



Trabajo de Grado

Licenciatura en Informática

ROMANO DAMIAN A.

<p>TES 00/13 DIF-02123 SALA</p>	<p> UNIVERSIDAD NACIONAL DE LA PLATA FACULTAD DE INFORMATICA Biblioteca 50 y 120 La Plata catalogo.info.unlp.edu.ar biblioteca@info.unlp.edu.ar</p> <p> DIF-02123</p>
---	---

Presentación Final

ALUMNO:

Analista de Computación UNLP **Romano, Damián A.** N°A1.34333/0

E-Mail: dromano@bigfoot.com

Tel. Particular: (0221) 421-8837 425-7249

Tel. Laboral (9 a 13 y 14 a 18): (011) 4316-3000 Int. 2684 ó 2517

DIRECTOR:

Antonelli, Gustavo

E-Mail: ganton@siemens-itron.com.ar

Tel. Laboral: (011) 4346-5947/5801

Indice Temático:

- Introducción
- Copia de la Propuesta de Trabajo de Grado Aprobada
- Copia del Informe de Avance del 50% Aprobado
- **Parte Práctica de la Tesis**
 - Sistema de Pago On_Line
 - _ Servidor de Tarjetas de Crédito
 - _ Servidor de Servicios
 - _ Cliente del Sistema de Pago On_Line
 - _ Base de Datos de las Aplicaciones
 - _ Instalación
 - _ Probando el Sistema
- Datos de la Implementación
- Por que P.G.P. (Pretty Good Privacy)
- Posibilidad de utilización / continuación para un uso mas “real”
- Sistema de Pago On_Line (Presentación Power Point)

- **Parte Teórica de la Tesis**

- Criptografía

- _ Prefacio
- _ Introducción
- _ Criptografía simétrica
- _ Criptografía asimétrica
- _ Otras herramientas criptográficas
- _ Certificados Digitales
- _ Infraestructura de claves públicas
- _ Protocolos de seguridad

- Conceptos de Firma Digital (Presentación Power Point)

- Certificación de Claves Publicas y Autoridades Certificadoras (Presentación Power Point)

- Extractos de los Boletines de Kriptópolis

- Vulnerabilidades del PGP

- Bibliografía

Introducción

Introducción:

Este es el Informe Final de Trabajo de Grado, que he realizado siguiendo los lineamientos de la Propuesta y el Informe de Avance del 50% aprobados en su oportunidad por la Comisión designada.

Este informe esta dividido en una Parte Práctica y una Teórica.

La Parte Práctica describe cada uno de los Módulos del Sistema de Pago On_Line, su instalación y su prueba, como así también se detallan algunos puntos del análisis y del desarrollo de los mismos.

La Parte Teórica de este Trabajo de Grado muestra parte de la información que he estudiado, aprendido y recopilado en estos dos últimos años.

Es de mencionar que mi Tesis se encuentra participando en la Categoría “Universitarios” del “Premio al Tecno-Emprendedor 2000” del Banco Francés. Los resultados de dicho concurso se darán a conocer a fines del mes de Septiembre del presente año.

También quería mencionar a la gente que trabaja en Kriptópolis (www.kriptopolis.com). Estos españoles me han ayudado respondiendo mis consultas y solucionando inconvenientes en mi Tesis. Tambien he participado de sus debates, newsgroups, foros e e-mails sobre Seguridad Informática.

Actualmente ellos poseen una copia de mi Trabajo que será evaluado y posteriormente incorporado al Site para que cualquier persona lo pueda bajar e instalar en su propia máquina.

Nota Importante:

Cualquier cambio, modificación o agregado este documento, tanto de la Parte Práctica como de la Teórica, será incluida en CD adjunto dentro del archivo \ReadMe.doc.

También se incluirán los datos de las Bases de Datos para facilitar las pruebas del Sistema de Pago On_Line.

Atte. Damián Romano (dromano@bigfoot.com)

Copia de la

Propuesta de Trabajo de

Grado Aprobada

Propuesta de Trabajo de Grado
Licenciatura en Informática

ALUMNOS:

Analista de Computación UNLP **Romano, Damián A.** N°AI.34333/0

E-Mail: dromano@bigfoot.com

Tel. Particular: 25-7249

Tel. Laboral (9 a 13 y 14 a 18): (01) 316-3000 Int.2517

DIRECTOR:

Antonelli, Gustavo

E-Mail: ganton@itron.com.ar

gustavo@nahuel.way.com.ar

MOTIVACIÓN:

El crecimiento explosivo de los sistemas de computación y sus conexiones vía redes ha aumentado la dependencia de las Organizaciones y los Individuos en la Información almacenada y las Comunicaciones usando estos sistemas. Esto ha llevado a la necesidad de proteger los datos y recursos, para garantizar la Autenticidad de los datos y los mensajes, y para proteger sistemas de ataques por red.

Las disciplinas de Criptografía y Seguridad en Redes han madurado, llegando al desarrollo de aplicaciones prácticas y disponibles para garantizar la Seguridad en las Redes.

Es por esto y mucho mas que queremos abordar un Trabajo de Grado estudiando las técnicas mas avanzadas de Criptografía y sus temas relacionados, como así también realizar una implementación de un Sistema (descripto mas adelante) que utilice estas técnicas.

Existen varios Conceptos de Seguridad Informática que deben ser estudiados y a los cuales intentaremos abordar en nuestra Tesis. Entre ellos tenemos:

- **Confidencialidad:** Proteger la Información.
- **Integridad:** Proteger la exactitud de la Información.
- **Autenticación:** Proteger la Información del Origen (Sender).
- **No Repudio (NonRepudiation):** Protegerse de la negación.
- **Identificación de Usuarios:** Asegurar la Identidad de los Usuarios.
- **Control de Acceso:** Controlar el Acceso a la Información y/o a los Recursos.
- **Disponibilidad:** Asegurar el envío de la Información.

OBJETIVO:

Nuestro Trabajo de Grado será una Investigación teórica de las Técnicas Criptográficas y Sistemas de Seguridad Informática, así como también, abordaremos la Implementación de un Sistema que se basará en Técnicas de Seguridad que se detallan a continuación.

Entre los temas que vamos a estudiar y analizar se encuentran:

- ❖ Criptografía de Clave Simétrica
- ❖ Criptografía de Clave Pública (Asimétrica)
- ❖ Distintos Algoritmos de Encriptación
- ❖ Autenticación y Firma Digital (Requisitos de Certificación de Autor, Contenido y Fecha)
- ❖ Estándares Criptográficos Internacionales
- ❖ Validez Legal de un Documento Electrónico
- ❖ Hashing para garantizar la Inalterabilidad
- ❖ Usos Actuales de Criptografía
- ❖ El Comercio Electrónico en Internet
- ❖ Pretty Good Privacy (PGP)
- ❖ Electronic Mail Security (PEM)
- ❖ Formas de Crackear los distintos Sistemas de Seguridad
- ❖ Posibles soluciones a los ataques del Sistema

Con el Objeto de no llevar la investigación de los temas antes mencionados hacia el infinito, vamos a basar el estudio en la forma de darle Seguridad al Sistema que detallamos a continuación y que titulamos:

Sistema de Pago de Servicios On-Line

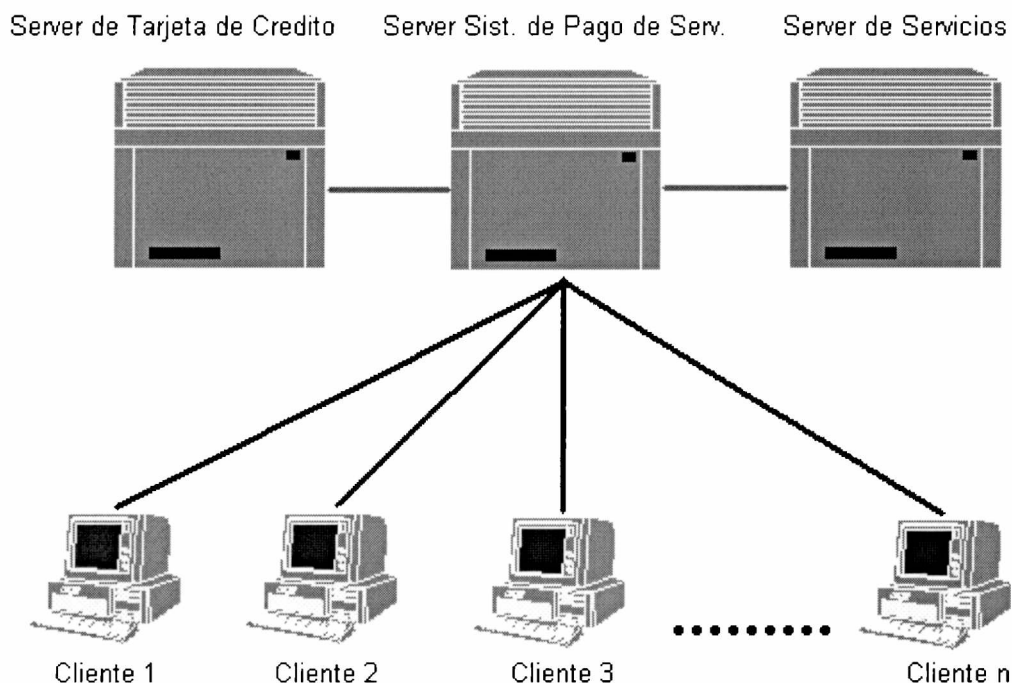
Consiste en un Sistema de Servidores y Clientes que se comunican con el Objeto de Pagar Servicios (luz, gas, teléfono, etc.) mediante Tarjeta de Crédito.

Existirán, en principio 3 Servidores y n Clientes. Según se detallan a continuación:

- Un Servidor de Pago, el cual tomará los pedidos de los Clientes, validará las Tarjetas de Crédito y hará la Transacción con los Servicios.
- Un Servidor de Validación de Tarjetas de Crédito (que podrían llegar a ser varios, 1 por cada tarjeta), que indicará si una Tarjeta es válida o no.
- Un Servidor de Servicios (que podrían llegar a ser varios, 1 por cada Servicio), el cual recibirá los pagos hechos por los clientes y verificará que los pagos que se hagan sean correctos.
- Y por último, los Clientes serán los que utilicen el Sistema de Pago para abonar los Servicios con Tarjeta de Crédito.

Nuestro Sistema de Pago utilizará distintas Técnicas Criptográficas que garantizarán los puntos vistos en el Objetivo de nuestra Tesis. La idea básica de esta implementación es que todas las comunicaciones entre servidores y clientes sean completamente seguras, cumpliendo los principios citados en la Motivación de esta Tesis.

Gráficamente el Sistema de Pago se podría ver así:



CLASIFICACIÓN:

El Proyecto se clasifica dentro del Area Redes y Sistemas de Seguridad Informática.

PLAZO DE EJECUCIÓN:

Estimamos la Presentación Final del Trabajo de Grado para Mayo de 1999.

También estimamos la Presentación del Informe del 50% de Avance para Marzo de 1999.

