevo programa: PGPdisk

El programa PGPmail es el antiguo PGPtools

Programa PGPdisk

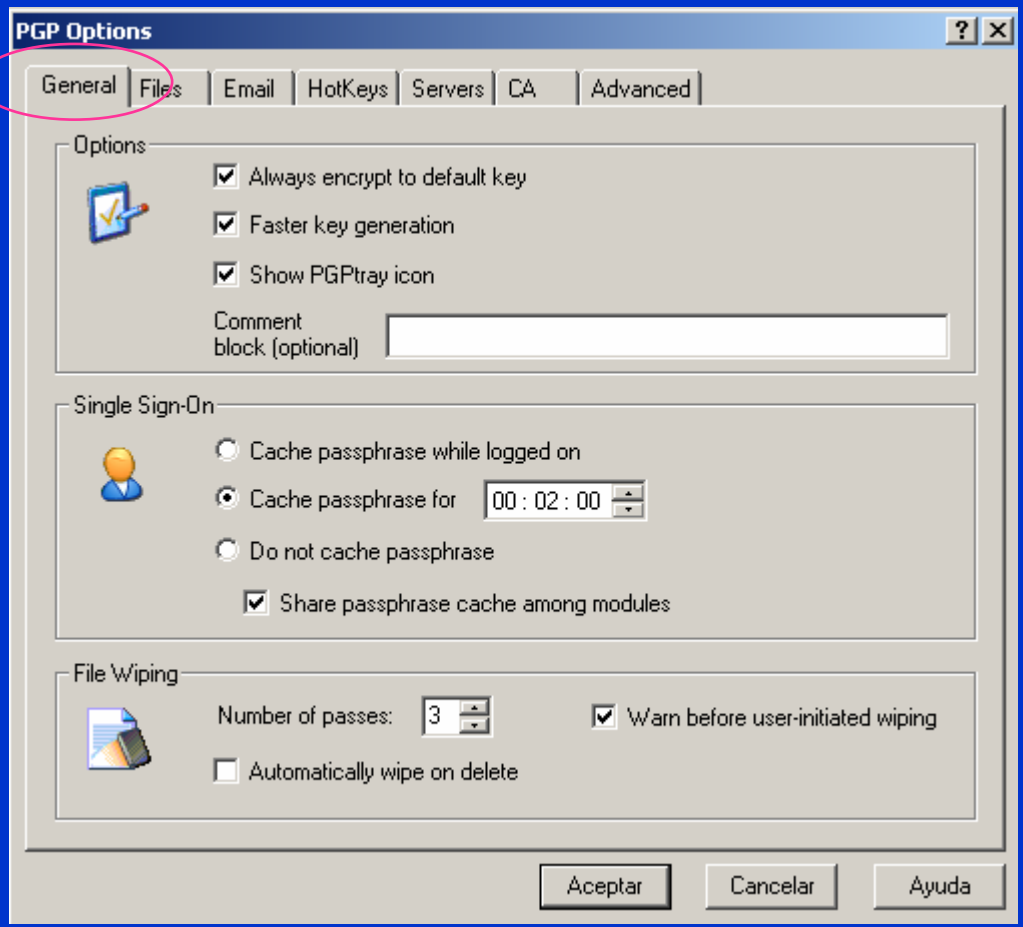


El programa PGPdisk permite montar una sección del disco como si se tratase de una unidad más en su PC.

De esta forma toda la información que allí se almacene estará protegida por una clave.

Desgraciadamente esta opción no viene incluida en la edición freeware.

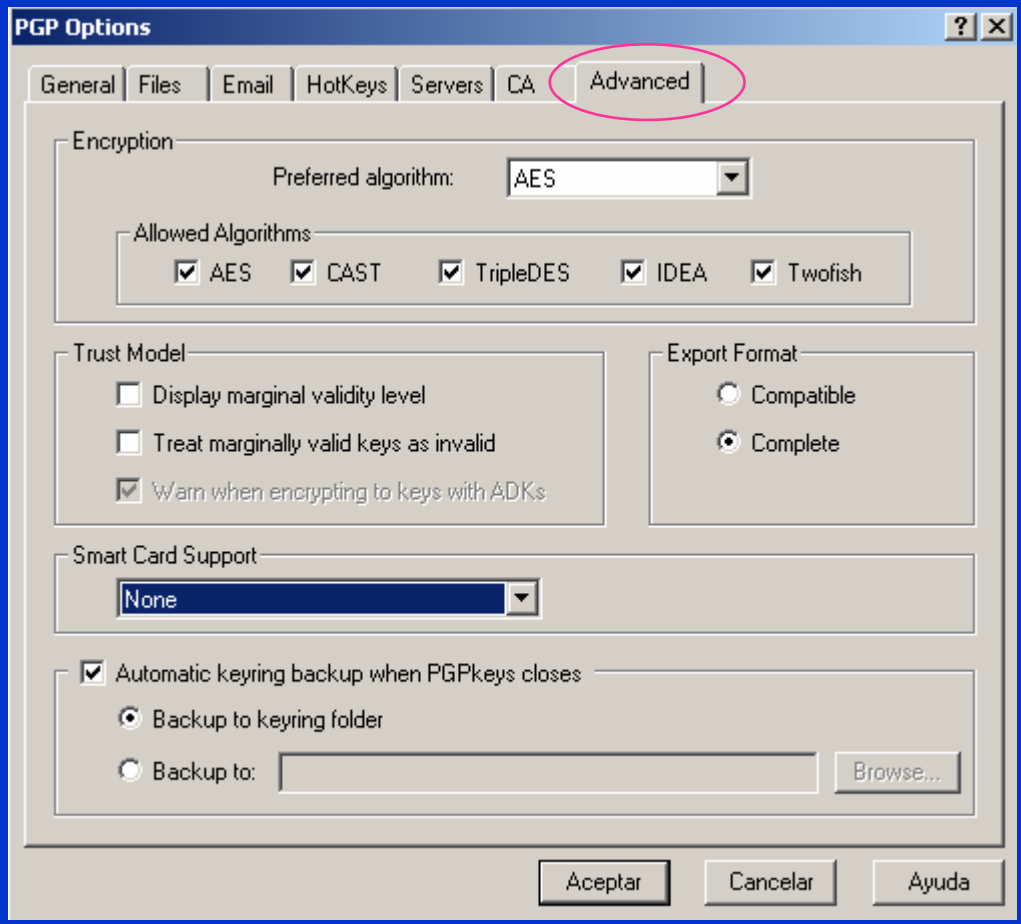
Opciones generales de PGP 8.0



Incluye al igual que en la versión 7.03 la opción de Single Sign On. Consiste en permitir la firma digital de documentos durante un tiempo dado, sin tener que introducir en cada uno de ellos la frase de paso para acceder a clave privada.

Las demás opciones son las mismas, con ligeras modificaciones.

Opciones avanzadas de PGP 8.0



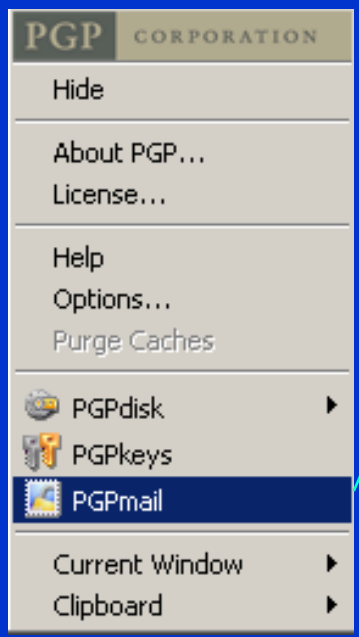
El algoritmo por defecto es el nuevo estándar AES (Rijndael) y cambia Blowfish por Twofish.

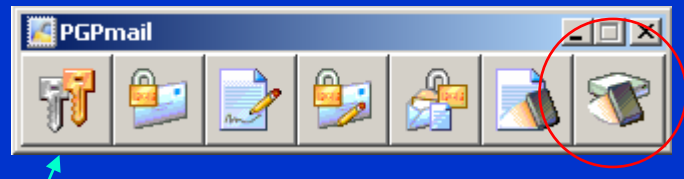
Incluye la opción de usar tarjetas inteligentes para almacenar y gestionar claves.

Se puede configurar un backup automático del anillo de claves al cerrar el programa.

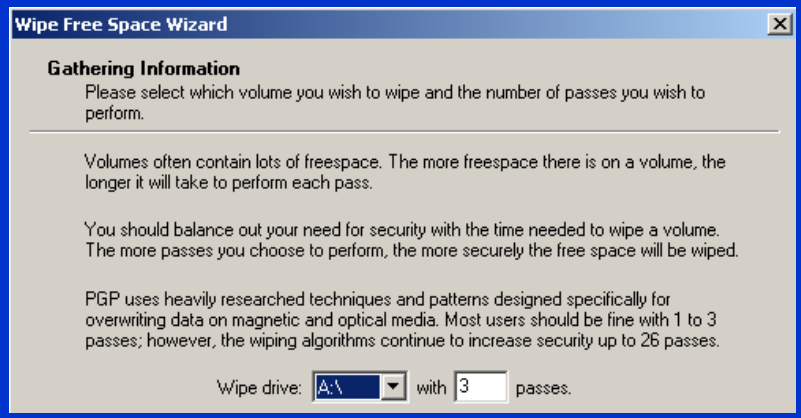
Acceso a Wipe Free Space desde PGPmail

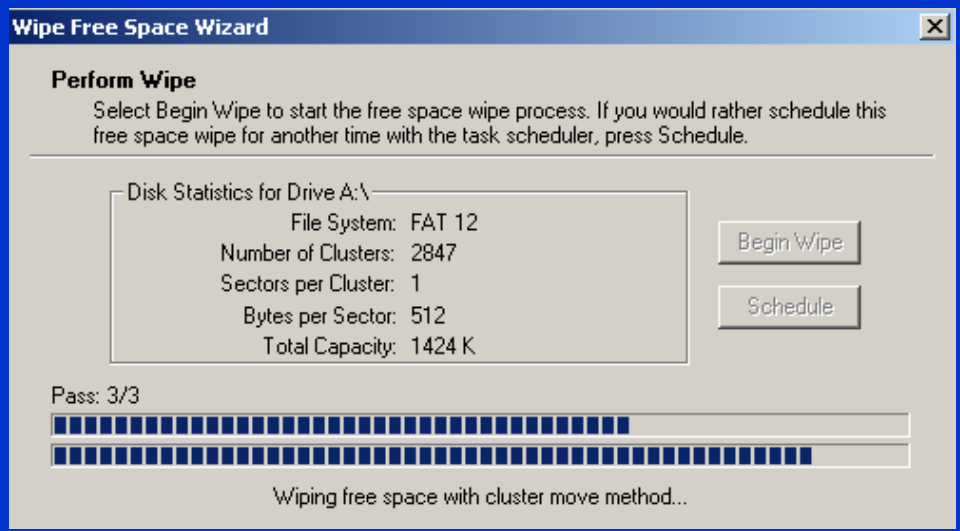
Desde PGPtray se accede a PGPmail





Free Space Wipe con PGP 8.0





Seleccionamos la unidad en la que vamos a borrar archivos temporales y el espacio libre al final del cluster de cada archivo del disco.

Se elige el número de pasos que hará el programa de borrado.

Nota: es una acción que toma bastante tiempo. En este caso de unidad A:\ y con sólo dos archivos, tres pasadas han significado más de 7 minutos

Estándar PKCS

- PKCS: Public-Key Cryptography Standards. Conjunto de especificaciones técnicas desarrolladas por Netscape, RSA y otros desarrolladores de informática con el objeto de uniformizar las técnicas de criptografía pública
- Publicación de la primera versión 1.0 en el año 1991.
- PKCS forma parte de distintos estándares de hecho como ANSI X9, PKIX, SET, S/MIME y SSL.
- A la fecha existen 14 documentos con títulos genéricos que van desde PKCS #1 a PKCS #15. Los puede descargar desde el servidor http://www.rsasecurity.com/rsalabs/pkcs/.
- Mantendremos los títulos originales en su versión en inglés de RSA Security Inc. Public-Key Cryptography Standards PCKS.

Documentos del estándar PKCS (2002)

- PKCS #1: RSA Cryptography Standard
- PKCS #2: Incluido ahora en PKCS #1
- PKCS #3: Diffie-Hellman Key Agreement Standard
- PKCS #4: Incluido ahora en PKCS #1
- PKCS #5: Password-Based Cryptography Standard
- PKCS #6: Extended-Certificate Syntax Standard
- PKCS #7: Cryptographic Message Syntax Standard
- PKCS #8: Private-Key Information Syntax Standard
- PKCS #9: Selected Attribute Types
- PKCS #10: Certification Request Syntax Standard
- PKCS #11: Cryptographic Token Interface Standard
- PKCS #12: Personal Information Exchange Syntax Standard
- PKCS #13: Elliptic Curve Cryptography Standard
- PKCS #15: Cryptographic Token Information Format Standard

Veremos sólo un breve resumen del contenido del estándar PKCS #1

PCKS #1 v2.1 del 14 de Junio 2002 (1)

PKCS #1: RSA Cryptography Standard

- Tipos de claves: definición de claves pública y privada.
- Conversión de primitivas: I2OSP (Integer-to-Octet-String primitive) y OS2IP (Octet-String-to-Integer primitive).
- Primitivas criptográficas cifra: RSAEP (RSA encryption primitive) y RSADP (RSA decryption primitive). En este último caso se especifica la operación típica para $n = p*q$ y la operación para $n = r_1*r_2*r_3*...*r_u$.
- Primitivas criptográficas firma: RSASP1 (RSA signature primitive 1) especificando la operación típica para $n = p*q$ y la operación en caso de que $n = r_1*r_2*r_3*...*r_u$ y RSAVP1 (RSA verification primitive 1).
- Esquemas de cifrado: RSAES-OAEP (RSA encryption scheme usando Optimal Asymmetric Encryption Padding). Especificación de las operaciones de cifrado y descifrado.

PCKS #1 v2.1 del 14 de Junio 2002 (2)

- Esquemas de firma con apéndice: RSASSA-PSS (RSA signature scheme with appendix - Probabilistic Signature Scheme). Define la firma y su verificación.
- Métodos de codificación para asignaturas con apéndices: EMSA-PSS (Encoding Method for Signatures with Appendix - Probabilistic Signature Scheme). Define operaciones de codificación y verificación.
- Sintaxis ASN.1: define los identificadores de objetos ANS.1 para las claves pública y privada RSA, RSAES-OAEP, RSASSA-PSS, etc.
- Técnicas soportadas: algoritmos funciones hash MD2, MD5, SHA-1 y los propuestos SHA-256, SHA-384 y SHA-512 así como funciones de generación de máscaras: MGF1 (Mask Generation Function 1).

☞ La patente de RSA ha expirado en septiembre de 2000 no así el nuevo sistema de cifra RSA con múltiples primos.

Fin del Tema 16

Cuestiones y ejercicios (1 de 4)

1. Usando la tabla correspondiente represente en base 64 los siguientes mensajes ASCII: M_1 = AMIGOS, M_2 = ¿Qué pasa?
2. Instale una versión de PGP, vaya a la carpeta del programa y luego imprima el documento que verá en la carpeta documentation.
3. Con cualquier versión de PGP cifre de forma local y con armadura (base 64) un archivo que haya creado con el bloc de notas y observe los rellenos que introduce en el criptograma y al final de él. Añada una letra al texto en claro y vuelva a comparar los textos ASC.
4. Cifre el documento TXT anterior y observe la salida ASC. Vuelva a cifrarlo y observe la salida. ¿Coinciden? ¿Qué ha pasado?
5. ¿Es posible que PGP cifre un documento y éste salga en claro?
6. ¿Por qué siempre se comprime el mensaje antes de cifrar?
7. ¿Qué puede decir de la gestión de claves públicas que ofrece PGP?

Cuestiones y ejercicios (2 de 4)

8. ¿Qué tipo de esquema de cifra es el que utiliza PGP?
9. Después de instalar PGP en nuestro computador, ¿qué es lo primero que nos sugiere?
10. ¿Qué diferencia hay entre elegir una clave DH/DSS y RSA?
11. Si creamos un nuevo par de claves asimétricas, ¿queda el nuevo usuario como usuario por defecto?
12. Cree tres nuevos usuarios Hugo, Paco y Luis con diferentes tipos y longitudes de clave. Haga que entre ellos se firmen sus claves.
13. Incluya en cada uno una fotografía en sus propiedades. Puede ser cualquier archivo con formato de imagen.
14. ¿Qué sucede si creamos que un nuevo usuario Ana con una clave tipo DH/DSS con una longitud exacta de 4.000 bits? ¿Podemos crear una clave RSA de 4.000 bits?

Cuestiones y ejercicios (3 de 4)

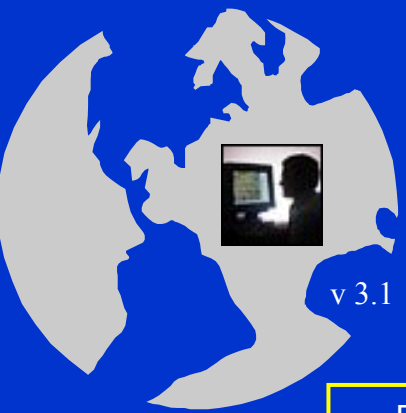
15. Revoque una clave y observe sus propiedades. ¿Se puede recuperar esa clave? ¿Puede deshabilitar una clave? ¿Qué significa esto?
16. Cree el grupo Sobrinos con los usuarios Hugo, Paco y Luis. Envíe un mensaje a ese grupo eligiendo desde PGPkeys Show Groups.
17. Cree el usuario FirmaEspecial con frase de paso UnaFirmaEspecial en la que intervengan 4 usuarios, cada uno con una porción igual de la clave y umbral 3. Cifre y firme un documento con dicha clave. ¿Podría alguien tener más de una participación de la clave?
18. Mediante el botón derecho del ratón cifre de forma local un archivo por ejemplo de Word, en modo formato compatible. Vuelva a cifrar el archivo original con otro nombre pero ahora sin formato base 64. ¿Cómo son ambos criptogramas? ¿Cómo son sus tamaños en bytes?
19. Si se cifra un archivo txt con la opción Secure Viewer, ¿se guarda el archivo descifrado? ¿Es seguro? ¿Puede hacerse con archivo Word?

Cuestiones y ejercicios (4 de 4)

20. Cree el nuevo usuario Pepillo y exporte su clave a un archivo. Borre ahora este usuario y desde PGPkey importe ese archivo de clave pública. ¿Se añade el usuario al anillo de claves públicas?
21. ¿Qué pasa en un cifrado local de un archivo con self decrypting? Compruébelo con un archivo cualquiera. ¿Podríamos usar esta opción si además de cifrar vamos a firmar ese documento?
22. ¿Qué significa actualizar una clave desde PGPkeys?
23. Añada un nuevo nombre a una clave. ¿Para qué puede servir esto?
24. ¿Qué son el KeyID y el Fingerprint? ¿Qué utilidad puede tener que la huella dactilar también esté dada como un conjunto de palabras?
25. Si una clave está revocada, ¿puede recibir archivos cifrados con su clave pública? ¿Puede firmar nuevos documentos? ¿Puede descifrar y/o comprobar documentos anteriores a la fecha de revocación?

Tema 17
Protocolos y Esquemas Criptográficos

Seguridad Informática y Criptografía



Material Docente de
Libre Distribución

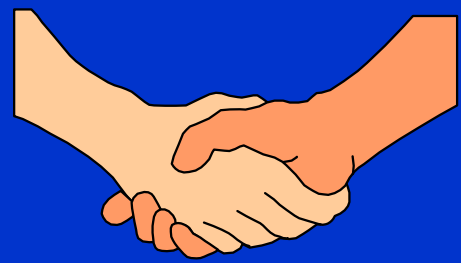
Ultima actualización: 03/03/03
Archivo con 72 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Definición de protocolo criptográfico

- Protocolo: es el conjunto de acciones coordinadas que realizan dos o más partes o entidades con el objeto de llevar a cabo un intercambio de datos o información.
- Protocolos criptográficos serán aquellos que cumplen esta función usando para ello algoritmos y métodos criptográficos.
- Permiten dar una solución a distintos problemas de la vida real, especialmente en aquellos en donde puede existir un grado de desconfianza entre las partes.

¿Qué es un protocolo?



Veamos 10 ejemplos

Ejemplos de protocolos criptográficos (1)

1.- El problema de la identificación del usuario

¿Cómo permitir que un usuario se identifique y autentique ante una máquina -y viceversa- sin que sea posible la obtención por un tercero de la clave o password?

2.- El problema del lanzamiento de la moneda

¿Cómo permitir que dos usuarios realicen una prueba con probabilidad ½ -como es el lanzamiento de una moneda- si éstos no se encuentran físicamente frente a frente y, a la vez, asegurar que ninguno de los dos hace trampa?

Ejemplos de protocolos criptográficos (2)

3.- El problema de la firma de contratos

¿Cómo permitir que dos o más usuarios que se encuentran físicamente alejados puedan realizar la firma de un contrato, asegurando que ninguno de ellos va a modificar las condiciones ni negarse a última hora a dicha firma?

4.- El problema del descubrimiento mínimo de un secreto

¿Cómo poder demostrar y convencer a otra persona o sistema que uno está en posesión de un secreto, sin por ello tener que desvelarlo ni a ella ni a un tercero?

Ejemplos de protocolos criptográficos (3)

5.- El problema del juego de póker mental o por teléfono

¿Cómo permitir que varios usuarios puedan jugar a través de la red un juego de póker -o cualquier otro- si no están físicamente en una misma mesa de juego y asegurando, al mismo tiempo, que ninguno de ellos va a hacer trampa?

6.- El problema de la división de un secreto o del umbral

Si tenemos un secreto único y por tanto muy vulnerable, ¿cómo permitir que ese secreto se divida en n partes, de forma que juntando k partes sea posible reconstruirlo y, en cambio, con k-1 partes imposible su reconstrucción?

Ejemplos de protocolos criptográficos (4)

7.- El problema del esquema electoral o voto electrónico

¿Cómo realizar unas elecciones a través de una red, de forma que pueda asegurarse que el voto es único y secreto, que los votantes estén autenticados y que ese voto se contabiliza adecuadamente en el cómputo final?

8.- El problema de la transmisión por canales subliminales

Dos usuarios desean intercambiar información a través de un tercero del cual desconfían. ¿Cómo pueden hacerlo sin cifrar la información de forma que este tercero sólo vea un mensaje con texto en claro aparentemente inocente?

Ejemplos de protocolos criptográficos (5)

9.- El problema del millonario

Dos usuarios desean conocer cuál de los dos tiene más dinero en su cuenta corriente. ¿Cómo pueden hacerlo de forma que, una vez terminado el protocolo, ambos sepan quién de los dos es más rico sin desvelar la cantidad de dinero del otro?

10.- El problema del correo electrónico con acuse de recibo

¿Cómo hacer que una vez recibido un correo electrónico, éste sólo pueda ser leído (abierto) si el receptor envía, con anterioridad al emisor, un acuse de recibo como sucede -de forma similar- con el correo normal certificado?

Transferencia inconsciente o trascordada

Algoritmo de TI propuesto por Michael Rabin en 1981.

- Un usuario A transfiere a un usuario B un dato o secreto con un cifrado probabilístico del 50%.
- El usuario B recibe el dato y tiene una probabilidad del 50% de descubrir el secreto. Una vez que ha recibido el dato, B sabe si éste es el secreto o no.
- No obstante, el usuario A no tiene forma de saber si el usuario B ha recibido el secreto o no.

Esta incertidumbre mutua forzará a los protagonistas a que terminen el protocolo sin hacer trampas.

Algoritmo de TI de Rabin (1)

Paso 1º A elige dos primos (p y q), calcula n = p*q y envía el valor n a B.

Paso 2º B elige un número aleatorio x del CCR(n) de forma que que mcd (x,n) = 1, y devuelve a A el valor $K = x^2 \bmod n$.

Paso 3º A calcula las cuatro raíces de $x^2 \bmod n$ y envía a B una de ellas. Las raíces de $x^2 \bmod n$ serán: x, n-x, y, n-y. Sólo A puede hacerlo porque conoce los valores de p y q.

Paso 4º B intenta descubrir el valor de p o q.

Conclusión del algoritmo de TI de Rabin

Si B recibe x o n-x no será capaz de encontrar p o q.

No tiene más información que la que tenía porque:

☹ x y n-x son valores que conoce (B ha elegido x).

Si B recibe y o n-y, podrá encontrar p o q.

En este caso, como $x^2 \bmod n = y^2 \bmod n$, entonces:

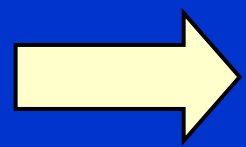
☺ $(x^2 - y^2) \bmod n = (x+y)(x-y) \bmod n = 0$

Luego $(x+y)(x-y) = k*n$ y se cumplirá que:

$p = mcd\ (x+y, n)$ y

$q = mcd\ (x-y, n)$

Para entenderlo mejor ... veamos un ejemplo



Ejemplo de algoritmo de TI de Rabin (1)

A Adela tiene como números secretos p y q, valores que corresponden a la factorización del valor n.

B Benito conoce el valor n y deberá descubrir, a partir del protocolo de transferencia inconsciente, p o q.

Ejemplo con valores:

Sea p = 7; q = 13. Luego, n = p*q = 7*13 = 91.

1.- A envía a B el valor n = 91.

2.- B elige al azar del CCR(91) el valor x = 15 y calcula $K = 15^2 \bmod 91 = 225 \bmod 91 = 43$. Se lo envía a A.

3.- A recibe K = 43 y calcula las 4 raíces de $x^2 \bmod n$.

Cálculo de raíces de la TI de Rabin

A calcula las dos raíces de $x^2 \bmod n = K$ de en p y q:

$x_1^2 = K \bmod p = 43 \bmod 7 = 1 \quad \Rightarrow \quad x_1 = 1$

$x_2^2 = K \bmod q = 43 \bmod 13 = 4 \quad \Rightarrow \quad x_2 = 2$

Con estos valores usa ahora el Teorema del Resto Chino

No siempre será tan fácil el cálculo de estas raíces como se verá más adelante

Si no recuerda el Teorema del Resto Chino, repase el archivo SItema05.

Teníamos que: $x_1 = 1$ y $x_2 = 2$.

Aplicando entonces la ecuación del TRC:

Aplicación del TRC en la TI de Rabin

$y_1 = \text{inv}(n/p, p) = \text{inv}(91/7, 7) = \text{inv}(13,7) \Rightarrow y_1 = 6$

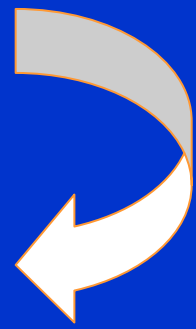
$y_2 = \text{inv}(n/q, q) = \text{inv}(91/13, 13) = \text{inv}(7,13) \Rightarrow y_2 = 2$

$$x = [(n/p)*y_1*x_1 + (n/q)*y_2*x_2] \bmod n$$

$$\therefore \quad x = (13*6*x_1 + 7*2*x_2) \bmod 91$$

Luego para todas las combinaciones x_i, p y q se tiene:

$\{x_1, x_2\}$	$\Rightarrow$	$[1,2]$	$\Rightarrow$	$x = 15$
$\{x_1, q-x_2\}$	$\Rightarrow$	$[1,13-2] = [1,11]$	$\Rightarrow$	$x = 50$
$\{p-x_1, x_2\}$	$\Rightarrow$	$[7-1,2] = [6,2]$	$\Rightarrow$	$x = 41$
$\{p-x_1, q-x_2\}$	$\Rightarrow$	$[7-1,13-2] = [6,11]$	$\Rightarrow$	$x = 76$



☺ $15^2 \bmod 91 = 50^2 \bmod 91 = 41^2 \bmod 91 = 76^2 \bmod 91 = 43$.

☺ Además se cumple que $15 + 76 = 91 = n$ y $50 + 41 = 91 = n$.

Conclusión del algoritmo de TI de Rabin

A recibirá cualquiera de estos cuatro valores:15, 50, 41, 76.

- Si A recibe el número 15 (el valor que había enviado a B) o bien n-15 = 91-15 = 76 (que llamaremos valores *x*) no tiene más datos que los que tenía al comienzo del protocolo y no podrá factorizar n.
- Si A recibe cualquiera de los otros dos valores enviados por B (50 ó 41) valores que llamaremos *y*, podrá factorizar n usando la expresión mcd (x+*y*, n) con x el valor elegido por A al comienzo del protocolo, es decir 15.
- Si $y = 50 \Rightarrow$ mcd (50+15, 91) = mcd (65, 91) = 13 q = 13
- Si $y = 41 \Rightarrow$ mcd (41+15, 91) = mcd (56, 91) = 7 p = 7

Elección de p y q en algoritmo de Rabin

Para facilitar el cálculo de las raíces de $x^2 \bmod p$ y $x^2 \bmod q$, el usuario A elegirá los valores de p y q de forma que cumplan:

- ✓ El valor (p+1) sea divisible por 4.
- ✓ El valor (q+1) sea divisible por 4.

Si $x^2 \bmod p = a \bmod p \Rightarrow$ dos soluciones: x_1 y $(p - x_1)$

Si $x^2 \bmod q = a \bmod q \Rightarrow$ dos soluciones: x_2 y $(q - x_2)$

Estas soluciones se obtienen aplicando el TRC, no obstante si (p+1) es divisible por 4 entonces para este primo p si $x = a^{(p+1)/4}$ se cumple:

$(a^{(p+1)/4})^2 \bmod p = a^{(p+1)/2} \bmod p = a(a^{(p-1)/2}) \bmod p = a$

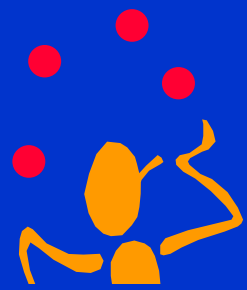
Esto es válido porque : $a^{(p-1)/2} \bmod p = 1$. Lo mismo sucede con q.

Luego: $x_1 = a^{(p+1)/4} \bmod p$ y $x_2 = a^{(q+1)/4} \bmod q$

Problema lanzamiento de la moneda (1)

Algoritmo propuesto por Mario Blum en 1982.

Se trata de resolver una apuesta entre dos personas A y B distantes entre sí mediante el lanzamiento de una moneda (cara o cruz).



Situaciones si A lanza la moneda al aire:

Caso 1

1º A lanza la moneda.
2º B hace su apuesta y se lo dice a A.
3º A le dice a B que ha salido "justo lo contrario"
... independientemente de lo que haya salido.
En este caso el usuario A hace trampa ...

Problema lanzamiento de la moneda (2)

Caso 2

1º A lanza la moneda.
2º B hace su apuesta y se lo dice a A.
3º No sale lo apostado por B y A se lo notifica.
4º B se desmiente y dice que "esa era su apuesta".
Ahora es el usuario B quien hace trampa ...

Si A y B están distantes y no hay un testigo de fe, ¿cómo puede desarrollarse el algoritmo para que ninguno de los dos pueda hacer trampa y, si lo hace, el otro lo detecte?

Esquema de Blum ⟶

El problema de la moneda según Blum

Soluciones al problema del lanzamiento de la moneda:

- Usar el protocolo de la transferencia inconsciente de Rabin con probabilidad del 50% ya visto, o bien...
- Usar el Esquema General de Blum:

1º A partir de un conjunto de números que la mitad son pares y la otra impares y una función unidireccional $f : x \rightarrow y$, el usuario A elige un valor x, calcula $y = f(x)$ y lo envía a B.

2º El usuario B apuesta por la paridad de x.

3º A le muestra a B el verdadero valor de x y su paridad.

Condiciones del esquema general de Blum

- B tendrá igual probabilidad de recibir un número par o impar.
- A deberá tener una probabilidad igual (50%) de recibir una apuesta par o impar por parte B.
- Ninguno de los dos podrá hacer trampa.