BIBLIOGRAFÍA:

- NETWORK SECURITY: PRIVATE COMMUNICATION IN A PUBLIC WORLD – Charlie Kaufman, Radia Perlman y Mike Speciner – PTR Prentice Hall.
- NETWORK AND INTERNETWORK SECURITY – PRINCIPLES AND PRACTICE William Stallings – Prentice Hall – IEEE Press.
- TECNICAS CRIPTOGRAFICAS DE PROTECCION – Fuster.
- PUBLIC-KEY CRYPTOGRAPHY – Salomaa.
- APPLIED CRYPTOGRAPHY – Schneier.
- THE OFFICIAL PGP USER'S GUIDE – Zimmermann Phillip.
- DIGITAL CASH – Peter Wayner.
- Distintas paginas Web que nos garantizan credulidad, como por ejemplo:
 - <http://www.jus.gov.ar/firma/> SubComité de Criptografia y Firma Digital
 - <http://www.pgpi.com/> The International PGP Home Page
 - <http://www.rsa.com/> RSA Data Security, Inc.

La Plata, 16 de Noviembre de 1998

Copia del

Informe de Avance del

50% Aprobado

Informe de Avance 50 %
Trabajo de Grado
Licenciatura en Informática

ALUMNO:

Analista de Computación UNLP **Romano, Damián A.** N°Al.34333/0

E-Mail: dromano@bigfoot.com

Tel. Particular: (0221) 425-7249

Tel. Laboral (9 a 13 y 14 a 18): (011) 4316-3000 Int. 2517 ó 2674

DIRECTOR:

Antonelli, Gustavo

E-Mail: ganton@itron.com.ar

gustavo@nahuel.way.com.ar

Introducción:

En estas hojas queremos informarles del Estado del Proyecto detallando lo que ya se encuentra realizado, lo que se encuentra en etapa de desarrollo y lo que falta lograr para poder exponer este Trabajo de Grado. Asimismo daremos una estimación de tiempo para su Entrega Final.

Rogamos tengan a bien responder a este Avance lo mas pronto posible.

Estado Actual del Proyecto:

_ Servidor de Tarjetas de Crédito:

Este Servidor simula ser un Servidor de Validación y Registro de Compras de Tarjetas de Crédito, es una Aplicación 32 bits que he realizado en Delphi 4. Corre bajo Windows 95 o NT. Obviamente atiende n clientes a la vez.

El Programa, mediante Sockets, recibe mensajes que debe procesar. Estos mensajes pueden ser de Testeo de Tarjetas de Crédito, o bien, son de Pago con Tarjetas de Crédito.

Por Ejemplo: si recibe un mensaje de tipo T (Testeo), con un Titular, un Número de Tarjeta y un Vencimiento, deberá verificar en su Base de Datos si esta Tarjeta existe y si esta habilitada. Luego informará el resultado al Programa Cliente que solicitó el dato.

Del mismo modo, si recibe un mensaje de tipo P (Pago), con los mismos datos de antes mas los datos del pago, deberá registrar el Pago en sus Bases (previa verificación de datos!).

El Servidor también maneja una Tabla Log para ir guardando las solicitudes que se le van haciendo y los resultados obtenidos.

Además se pueden ver los datos de las Tarjetas de Crédito de la Base de Datos del Servidor, dar de Alta nuevas Tarjetas o Modificar los Datos de alguna ya ingresada previamente. Esto último es muy útil cuando se produce el robo de una Tarjeta de Crédito: simplemente se pone el Estado en "ROBADA" y listo.

Otra de las funciones que tiene es mostrar los Pagos que se registraron en este Servidor. Se muestra la fecha, el Monto y el Servicio que se abonó.

Servidor de Tarjetas de Credito

Listado de Consultas a este Servidor: Servidor Activo

DATOS	RESULTADO	FECHA Y HORA
▶ MASTERCARD damian 1 01/01/2000	MASTERCARD damian 1	14/02/1999 18:43:29
MASTERCARD damian 1 01/01/2000	HABILITADA	14/02/1999 18:44:54
MASTERCARD damian 1 01/01/2000	HABILITADA	14/02/1999 18:45:00
VISA damian 1 01/01/2000	Datos No Validos	14/02/1999 18:45:18
VISA damian 1 01/01/2000	Datos No Validos	14/02/1999 18:45:34
MASTERCARD damian 1 01/01/2000	HABILITADA	14/02/1999 18:45:36
MASTERCARD damian 1 01/01/2000	HABILITADA	14/02/1999 18:56:24
MASTERCARD Damian Romano 23 14/02/1999	DATOS NO VALIDOS	14/02/1999 18:56:59
MASTERCARD sfdsgfd lasfdasdf 14/02/1999	DATOS NO VALIDOS	14/02/1999 18:58:23
VISA DAMIAN 3 01/01/2000	ROBADA	14/02/1999 19:00:04
MASTERCARD Damian 1 01/01/2000	HABILITADA	16/02/1999 11:01:57
MASTERCARD Damian Romano 123 16/02/1999	DATOS NO VALIDOS	16/02/1999 11:02:24
VISA 2 Damian 01/01/2000	DATOS NO VALIDOS	16/02/1999 11:02:38
MASTERCARD DAMIAN 1 01/01/2000	HABILITADA	23/02/1999 16:28:43
MASTERCARD ASDF SFA 23/02/1999	DATOS NO VALIDOS	23/02/1999 16:28:54

Ver Datos Tarjetas Pagos Registrados Bajar Servidor

Servidor de Tarjetas de Crédito Activo.

_ Servidor de Pago de Servicios:

Este Servidor simula ser un Servidor de Validación y Registro de Pago de Servicios, es una Aplicación 32 bits que he realizado en Delphi 4. Corre bajo Windows 95 o NT. Obviamente atiende n clientes a la vez.

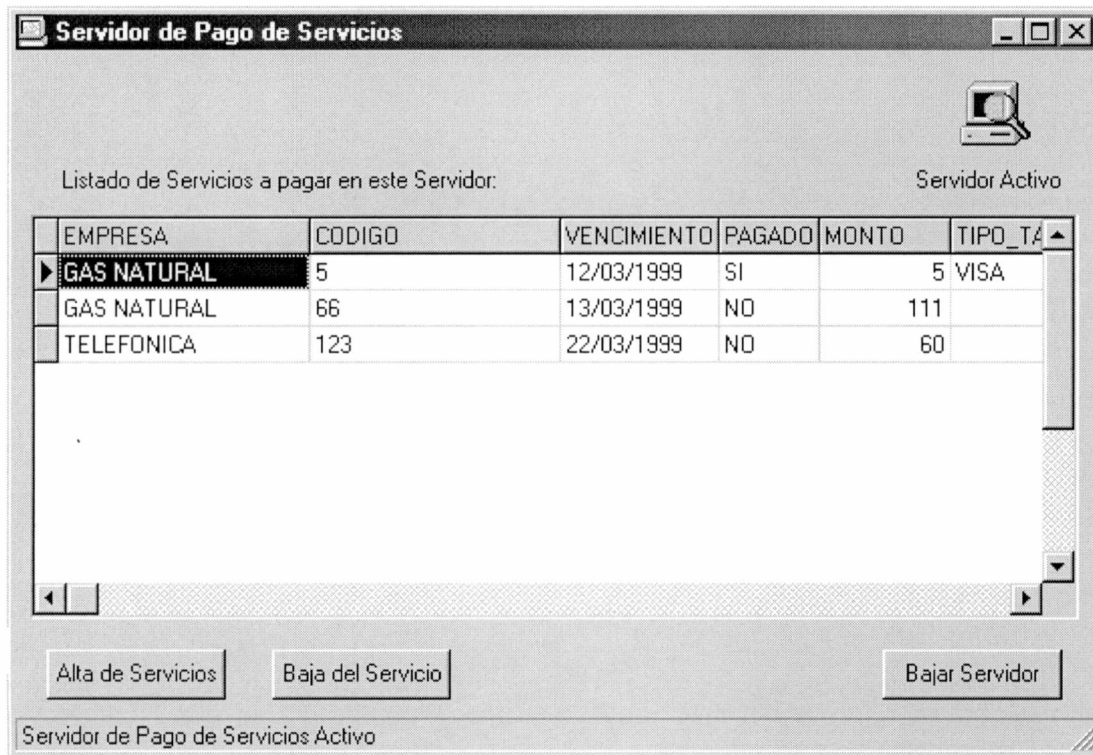
Esta Aplicación, mediante Sockets, recibe mensajes que debe procesar. Estos mensajes pueden ser de Testeo de Servicios a Pagar, o bien de Pagos.

Por ejemplo: de recibir un mensaje de tipo T (Testeo), el Servicio TELEFONICA, un Vencimiento y un Monto, buscará en sus Bases si existe tal Servicio y responderá al Programa Cliente que hizo la pregunta.

Ahora, si se le envía un mensaje de tipo P (Pago), con los datos anteriores mas los datos de una Tarjeta de Crédito válida, registrará el pago, previa verificación.

El Servidor, al registrar el pago de algún servicio, devuelve un Serial a modo de Constancia de Pago para el Cliente.

Además, el programa permite dar Altas y Bajas de Servicios.



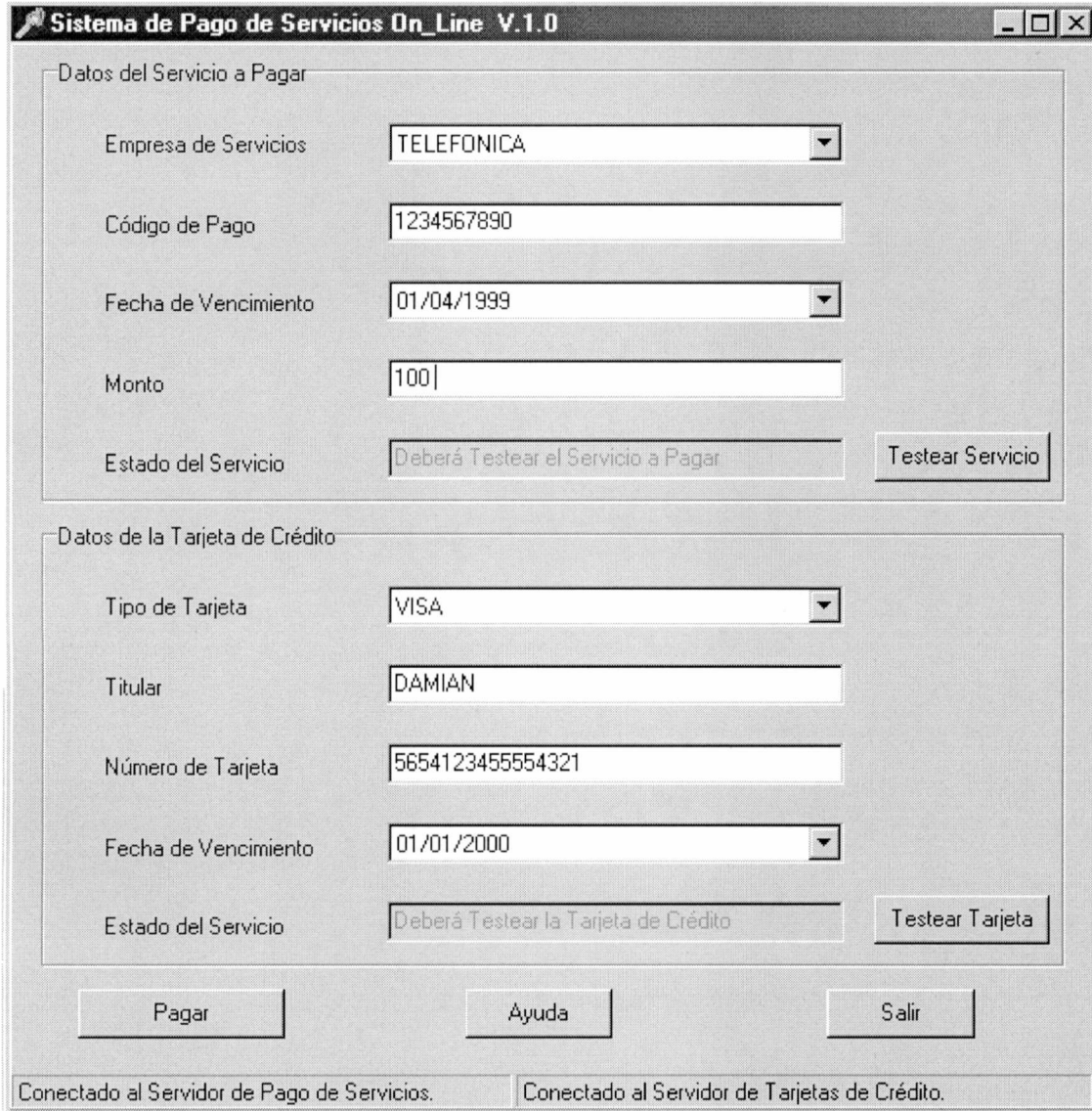
_ Cliente de Pago On_Line:

El Cliente de Pago On_Line es otra Aplicación Delphi 4 que utiliza Sockets para comunicarse con los Servidores de Tarjetas de Crédito y de Pago de Servicios.

El Programa Cliente será un programa muy fácil de usar, en el cual se deben ingresar los datos del Servicio a Pagar y los datos de la Tarjeta de Crédito que se utilizará para este pago.

Obviamente, detrás de esta pantalla, el Cliente se esta comunicando con los Servidores validando la información ingresada. Así también recibe las respuestas de estos Servidores y las procesa.

La interfaz del Programa Cliente todavía esta en desarrollo, debido a que se necesita una Interfaz lo mas amigable y fácil de usar posible. La próxima imagen es de la versión de desarrollo de la Interfaz del Cliente.



Sistema de Pago de Servicios On_Line V.1.0

Datos del Servicio a Pagar

Empresa de Servicios: TELEFONICA

Código de Pago: 1234567890

Fecha de Vencimiento: 01/04/1999

Monto: 100

Estado del Servicio: Deberá Testear el Servicio a Pagar

Testear Servicio

Datos de la Tarjeta de Crédito

Tipo de Tarjeta: VISA

Titular: DAMIAN

Número de Tarjeta: 5654123455554321

Fecha de Vencimiento: 01/01/2000

Estado del Servicio: Deberá Testear la Tarjeta de Crédito

Testear Tarjeta

Pagar Ayuda Salir

Conectado al Servidor de Pago de Servicios. Conectado al Servidor de Tarjetas de Crédito.

_ Sistema Seguro de Transmisión:

Para la Transmisión elegí el Encriptado de Clave Publica P.G.P. (Pretty Good Privacy) para Garantizar no solo que el Mensaje sea leído solamente por el Destinatario (Confidencialidad e Integridad), si no también, que se garantice que el Emisor sea quien dice ser (Autenticidad y Control de Usuarios).

Por eso, antes de enviar cualquier paquete de Información se Encriptan y se Firman los datos y luego recién se envían. Del otro lado, el que recibe los datos, debe desencriptarlos y luego, si se verifica la integridad y autenticidad de los mismos, procesarlos.

Por ejemplo, al mandar un Mensaje al Servidor de Tarjetas de Crédito, el Programa Cliente1 Encripta el Mensaje con la Clave Publica del Servidor de Tarjetas, luego lo Encripta con su Clave Privada (Firma Digital), y por último lo envía. En el otro extremo, el Servidor de Tarjetas de Crédito recibe el Mensaje, lo Desencripta con la Clave Publica del Cliente1, luego lo Desencripta con su Clave Privada y así obtiene el Mensaje para Trabajar.

Todo esto último es transparente al Usuario, salvo cuando hay que hacer mantenimiento de Claves Publicas, o cuando se piense que esta comprometida la propia clave secreta. Igualmente este mantenimiento es muy simple.

_ Parte Teórica:

La parte Teórica de esta Tesis consistirá de Documentos y Presentaciones tipo PowerPoint, que abarcarán los temas de:

- ❖ Criptografía de Clave Simétrica
- ❖ Criptografía de Clave Pública (Asimétrica)
- ❖ Distintos Algoritmos de Encriptación
- ❖ Autenticación y Firma Digital (Certificación de Autor, Contenido y Fecha)
- ❖ Estándares Criptográficos Internacionales
- ❖ Validez Legal de un Documento Electrónico
- ❖ Hashing para garantizar la Inalterabilidad
- ❖ Usos Actuales de Criptografía
- ❖ El Comercio Electrónico en Internet
- ❖ Pretty Good Privacy (PGP)
- ❖ Formas de Crackear los distintos Sistemas de Seguridad
- ❖ Posibles soluciones a los ataques del Sistema

Puntos a completar del Proyecto:

- _ Analizar si falta alguna Tabla en alguno de los Módulos.
- _ Terminar de Procesar el Material Teórico y pasarlo a .Doc o a .Ppt.
- _ Hacer programas para ABM de Tablas que contienen algunos parámetros tipo dirección IP de los Servidores, ports de comunicación, etc.
- _ Eliminar HardCodings.
- _ Agregar archivos .Ini para datos que necesitan los Programas.
- _ Terminar de Probar y Mejorar algunos puntos "flojos" en la implementación.
- _ Terminar la Interfaz del Programa Cliente haciéndola mas amigable y fácil de usar.
- _ Si bien todo el Sistema fue desarrollado en una sola maquina, ya lo he probado en Red. Pienso que para la Exposición Final seria interesante tener al menos tres maquinas conectadas en Red, para que sea mas ilustrativa.

Fecha de Entrega Final:

Estimamos la fecha de la Entrega Final para fines de Junio del presente año.

La Plata, 26 de Abril de 1999.

Parte Práctica de la **Tesis**

Trabajo de Grado Licenciatura en Informática
Romano Damián A.

Sistema de Pago On_Line

Estado final del proyecto:

Parte Práctica:

Sistema de Pago On_Line

Esta aplicación se pensó como un modelo de un Sistema de Pago de Servicios (luz, gas, teléfono, etc.) que sería instalado en puestos diseminados por distintos negocios de la ciudad a fin de que el público en general abone sus boletas.

Siendo que se utiliza para la comunicación entre módulos el protocolo TCP/IP, la plataforma donde corra el sistema podría ser tanto una Intranet como la misma Internet, con lo que los clientes podrían pagar desde sus casas.

Se quiso asegurar la seguridad de las comunicaciones entre los distintos módulos del Sistema y para ello se utilizó PGP (Pretty Good Privacy). Veremos más adelante como lo utilizan los distintos módulos.

Les solicitamos a los que utilicen este sistema que tengan en cuenta que el mismo no fue desarrollado (al menos en esta versión) para ningún cliente en especial, por lo que quizás se encuentren partes que podrían estar un poco más funcionales / amigables. No fue el objetivo lograr una interfaz del todo amigable, lo que se buscó es que el sistema funcione correctamente asegurando que la información que se envía entre los módulos, de ser interceptada, no sea descifrable.

El lenguaje de Programación utilizado para desarrollar la parte práctica de esta Tesis es Delphi 4.

El Sistema de Pago On_Line, consiste en 3 aplicaciones que describiremos a continuación:

Servidor de Tarjetas de Crédito:

Este servidor simula ser un servidor de validación y registro de compras de Tarjetas de Crédito. Posee un conjunto de tablas, de donde obtiene y almacena la información.

Base de Datos: Paradox V.7.

Plataforma: Este programa corre sobre Windows 95, 98 o NT.

Protocolo de Comunicación: TCP/IP con Ports parametrizados desde un archivo .INI. (Ver Descripción del Configuracion.Ini mas abajo).

El Servidor de Tarjetas de Crédito cumple una serie de funciones cada vez que se comunican con el:

- Paso 1. Recibe la Solicitud Encriptada enviada por un programa cliente
- Paso 2. Desencripta la Solicitud utilizando su clave Privada y la clave Pública del Cliente
- Paso 3. Verifica que el Formato de la Solicitud este correcto
- Paso 4. Verifica si los datos de la Solicitud (Tipo de Tarjeta, Número de Tarjeta, Titular y Vencimiento) se encuentran en su base de datos local y en que estado esta (Habilitada, Robada, etc).
- Paso 5. Se encripta el resultado de la búsqueda con la clave Privada del Servidor de Tarjetas y con la clave Pública del programa Cliente.
- Paso 6. Se envía al programa Cliente el resultado encriptado.

El Servidor de Tarjetas de Crédito utiliza PGP Versión 2.6.3i enviándole parámetros para que encripte o desencripte información. Una vez que PGP devuelve un resultado el Servidor lo procesa.

Todos los mensajes de consulta que recibe el Servidor han sido firmados con la clave privada del remitente y con la Pública del destinatario (en este caso el Servidor de Tarjetas de Crédito). Así también, cuando el Servidor envía una respuesta lo hace previamente firmándola con su clave privada y con la clave Pública del destinatario. El proceso de firmado se encuentra bien explicado en la parte de Criptografía de esta Tesis.

Esta aplicación también sirve de ABM de Tarjetas de Crédito: se pueden dar de alta tarjetas, modificarlas (por ejemplo si se produce un robo o si hubiera un error de carga de datos) o darlas de baja.

Todos los mensajes que reciben deben estar (y efectivamente lo están) encriptados y firmados digitalmente con PGP (Pretty Good Privacy) a fin de garantizar el origen de los datos. Asimismo, todos los resultados que el servidor envía se firman con su clave privada y luego se encriptan con la clave Pública del destinatario.

Configuracion.Ini

Este archivo tiene los parámetros necesarios para el Correcto Funcionamiento del Servidor de Tarjetas. Es leído automáticamente por la aplicación en el momento en que se ejecuta.

El archivo Configuracion.Ini del Servidor de Servicios y del Programa Cliente de Pago On_Line son muy similares al descripto acá para el Servidor de Tarjetas, al menos en estructura.