¿Búsqueda de esa función f?

Antes deberemos explicar qué se entiende por restos cuadráticos y enteros de Blum 

Restos cuadráticos de Blum

Buscamos una función unidireccional con trampa que cumpla las características del protocolo anterior.

El valor *a* es un resto cuadrático de Blum R_2 mod n si:

$$x^2 \bmod n = a$$

siendo mcd (*a*,n) = 1 → solución

¿Algún problema? Sí ⇒ No sigue la paridad deseada.

Por ejemplo, el resto cuadrático $R_2 = 4$ en mod 11 se obtiene para x = 2 (par) y x = 9 (impar) ya que:

$2^2 \bmod 11 = 4 \bmod 11 = 4$ y $9^2 \bmod 11 = 81 \bmod 11 = 4$

Enteros de Blum

Un entero de Blum es un número resultado del producto de dos primos p y q, ambos congruentes con 3 módulo 4.

En este caso se cumplirá que:

$y = x^2 \bmod n$ mantendrá la paridad con $z = y^2 \bmod n \quad \forall\, x \in Z_n$

Ejemplo: sea $n = 11*19 = 209$ y el valor $x = 24$

11 mod 4 = 3; 19 mod 4 = 3 (cumplen congruencia 3 mod 4 👍)

$y = x^2 \bmod n = 24^2 \bmod 209 = 576 \bmod 209 = 158$

$z = y^2 \bmod n = 158^2 \bmod 209 = 24.964 \bmod 209 = 93$

Como se observa, en este caso y es par y z es impar.

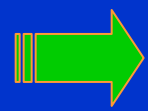
Luego, para todos los restos principales de $y = 158$ (par) que se obtengan con valores de x diferentes, el resto cuadrático z_2 será siempre el valor 93 (impar).

Paridad en enteros de Blum

Es importante recalcar que:

- Existirá igual número de soluciones y (pares o impares) que de soluciones z (pares o impares).
- Esto no sucederá con enteros que no sean de Blum.
- Por lo tanto, esta igualdad de paridad en los valores de los restos de z y de y, hará que desde el punto de vista del usuario B que recibe como dato el valor z o resto R_2 enviado por A, exista una equiprobabilidad.

  El siguiente cuadro indica la paridad de R_2 para algunos módulos enteros y no enteros de Blum.



Ejemplo de paridad en enteros de Blum

Paridad de elementos de R_2 para módulos enteros de Blum

n	p	q	*y* (pares)	*y* (impares)	*z* (pares)	*z* (impares)
21	3	7	10	10	10	10
33	3	11	12	20	12	20
57	3	19	24	32	24	32
69	3	23	36	32	36	32
77	7	11	36	40	36	40

Observe que se obtiene igual cantidad de valores y pares que de z pares. De la misma forma, se obtiene igual cantidad de valores y impares que de z impares.

Ejemplo de paridad en no enteros de Blum

Paridad de elementos de R_2 para módulos no enteros de Blum

n	p	q	y (pares)	y (impares)	z (pares)	z (impares)
15	3	5	8	6	6	8
35	5	7	14	20	8	26
39	3	13	22	16	16	22

En este caso no se obtienen cantidades iguales de valores y, z.

Como ejercicio, compruebe que los números 21, 33, 57, 69 y 77 del ejemplo anterior son enteros de Blum y que, por el contrario, 15, 35 y 39 no lo son.

El algoritmo de Blum

1) A elige dos primos p y q de forma que n = p*q es un entero de Blum (p y q son congruentes con 3 mod 4)
2) A elige un elemento x de Z_n y calcula $y = x^2 \bmod n$. Luego calcula $z = y^2 \bmod n$, valor que envía a B.
3) B recibe z y apuesta por la paridad del valor y.
4) A le informa a B si ha acertado o no en su apuesta. Le muestra también el valor x elegido y el valor de y. Además le comprueba que n es un entero de Blum.
5) B comprueba que $y = x^2 \bmod n$ y que $z = y^2 \bmod n$.
6) A y B han actuado con una probabilidad del 50% en los pasos 2 y 3, respectivamente.

Ejemplo del algoritmo de Blum

Sean los primos p = 7 y q = 19
Luego, n = p*q = 7*19 = 133
Comprobación de que n = 133 es un entero de Blum:
7 mod 4 = 3; 19 mod 4 = 3 

- A elige el valor x = 41 y calcula:
  - $y = x^2 \bmod n$
    - $y = 41^2 \bmod 133 = 1.681 \bmod 133 = 85$
  - $z = y^2 \bmod n$
    - $z = 85^2 \bmod 133 = 7.225 \bmod 133 = 43$
- A envía a B el valor z = 43.
- B debe apostar por la paridad de y.

Conclusión del ejemplo de Blum

Situación 1 (B acierta)

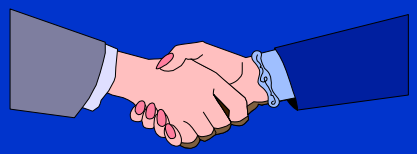
- Si B acierta y dice que y es impar, A no puede negarle que ha ganado. A debe mostrarle a B los valores x e y. Además debe demostrarle a B que n era un entero de Blum.

Situación 2 (B no acierta)

- Si B no acierta y dice que y es par, A le dice a B que ha perdido, le demuestra que n era un entero de Blum y le muestra el valor x elegido así como el valor y.

Compruebe que a iguales valores de resto principal y resto cuadrático se llega para x = 22, x = 92 y x = 111. Es decir, si se recibe z = 43 (impar) la única posibilidad es que el valor de y sea 85 (impar) y que A haya elegido como valor x alguno de éstos: 22, 41, 92 ó 111.

La firma de contratos



Dos personas desean firmar un contrato sin un ministro de fe.

- Deben cumplirse dos condiciones:

- Que los firmantes queden obligados a culminar la firma sólo a partir de un punto del protocolo. Esto se conoce como compromiso de los contratantes.
- Que la firma no pueda falsificarse y que, además, pueda ser comprobada por la otra parte.

Un posible algoritmo

Algoritmo básico de firma de contratos (1)

1. El usuario A elige dos claves i_A y j_A en un sistema de clave pública y calcula sus claves privadas i_A^{-1} y j_A^{-1}.
2. El usuario B elige una clave secreta K_B.
3. A envía a B sus dos claves públicas i_A y j_A.
4. B elige una de las dos claves recibidas y con ella cifra su clave K_B, enviando el resultado al usuario A.
5. A elige al azar una de sus dos claves privadas i_A^{-1} y j_A^{-1} y descifra con dicha clave el valor recibido en el punto 4.
6. A cifra el primer bloque del mensaje de firma usando el valor elegido en el punto 5 como clave y lo envía a B.
7. B descifrará con la clave recibida el bloque de firma.

Algoritmo básico de firma de contratos (2)

Observe que los siete pasos anteriores corresponden básicamente al algoritmo de transferencia inconsciente entre los usuarios A y B.

Finalización del protocolo

8. A repite la operación de los pasos 5 y 6 para cada uno de los bloques de su firma y B el paso 7.
9. Terminados los bloques de su firma, A repite el paso 6 utilizando ahora su otra clave privada y B el paso 7.

Algoritmo básico de firma de contratos (3)

- Si A y B han elegido al azar la misma clave con una probabilidad del 50% para cada uno, B descifrará un mensaje con sentido en la primera vuelta. En caso contrario, B recibe un texto sin sentido y deberá esperar hasta recibir el último bloque de la segunda vuelta para obtener el texto en claro.
- Sin embargo, A no tiene cómo saber en cuál de los dos pasos (en la primera o la segunda vuelta) ha logrado B descifrar el criptograma y obtener un texto con sentido lo que fuerza a ambas partes a terminar el algoritmo.

Firma de contratos: algoritmo de Even (1)

En el año 1985 Even, Goldreinch y Lempel proponen el uso de sistemas de cifra simétricos para la firma de contratos.

1. A elige un conjunto de 2n claves en un sistema simétrico: $C_1, C_2, ..., C_n, C_{n+1}, ..., C_{2n}$. Las claves se tomarán como parejas, esto es (C_1, C_{n+1}), (C_2, C_{n+2}), ..., (C_n, C_{2n}) aunque no tengan ninguna relación entre sí.
2. A cifra un mensaje estándar M_A conocido por B con 2n claves $E_{C1}(M_A), E_{C2}(M_A), ..., E_{C2n}(M_A)$ y le envía a B ordenados los 2n criptogramas.
3. A se comprometerá más adelante a la firma del contrato si B puede presentarle para algún i el par (C_i, C_{n+i}).

Firma de contratos: algoritmo de Even (2)

4. B elige también un conjunto de 2n claves de un sistema simétrico: D_1, D_2,..., D_n, D_{n+1}, ..., D_{2n} y las claves se tomarán como parejas (D_1, D_{n+1}), (D_2, D_{n+2}), ..., (D_n,D_{2n}). B cifra un mensaje estándar M_B conocido por A con las 2n claves $E_{D1}(M_B)$, $E_{D2}(M_B)$, ..., $E_{D2n}(M_B)$ y envía a A 2n criptogramas ordenados. B se comprometerá a la firma en los mismos términos que lo hizo A en el punto anterior.
5. A envía a B cada par (C_i, C_{n+i}) ordenados mediante una transferencia inconsciente; es decir enviando C_i o C_{n+i} con igual probabilidad. Lo mismo hace B enviando a A ordenadamente uno de los dos valores del par (D_i, D_{n+i}).

   En este punto A y B tienen la mitad de las claves del otro.

Firma de contratos: algoritmo de Even (3)

6. Si la longitud de cada clave C_i o D_i es de L bits, A y B realizan el siguiente bucle con $1 \leq i \leq 2n$ para la clave C_i y D_i que no han usado en los pasos anteriores:

   for $1 \leq j \leq L$
   begin
   A envía a B el bit jésimo de todas esas claves C_i
   B envía a A el bit jésimo de todas esas claves D_i
   end (Esto se conoce como compromiso bit a bit)

7. Al realizar el bucle completo, A y B tienen las 2n claves del otro y se supone firmado el contrato.

A y B pueden generar mensajes del tipo "Esta es mi mitad izquierda i de mi firma" para cifrar con la clave C_i y D_i y "Esta es mi mitad derecha i de mi firma" para cifrar con la clave C_{n+i} y D_{n+i}

Protocolo de firma ciega

Supongamos que Adela desea que Benito le firme algo pero sin que Benito se entere de qué es lo que está firmando. En este caso Benito actúa como un ministro de fe, autenticando a Adela.

Protocolo:

- Adela pone un documento dentro de un sobre.
- Adela cierra el sobre y se lo envía a Benito.
- Benito firma el sobre autenticando a Adela y se lo devuelve.
- Adela abre el sobre y demuestra que Benito al firmar en el sobre cerrado también ha firmado el documento que estaba en su interior.

En el anterior algoritmo, si Benito necesita una comprobación de la identidad Adela, ésta sencillamente incluye una firma digital suya en el sobre que le permita a Benito comprobar su autenticidad.

Algoritmo de firma ciega RSA (Chaum)

Adela desea que Benito le firme un documento M

- Adela (A) conoce las claves públicas de Benito (B: n_B, e_B)
- A elige un valor k de forma que mcd (k, n_B) = 1, calcula k^{-1} = inv (k, n_B) y luego enmascara su mensaje mediante la siguiente operación en n_B:
  - $t_A = M * k^{eB} \bmod n_B$ → y lo envía a B
- B firma el valor: $t_B = t_A^{dB} \bmod n_B$ → y lo envía a A
- A quita la máscara haciendo $s = t_B * inv(k, n_B) \bmod n_B$
- El resultado es que A tiene $M^{dB} \bmod n_B$, la firma de B.

Comprobación: $t_B = (M * k^{eB})^{dB} \bmod n_B = M^{dB} * k \bmod n_B$

Luego: $[M^{dB} * k * inv(k, n_B)] \bmod n_B = M^{dB} \bmod n_B$

Ejemplo de algoritmo de firma ciega

- Adelaida (A) desea que Benito (B) le firme el mensaje M = 65
- Claves públicas de B: $n_B = 299$, $e_B = 7$
- Clave privada y datos de B: $p_B = 13$; $q_B = 23$; $\phi(n_B) = 264$, $d_B = 151$
- A elige k / mcd (k, n_B), por ejemplo k = 60. Luego inv (k, n_B) = 5
- A enmascara el mensaje: $t_A = M * k^{eB} \bmod n_B = 65 * 60^7 \bmod 299$
- A envía a B: $t_A = 65*226 \bmod 299 = 39$
- B firma t_A con clave privada: $t_B = t_A^{eB} \bmod n_B = 39^{151} \bmod 299 = 104$
- A quita la máscara: $s = t_B * \text{inv}(k, n_B) = 104 * 5 \bmod 299 = 221$
- Este valor (221) es el mismo que se obtendría si B firmase su con clave privada el mensaje M, es decir $65^{151} \bmod 299 = 221$

¿Existe el correo electrónico certificado?

¿Cómo podemos estar seguros que un mensaje enviado por correo electrónico ha sido abierto y su contenido conocido sólo por su destinatario autorizado?

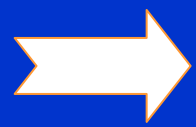
Para evitar estas situaciones podemos usar el protocolo del correo certificado

Los sistemas actuales de *e-mail* permiten emitir desde el cliente de correo del receptor un *acuse de recibo*. No obstante, esto sólo significa que "alguien" en extremo receptor desde el buzón de entrada pincha sobre un mensaje nuevo y a la pregunta ¿enviar acuse recibo al emisor? pulsa *Enter* eligiendo la opción Sí.

El correo electrónico certificado

- El usuario A desea enviar un mensaje electrónico como correo certificado al usuario B.
- El usuario A le descubre el mensaje (le envía la clave) sólo después de que el usuario B le envíe el acuse de recibo correspondiente. De la misma manera que actuamos ante un correo certificado: nos entregan "la multa" ✉ si primero firmamos.
- El algoritmo será muy similar al anterior de firma de contratos propuesto por Even.

Veamos una implementación del algoritmo



Un algoritmo de correo certificado (1)

- A elige de forma aleatoria n+1 claves (a_0, a_1, a_2, ... a_n) de un sistema de cifra simétrico. Las claves a_i no están relacionadas.
- Con la clave a_0 A cifrará el documento o carta, $C_0 = E_{a0}(M)$ y se lo envía a B.
- Las claves (a_1, a_2, ... a_n) serán la parte izquierda de la clave KI_{Ai}.
- A calcula $a_{n+i} = a_0 \oplus a_i$ para $1 \leq i \leq n$, obteniendo así la parte derecha de la clave (a_{n+1}, a_{n+2}, ... a_{2n}) es decir KD_{Ai}.
- A y B se ponen de acuerdo en un mensaje estándar de validación, por ejemplo V = "Mensaje de Validación".
- A cifra el mensaje de validación V con las 2n claves secretas, es decir n claves KI_{Ai} y n claves KD_{Ai}.
- Cifrado de validación de la parte izquierda: $VI_{Ai} = E_{KIAi}(V)$.
- Cifrado de validación de la parte derecha: $VD_{Ai} = E_{KDAi}(V)$.
- A envía a B los pares ordenados (VI_{Ai}, VD_{Ai}) para i = 1, 2, ... n.

Un algoritmo de correo certificado (2)

- B genera de forma similar n parejas de claves KI_{Bi} y KD_{Bi}, 2n claves.
- B genera n parejas de mensajes "Acuse de Recibo de la parte i Izquierda" (RI_i) y "Acuse de Recibo de la parte i Derecha" (RD_i).
- B cifra las parejas (RI_i, RD_i) con un sistema simétrico usando las claves KI_{Bi} y KD_{Bi}.
- B envía a A las parejas ordenadas (IB_i, DB_i) = [E_{KIBi}(RIi), E_{KDBi}(RDi)].
- Mediante una trasferencia trascordada A envía a B una de las dos claves secretas (K_{IA1} o K_{DA1}) y lo mismo hace B que envía a A (K_{IB1} o K_{DB1}).
- Este proceso se repite hasta que se envían los n valores de claves.
- B usa las claves enviadas por A en el paso anterior para comprobar que al descifrar $D_{KIAi}(V_{Ai})$ o $D_{KDAi}(V_{Ai})$ obtiene el Mensaje de Validación.
- A usa las claves enviadas por B en el paso anterior para comprobar que al descifrar $D_{KIBi}(I_{Bi})$ o $D_{KDBi}(I_{Bi})$ obtiene siempre RI_i o RD_i.