Veamos los datos que contiene el archivo Configuracion.Ini (los comentarios se hacen entre corchetes ({})):

CONFIGURACION.INI**[SERVER TARJETAS]**

Port=2440 {Port de Comunicación del Servidor de Tarjetas}
Ip=128.102.51.244 {Dirección IP del Servidor de Tarjetas}

[SERVER SERVICIOS]

Port=2441 {Port de Comunicación del Servidor de Servicios}
Ip=128.102.51.244 {Dirección IP del Servidor de Servicios}

[PATH TABLAS]

Path=C:\Tarjetas\Tablas {Path donde se encuentran las Tablas Locales de la Aplicación}

[NOMBRES]

Cliente=cliente {Nombre del Cliente para PGP}
Tarjetas=tarjetas {Nombre del Servidor de Tarjetas para PGP}
Servicios=servicios {Nombre del Servidor de Servicios para PGP}

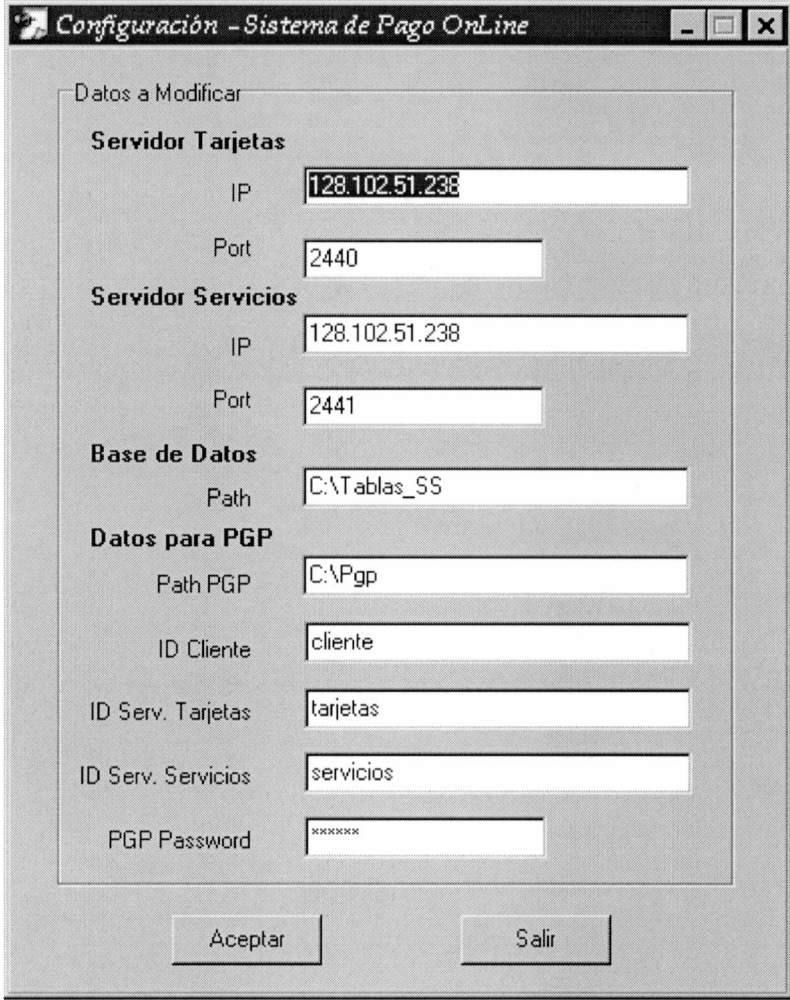
[PASSWORD]

MyPgpPass=` iCÃë {Password de la Aplicación para acceder a su KeyRing y obtener su propia Clave Privada}

[Path PGP]

Path=C:\Tarjetas\Pgp {Path donde se encuentra el programa PGP}

Para modificar los datos del archivo Configuracion.Ini existe en cada directorio de aplicación un ejecutable llamado Configuracion.Exe. A continuación mostramos como se ve la aplicación:



The image shows a Windows-style configuration window titled "Configuración - Sistema de Pago OnLine". The window contains several sections of configuration data:

- Datos a Modificar**
 - Servidor Tarjetas**
 - IP: 128.102.51.238
 - Port: 2440
 - Servidor Servicios**
 - IP: 128.102.51.238
 - Port: 2441
 - Base de Datos**
 - Path: C:\Tablas_SS
 - Datos para PGP**
 - Path PGP: C:\Pgp
 - ID Cliente: cliente
 - ID Serv. Tarjetas: tarjetas
 - ID Serv. Servicios: servicios
 - PGP Password: xxxxxxxx

At the bottom of the window are two buttons: "Aceptar" and "Salir".

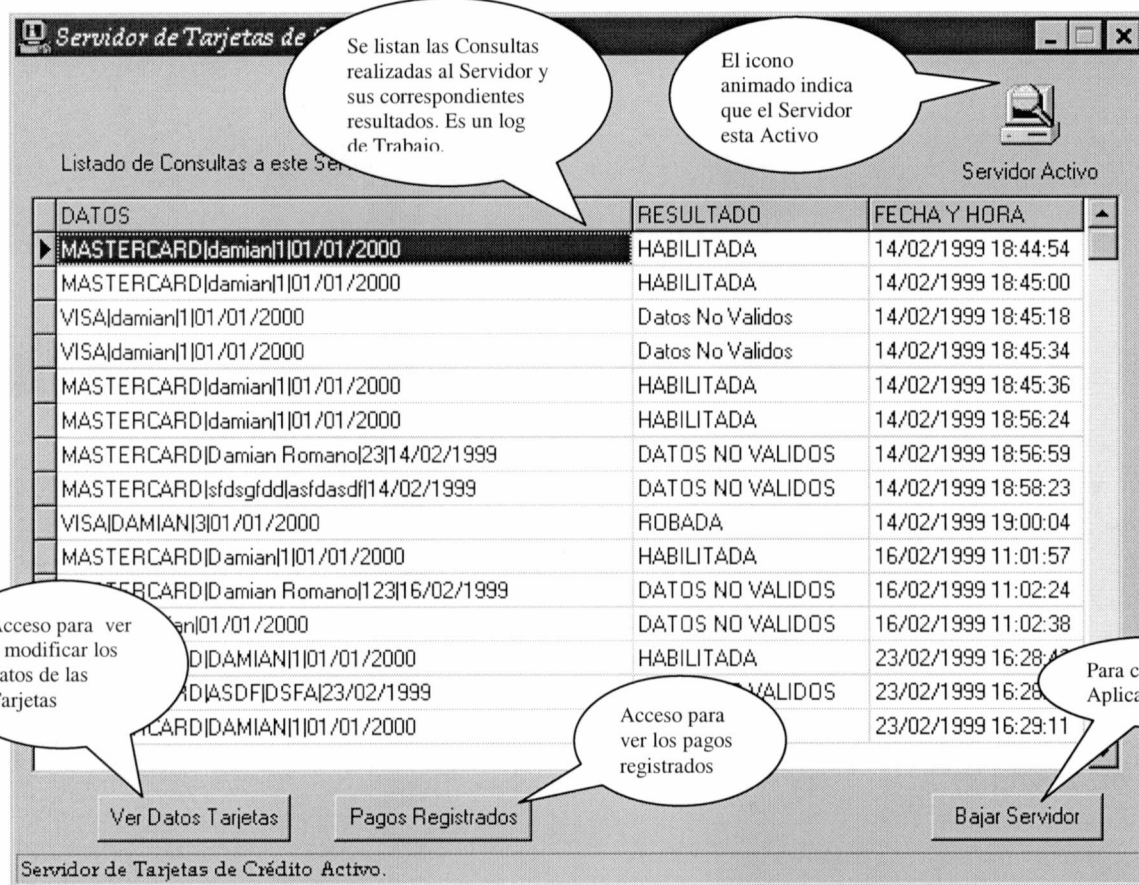
El Servidor de Tarjetas tiene una Interfaz simple e intuitiva. En su pantalla principal muestra el log de trabajo, en el se reflejan todas las consultas atendidas por el, la fecha en que fueron realizadas y su resultado.

En su esquina superior derecha posee un icono animado para indicar que el Servidor esta Activo.

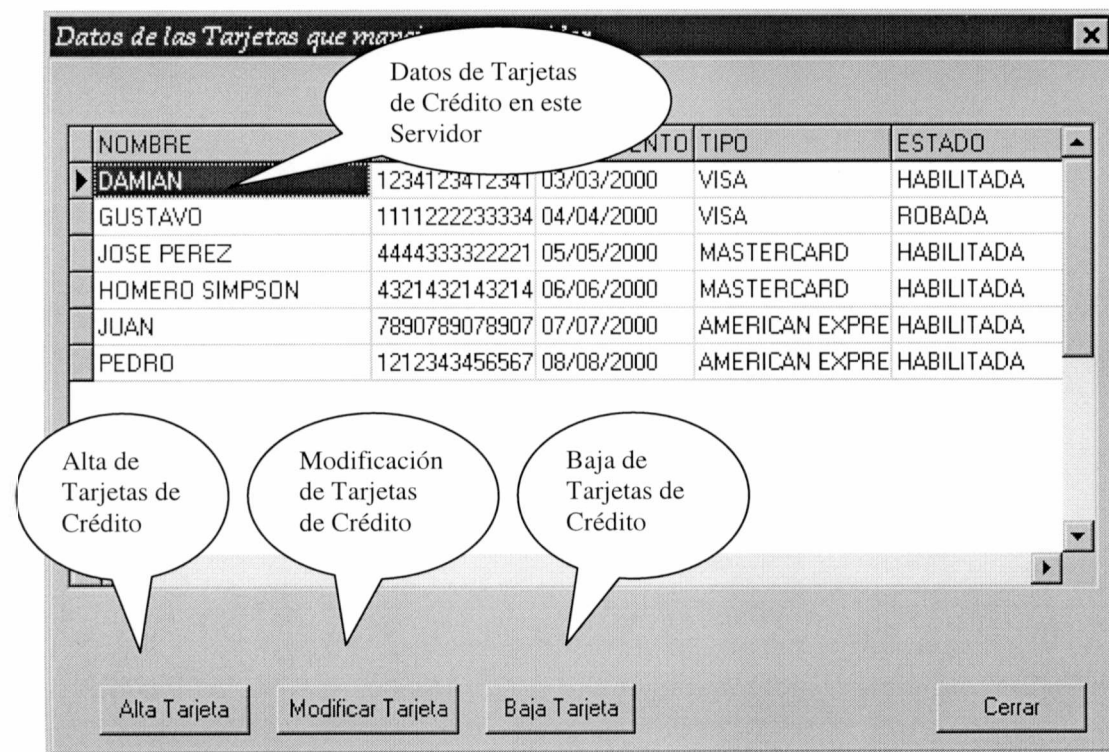
Si lo minimizamos el programa sigue ejecutando pero visualmente lo vemos como un Tray Icon (iconito pequeño en la esquina inferior derecha del Escritorio de Windows).

Al presionar el Botón llamado "Ver Datos Tarjetas" se abre otra pantalla permitiendo el Alta, Baja y Modificación de Tarjetas de Crédito, y al presionar el Botón llamado "Pagos Registrados" se muestra un listado de los pagos realizados en este Servidor a fin de que se cobren a los usuarios de las Tarjetas.