Un algoritmo de correo certificado (3)

- No pueden hacer trampa. A y B ya tienen información suficiente para demostrar ante un ministro de fe que el otro no ha seguido el protocolo.
- A y B se intercambian ahora bit a bit todos los bits de las claves de forma alterna. El primer bit de KI_{A1}, el primer bit de KI_{B1}, el primer bit de KI_{A2}, el primer bit de KI_{B2}, ... el primer bit de KD_{A1}, etc.
- Este paso se conoce como compromiso bit a bit entre A y B.
- A obtiene todas las claves de B y comprueba todos los Acuse de Recibo pareados, la parte i Izquierda y su correspondiente parte i Derecha.
- B obtiene todas las claves de A y comprueba que todos los envíos de A contienen el Mensaje de Validación. A deberá mostrar todas sus claves a B para que B compruebe que A ha usado la función $a_{n+1} = a_0 \oplus a_i$.
- Como B tiene todas las claves de A calcula ahora $a_0 = KI_{Ai} \oplus KD_{Ai}$. Para ello cualquiera de las parejas de Acuse de Recibo son válidas.
- B descifra el criptograma $Da_0(C_0) = M$ y recupera el mensaje ✉.

El póker mental con cifra simétrica (1)

♣ ♦ ♠ ♥ Se trata de encontrar un protocolo que permita el juego del póker a través de una red de computadores. Se debe asegurar un juego limpio y sin trampas. Aunque el número de jugadores puede ser cualquiera,veremos un ejemplo sólo para dos jugadores.

1. A y B usan un sistema de cifra simétrica que tenga propiedades conmutativas, usando claves K_A y K_B respectivamente.
2. B cifra -acción mezcla- las 52 cartas (codificadas con un número aleatorio c_i) con su clave secreta K_B: $E_{KB}(c_i)$ y las envía a A.
3. A elige al azar 5 valores y envía a B: $E_{KB}(c_{B1}, c_{B2}, c_{B3}, c_{B4}, c_{B5})$.
4. B recibe estos valores y los descifra con su clave secreta K_B. Así obtiene: $D_{KB}[E_{KB}(c_{B1}, c_{B2}, c_{B3}, c_{B4}, c_{B5})] = c_{B1}, c_{B2}, c_{B3}, c_{B4}, c_{B5}$. Estas cinco cartas c_{Bi} corresponden a la mano de B.

El póker mental con cifra simétrica (2)

5. A elige otras cinco cartas de las 47 restantes, las cifra con su clave secreta K_A y envía a B: $E_{KA}[E_{KB}(c_{A1}, c_{A2}, c_{A3}, c_{A4}, c_{A5})]$.
6. B descifra con su clave secreta K_B la cifra anterior y envía a A el resultado: $E_{KA}(c_{A1}, c_{A2}, c_{A3}, c_{A4}, c_{A5})$.
7. A descifra lo anterior con su clave secreta K_A y obtiene su mano c_{Ai}: $D_{KA}[E_{KA}(c_{A1}, c_{A2}, c_{A3}, c_{A4}, c_{A5})] = c_{A1}, c_{A2}, c_{A3}, c_{A4}, c_{A5}$.
8. Las restantes 42 cartas permanecerán en poder de A que es quien las reparte. Estas cartas siguen cifradas con la clave de B.
9. Si los jugadores desean cambiar algunas cartas, siguen el mismo procedimiento anterior.

Como ejercicio, intente generalizar este esquema para cuatro jugadores.

Esquema con cifra asimétrica

Póker mental con cifra asimétrica RSA (1)

En este caso se usará un sistema RSA en el que el módulo de trabajo n será compartido y el par de claves asimétricas de cada jugador, e y d, serán ambas secretas. Veamos un ejemplo para 4 jugadores.

1. El jugador A que repartirá las cartas, todas ellas codificadas con un número aleatorio c_i, las mezclará cifrándolas con su clave pública e_A: $E_{eA}[c_1, c_2, c_3, \ldots c_{50}, c_{51}, c_{52}]$ y las envía a B.
2. B elige cinco cartas, las cifra con su clave pública e_B y devuelve a A : $E_{eB}\{E_{eA}[c_{B1}, c_{B2}, c_{B3}, c_{B4}, c_{B5}]\}$.
3. A descifra lo recibido con su clave privada d_A y se lo envía a B: $E_{eB}[c_{B1}, c_{B2}, c_{B3}, c_{B4}, c_{B5}]$.
4. B descifra ahora con su clave privada d_B lo recibido y se queda con su mano $c_{Bi} = c_{B1}, c_{B2}, c_{B3}, c_{B4}, c_{B5}$.

Póker mental con cifra asimétrica RSA (2)

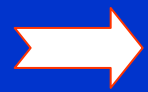
5. El jugador B pasa las restantes 47 cartas al jugador C y se repiten los pasos 2 al 4 anteriores entre C y A, usando ahora las claves e_C, d_A y d_C.
6. Terminado el paso 5, el jugador C tendrá entonces como mano $c_{Ci} = c_{C1}, c_{C2}, c_{C3}, c_{C4}, c_{C5}$.
7. El jugador C pasa las restantes 42 cartas al jugador D y se repiten los pasos 2 al 4 entre D y A, usando ahora las claves e_D, d_A y d_D.
8. Terminado el paso 7, el jugador D tendrá entonces como mano $c_{Di} = c_{D1}, c_{D2}, c_{D3}, c_{D4}, c_{D5}$.
9. El jugador D devuelve las 37 cartas que quedan y que están cifradas con su clave pública: $E_{eD}\{E_{eA}[c_1, c_2, c_3, \ldots c_{36}, c_{37}]\}$ al jugador A.

Póker mental con cifra asimétrica RSA (3)

10. El jugador A elige 5 cartas entre las 37 y devuelve al jugador D: $E_{eD}\{E_{eA}[c_{A1}, c_{A2}, c_{A3}, c_{A4}, c_{A5}]\}$.
11. El jugador D descifra con su clave privada d_D lo recibido y envía a A: $E_{eA}[c_{A1}, c_{A2}, c_{A3}, c_{A4}, c_{A5}]$.
12. El jugador A descifra con su clave privada d_A lo recibido y se queda con su mano $c_{Ai} = c_{A1}, c_{A2}, c_{A3}, c_{A4}, c_{A5}$.
13. Todos tienen su mano de juego. Las restantes 32 cartas quedan en poder de A cifradas por D y A: $E_{eD}\{E_{eA}[c_1, c_2, \ldots c_{31}, c_{32}]\}$.
14. Si un jugador X desea descarte, pide las cartas a A, elige las que desea, las cifra con su clave pública e_X y se las devuelve a A, quien las envía a D para que descifre con su clave privada d_D. D las devuelve a A para que descifre con su clave privada d_A y A envía a X: E_{eX}[cartas elegidas en su descarte].

El canal subliminal

- Como ejemplo de canal subliminal, en un supermercado podrían incluir en la música ambiental una información no audible y que sólo nuestro subconsciente sea capaz de interpretar. No se extrañe de ello, este tipo de experimentos se han probado hace muchos años atrás.
- El concepto de canal subliminal fue propuesto por Gustavus Simmons en 1983. Se conoce también como el problema de los prisioneros.



- Dos prisioneros cómplices de un delito son encarcelados en celdas separadas. Si entre ellos pueden intercambiarse mensajes a través de un carcelero que los puede leer, ¿cómo hacen para que esos mensajes en principio inocentes, lleven de forma subliminal un mensaje cifrado y que el carcelero sea incapaz de dilucidar ese secreto?

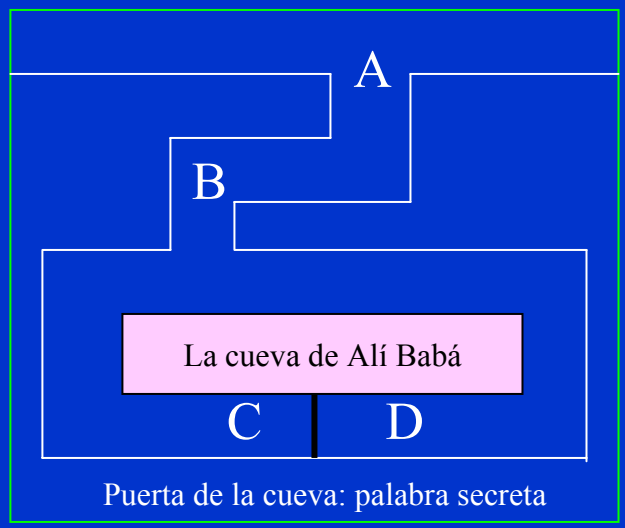
  La técnica denominada esteganografía, hoy en día de moda, realiza una operación similar, normalmente ocultando un texto bajo una fotografía.

El problema de los prisioneros

- El prisionero A genera un mensaje inocente M que desea enviar al prisionero B a través del guardia.
- Utilizando una clave secreta K acordada con anterioridad, el prisionero A "firma" el mensaje de forma que en esa firma se esconda el mensaje subliminal.
- El guardia recibe el mensaje "firmado" por A y como no observa nada anormal se lo entrega al prisionero B.
- El prisionero B comprueba la firma de su compañero A, autentica el mensaje y lee la información subliminal en M.

Existen varios esquemas de uso del canal subliminal para proteger la información, entre ellos el propio esquema de Simmons basado en la factorización de un número grande n compuesto por tres primos p, q y r. Habrá por tanto $2^3 = 8$ raíces de las cuales sólo algunas se usarán como valores válidos y otras no. Hace uso del Teorema del Resto Chino.

Transferencia con conocimiento nulo TCN



Este modelo fue presentado por J. Quisquater y L. Guillou en Crypto '89 para explicar el protocolo de transferencia con conocimiento cero o nulo.

Algoritmo:

1. Mortadelo y Filemón se acercan a la cueva en el punto A.
2. Mortadelo se adentra en la cueva hasta llegar al punto C o D.
3. Filemón se acerca al punto B de la cueva y le pide a Mortadelo que salga por la ladera derecha o izquierda, según desee.
4. Mortadelo satisface la petición de Filemón y sale por la ladera que éste le ha solicitado, usando si es menester la palabra secreta para abrir la puerta.
5. Se repite el proceso desde el comienzo hasta que Filemón se convence que Mortadelo conoce la palabra secreta.

Esquema de TCN de Koyama

- A desea demostrar a B que conoce la clave secreta RSA de un tercer usuario C, es decir d_C. Como es lógico también conocerá p_C, q_C y $\phi(n_C)$. Las claves públicas de C son n_C y e_C que conocen tanto A como B.
- A y B se ponen de acuerdo y eligen dos valores aleatorios k y m con la condición de que $k*m = e_C \bmod \phi(n_C)$.
- Como A debe mantener en secreto el valor de $\phi(n_C)$ le propone a B que en cada ejecución del algoritmo elija un número m primo por lo que A calcula $k = [\{inv\ (m, \phi(n_C)\}*e_C] \bmod \phi(n_C)$.
- A propone a B un texto aleatorio M o bien A y B generan este texto usando, por ejemplo, un algoritmo de transferencia trascordada.
- Usando la clave privada d_C de C, ahora A calcula $C = M^{dC} \bmod n_C$. Luego calcula $X = C^k \bmod n_C$ y envía el valor X a B.
- B recibe X y comprueba si $X^m \bmod n_C$ es igual al texto M. Si es así, quiere decir que A ha usado d_C, la clave privada de C.
- Se repite el proceso las veces que haga falta hasta que B acepte que A conoce clave privada de C.

¿Por qué funciona el esquema de Koyama?

Por simplicidad supondremos que los datos de C no tienen subíndice:

1. A conoce n, e, d, p, q, $\phi(n)$ y el texto M; B conoce n, e y el texto M.
2. B elige un primo m y se lo envía envía a A.
3. A calcula $k = [\{inv\ (m, \phi(n)\} * e] \bmod \phi(n)$.
4. A calcula $C = M^d \bmod n$ y $X = C^k \bmod n = M^{dk} \bmod n$ y envía este valor X a B.
5. B recibe X y calcula $X^m \bmod n = M^{(dk)m} \bmod n = M^{km*d} \bmod n$, pero como $k*m = e \bmod \phi(n)$ entonces $M^{km*d} \bmod n = M^{e*d} \bmod n = M$.
6. La única posibilidad para que B recupere el texto M en el paso 5, es que A haya usado en la cifra del paso 4 la clave privada d.
7. Si B no se convence en el primer intento, ambos repiten el algoritmo con valores primos m distintos en cada iteración, hasta que se cumpla un umbral ante el que B acepte que A está en posesión de ese secreto.

Ejemplo del esquema de TCN de Koyama

- Supongamos que A desea demostrar a B que conoce la clave privada de C. Los valores públicos de C son n = 77, e = 13.
- El mensaje M acordado por A y B es la palabra PADRINO con la codificación que se muestra a continuación:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

- Supongamos que B elige como valor aleatorio m = 29.
- A calcula k según el algoritmo de Koyama y para cada valor M_i del mensaje (P = 18, A = 2, D = 5, etc.) calcula primero $C = M_i^d \bmod n$ y luego $X = C^k \bmod n = 30, 39, 31, 27, 54, 36, 68$ que envía a B.
- B calcula $30^{29} \bmod 77$, $39^{29} \bmod 77$, $31^{29} \bmod 77$, $27^{29} \bmod 77$, $54^{29} \bmod 77$, $36^{29} \bmod 77$, $68^{29} \bmod 77$ y obtiene la cadena de caracteres PADRINO.
- El protocolo puede repetirse para otros valores primos m que elija B y siempre se obtendrá como resultado el mismo mensaje M. ✌

Solución del ejemplo de TCN de Koyama

- Como n = 77, es obvio que p = 7, q = 11, $\phi(n) = 60$. Por lo tanto, puesto que e = 13 entonces d = inv {e, $\phi(n)$} = inv (13, 60) = 37.
- $M_1 = 18$; $M_2 = 2$; $M_3 = 5$; $M_4 = 20$; $M_5 = 10$; $M_6 = 15$; $M_7 = 17$.
- $C_1 = 18^{37} \bmod 77 = 39$; $C_2 = 2^{37} \bmod 77 = 51$; $C_3 = 5^{37} \bmod 77 = 47$; $C_4 = 20^{37} \bmod 77 = 48$; $C_5 = 10^{37} \bmod 77 = 10$; $C_6 = 15^{37} \bmod 77 = 71$; $C_7 = 17^{37} \bmod 77 = 52$.
- $k = [\{\text{inv}(m, \phi(n)\} * e] \bmod \phi(n) = \text{inv}(29, 60) * 13 \bmod 60 = 17$.
- $X_1 = 39^{17} \bmod 77 = 30$; $X_2 = 51^{17} \bmod 77 = 39$; $X_3 = 47^{17} \bmod 77 = 31$; $X_4 = 48^{17} \bmod 77 = 27$; $X_5 = 10^{17} \bmod 77 = 54$; $X_6 = 71^{17} \bmod 77 = 36$; $X_7 = 52^{17} \bmod 77 = 68$. Luego X = 30, 39, 31, 27, 54, 36, 68.
- $30^{29} \bmod 77 = 18 =$ P; $39^{29} \bmod 77 = 2 =$ A; $31^{29} \bmod 77 = 5 =$ D
  $27^{29} \bmod 77 = 20 =$ R; $54^{29} \bmod 77 = 10 =$ I; $36^{29} \bmod 77 = 15 =$ N
  $68^{29} \bmod 77 = 17 =$ O.

En este ejemplo el valor de n es muy pequeño y resulta muy fácil romper la clave privada simplemente factorizando el módulo ☺.