Gráficamente, el Servidor de Tarjetas se ve así:



Pantalla Ver Datos Tarjetas:



Pantalla de Alta de Tarjetas (la pantalla de modificación de Tarjetas es muy similar):

Alta de Tarjetas

Complete los Datos de la Tarjeta

Nombre ROMANO DAMIAN

Número 1234432156788765

Vencimiento Lunes , 01 de Enero de 2001

Tipo AMERICAN EXPRESS

Estado HABILITADA

Aceptar Cancelar

Alta de Tarjetas de Credito

Servidor de Servicios:

Este servidor simula ser un servidor de validación y registro de pago de Servicios tales como Teléfono, Gas, Luz, etc. Posee un conjunto de tablas, de donde obtiene y almacena la información.

Base de Datos: Paradox V.7.

Plataforma: Este programa corre sobre Windows 95, 98 o NT.

Protocolo de Comunicación: TCP/IP con Ports parametrizados desde un archivo .INI. (Ver Descripción del Configuración.Ini en la descripción del Servidor de Tarjetas, es muy similar).

El Servidor de Servicios cumple una serie de funciones cada vez que se comunican con el:

- Paso 7. Recibe la Solicitud Encriptada enviada por un programa cliente
- Paso 8. Desencripta la Solicitud utilizando su clave Privada y la clave Pública del Cliente
- Paso 9. Verifica que el Formato de la Solicitud este correcto
- Paso 10. Verifica si los datos de la Solicitud que le envía el cliente (Empresa, Código de Pago, Vencimiento, Monto y si fue abonado o no) se encuentran en su base de datos local y en que estado están (Abonado / No Abonado).
- Paso 11. Se encripta el resultado de la búsqueda con la clave Privada del Servidor de Servicios y con la clave Pública del programa Cliente.
- Paso 12. Se envía al programa Cliente el resultado encriptado.

El Servidor de Servicios utiliza PGP Versión 2.6.3i enviándole parámetros para que encripte o desencripte información. Una vez que PGP devuelve un resultado el Servidor lo procesa.

Todos los mensajes de consulta que recibe el Servidor han sido firmados con la clave privada del remitente y con la Pública del destinatario (en este caso el Servidor de Servicios). Así también, cuando el Servidor envía una respuesta lo hace previamente firmándola con su clave privada y con la clave Pública del destinatario. El proceso de firmado se encuentra bien explicado en la parte de Criptografía de esta Tesis.

Esta aplicación también sirve de ABM de Servicios: se pueden dar de alta Servicios o darlos de baja.

Todos los mensajes que reciben deben estar (y efectivamente lo están) encriptados y firmados digitalmente con PGP (Pretty Good Privacy) a fin de garantizar el origen de los datos. Asimismo, todos los resultados que el servidor envía se firman con su clave privada y luego se encriptan con la clave Pública del destinatario.

Configuracion.Ini

Este archivo tiene los parámetros necesarios para el Correcto Funcionamiento del Servidor de Servicios. Es leído automáticamente por la aplicación en el momento en que se ejecuta.

Para modificar los datos del archivo Configuracion.Ini existe en cada directorio de aplicación un ejecutable llamado Configuracion.Exe.

El detalle de los parámetros que contiene el Archivo .Ini lo podemos ver mas arriba en la sección que habla sobre el Servidor de Tarjetas. Los archivos .Ini de las aplicaciones son muy similares.

El Servidor de Servicios tiene una Interfaz simple e intuitiva. En su pantalla principal muestra los Servicios abonados y sin abonar.

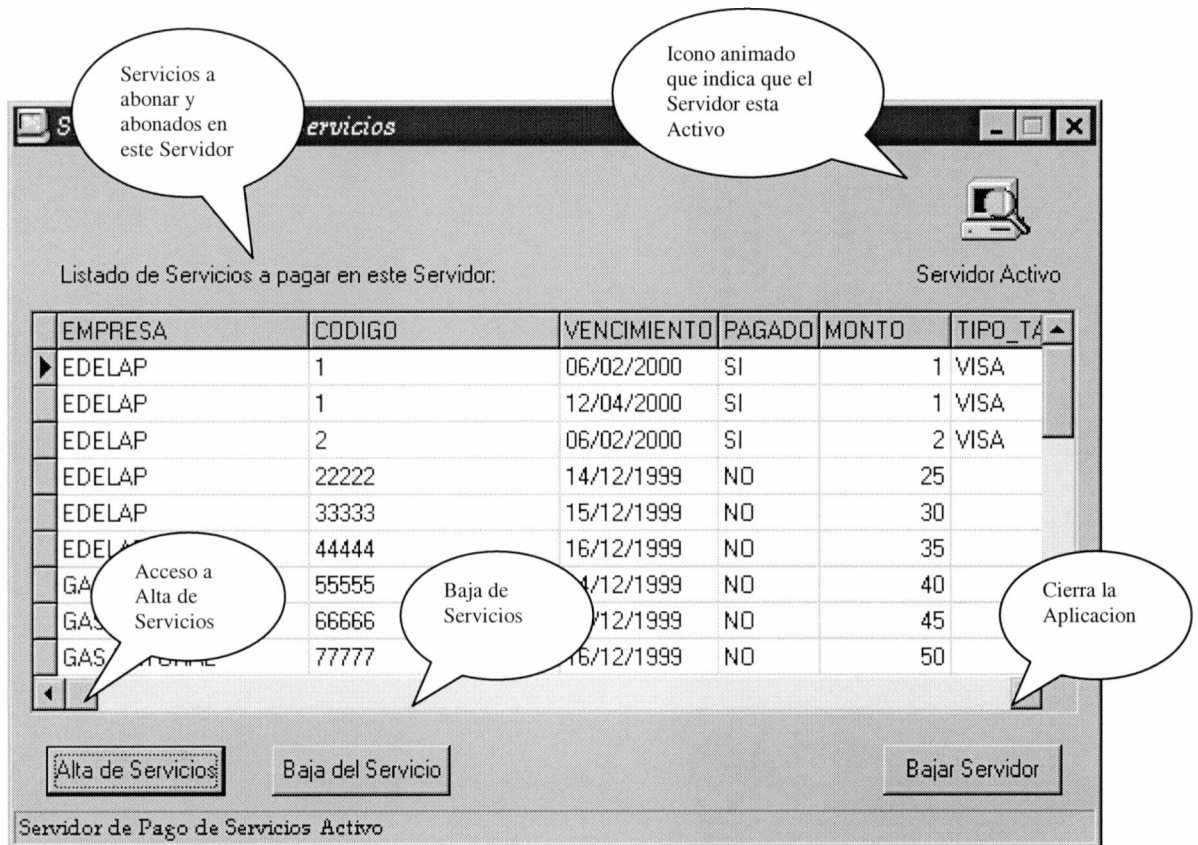
El programa en su esquina superior derecha posee un icono animado para indicar que el Servidor esta Activo.

Si lo minimizamos el programa sigue ejecutando pero visualmente lo vemos como un Tray Icon (iconito pequeño en la esquina inferior derecha del Escritorio de Windows).

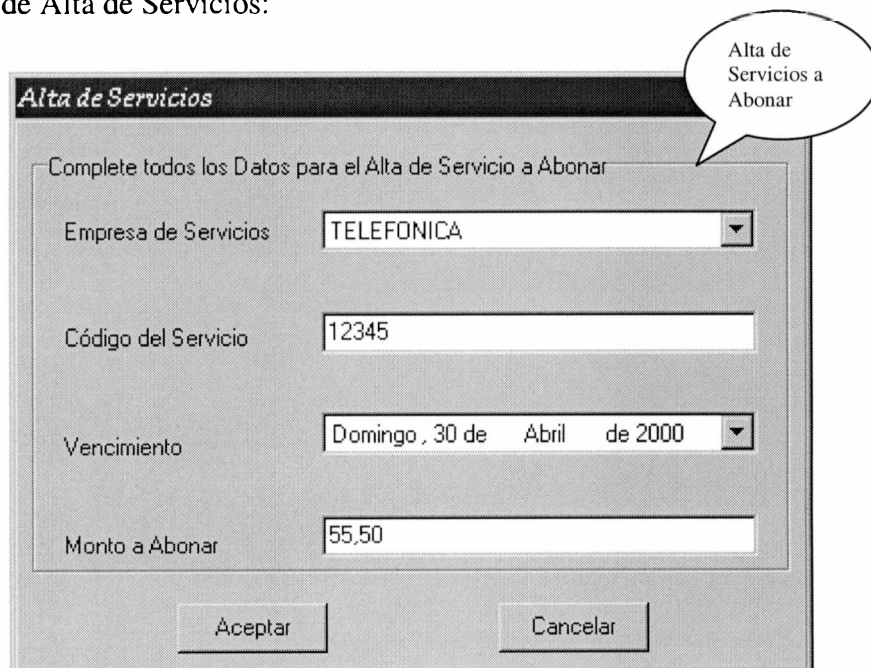
Al presionar el Botón llamado “Alta de Servicios” se abre una pantalla que nos permite dar de alta un nuevo Servicio que este sin abonar, quedando a la espera de que alguien lo haga.

Gráficamente, el Servidor de Servicios se ve así:

Pantalla Principal:



Pantalla de Alta de Servicios:



Cliente del Sistema de Pago On Line:

Esta es la interfaz con la que se encontrará el usuario que utilice el sistema. Su manejo es muy fácil e intuitivo. Se maneja tanto con teclas como con mouse y posee ayudas al usuario.

Base de Datos: DBase IV.

Plataforma: Este programa corre sobre Windows 95, 98 o NT.

Protocolo de Comunicación: TCP/IP con Ports parametrizados desde un archivo .INI. (Ver Descripción del Configuración.Ini en la descripción del Servidor de Tarjetas, es muy similar).

El programa cliente se comunica con el servidor de tarjetas de crédito y con el de servicios para validar los datos que se le ingresan y para realizar los pagos.

El Usuario Final realiza una serie de Pasos antes de abonar cada boleta de un Servicio con su Tarjeta de Crédito:

- Paso 1. Ingresar los datos del Servicio a abonar. Luego del ingreso de todos los datos necesarios el Sistema se comunica con el Servidor de Servicios para verificar que el servicio existe y aun no ha sido abonado.
- Paso 2. Si el Paso 1 es exitoso se ingresa al Paso 2 donde se ingresan y verifican los datos de la Tarjeta de Crédito. El Sistema se comunica automáticamente con el Servidor de Tarjetas para hacer dicha verificación.
- Paso 3. Si el Paso 2 es exitoso se ingresa al Paso 3 donde se muestran todos los datos para un ultimo control y se envía toda la información al los Servidores para abonar y registrar el pago del Servicio.

Nota: en todos los pasos se puede volver al paso anterior.

La seguridad de los datos que envía y recibe se obtiene gracias a que se utiliza PGP para encriptar los datos antes de ser enviados. Con este sistema, el usuario final puede estar seguro de que si sus datos son interceptados por un tercero, este no podrá descifrarlos.

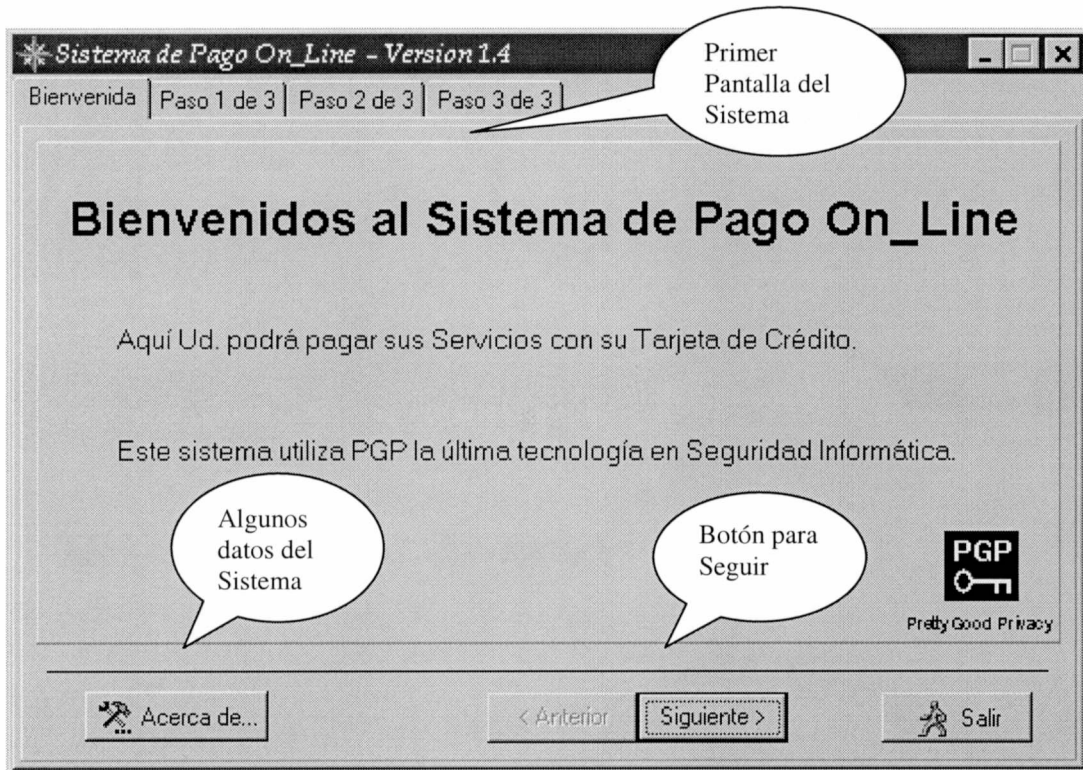
El Sistema de Pago On_Line utiliza PGP Versión 2.6.3i enviándole parámetros para que encripte o desencripte información. Una vez que PGP devuelve un resultado el Sistema lo procesa.

Todos los mensajes que recibe el Sistema han sido firmados con la clave privada del remitente y con la Pública del destinatario. Así también, cuando el Sistema envía una consulta a los Servidores lo hace previamente firmándola con su clave privada y con la clave Pública del destinatario. El proceso de firmado se encuentra bien explicado en la parte de Criptografía de esta Tesis.

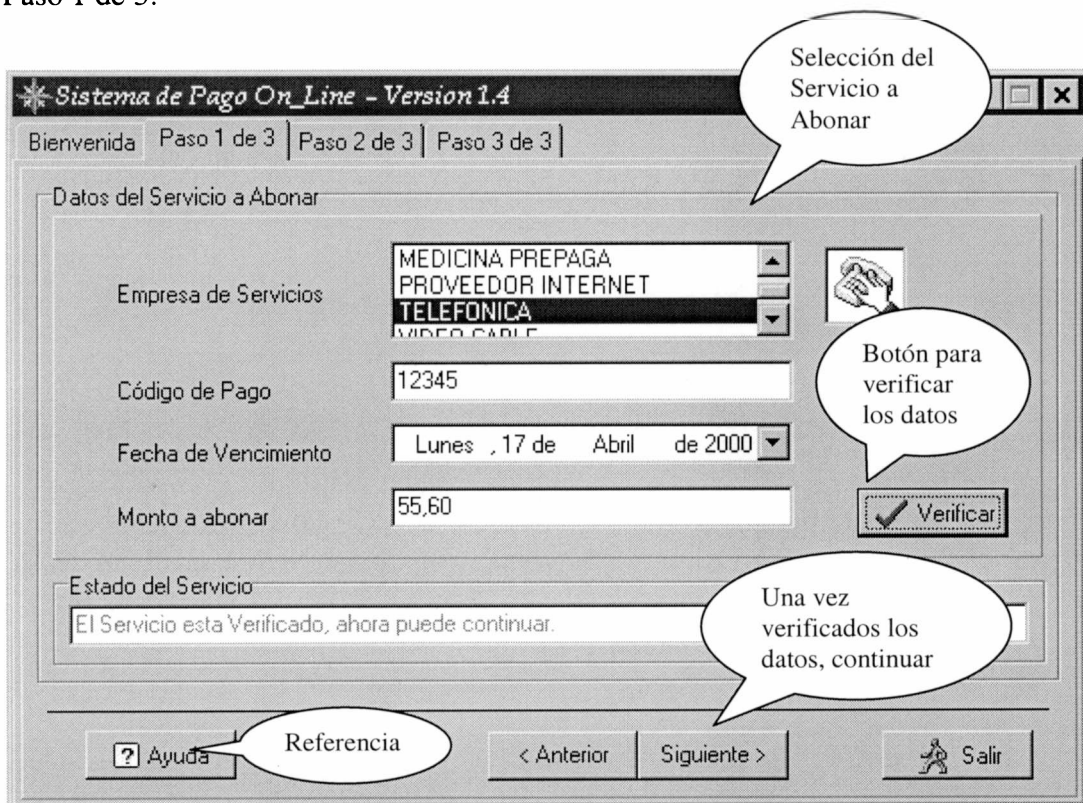
La interfaz del cliente de Pago On_Line se modificó desde el último informe logrando una aplicación mas amigable, simple y funcional.

Gráficamente, el Sistema de Pago On_Line se ve así:

Pantalla Principal:



Paso 1 de 3:



Paso 2 de 3:

Sistema de Pago On Line - Version 1.4

Bienvenida | Paso 1 de 3 | Paso 2 de 3 | Paso 3 de 3

Datos de la Tarjeta de Crédito

Tarjeta de Crédito	AMERICAN EXPRESS MASTERCARD VISA	
Titular	DAMIÁN	
Número de Tarjeta	1234123412341234	
Fecha de Expiración	Viernes, 03 de Marzo de 2000	

Verificar

Estado de la Tarjeta
Tarjeta de Crédito sin Verificar. Por favor, complete los datos y pres

Ayuda | Referencia | < Anterior | Siguiete > | Salir

Paso 3 de 3:

Sistema de Pago On Line - Version 1.4

Bienvenida | Paso 1 de 3 | Paso 2 de 3 | Paso 3 de 3

Verificación de Datos y Confirmación del Pago

Asegúrese de la Correctitud de los datos y presione el botón Pagar.
Vuelva atrás para modificar algún dato o Cancele la Operación con botón Cancelar.

Servicio	Tarjeta de Crédito
Empresa: TELEFONICA	Tarjeta: VISA
Código: 12345	Titular: DAMIÁN
Vencimiento: 17/04/2000	Número: 1234123412341234
Monto: 55,60	Expiración: 03/03/2000

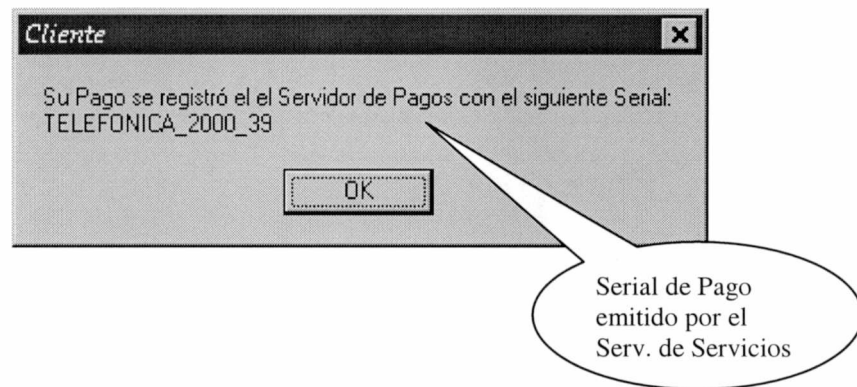
Cancelar Pagar

Ayuda | Referencia | < Anterior | Siguiete > | Salir

Informe del Servidor de Tarjetas registrando el pago:



Informe del Serial asignado al pago por el Servidor de Servicios:



El Servidor de Servicios devuelve un Serial asignado a la operación que el Usuario Final pueda hacer cualquier tipo de reclamo.

Base de Datos de las Aplicaciones:

El Servidor de Tarjetas de Crédito y el Servidor de Servicios utilizan un conjunto de Tablas Paradox.

Para el Cliente de Pago On_Line se había pensado utilizar también tablas Paradox, pero estas últimas no permitían guardar datos binarios como gráficos, por lo que se decidió cambiar a Dbase para permitir guardar los pequeños gráficos de las Tarjetas de Crédito y de los distintos Servicios aceptados por el Sistema de Pago On_Line.

A continuación describiré cada una de las Estructuras y Datos de las distintas Bases de Datos:

Base de Datos del Servidor de Tarjetas de Crédito:

Tabla	Estructura	Descripción
Estados	ID_ESTADO (N) DESCRIPCION (A 50)	Posee los estados posibles que puede tener una tarjeta (habilitada, robada, etc.)
Log	DATOS (A 50) RESULTADO (A 20) FECHA Y HORA (@)	Aquí guarda las consultas que le hicieron y el resultado obtenido para cada una de ellas.
Reg_Pagos	TIPO (A 20) NUMERO_TARJETA (A 20) MONTO (N) DESCRIPCION (A 20) FECHA_PAGO (D)	Aquí se almacenan los pagos realizados con las tarjetas con el detalle de lo que se abono.
Tarjetas	NOMBRE (A 50) NUMERO (A 16) VENCIMIENTO (D) TIPO (A 20) ESTADO (A 50)	Guarda la información de una Tarjeta de Crédito.
Tipos_Tarjetas	ID_TARJETA (N) NOMBRE (A 20)	Tiene los tipos de Tarjetas (Visa, Mastercard, etc.)

Base de Datos del Servidor de Servicios:

Tabla	Estructura	Descripción
Serial_SS	SERIAL (N)	El serial es un valor que se asigna a una operación de pago exitosa. Se devuelve para control del Usuario.
Servicios	EMPRESA (A 20) CODIGO (A 20)	Mantiene la información de los Servicios a abonar y los que han sido

	VENCIMIENTO (D) MONTO (N) PAGADO (A 2) TIPO_TARJETA (A 20) TITULAR (A 50) NUMERO_TARJETA(A16) VENCIMIENTO_TARJ (D) SERIAL (N)	abonados. Los que han sido abonados tendrán asignados un serial y los datos de la Tarjeta de Crédito con la que se pago el mismo.
Tipos_Servicios	ID_EMPRESA (N) NOMBRE (A 20)	Posee los nombres de las Empresas de las que se pueden abonar servicios.