El voto electrónico o por ordenador

- Todos tenemos de una u otra forma una idea intuitiva, aunque quizás no completa, sobre cómo se desarrolla un proceso electoral.
- La pregunta es si es posible realizar este tipo de eventos desde Internet, lo que se conoce como esquema electoral.
- La respuesta es sí con la ayuda de técnicas y protocolos criptográficos aunque no se trata sólo de un problema de implementación técnica; es menester tener en cuenta otros factores importantes, a saber:
  - Socio-políticos, económicos, jurídicos, legislativos...

Definición de esquema electoral

"Un esquema de votación electrónica es una aplicación distribuida y constituida por un conjunto de mecanismos criptográficos y protocolos que, de forma conjunta, permiten que se realicen elecciones en una red de computadores, de forma segura, incluso suponiendo que los electores legítimos pueden tener un comportamiento malicioso."

Andreu Riera (Tesis Doctoral, UAB, 1999)

Requisitos de un esquema electoral (1)

Requisitos de un esquema electoral:

- Sólo pueden votar quienes estén censados.
- El voto debe ser secreto.
- El voto debe ser único por cada votante.
- Se contabilizarán todos los votos válidos.
- El recuento parcial no debe afectar a votos que se emitan con posterioridad.

sigue

Requisitos de un esquema electoral (2)

Requisitos de un esquema electoral:

Cada votante podrá comprobar que su voto ha sido tenido en cuenta en el escrutinio.

Esto último es muy importante

Y, además:

Se debe proteger el proceso contra ataques en red.

El proceso debe ser factible, práctico y dentro de lo posible de uso universal.

Primera aproximación del voto electrónico

MCV = Mesa Central de Votación

- El votante cifra su voto con la clave pública de MCV.
- El votante envía su voto a la MCV.
- La MCV descifra el voto y lo contabiliza.
- La MCV hace público el resultado.

¿Qué problemas presenta este esquema? TODOS... ☹

La MCV no sabe de dónde vienen los votos, si éstos son válidos o no y si alguien vota más de una vez. Además puede conocer la identidad del votante por lo que se vulnera el secreto del voto. Lo único que aquí se protege es el secreto del voto ante terceros.

Segunda aproximación del voto electrónico

MCV = Mesa Central de Votación

- El votante firma su voto con su clave privada y lo cifra luego con la clave pública de MCV.
- El votante envía su voto a la MCV.
- La MCV descifra el voto, lo contabiliza y hace público el resultado.

¿Qué problema tenemos ahora?

En este nuevo esquema se satisface que cada votante autorizado vote una sola vez, no obstante seguimos vulnerando el secreto del voto ante la MCV.

Tercera aproximación del voto electrónico

El tercer esquema contempla dos mesas:

- MCV = Mesa Central de Votación
- MCL = Mesa Central de Legitimación

  - Evita que la MCV conozca a quién ha votado el votante, mediante un protocolo entre ambas, y además gestionan una lista de votantes censados.

☞ MCV y MCL deben ser órganos independientes

Veamos cómo funciona este esquema

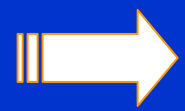
Un protocolo de voto electrónico (1)

1. El votante A envía a la MCL el mensaje:

   Buenos días, soy A y vengo a votar.
2. La MCL verifica si A está censado. Si no es un votante legítimo rechaza la solicitud. Si es legítimo, le envía un número aleatorio de identificación único i(A) y le borra de la lista para impedir que vuelva a votar.

Toda la información irá cifrada y firmada

Características de i(A)



Un protocolo de voto electrónico (2)

Un protocolo de voto electrónico (3)

3. La MCL envía a la MCV la lista de números de validación.
4. El votante A escoge una identificación secreta s(A) y envía a la MCV el mensaje formado por el trío [i(A), v(A), s(A)] es decir:
   - su identificación i(A)
   - su voto v(A)
   - su número secreto s(A)

Puede generarlo internamente con su sistema de cifra. Será también un valor de muchos dígitos.

Un protocolo de voto electrónico (4)

5. La MCV verifica que el número i(A) de identificación se encuentra en el conjunto N de los números censados y cruza los datos para evitar que se vote más de una vez. Quita i(A) del conjunto N y añade s(A) al conjunto de electores que han optado por la opción v(A).
6. La MCV contabiliza los votos y hace público el resultado, junto con la lista de números secretos s(A) que han votado a la opción v(A) ... luego →

Un protocolo de voto electrónico (5)

- ☺ Cada elector puede comprobar si su voto ha sido contabilizado sin hacer pública su opción.

  ¿Qué pasa si MCV y MCL no son independientes?

  Si las dos mesas, MCV y MCL, no tienen la idoneidad y la integridad que se presume, la solución está en el uso de una diversidad de esquemas más desarrollados que evitan esta anomalía mediante protocolos, entre ellos ANDOS (All-or-Nothing Disclosure Of Secrets) Distribución Anónima de Números de Validación, pero esto ya se escapa del objetivo de este libro.

Otros esquemas de mesas electorales

Hay muchos otros esquemas con dos mesas, una única mesa e incluso ninguna, cada uno con sus características propias. Entre ellos tenemos:

- Modelo de Cohen y Fisher (1985)
- Modelo de Fujioka y otros (1992)
- Modelo de Park y otros (1993)
- Modelo de Sako y Killian (1995)
- Modelo de Borrel y Rifà (1996)

Observe que son modelos y esquemas muy recientes.

Estado del arte en voto electrónico

- Existen diversos modelos y esquemas, algunos de ellos probados con éxito con un número reducido de electores.
- No está todavía bien solucionado el problema de la protección física y lógica de la red ante ataques masivos, denegación de servicio, etc. Es el principal problema al que se enfrentan estos esquemas, su difícil escalabilidad en sistemas grandes y abiertos.
- No obstante, el proceso de unas elecciones vía Internet realizable, práctico y seguro en cuanto a la privacidad y autenticidad, es completamente factible.

Fin del Tema 17

Cuestiones y ejercicios (1 de 4)

1. ¿Qué diferencia hay entre un protocolo de red como por ejemplo TCP/IP con un protocolo criptográfico?
2. En una transferencia inconsciente de Rabin, A y B se intercambian lo siguiente. A envía a B el número compuesto n = 55, B elige el valor x = 9 y envía x^2 mod n a A. ¿Qué valores de los 4 que puede devolver A a B permiten a este último factorizar el cuerpo n?
3. ¿Qué sucede si en el ejemplo anterior B elige x = 10?
4. ¿En el ejemplo anterior, están bien elegidos por A los valores de p y q? Qué valores usaría si p y q fuesen números mayores que 10?
5. Presente una solución al problema del lanzamiento de la moneda a través del esquema de transferencia inconsciente de Rabin.
6. Calcule todos los valores de x^2 mod 13. Sea a = 2, 3, 4, 5, 6. ¿Cuáles son restos cuadráticos de Blum en el cuerpo n = 13?, ¿por qué?

Cuestiones y ejercicios (2 de 4)

7. Para los restos cuadráticos encontrados en el ejercicio anterior, ¿se cumple la paridad en el valor de x? ¿Qué significa esto?
8. ¿Cuáles de los siguientes siete números compuestos son enteros de Blum: 69, 143, 161, 189, 319, 713, 1.333? ¿Justifíquelo?
9. Encuentre todos los restos de y, z para el entero de Blum n = 33.
10. En un protocolo con enteros de Blum, A trabaja en n = 77 y elige el valor x = 15. Calcula $y = x^2 \bmod n$ y luego $z = y^2 \bmod n$. Envía el valor z a B. ¿Cuál es el escenario del protocolo y cómo trabaja?
11. ¿Qué sucede si en el esquema anterior de Blum el usuario B conoce el valor de los primos p y q? ¿Funciona así el protocolo?
12. En el algoritmo de firma de contratos con claves asimétricas y una clave simétrica, ¿cómo puede comprobar el usuario B que A está usando en cada vuelta una clave privada distinta y no hace trampa?

Cuestiones y ejercicios (3 de 4)

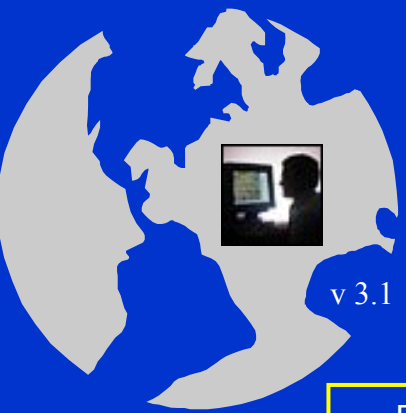
13. ¿Cómo se entiende el compromiso de firma de A y B en el esquema de firma de contratos de Even?
14. En el esquema anterior de Even ¿qué relación tiene el compromiso bit a bit con el término correcto del protocolo? ¿Por qué están A y B obligados a terminar el protocolo hasta el último bit?
15. Se desea que el usuario B le firme de forma ciega al usuario A el mensaje M = 100. Si $n_B = 253$, $e_B = 19$ y el usuario A elige k = 25, realice y compruebe el protocolo de firma ciega.
16. ¿Para qué podría servir un protocolo como el de firma ciega?
17. ¿Por qué decimos que el actual acuse de recibo de los clientes de correo electrónico no corresponde a uno verdadero?
18. En el algoritmo de correo con acuse de recibo, compruebe que B obtiene la clave de descifrado del mensaje haciendo $KI_{Ai} \oplus KD_{Ai}$.

Cuestiones y ejercicios (4 de 4)

19. Generalice el póker mental con cifra simétrica para 4 jugadores.
20. ¿Qué diferencia hay en cuanto a la elección de cartas de una mano entre el esquema de póker mental con cifra simétrica y el esquema con cifra asimétrica? ¿Es esto un inconveniente o no?
21. En el esquema de Quisquater y Guillou de conocimiento nulo, si Mortadelo y Filemón repiten el protocolo 20 veces, ¿cuál es la probabilidad de que el primero engañe al segundo?
22. Usando el software Fortaleza de la asignatura (ver Web) , repita el ejercicio de TCN de Koyama con n = 465.256.980.233 y e = 4.171. B elige el valor m = 131, el mensaje M es el mismo y se recibe: X_1 = 394.106.275.745; X_2 = 342.981.204.125; X_3 = 49.911.481.740; X_4 = 366.983.136.296; X_5 = 56.903.681.682; X_6 = 246.374.030.904; X_7 = 254.152.395.874. ¿Qué valor tiene la clave privada d?

Tema 18
Bibliografía, Enlaces, SW y Tablas

Seguridad Informática y Criptografía



Material Docente de
Libre Distribución

Ultima actualización: 03/03/03
Archivo con 50 diapositivas

Jorge Ramió Aguirre
Universidad Politécnica de Madrid

Bibliografía recomendada en castellano (1)

Bibliografía en castellano comentada ordenada alfabéticamente

Caballero, Pino

INTRODUCCIÓN A LA CRIPTOGRAFÍA. SEGUNDA EDICIÓN

Editorial Ra-Ma, Textos Universitarios, Madrid

Año 2002 (133 páginas)

Libro de introducción a las técnicas criptográficas que presenta estos temas bajo una orientación matemática clara y precisa. Actualización de la primera edición de 1996 en la que trata los temas de criptografía teórica, criptografía de clave secreta y pública, problemas de autenticación y accesos y algunas aplicaciones criptográficas. Para un buen seguimiento de la lectura, en algunos apartados es recomendable contar con una base de conocimientos en matemáticas a nivel universitario.

Bibliografía recomendada en castellano (2)

Fúster, Amparo; De la Guía, Dolores; Hernández, Luis; Montoya, Fausto; Muñoz, Jaime

TÉCNICAS CRIPTOGRÁFICAS DE PROTECCIÓN DE DATOS. SEGUNDA EDICIÓN

Editorial Ra-Ma

Año 2000 (372 páginas)

Detallado y actualizado resumen de las técnicas de cifra modernas, profundizando en los sistemas de clave secreta con cifra en flujo y en bloque, de clave pública y sus aplicaciones en redes. De especial interés resulta el capítulo dedicado a protocolos criptográficos y sus apéndices en donde explica los métodos matemáticos usados en criptografía y nociones sobre complejidad computacional. La segunda edición incluye además una colección de problemas y sus soluciones en un disquete.

Bibliografía recomendada en castellano (3)

Morant Ramón, J.L.; Ribagorda Garnacho, A.; Sancho Rodríguez J.

SEGURIDAD Y PROTECCIÓN DE LA INFORMACIÓN

Colección de Informática, Editorial Centro de Estudios Ramón Areces, S.A., Madrid

Año 1994 (388 páginas)

Libro que trata, además de los temas genéricos de la criptografía clásica y moderna, aspectos de seguridad en sistemas operativos, en bases de datos y en redes de computadores. Buen texto descriptivo que profundiza en ciertos aspectos matemáticos y hace un buen estudio de la gestión de claves. Tiene además como característica ser el primer libro con formato universitario sobre criptografía y seguridad informática en España, y seguramente de lengua española.

Bibliografía recomendada en castellano (4)

Pastor, José; Sarasa, Miguel Angel

CRIPTOGRAFÍA DIGITAL. FUNDAMENTOS Y APLICACIONES

Prensas Universitarias de Zaragoza

Año 1998 (597 páginas)

Extenso y completo texto sobre técnicas criptográficas modernas, con una buena cantidad de ejemplos y profusión de tablas con datos de interés. Destacan los capítulos de cifra con clave secreta, en donde se estudian más de una docena de criptosistemas, los de clave pública y su aplicación en firmas digitales y un capítulo dedicado a protocolos criptográficos. En particular, están muy bien tratados los apéndices con temas matemáticos así como los algoritmos de factorización y del logaritmo discreto, algo no muy común y que se agradece. Para el buen seguimiento es necesario contar con una buena base matemática.

Bibliografía recomendada en castellano (5)

Singh, Simon (traducción de José Ignacio Moraza)

LOS CÓDIGOS SECRETOS

Editorial Debate S.A.

Año 2000 (382 páginas)

Interesante y completo libro de Simon Singh editado en 1999 en el que se hace un repaso extenso de la criptografía denominada clásica, máquinas y artilugios de cifra, máquina Enigma, etc., desde una perspectiva un tanto novelesca que, sin desmerecer en absoluto la calidad técnica del mismo, lo convierte en un excelente libro de lectura. Encontrará en él una interesante presentación de los sistemas de cifra asimétrica, la historia inmersa en la búsqueda de la criptografía de clave pública y el intercambio de clave, para terminar con PGP y un capítulo dedicado a la criptografía cuántica.

Bibliografía recomendada en inglés (1)

Bibliografía en inglés comentada ordenada alfabéticamente

Denning, Dorothy

CRYPTOGRAPHY AND DATA SECURITY

Addison-Wesley Publishing Company, London

Año 1982 (400 páginas)

Libro clásico de criptografía, si bien antiguo en comparación con la bibliografía actual. Trata los algoritmos y técnicas criptográficas clásicas y modernas, control de accesos y de flujo. Tiene un buen número de ejemplos resueltos y ejercicios propuestos aunque sin sus soluciones. De especial interés son algunos algoritmos desarrollados en PASCAL. Es, no obstante, un referente básico de acuerdo a su fecha de edición.

Bibliografía recomendada en inglés (2)

Menezes, Alfred; Oorsschof, Paul; Vanstone, Scott

HANDBOOK OF APPLIED CRYPTOGRAPHY

CRC Press Inc.

Año 1997 (780 páginas)

Interesante y completo libro dedicado al estudio de los algoritmos con una visión matemática de alto nivel. Sus capítulos están orientados a bases matemáticas para la criptografía, sistemas de claves secretas y públicas, cifradores de flujo y de bloque, hash, autenticación, firma digital y gestión de claves. Obra imprescindible para el estudiante universitario que desea profundizar en el análisis de los algoritmos, si bien el seguimiento del mismo puede resultar algo complejo por el nivel matemático comentado. Le recomiendo que descargue el libro de forma gratuita desde la página Web de su autor http://www.cacr.math.uwaterloo.ca/hac/.