Base de Datos del Cliente de Pago On Line:

Tabla	Estructura	Descripción
Tipos_Tarjetas	ID_TARJETA (N 3) DESC (C 20) GRAFICO (B)	Posee los tipos de tarjetas aceptados por el Sistema de Pago On_Line.
Tipos_Servicios	ID_EMPRESA (N 3) NOMBRE (C 20) GRAFICO (B)	Posee las Empresas de Servicios que tiene el Sistema de Pago On_Line.
Ayuda	AYUDA1 (M) AYUDA2 (M) AYUDA3 (M)	Almacena las ayudas del Sistema.

Instalación:

REQUISITOS MINIMOS:

Pentium 100 Mhz

Windows NT, 95 o 98

32 MB Ram

25 MB libres en Disco Duro

Protocolo TCP/IP instalado, configurado con una dirección IP válida.

Configuración Regional: Español (Argentina)

Instale las aplicaciones siguientes en el orden que se muestran. Y respete las instrucciones de configuración que tienen.

Instalación del PGP - Pretty Good Privacy

- Ejecute: \\Pretty Good Privacy\144mb\Disk1\Setup.exe
- Acepte todas las opciones por defecto que se le presenten. Si se modificara alguno luego habrá que modificar los archivos .Ini.

Eventualmente se podrá cambiar el resto de los datos.

NOTA MUY IMPORTANTE: El PGP se debe instalar 1 sola vez en cada maquina donde se ejecuten cualquiera de las 3 aplicaciones siguientes.

Instalación del Servidor de Tarjetas de Crédito

- Ejecute: \\Servidor de Tarjetas\144mb\Disk1\Setup.exe
- Acepte todas las opciones por defecto que se le presenten.
- Una vez instalado ingresar a la Configuración (Inicio, Programas, Servidor de Tarjetas, Configuración)
- Poner la IP correcta en donde corresponde
- Al dar Ok se pedirá que se reinicie la maquina de ser necesario. NO PASAR POR ALTO ESTE PASO SI SE SOLICITA.
- Ejecutar la aplicación (Inicio, Programas, Servidor de Tarjetas, Servidor de Tarjetas de Crédito)

Eventualmente se podrá cambiar el resto de los datos.

Instalación del Servidor de Servicios

- Ejecute: \\Servidor de Servicios\Disk1\Setup.exe
- Acepte todas las opciones por defecto que se le presenten.

- Una vez instalado ingresar a la Configuración (Inicio, Programas, Servidor de Servicios, Configuración)
- Poner la IP correcta en donde corresponde
- Al dar Ok se pedirá que se reinicie la maquina de ser necesario. NO PASAR POR ALTO ESTE PASO SI SE SOLICITA.
- Ejecutar la aplicación (Inicio, Programas, Servidor de Servicios, Servidor de Servicios)

Eventualmente se podrá cambiar el resto de los datos.

Instalación del Cliente (Sistema de Pago On Line)

- Ejecute: \\Sistema de Pago On_Line\Disk1\Setup.exe
- Acepte todas las opciones por defecto que se le presenten.
- Una vez instalado ingresar a la Configuración (Inicio, Programas, Cliente, Configuración)
- Poner la IP correcta del Servidor de Servicios y de Tarjetas en donde corresponde
- Al dar Ok se pedirá que se reinicie la maquina de ser necesario. NO PASAR POR ALTO ESTE PASO SI SE SOLICITA.
- Ejecutar la aplicación solo si los Servidores están activos (Inicio, Programas, Cliente, Sistema de Pago On_Line), de lo contrario indicara un error. Si esto ultimo sucede revise el IP y los Ports en las configuraciones de las 3 aplicaciones y de la/s maquina/s.

Eventualmente se podrá cambiar el resto de los datos.

Nota Importante: Obviamente, y esa es la idea, se pueden hacer correr las distintas aplicaciones en distintos equipos, o sea, poner el Servidor de Tarjetas en el equipo 1, el Servidor de Servicios en el Equipo 2 y el Cliente en los equipos 3, 4, 5...n. Para que esto funcione correctamente se deberán configurar correctamente todas aplicaciones y se deberá instalar el PGP en todos los equipos donde corran las aplicaciones.

En el directorio Documentación se encuentra la Documentación del Sistema.

En el directorio Fuentes se encuentran los fuentes (Delphi 4)

Se hicieron varias pruebas de instalación sobre equipos de distintas velocidades y con distintos sistemas operativos, obviamente dentro de los Requisitos Mínimos descriptos, sin que se registraran mayores inconvenientes.

Probando el Sistema:

Una vez instalados todos los módulos del Sistema de Pago On_Line, se deberán ejecutar las distintas aplicaciones.

El Servidor de Servicios y el Servidor de Tarjetas deberán ejecutarse antes que el cliente del Sistema de Pago On_Line, de lo contrario este último mostrará un mensaje de error y se cerrará.

La aplicación Cliente buscará a los Servidores en la IP y Ports que se indiquen en su archivo Configuracion.Ini. Si hubiera algún inconveniente se deberá revisar todos los archivos Configuracion.Ini.

El Sistema se ha instalado en varios equipos sin que se presentaran mayores errores. Por favor, remitirse a Requisitos Mínimos ante cualquier inconveniente.

Datos de la Implementación:

Passwords para abrir el Key Ring PGP:

Para abrir el Key_Ring PGP y extraer su propia clave secreta cada aplicación debe saber su password. Para proteger este password utilizo una librería llamada Crypt32.Pas (que se puede examinar en el CD adjunto).

La clave que se ingresa desde el módulo de Configuración se encripta con Crypt32 y se guarda localmente en las Tablas Paradox del modulo correspondiente. Cuando se la necesita se la desencripta con la misma clave.

Crypt32 es excelente para encriptar Passwords y Texto en General, utiliza un modulo de 32 bits para encriptar y desencriptar. Posee 2^{96} variantes. Es muy difícil de Hackear. Aquí estamos utilizando Clave Simétrica, ya que para este caso en particular encuadra perfectamente pues no la necesitamos compartir con nadie mas.

Mensajes entre los Clientes y los Servidores:

A continuación se muestran los mensajes (en texto plano: sin encriptar) que viajan entre los servidores y los clientes del Sistema de Pago On_Line:

Mensajes desde el Cliente al Servidor de Servicios:

Mensaje de Testing:

El Programa Cliente envía un Mensaje de Testing (T) al Servidor de Servicios con el objeto de que este ultimo chequee si los datos enviados (Empresa, Código de Pago, Fecha de Vencimiento y Monto) son correctos.

T|EMPRESA|CODIGO|FECHAVENC|MONTO

Mensaje de Pago:

El Mensaje de Pago, es el que el Programa Cliente envía al Servidor de Servicios indicándole los detalles del Pago que se esta realizando con los datos de la Tarjeta de Crédito.

P|EMPRESA|CODIGO|FECHAVENC|MONTO|TIPOTARJETA|#TARJETA|VENC_TARJ

Mensajes desde el Servidor de Servicios al Cliente:**Mensaje de Respuesta al Testing:**

T SI	Si ya esta pagado el Servicio
T NO	Si NO esta pagado y existe en la Base de Datos para pagarlo
T NOVALIDO	Si NO Existe en la Base de Datos del Servidor

Mensaje de Respuesta al Pago:

P|EMPRESA|AÑO|SERIAL **Es el Comprobante del Cliente**

Mensajes desde el Cliente al Servidor de Tarjetas de Crédito:**Mensaje de Testing:**

El Programa Cliente envía un Mensaje de Testing (T) al Servidor de Tarjetas con el objeto de que este ultimo chequee si los datos enviados (Titular, Numero de Tarjeta y Fecha de Vencimiento) son correctos.

T|TIPO|TITULAR|#TARJETA|VECHAVENC

Mensaje de Pago:

El Mensaje de Pago, es el que el Programa Cliente envía al Servidor de Tarjetas indicándole los detalles del Pago que se realizó con los datos con la Tarjeta de Crédito.

P|TIPO|TITULAR|#TARJETA|VECHAVENC|MONTO|EMPRESA

Mensajes desde el Servidor de Tarjetas de Crédito al Cliente:**Mensaje de Respuesta al Testing:**

T NOVALIDO	No Existe en las Bases de Datos del Servidor de Tarjetas
T HABILITADA	Existe y esta Habilitada
T ROBADA	Obvio
T VENCIDA	Obvio
T DADA DE BAJA	Obvio

Mensaje de Respuesta al Pago:

P|OK

User Names y Passwords para utilizar PGP

PGP permite tres tamaños de Clave RSA:

- **512 Bits** – Low Comercial Grade, rápida pero menos segura
- **768 Bits** – High Comercial Grade, mediana velocidad, buena seguridad
- **1024 Bits** – “Military” Grade, lento pero de máxima velocidad

En este proyecto, tanto para los Servidores como para los Clientes, utilizamos el tamaño **768 bits para la Clave RSA: High Commercial Grade**, medium speed, good security. Se pensó en este tamaño de Clave por estar pensada para Comercio (High Comercial Grade) y por ser medianamente rápido tanto para encriptar como para desencriptar.

Usuarios y Passwords:

Servidor de Tarjetas de Crédito:

UserName: servidor tarjetas <servidor@tarjetas.com>

Password: tarjetas

Servidor de Pago de Servicios:

UserName: servidor servicios <servidor@servicios.com>

Password: servicios

Cliente:

UserName: cliente prueba <cliente@prueba.com>

Password: prueba

Por que P.G.P. (Pretty Good Privacy)

Por que P.G.P. (Pretty Good Privacy):

Un protocolo de seguridad es la parte visible de una aplicación, es el conjunto de programas y actividades programadas que cumplen con un objetivo específico y que usan esquemas de seguridad criptográfica.

El **PGP** es un protocolo libre ampliamente usado de intercambio de archivos y de correo electrónico seguros, es uno de los más conocidos.

Este y cualquier protocolo de seguridad procura resolver algunos de los problemas de la seguridad como la integridad, la confidencialidad, la autenticación y el no rechazo, mediante sus diferentes características.

Las características de los protocolos se derivan de las múltiples posibilidades con que se puede romper un sistema, es decir, robar información, cambiar información, leer información no autorizada, y todo lo que se considere no autorizado por los usuarios de una comunicación por red.

Elegí PGP por ser un software de Criptografía de alta seguridad existente para la mayoría de los sistemas operativos (PC DOS, UNIX, VAX/VMS, Macintosh y Amiga, entre otros), por su flexibilidad y por sus características prácticas para encriptar, desencriptar y manipular archivos. Fue creado por Phil Zimmerman, uno de los héroes de Internet, que corre riesgo de ir a la cárcel por haber hecho público un algoritmo de encriptación secreto. El PGP es un algoritmo de clave pública, que permite que las personas intercambien mensajes con privacidad, autenticación y conveniencia.

- Privacidad porque sólo aquellos para los cuales estaba destinado un mensaje pueden recibirlo.
- Autenticación porque puede verificarse que un mensaje público publicado por una determinada persona haya sido escrito sólo por ella (esto es lo que suele llamarse firma electrónica).
- Conveniencia porque la privacidad y la autenticación se obtienen sin los problemas de manejar claves que se suelen tener con el software común de Criptografía. No se requiere un canal seguro para intercambiar claves entre usuarios.