Bibliografía recomendada en inglés (3)

Pflegger P., Charles

SECURITY IN COMPUTING

Prentice-Hall International Editions, London

Año 1989 (538 páginas)

Texto de consulta general sobre seguridad informática que trata los criptosistemas y además estudia la seguridad en los programas, en informática personal, en las comunicaciones, análisis de riesgos, así como los aspectos legales y éticos de esta especialidad. Incluye varios ejercicios propuestos sin incluir sus soluciones. Debido a la fecha de edición, no profundiza en aspectos de cifra asimétrica y algoritmos actuales.

Bibliografía recomendada en inglés (4)

Salomaa, Arto

PUBLIC-KEY CRYPTOGRAPHY. SECOND EDITION

EATCS Monographics on Theoretical Computer Science

W. Brauer, G. Rozenberg, A. Salomaa (Eds.), Springer-Verlag, New York

Año 1996 (268 páginas)

Interesante y ameno, profundiza en los criptosistemas de clave pública y protocolos criptográficos en esta segunda edición. En algunos capítulos es necesario contar con una base matemática de nivel universitario para una mejor comprensión. No obstante, su lectura es muy agradable y presenta algunos ejemplos resueltos. Incluye un amplio estudio de los sistemas de mochilas, su implementación, debilidades, tipos de ataques, etc.

Bibliografía recomendada en inglés (5)

Schneier, Bruce

APPLIED CRYPTOGRAPHY: PROTOCOLS, ALGORITHMS, AND SOURCE CODE IN C. SECOND EDITION

John Wiley & Sons, Inc., New York

Año 1996 (758 páginas)

Segunda edición del libro con mismo título, con mayor énfasis en los algoritmos de cifra, protocolos, firmas, etc., especialmente en los sistemas con clave privada y pública. El texto de Schneier resulta fundamental para los estudiantes universitarios de asignaturas de criptografía. Incluye una bibliografía es muy completa, hay profusión de tablas y estudia una infinidad de criptosistemas, adjuntando el código fuente en C de muchos de los algoritmos estudiados. Se echa en falta, no obstante, que no tenga algunos ejemplos resueltos en cada capítulo.

Bibliografía recomendada en inglés (6)

Seberry, Jennifer; Pieprzyk, Josef

CRYPTOGRAPHY. AN INTRODUCTION TO COMPUTER SECURITY

Prentice-Hall, New York

Año 1989 (375 páginas)

Además de los temas propios de criptografía clásica y moderna, incluye un capítulo de introducción a la aritmética modular bien estructurado. Trata también la seguridad informática en bases de datos, en sistemas operativos y en redes. Como lector se agradece en especial la gran cantidad de ejercicios propuestos y resueltos en cada capítulo, incluyendo el código en PASCAL.

Bibliografía recomendada en inglés (7)

Stallings, William

CRYPTOGRAPHY AND NETWORK SECURITY. PRINCIPLES AND PRACTICE. THIRD EDITION

Prentice-Hall Inc.

Año 2002 (696 páginas)

Con 130 páginas más que la segunda edición de mismo título (1999), el texto está estructurado de una forma óptima que permite una agradable lectura. Además de cifra simétrica y asimétrica, algoritmos, firmas, hash, autenticación y seguridad en redes, esta edición se centra en la seguridad en Internet, profundizando en temas como correo seguro, protocolos de redes, IP seguro, seguridad en Web, intrusiones, cortafuegos, etc. Incluye algunos ejemplos y ejercicios. Es para muchos probablemente el mejor libro de criptografía y seguridad informática en estos momentos.

Bibliografía de consulta y de interés (1)

Bibliografía interesante ordenada alfabéticamente por nombre del autor

- Amoroso, Edward. Fundamental of Computer Security Technology. Prentice Hall, 1994.
- Anderson, Ross. Security Engineering. John Wiley & Sons, 2001.
- Bauer, Friedrich. Decrypted Secrets. Methods and Maxims of Cryptology. Springer-Verlag, 1997.
- Brassars, Gilles. Cryptologie Contemporaine. Edit. Masson, Paris, 1993.
- Braun, Christoph. Unix System Security Essentials. Addison-Wesley, 1994.
- Buck, Edward. Introduction to Data Security & Controls. QED Technical Publishing Group, 1991.
- Contreras, J.M.; González, M.; Chamorro, R. Correo Electrónico en Internet. Paraninfo, 1997.
- Cobb, Sthephen. Manual de Seguridad para PC y Redes Locales. McGraw-Hill, 1994.

Bibliografía de consulta y de interés (2)

- Chapman, Brent; Zwicky, Elizabeth. Construya Firewalls para Internet. McGraw-Hill, 1997.
- Cheswick, William; Bellovin, Steven. Firewalls and Internet Security. Addison-Wesley, 1994.
- De Marcelo, Jesús. Virus de Sistemas Informáticos e Internet. Ra-Ma, 2000.
- Deavours, C.; Khan, D.; Kruh, L.; Mellen, G. Winkel, B. Cryptology: Machines, History & Methods. Artech House, 1989.
- Del Peso Navarro, Emilio; Ramos González, Miguel Angel. Confidencialidad y Seguridad de la Información: La LORTAD y sus implicaciones socioeconómicas. Díaz de Santos, 1994.
- Diffie, Whitfield; Landau, Susan. Privacy on the Line. The Politics of Wiretapping and Encryption. MIT Press, 1998.
- Farley, Marc; Stearns, Tom; Hsu, Jeffrey. Seguridad e Integridad de los Datos. McGraw-Hill, 1998.

Bibliografía de consulta y de interés (3)

- Feghhi, Jalal; Feghhi, Jalil; Williams, Peter. Digital Certificates. Applied Internet Security. Addison-Wesley, 1998.
- Fisher, Royal. Seguridad en los Sistemas Informáticos. Ediciones Díaz de Santos, 1988.
- Ford, Warmick. Computer Communications Security. Principles, Standards Protocols and Techniques. Prentice Hall, 1994.
- Hunter, John. An Information Security Handbook. Springer, 2001.
- Garfinkel, Simson; Spafford, Gene. Seguridad Práctica en Unix e Internet. McGraw-Hill, 1999.
- Giblin, Peter. Primes and Programming. An Introduction to Number Theory with Computing. Cambridge University Press, 1994.
- Golomb, Solomon. Shift Register Sequences. Aegean Park Press, 1982.
- Gollman, Dieter. Computer Securiry. John Wiley & Sons, 1999.
- Graff, Jon. Cryptography and E-Commerce. John Wiley & Sons, 2001.
- Jaworski, Jamie; Perrone, Paul. Seguridad en Java. Prentice-Hall, 2001.

Bibliografía de consulta y de interés (4)

- Klander, Lars. Hacker Proof. The Ultimate Guide to Network Security. Gulf Publishing Company, 1997. A Prueba de Hackers. Anaya, 1998.
- Knudsen, Jonathan. Java Cryptography. O'Reilly, 1998.
- Konheim, Alan G. Cryptography: A Primer. John Wiley & Sons, 1981.
- Loshin, Pete. Big Book of IPSec RFCs. Internet Security Architecture. Morgan Kaufmann, 2000.
- Madron, Thomas W. Network Security in the '90s. Issues and Solutions for Managers. John-Wiley & Sons, Inc., 1992.
- Menezes, Alfred. Elliptic Curve Public Key Cryptosystems. Kluwer Academic Publishers, Fourth Printing, 1997.
- Meyer, C.; Matyas, S. Cryptography: A New Dimension in Computer Data Security. John Wiley & Sons, 1982.
- New Riders (varios autores). Implementing Internet Security. New Riders Publishing NRP, 1995.
- Khan, David. The Codebreakers. The Story of Secret Writing. Macmillan Publishing Company, New York, 1967.

Bibliografía de consulta y de interés (5)

- Northcutt, Stephen; Novak, Judy. Detección de Intrusos. Segunda Edición. Guía Avanzada. Prentice Hall, 2001.
- Oppliger, Rof. Sistemas de Autentificación para Seguridad en Redes. Ra-Ma, 1998.
- Piattini, Mario, Del Peso, Emilio. Auditoría Informática. Un Enfoque Práctico. Ra-ma, 1998.
- Poe, Edgar Allan. El Escarabajo de Oro. Anaya Tus Libros, 1995.
- Ribagorda, Arturo. Glosario de Términos de Seguridad de las T.I. Ediciones CODA, 1997.
- Ribenboim, Paulo. The Little Book of Big Primes. Springer-Verlag, 1991.
- Rodríguez Prieto, A. Protección de la Información. Paraninfo, 1986.
- Rueppel, Rainer A. Analysis and Design of Stream Ciphers. Springer-Verlag, 1986.
- Russell, Deborah; Gangemi Sr., G.T. Computer Security Basics. O'Reilly & Associates, Inc., 1991.

Bibliografía de consulta y de interés (6)

- Schneier, Bruce. E-Mail Security. How to Keep your Electronic Messages Private. John-Wileys & Sons, Inc., 1995.
- Sgarro, A. Códigos Secretos. Pirámide, 1989.
- Smith, Richard E. Internet Cryptography. Addison-Wesley, 1997.
- Stallings, William. Network Security Essential. Aplications and Standards. Prentice Hall, 2000.
- Stallings, William. Protect Your Privacy. A Guide for PGO Users. Prentice Hall, 1995.
- Stein, Lincoln D. Web Security. A Step-by-Step Reference Guide. Addison-Wesley, 1998.
- Tena, Juan. Codificación de la Información . Universidad de Valladolid, 1997.
- Thomas, Stephen. SSL and TLS Essentials. John Wiley & Sons, 2000.
- Tung, Brian. Kerberos. A Network Authentication System, Addison-Wesley, 1999.

Bibliografía de consulta y de interés (7)

- Van Tilborg, H.C.A. An Introduction to Cryptology. Kluwer Academic Publishers, 1988.
- Wadlow, Thomas. The Pricess of Network Security. Addison-Wesley, 2000.
- White, Gregory, Fisch, Eric, Pooch, Udo. Computer System and Network Security. CRC Press Inc., 1996.
- Zimmermann, Philip R. The Official PGP User's Guide. MIT Press, 1995.

En los últimos dos o tres años, la bibliografía relacionada con la criptografía y seguridad informática ha aumentado de una forma espectacular. La bibliografía presentada en este capítulo como de consulta está actualizada sólo hasta finales del año 2001. Hoy cada mes aparecen tres o cuatro libros sobre esta temática en el mercado. No obstante, los libros que se han recomendado y comentado sí son actuales y permiten estar al día en este tema.

Enlaces de interés en Internet (1)

Los enlaces que puede encontrar en Internet sobre temas de seguridad informática son muchos. Por ejemplo, una consulta al buscador Google por "criptografía" entrega 45.000 enlaces; sobre "computer security" 417.000; por "cryptography" 937.000; por "hackers" más de 4 millones, etc. En estas tres diapositivas pondremos sólo unos cuantos enlaces que se recomiendan al lector, en los que encontrará información interesante.

- RedIRIS
  http://www.rediris.es/
- CSIC – Dpto. de Tratamiento de la Información
  http://www.iec.csic.es/
- Criptonomicón
  http://www.iec.csic.es/criptonomicon/
- Hispasec
  http://www.hispasec.com/
- VirusProt
  http://www.virusprot.com/
- Kriptópolis
  http://www.kriptopolis.com/
- CriptoRed
  http://www.criptored.upm.es/
- Revista SIC
  http://www.revistasic.com/
- Organización ISACA
  http://www.isaca.org/
- Computer Security Resource Center del NIST
  http://csrc.nist.gov/
- CERT de Carnegie Mellon
  http://www.cert.org/
- National Security Agency
  http://www.nsa.gov/

Enlaces de interés en Internet (2)

- Página Web de Alfred Menezes
  http://www.cacr.math.uwaterloo.ca/~ajmeneze/
- Página Web de Bruce Schneier
  http://www.counterpane.com/schneier.html
- Página Web del libro de William Stallings
  http://williamstallings.com/Crypto3e.html
- Página Web de Solomon Golomb
  http://csi.usc.edu/faculty/golomb.html
- Página Web de Vincent Rijmen: Rijndael
  http://www.esat.kuleuven.ac.be/~rijmen/rijndael/
- Página Web de Philip Zimmermann
  http://www.philzimmermann.com/
- Crypto++ Library
  http://www.eskimo.com/~weidai/cryptlib.html
- Conceptos sobre voto electrónico
  http://oasis.dit.upm.es/~jantonio/documentos/voto-electronico/article.html
- Esteganografía
  http://www.stegoarchive.com/
- Software libre de esteganografía
  http://members.tripod.com/steganography/stego/software.html
- Página de criptografía visual de Doug Stinson
  http://www.cacr.math.uwaterloo.ca/~dstinson/visual.html
- Quantum Cryptography Tutorial
  http://www.cs.dartmouth.edu/~jford/crypto.html
- PGP internacional
  http://www.pgpi.org/
- Computer Security Information
  http://www.alw.nih.gov/Security/security.html
- Cryptographic Compendium de John Savard
  http://home.ecn.ab.ca/~jsavard/crypto/entry.htm
- PKCS standards: RSA Security Corporation
  http://www.rsasecurity.com/rsalabs/pkcs/
- The Prime Pages
  http://www.utm.edu/research/primes/
- Polinomios primitivos
  http://mathworld.wolfran.com/primitivepolynomial.html
- SW para generación de polinomios primitivos
  http://www.theory.csc.uvic.ca/~cos/gen/poly.html
- Cryptome: documentos sobre criptografía
  http://cryptome.org/

Enlaces de interés en Internet (3)

- Delitos Informáticos y Leyes
  http://www.delitosinformaticos.com/
- Electronic Frontier Foundation EFF
  http://www.eff.org/
- Leyes, LOPD, 17799, etc. CSI – MAP - España
  http://www.map.es/csi/fr600001.htm#5
- SSH – Cryptography A-Z
  http://www.ssh.com/support/cryptography/
- Linux Security
  http://www.linuxsecurity.com/
- Ley Orgánica de Protección de Datos LOPD
  http://www.igsap.map.es/cia/dispo/lo15-99.htm
- The Internet Engineering Task Force
  http://www.ietf.org/
- FIPS Federal Inf. Process. Standards Publications
  http://www.itl.nist.gov/fipspubs/
- RFC Editor Homepage
  http://www.rfc-editor.org/
- Open SSL
  http://www.openssl.org/
- ANSI American National Standards Institute
  http://www.ansi.org/
- ETSI. Política y normas de seguridad Europa
  http://www.etsi.org/technicalfocus/home.htm
- Cryptography and Inform. Security Group MIT
  http://theory.lcs.mit.edu/~cis/
- Computer Forensics Laboratory
  http://www.dcfl.gov/index.shtml
- VeriSign
  http://www.verisign.com/
- The Hacker Quarterly
  http://www.2600.com/
- Web spoofing. Universidad de Princeton
  http://www.cs.princeton.edu/sip/pub/spoofing.html
- Apache Software Foundation
  http://www.apache.org/
- GNU Software de seguridad
  http://www.gnu.org/directory/security/
- Netscape Security
  http://wp.netscape.com/security/
- Windows Security
  http://www.microsoft.com/security/

Software libro electrónico de cifra clásica

- Autor: Ana María Camacho Hernández (1998).
- Enlace: http://www.criptored.upm.es/paginas/software.htm#propio.
- Libro electrónico realizado con ToolBook que hace un repaso a los temas principales de la criptografía clásica, incluyendo fotografías de máquinas de cifrar así como de los algoritmos más característicos para la realización de prácticas sencillas en un entorno Windows. Cuenta con cinco secciones:
- Sección 0: Historia de la criptografía. Principios de las técnicas criptográficas. Presentación de los algoritmos escítala, de Polybios y del César.
- Sección 1: Máquinas de cifrar. Rueda de Jefferson, discos de Alberti y Wheatstone, cifrador de Vernam, máquinas Enigma, M-325 y Hagelin.
- Sección 2: Cifrados por sustitución. Monoalfabética, polialfabética, cifra de Vigenère, de Beaufort, por homofonías, de Beale, de Playfair y de Hill.
- Sección 3: Cifrados por transposición. Cifra por grupos, series, columnas y filas.
- Sección 4: Algoritmos. Están implementados en la misma aplicación todos los algoritmos de cifrado y descifrado por sustitución y transposición para poder ejercitarse con ejemplos de textos introducidos por teclado.

Software CriptoClásicos

- Autor: Luis Miguel Motrel Berjano (1999).
- Enlace: http://www.criptored.upm.es/paginas/software.htm#propio.
- Aplicación para prácticas de sistemas de cifra clásicos que incluye algoritmos de cifra monoalfabética por sustitución y por transposición, cifra polialfabética con los cifradores de Vigenère, de Beaufort y de clave continua.
- Permite, además de cifrar y descifrar, realizar ataques por criptoanálisis a los sistemas anteriores mediante el uso de técnicas de estadísticas del lenguaje.
- Incluye además el cifrador de Vernam, el cifrador de Playfair, el cifrador de Hill digrámico y cifrado por transposiciones.
- Todas las operaciones de cifra pueden realizarse con los alfabetos castellano módulo 27 (letras mayúsculas), castellano módulo 37 (letras y dígitos) e inglés módulo 26 y módulo 36.
- Incluye un apartado con herramientas características de trabajo dentro de un cuerpo finito y estadísticas básicas del lenguaje. Las operaciones están limitadas a cuerpos menores que 65.536.
- Ayuda en formato Windows estándar y contextual mediante la tecla F1.

Software cifrador de Hill

- Autor: Mª Carmen Cogolludo Alcarazo (2001).
- Enlace: http://www.criptored.upm.es/paginas/software.htm#propio.
- Aplicación para prácticas de laboratorio con el cifrador poligrámico de Hill.
- Permite cifrar y descifrar archivos txt con una matriz clave de tamaño 2x2 hasta 10x10 dentro de los siguientes cuerpos: alfabeto castellano con letras en mayúsculas (mod 27), alfabeto incluyendo además los dígitos (mod 37) y por último un subconjunto de caracteres ASCII imprimibles (mod 191). En este último caso, la salida puede guardarse como un archivo en formato base 64.
- Las matrices clave pueden guardarse como un archivo.
- Permite además realizar ataques por criptoanálisis según el método de Gauss-Jordan. Una vez criptoanalizada la matriz clave, entrega un seguimiento de las ecuaciones que han permitido romper el sistema.
- El programa incluye una herramienta para el cálculo del determinante de una matriz, la matriz inversa y el número de matrices válidas dentro de un cuerpo.
- Ayuda en formato Windows estándar y contextual mediante la tecla F1.

Software cifrador de mochilas M-H

- Autor: Juan Carlos Rodríguez Castellano (1997).
- Enlace: http://www.criptored.upm.es/paginas/software.htm#propio.
- Software para prácticas de cifrado y descifrado del sistema de cifra con mochila de Merkle - Hellman realizado en Delphi.
- Se ha incluido una librería para trabajar con números grandes: decenas o centenas de dígitos, y herramientas básicas de trabajo dentro de un cuerpo.
- Permite el diseño de mochilas del tamaño y datos que desee el usuario, con tamaño recomendable M-H o bien mochilas con un tamaño proporcional al modelo recomendado por Merkle y Hellman.
- Una vez creada una mochila, el programa incluye la opción de criptoanálisis de la msima según el método de Shamir y Zippel, indicando luego de romper la mochila clave los valores analizados hasta lograr su objetivo
- Ayuda en formato Windows estándar y contextual mediante la tecla F1.

Software: CripMod sistemas modernos

- Autores: José Alberto Charfolé Sancho y D. Abel Gregorio Palomino (1997).
- Enlace: http://www.criptored.upm.es/paginas/software.htm#propio.
- Software para prácticas de sistemas de cifra modernos de clave privada y pública realizado en Delphi y con entorno de ayuda en castellano e inglés.
- Incluye cifrado y descifrado con DES, Data Encryption Standard, en todos sus modos de cifra, además de Triple DES.
- La generación de claves puede ser por parte del usuario o bien la genera el programa. Detecta claves débiles y semidébiles.
- Incluye una aplicación de cifra y firma RSA y ElGamal, limitados a cuerpos de tamaño menor que 65.536.
- Incluye un sistema básico de cifra con curvas elípticas.
- Cuenta con un apartado dedicado a herramientas típicas de trabajo dentro de un cuerpo: primalidad, máximo común divisor, inversos, indicador de Euler.
- Ayuda en formato Windows estándar y contextual mediante la tecla F1.
- Presentación de entradas y resultados en ASCII y hexadecimal.

Software: Fortaleza de cifrados

- Autor: Cristina Chércoles Larriba (1999).
- Enlace: http://www.criptored.upm.es/paginas/software.htm#propio.
- Software de prácticas realizado en Visual Basic que permite realizar y simular las operaciones dentro de un cuerpo más características en sistemas de cifra exponencial como son RSA y ElGamal: operaciones básicas de suma, resta, multiplicación y división, con o sin módulo, raíz, máximo común divisor, cálculo de inversos, potencia y primalidad.
- Usa una librería de números grandes: decenas hasta centenas de dígitos.
- Incluye un módulo de factorización de números compuestos por dos primos mediante los métodos de Pollard Rho, Dixon y Fraccciones Continuas. Además se muestra una lista de primos para poder trabajar con ellos, un conjunto de ejemplos y una tabla de primos titánicos.
- Incluye un módulo de cálculo del logaritmo discreto mediante los métodos de Búsqueda Exhaustiva, Paso Gigante - Paso Enano y Pohlig - Hellman. Además presenta un conjunto de ejemplos.
- Ayuda en formato Windows estándar y contextual mediante la tecla F1.

Software: CriptoRES funciones hash

- Autor: José Azaña Alonso (2001).
- Enlace: http://www.criptored.upm.es/paginas/software.htm#propio.
- Software para el estudio y seguimiento de las funciones hash más conocidas usadas en la compresión del documento para la firma digital y también en otras aplicaciones de autenticación como los certificados digitales.
- Tanto para MD5 como para SHA-1, la aplicación obtiene los resúmenes de documentos de archivos o texto introducido por teclado.
- Permite hacer un seguimiento del algoritmo de resumen, a nivel de bloques mostrando todas las operaciones en hexadecimal o bien siguiendo los pasos de las operaciones en bajo nivel. Como en este último caso la información mostrada está en bits, por su gran extensión muestra sólo el primer bloque de resumen que es precisamente donde se incluyen los rellenos para congruencia con el tamaño del bloque de 512 bits.
- Incluye una representación gráfica de la ecuación matemática del problema del ataque por la paradoja del cumpleaños.
- Ayuda en formato Windows estándar y contextual mediante la tecla F1.

Software: CryptoIDEA

- Autor: Esther Sánchez Mellado (1999).
- Enlace: http://www.criptored.upm.es/paginas/software.htm#propio.
- Software de prácticas realizado en Delphi que permite cifrar y descifrar con el algoritmo IDEA así como realizar un seguimiento del proceso y la generación de claves directas e inversas de cifra.
- Muestra además los ficheros en números enteros y en binario.
- Incluye un sistema de gestión de bases de datos tipo Paradox 5.0 para el mantenimiento (creación, modificación, borrado, etc.) de claves.
- La generación de claves se hace a partir de un texto o clave ASCII.
- Incluye un apartado de herramientas para cálculos típicos dentro de un cuerpo, en especial mod 65.536 y mod 65.537, valores usados en IDEA. Además el cálculo del máximo común divisor, inversos y conversiones de carácter a ASCII, de entero a binario y de binario a entero.
- Ayuda en formato Windows estándar y contextual mediante la tecla F1.

Otro software que se usa en la asignatura

- El en proyecto de desarrollo de software de prácticas para la asignatura de Seguridad Informática, hay otros programas que se están llevando a cabo y que en los siguientes meses se pondrán en el servidor Web ya comentado, entre ellos:
- Laboratorio de cifra RSA que permite comprobar las debilidades asociadas a este tipo de cifra: claves parejas, mensajes no cifrables, ataque cíclico, etc.
- Laboratorio de ataque al DES. Además de cifrado y descifrado con claves en ASCII o hexadecimal, muestra la debilidad de las claves del DES ante un ataque monousuario, distribuido simulado o distribuido a través de la red.
- Simulación de las operaciones de autenticación de Kerberos en una aplicación de tipo Windows para prácticas.
- Laboratorio de seguimiento del algoritmo Rijndael.
- Etc.
- Otros programas ya desarrollados y que se usan de forma interna, no se han hecho públicos al no cumplir alguna condición básica para su libre distribución como puede ser el tamaño de la aplicación, su funcionalidad, etc.

Tabla de frecuencia de monogramas

A	7,49	A	10,60	A	9,83
B	1,29	B	1,16	B	0,86
C	3,54	C	4,85	C	4,15
D	3,62	D	5,87	D	4,04
E	14,00	E	13,11	E	11,41
F	2,18	F	1,13	F	0,81
G	1,74	G	1,40	G	0,85
H	4,22	H	0,60	H	0,57
I	6,65	I	7,16	I	6,04
J	0,27	J	0,25	J	0,17
K	0,47	K	0,11	K	0,00
L	3,57	L	4,42	L	4,34
M	3,39	M	3,11	M	2,42
N	6,74	N	7,14	N	6,03
Ñ	-,--	Ñ	0,10	Ñ	0,07
O	7,37	O	8,23	O	7,75
P	2,43	P	2,71	P	2,41
Q	0,26	Q	0,74	Q	0,73
R	6,14	R	6,95	R	5,26
S	6,95	S	8,47	S	7,13
T	9,85	T	5,40	T	3,62
U	3,00	U	4,34	U	3,24
V	1,16	V	0,82	V	0,69
W	1,69	W	0,12	W	0,00
X	0,28	X	0,15	X	0,18
Y	1,64	Y	0,79	Y	0,63
Z	0,04	Z	0,26	Z	0,29
				Blanco	6,33

Valores de frecuencia en tanto por ciento en archivos de 50.000 caracteres.
Columnas: 1º Inglés (mod 26); 2ª Castellano (mod 27); 3ª Castellano (mod 28)

Monogramas más frecuentes mod 27

E	13,11	C	4,85	Y	0,79
A	10,60	L	4,42	Q	0,74
S	8,47	U	4,34	H	0,60
O	8,23	M	3,11	Z	0,26
I	7,16	P	2,71	J	0,25
N	7,14	G	1,40	X	0,15
R	6,95	B	1,16	W	0,12
D	5.87	F	1,13	K	0,11
T	5,40	V	0,82	Ñ	0,10

Frecuencia alta Frecuencia media Frecuencia baja

Con los 9 caracteres más frecuentes podemos formar la palabra ESTIRANDO

Alfabeto castellano y sus inversos mod 27

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Alfabeto y tabla módulo 27

a	inv (a,27)	a	inv (a,27)	a	inv (a,27)
1	1	2	14	4	7
5	11	7	4	8	17
10	19	11	5	13	25
14	2	16	22	17	8
19	10	20	23	22	16
23	20	25	13	26	26

Alfabeto castellano y sus inversos mod 37

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Alfabeto y tabla módulo 37

0	1	2	3	4	5	6	7	8	9
27	28	29	30	31	32	33	34	35	36

a	inv (a,37)	a	inv (a,37)	a	inv (a,37)	a	inv (a,37)	a	inv (a,37)	a	inv (a,37)
1	1	2	19	3	25	4	28	5	15	6	31
7	16	8	14	9	33	10	26	11	27	12	34
13	20	14	8	15	5	16	7	17	24	18	35
19	2	20	13	21	30	22	32	23	29	24	17
25	3	26	10	27	11	28	4	29	23	30	21
31	6	32	22	33	9	34	12	35	18	36	36

Tabla de Vigenère

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	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	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	Ñ	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
Ñ	Ñ	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	N
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	N	Ñ
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	Ñ	O
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	Ñ	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y

Código Baudot (cifrador de Vernam)

Código Binario	Carácter	Código Binario	Carácter
00000	Blanco	10000	T
00001	E	10001	Z
00010	=	10010	L
00011	A	10011	W
00100	Espacio	10100	H
00101	S	10101	Y
00110	I	10110	P
00111	U	10111	Q
01000	<	11000	O
01001	D	11001	B
01010	R	11010	G
01011	J	11011	↑
01100	N	11100	M
01101	F	11101	X
01110	C	11110	V
01111	K	11111	↓

Código ASCII/ANSI de nivel bajo (1)

Byte	carácter
0010 0000	Espacio
0010 0001	!
0010 0010	“
0010 0011	#
0010 0100	$
0010 0101	%
0010 0110	&
0010 0111	‘
0010 1000	(
0010 1001	)
0010 1010	*
0010 1011	+
0010 1100	,
0010 1101	-
0010 1110	.
0010 1111	/

Byte	carácter
0011 0000	0
0011 0001	1
0011 0010	2
0011 0011	3
0011 0100	4
0011 0101	5
0011 0110	6
0011 0111	7
0011 1000	8
0011 1001	9
0011 1010	:
0011 1011	;
0011 1100	<
0011 1101	=
0011 1110	>
0011 1111	?

Byte	carácter
0100 0000	@
0100 0001	A
0100 0010	B
0100 0011	C
0100 0100	D
0100 0101	E
0100 0110	F
0100 0111	G
0100 1000	H
0100 1001	I
0100 1010	J
0100 1011	K
0100 1100	L
0100 1101	M
0100 1110	N
0100 1111	O

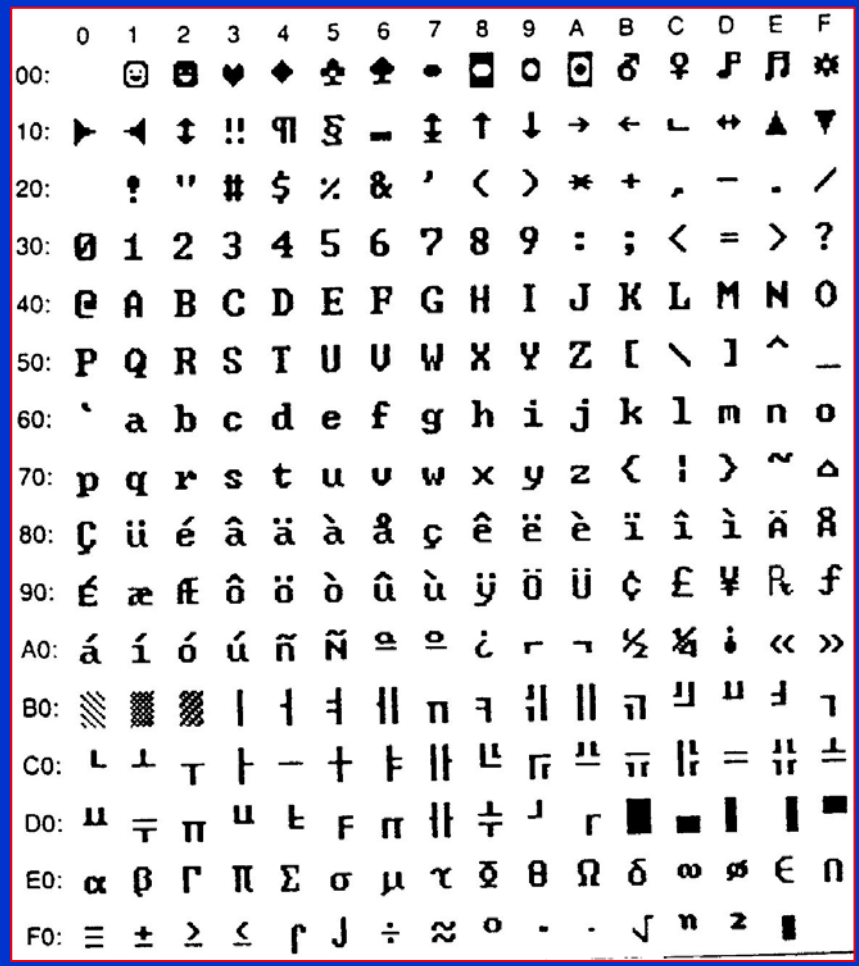
Código ASCII/ANSI de nivel bajo (2)

Byte	carácter
0101 0000	P
0101 0001	Q
0101 0010	R
0101 0011	S
0101 0100	T
0101 0101	U
0101 0110	V
0101 0111	W
0101 1000	X
0101 1001	Y
0101 1010	Z
0101 1011	[
0101 1100	\
0101 1101	]
0101 1110	^
0101 1111	_

Byte	carácter
0110 0000	`
0110 0001	a
0110 0010	b
0110 0011	c
0110 0100	d
0110 0101	e
0110 0110	f
0110 0111	g
0110 1000	h
0110 1001	i
0110 1010	j
0110 1011	k
0110 1100	l
0110 1101	m
0110 1110	n
0110 1111	o

Byte	carácter
0111 0000	p
0111 0001	q
0111 0010	r
0111 0011	s
0111 0100	t
0111 0101	u
0111 0110	v
0111 0111	w
0111 1000	x
0111 1001	y
0111 1010	z
0111 1011	{
0111 1100	|
0111 1101	}
0111 1110	~
0111 1111	

Tabla de código ASCII extendido



00-0F: Valor decimal: 000-015

10-1F: Valor decimal: 016-031

20-2F: Valor decimal: 032-047

30-3F: Valor decimal: 048-063

40-4F: Valor decimal: 064-079

50-5F: Valor decimal: 080-095

60-6F: Valor decimal: 096-111

70-7F: Valor decimal: 112-127

80-8F: Valor decimal: 128-143

90-9F: Valor decimal: 144-159

A0-AF: Valor decimal: 160-175

B0-BF: Valor decimal: 176-191

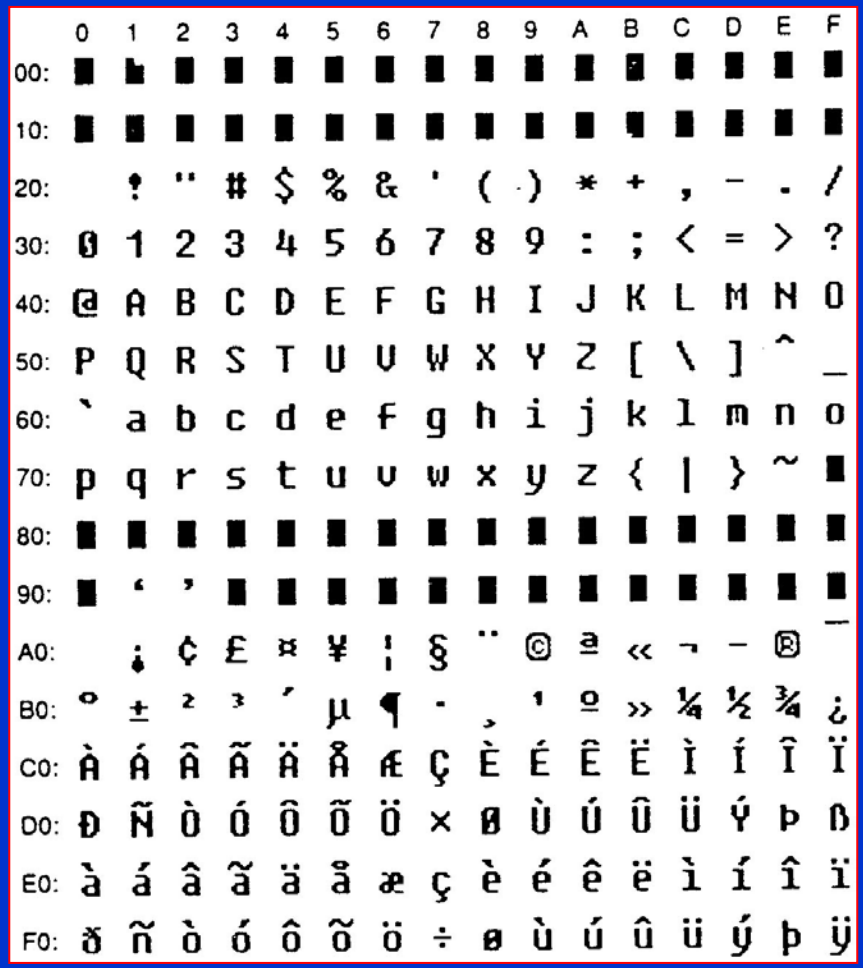
C0-CF: Valor decimal: 192-207

D0-DF: Valor decimal: 208-223

E0-EF: Valor decimal: 224-239

F0-FF: Valor decimal: 240-255

Tabla de código ANSI extendido



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00:	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
10:	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
20:		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30:	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40:	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50:	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60:	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70:	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	■
80:	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
90:	■	‘	’	■	■	■	■	■	■	■	■	■	■	■	■	■
A0:		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
B0:	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0:	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0:	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0:	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0:	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

00-0F: Valor decimal: 000-015

10-1F: Valor decimal: 016-031

20-2F: Valor decimal: 032-047

30-3F: Valor decimal: 048-063

40-4F: Valor decimal: 064-079

50-5F: Valor decimal: 080-095

60-6F: Valor decimal: 096-111

70-7F: Valor decimal: 112-127

80-8F: Valor decimal: 128-143

90-9F: Valor decimal: 144-159

A0-AF: Valor decimal: 160-175

B0-BF: Valor decimal: 176-191

C0-CF: Valor decimal: 192-207

D0-DF: Valor decimal: 208-223

E0-EF: Valor decimal: 224-239

F0-FF: Valor decimal: 240-255

La codificación en Base 64

Valor 6 bits		Carácter codificado	Valor 6 bits		Carácter codificado	Valor 6 bits		Carácter codificado	Valor 6 bits		Carácter codificado
0	000000	A	16	010000	Q	32	100000	g	48	110000	w
1	000001	B	17	010001	R	33	100001	h	49	110001	x
2	000010	C	18	010010	S	34	100010	i	50	110010	y
3	000011	D	19	010011	T	35	100011	j	51	110011	z
4	000100	E	20	010100	U	36	100100	k	52	110100	0
5	000101	F	21	010101	V	37	100101	l	53	110101	1
6	000110	G	22	010110	W	38	100110	m	54	110110	2
7	000111	H	23	010111	X	39	100111	n	55	110111	3
8	001000	I	24	011000	Y	40	101000	o	56	111000	4
9	001001	J	25	011001	Z	41	101001	p	57	111001	5
10	001010	K	26	011010	a	42	101010	q	58	111010	6
11	001011	L	27	011011	b	43	101011	r	59	111011	7
12	001100	M	28	011100	c	44	101100	s	60	111100	8
13	001101	N	29	011101	d	45	101101	t	61	111101	9
14	001110	O	30	011110	e	46	101110	u	62	111110	+
15	001111	P	31	011111	f	47	101111	v	63	111111	/
										(Relleno)	=

Tabla de codificación en Base 64

Cada 3 bytes ANSI (24 bits) se convierten en 4 elementos Base 64 de 6 bits cada uno. El fichero aumenta un 33% pero ello se compensará al usar la compresión zip.

Ejemplo de codificación Base 64

$Hola_{ANSI}$ = 01001000 01101111 01101100 01100001

$Hola_{B64}$ = 010010 000110 111101 101100 011000 01 (00 00) = SG9sYQ==

Valor 6 bits		Carácter codificado	Valor 6 bits		Carácter codificado	Valor 6 bits		Carácter codificado	Valor 6 bits		Carácter codificado
0	000000	A	16	010000	Q	32	100000	g	48	110000	w
1	000001	B	17	010001	R	33	100001	h	49	110001	x
2	000010	C	18	010010	S	34	100010	i	50	110010	y
3	000011	D	19	010011	T	35	100011	j	51	110011	z
4	000100	E	20	010100	U	36	100100	k	52	110100	0
5	000101	F	21	010101	V	37	100101	l	53	110101	1
6	000110	G	22	010110	W	38	100110	m	54	110110	2
7	000111	H	23	010111	X	39	100111	n	55	110111	3
8	001000	I	24	011000	Y	40	101000	o	56	111000	4
9	001001	J	25	011001	Z	41	101001	p	57	111001	5
10	001010	K	26	011010	a	42	101010	q	58	111010	6
11	001011	L	27	011011	b	43	101011	r	59	111011	7
12	001100	M	28	011100	c	44	101100	s	60	111100	8
13	001101	N	29	011101	d	45	101101	t	61	111101	9
14	001110	O	30	011110	e	46	101110	u	62	111110	+
15	001111	P	31	011111	f	47	101111	v	63	111111	/
										(Relleno)	=

Tabla de codificación en Base 64

Tabla de primos del 1 al 1000

2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	61	67	71	73	79	83	89	97
101	103	107	109	113	127	131	137	139	149	151	157	163	167	173	179	181	191	193	197	199				
211	223	227	229	233	239	241	251	257	263	269	271	277	281	283	293									
307	311	313	317	331	337	347	349	353	359	367	373	379	383	389	397									
401	409	419	421	431	433	439	443	449	457	461	463	467	479	487	491	499								
503	509	521	523	541	547	557	563	569	571	577	587	593	599											
601	607	613	617	619	631	641	643	647	653	659	661	673	677	683	691									
701	709	711	727	733	739	743	751	757	761	769	773	787	797											
809	811	821	823	827	829	839	853	857	859	863	877	881	883	887										
907	911	919	929	937	941	947	953	967	971	977	983	991	997											

Tabla de primos del 1001 al 2000

1009	1013	1019	1021	1031	1033	1039	1049	1051	1061	1063	1069	1087	1091	1093	1097
1103	1109	1117	1123	1129	1151	1153	1163	1171	1181	1187	1193				
1201	1213	1217	1223	1229	1231	1237	1249	1259	1277	1279	1283	1289	1291	1297	
1301	1303	1307	1319	1321	1327	1361	1367	1373	1381	1399					
1409	1423	1427	1429	1433	1439	1447	1451	1453	1459	1471	1481	1483	1487	1489	1493	1499
1511	1523	1531	1543	1549	1553	1559	1567	1571	1579	1583	1597				
1601	1607	1609	1613	1619	1621	1627	1637	1657	1663	1667	1669	1693	1667	1699	
1709	1721	1723	1733	1741	1747	1753	1759	1777	1783	1787	1789				
1801	1811	1823	1831	1847	1861	1867	1871	1873	1877	1879	1889				
1901	1907	1913	1931	1933	1949	1951	1973	1979	1987	1993	1997	1999			

Polinomios primitivos

Polinomios para n = 3: $x^3 + x + 1$; $x^3 + x^2 + 1$
Polinomios para n = 4: $x^4 + x + 1$; $x^4 + x^3 + 1$
Polinomios para n = 5: $x^5 + x^2 + 1$; $x^5 + x^4 + x^3 + x^2 + 1$; $x^5 + x^4 + x^2 + 1$; $x^5 + x^3 + x^2 + x + 1$; $x^5 + x^4 + x^3 + x + 1$; $x^5 + x^3 + x + 1$
Polinomios para n = 6: $x^6 + x + 1$; $x^6 + x^5 + x^2 + x + 1$; $x^6 + x^5 + x^3 + x^2 + 1$; $x^6 + x^4 + x^3 + x + 1$; $x^6 + x^5 + x^4 + x + 1$; $x^6 + x^5 + 1$

(1, 0)	(11, 2, 0)	(21, 2, 0)	(31, 3, 0)	(41, 3, 0)
(2, 1, 0)	(12, 6, 4, 1, 0)	(22, 1, 0)	(32, 7, 6, 2, 0)	(42, 5, 4, 3, 2, 1, 0)
(3, 1, 0)	(13, 4, 3, 1, 0)	(23, 5, 0)	(33, 13, 0)	(43, 6, 4, 3, 0)
(4, 1, 0)	(14, 5, 3, 1, 0)	(24, 4, 3, 1, 0)	(34, 8, 4, 3, 0)	(44, 6, 5, 2, 0)
(5, 2, 0)	(15, 1, 0)	(25, 3, 0)	(35, 2, 0)	(45, 4, 3, 1, 0)
(6, 1, 0)	(16, 5, 3, 2, 0)	(26, 6, 2, 1, 0)	(36, 11, 0)	(46, 8, 5, 3, 2, 1, 0)
(7, 1, 0)	(17, 3, 0)	(27, 5, 2, 1, 0)	(37, 6, 4, 1, 0)	(47, 5, 0)
(8, 4, 3, 2, 0)	(18, 7, 0)	(28, 3, 0)	(38, 6, 5, 1, 0)	(48, 7, 5, 4, 2, 1, 0)
(9, 4, 0)	(19, 5, 2, 1, 0)	(29, 2, 0)	(39, 4, 0)	(49, 9, 0)
(10, 3, 0)	(20, 3, 0)	(30, 6, 4, 1, 0)	(40, 5, 4, 3, 0)	(50, 4, 3, 2, 0) (*)

Algunos polinomios de x^n para n = 1 hasta n = 50

(*) Explicación: $(50, 4, 3, 2, 0) \Rightarrow p(x) = x^{50} + x^4 + x^3 + x^2 + 1$

Magnitudes de tiempo y criptoanálisis

Longitud de la clave	Tiempo necesario para romper la clave
40 bits	2 segundos
48 bits	9 minutos
56 bits	40 horas
64 bits	14 meses
72 bits	305 años
80 bits	78.250 (2^{16}) años
96 bits	5.127.160.311 (2^{32}) años
112 bits	336.013.578.167.538 (2^{48}) años
128 bits	22.020.985.858.787.784.059 (2^{64}) años

Valores de tiempo con números grandes	
Edad del planeta	10.000.000.000 ($10^{10} = 2^{34}$) años
Edad del universo	100.000.000.000 ($10^{11} = 2^{37}$) años

La tabla muestra el tiempo medio de criptoanálisis necesario para romper una clave del DES por fuerza bruta, usando la potencia de cálculo alcanzada en el DES Challenge III, unos 250.000 millones de claves por segundo con máquina DES Cracker y 100.000 PCs conectados en red.

Fin del Tema 18

Cuestiones y ejercicios (1 de 2)

1. En un texto cifrado por sustitución afín, se tiene como criptograma C = PCERC QBCSK AQBGR LGFJQ KCXKÑ LCECN RKZHL KKXGF LCAFB. ¿Qué es lo primero que hacemos? Ataque el criptograma con el software de la asignatura. ¿Qué puede concluir?
2. Con el cifrador de Vernam y código Baudot ciframos M = SOL con la clave K = MAR. ¿Qué se obtiene como criptograma?
3. Observando el código ASCII/ANSI de nivel bajo en binario, ¿Por qué es desaconsejable cifrar con mochilas de tamaño 4 u 8?
4. La clave DES escrita en código ASCII es K = HOLAhola. ¿Cuál sería en este caso la clave de 56 bits? ¿Qué puede decir de esto?
5. Observando la tabla de primos del 1 al 1.000 y de 1.001 al 2.000, ¿podríamos concluir que en una ventana igual (2.001 al 3.000; 3.001 al 4.000, etc.) cada vez hay más números primos? ¿Por qué?

Cuestiones y ejercicios (2 de 2)

6. ¿Por qué en módulo 37 existen más inversos que en módulo 27?
7. Codifique en base 64 el texto de 10 caracteres M = ¡Qué tal!
8. ¿Qué mensaje hay en M' = v01lb nNham Ugb2N 1bHRv Pw== que está codificado en base 64? ¿Por qué el relleno es dos?
9. ¿En cuánto aumenta el tamaño cuando convertimos ASCII/ANSI a código base 64? ¿Es eso significativo en PGP? ¿Por qué sí o no?
10. ¿En qué zona podemos decir que el código ASCII y el ANSI son iguales?
11. ¿Se codificará igual en ASCII que en ANSI el mensaje M_1 = "Voy a entrar"? Y si ahora el mensaje es M_2 = "Pasa, está abierto" ¿Qué consecuencias puede tener esto en un programa?
12. Un mensaje codificado en ANSI hexadecimal es 43 72 69 70 74 6F 67 72 61 66 ED 61, ¿cuál es el texto en castellano?