Puede obtener el PGP versión shareware en <http://www.pgp.com>

Posibilidad de utilización / continuación para un uso mas “real”

Posibilidad de utilización / continuación para un uso mas “Real”:

El sistema de Pago On_Line se penso originalmente para instalar equipos en distintos lugares físicos, como por ejemplo: supermercados, estaciones de servicio, etc. a fin de que el público en general pueda pagar sus facturas. También se penso para ser utilizado desde el hogar.

Para insertar en el mercado la idea del Sistema de Pago On_Line, habría que darle a este último un enfoque mas comercial, el cual no posee por no ser este el objetivo de este Trabajo de Grado.

Si una empresa se interesara en el sistema se deberían tener en cuenta mejoras como las siguientes, entre muchas otras:

- Configurar el Sistema para poder ingresar telefónicamente a una Intranet / Internet.
- Diseñar el programa cliente desde un punto de vista mas comercial (una interfaz mas gráfica, con mas interacción con el usuario, etc.)
- Hacer que el sistema maneje su ganancia (obviamente el sistema tendría un costo y una ganancia, y habría que estimarlos).
- Utilizar otro tipo de Tablas sería muy recomendable. Las Tablas Paradox, aquí utilizadas, son muy inferiores a cualquier motor de bases de datos como por ejemplo Oracle 8.0. La parte buena de Paradox es que es totalmente gratuita su utilización tanto para estudio como para cualquier actividad comercial.
- Se puede lograr, sin demasiado esfuerzo, que los dos Servidores (el de Tarjetas y el de Servicios) corran como Servicios en Windows NT.
- Sería útil estandarizar y mejorar el sistema de Seriales que emite el Servidor de Servicios, esto daría mas confianza / seguridad al Usuario Final.
- Para lograr que el sistema sea un poco mas “fool proof” se podrían eliminar los archivos .INI y en su lugar utilizar secciones en el Registro de Windows.
- Testear todo el producto con un equipo de testeo a fin de encontrar todos los “bugs” que pudiera tener y eliminarlos.

Sistema de Pago On_Line

(Presentación Power Point)

Sistema de Pago On_Line

Criptografía de Clave Pública

Damián Romano

Información general del proyecto

Metas del proyecto

Descripción del proyecto

Sistema Seguro

Sistema de Comunicación

Servidor de Tarjetas de Crédito

Servidor de Servicios

Clientes

Comentarios

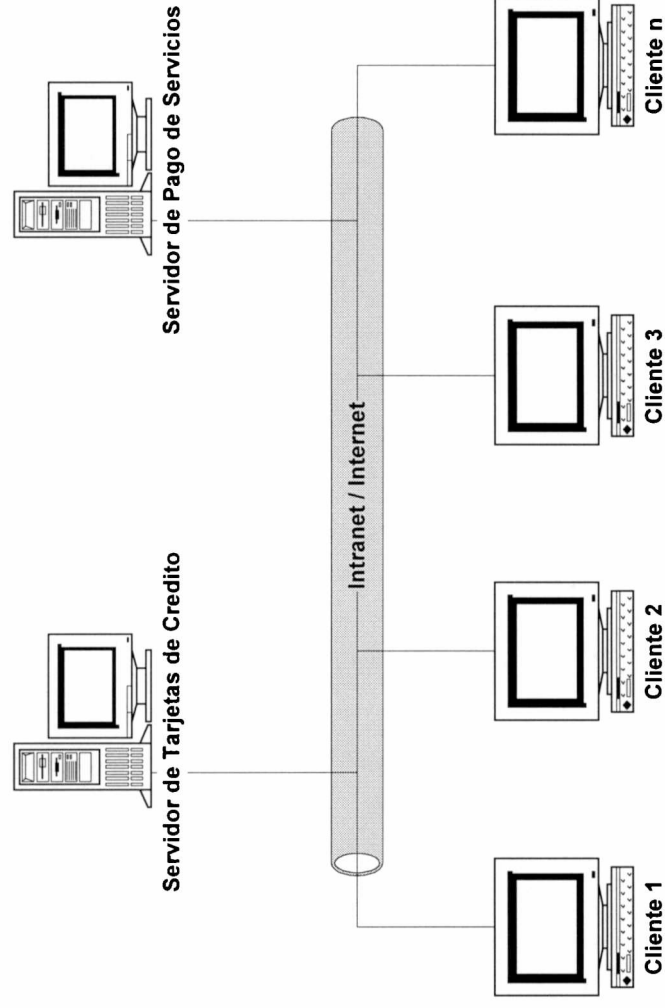
Metas del proyecto

- **Realizar un Sistema de Pago de Servicios On_Line**
- **Asegurar que todos los Movimientos de Información sean Seguros**
- **Utilizar Criptografía de Clave Pública para el Envío de Datos entre Clientes y Servidores**

Descripción

- **El Sistema posee dos Servidores y n Clientes**
- **Un Servidor verifica la validez de las Tarjetas de Crédito y registra los pagos, y el otro valida los Servicios y almacena los pagos.**
- **Cada Cliente chequea sus Tarjetas y Servicios y efectúa el pago de Servicios mediante Tarjeta de Crédito.**
- **Gráficamente...**

Sistema de Pago On_Line



Todas las Transacciones entre los Clientes y los Servidores son totalmente seguras gracias al Sistema de Criptografia Publica PGP (Pretty Good Privacy)

Sistema Seguro

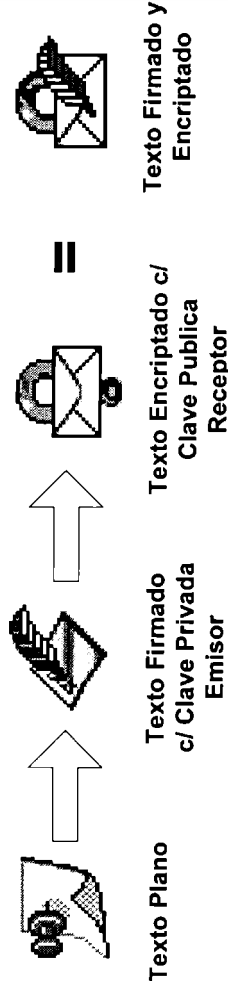
- **Antes de efectuar un envío de información se Encripta y Firma el Texto Plano...**
- **...y cada vez que se recibe información se Desencripta y se Verifica la identidad del que origino el mensaje.**
- **Obviamente, si la Autenticidad de la información no es verificada se descarta.**
- **Gráficamente...**

Sistema Seguro

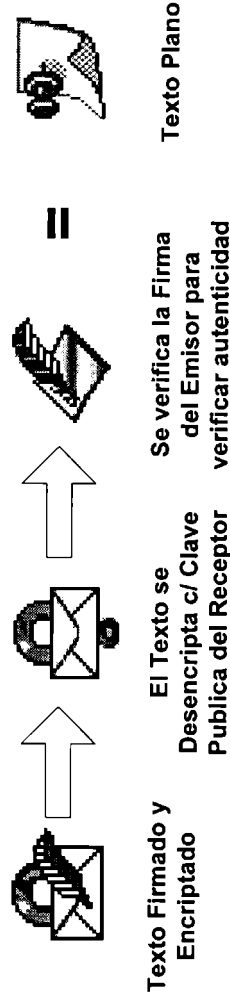
El sistema, cada vez que necesite enviar información, la firmará y la encriptará. Por lo que, al llegar al receptor, este tendrá que desencriptarla y verificar la firma digital.

En ningún momento se transmite un texto plano!

En el Emisor:



En el Receptor:



Sistema de Comunicación

- **El sistema utiliza Sockets TCP/IP para mantener la comunicación entre Clientes y Servidores.**
- **Los Ports que utilizan estan parametrizados en un Archivo INI, y existe una aplicación para modificarlos en el Cliente.**

Servidor de Tarjetas de Crédito

- **Es una aplicación, que mediante sockets TCP/IP recibe datos sobre tarjetas de crédito y responde indicando si los datos son válidos o no, de acuerdo a su propia base de datos.**
- **Atiende varios clientes a la vez.**
- **Además, almacena los pagos que se hacen con cada tarjeta.**

Servidor de Servicios

- **Es una aplicación, que recibe información sobre distintos servicios (Telefónica, Edelap, etc) y responde si los datos ingresados existen en su base de datos.**
- **Atiende varios clientes a la vez.**
- **El Servidor se encarga de registrar las operaciones.**

Cientes

- **Esta aplicación, es la interfaz para introducir los datos de la tarjeta de crédito y del servicio a pagar, y así realizar el pago.**

Comentarios

- **PGP proporciona un nivel de seguridad que nada tiene que envidiarle a cualquier otro sistema criptográfico jamás desarrollado.**
- **Proporcionará un gran rendimiento si se emplea correctamente, pero su uso inadecuado podría convertirlo en una protección totalmente inútil.**

24/07/2000

Damian Romano

dromano@bigfoot.com

Gracias...

Damián A. Romano
dromano@bigfoot.com

Anterior

Siguiente

Parte Teórica de la Tesis

Trabajo de Grado Licenciatura en Informática
Romano Damián A.

Criptografía

Criptografía

Objetivo: *Esta sección tiene como propósito explicar las herramientas de seguridad informática más comunes, tratando de enfatizar la importancia de la criptografía, y dando una explicación lo más sencilla posible.*

Prefacio

El uso de técnicas criptográficas tiene como propósito prevenir algunas faltas de seguridad en un sistema computarizado. La seguridad en general debe de ser considerada como un aspecto de gran importancia en cualquier corporación que trabaje con sistemas computarizado. El hecho que gran parte de actividades humanas sea cada vez más dependiente de los sistemas computarizados hace que la seguridad juegue un papel importante.

Quizá antes sea importante mencionar algunos datos relacionados con la seguridad antes de comenzar con el desarrollo del tema

En el reporte reciente “Computer Crime Survey” del **FBI**, proporcionado por Secure Site E-News del 22 de mayo de 1999, de la compañía VeriSign, se dieron los siguientes datos:

Se estudiaron 521 compañías de varias ramas de la industria y de diferentes tamaños. Estas están actualmente trabajando para que su sistema computarizado sea seguro.

El 61% de estas compañías ha tenido experiencias de perdida debido al uso de no autorizado de su sistema computarizado.

El 32 % de estas organizaciones están usando ahora métodos de identificación segura en su sitio de internet.

El promedio de perdida de robo o perdida de información esta sobre \$1.2 M de dólares.

El promedio de perdida por sabotaje esta sobre \$1.1 M dólares.

El 50% de todas las compañías reportaron abuso de del uso de la red

El 94% de las organizaciones tiene actualmente un sitio en la web.

A la pregunta ¿qué tipo de tecnología de seguridad usa? Se contesto con lo siguiente:

Se cuenta con un control en el acceso, el 89%.

Cuenta con archivos cifrados, el 59%.

Cuenta con sistema de passwords, el 59%.

Usa Firewalls, el 88%.

Una sistema de log-in cifrados, el 44%.

Usa smart-cards, 37%.

Detención de intrusos, 40%.

Certificados digitales para la autenticación, 32%.

A la pregunta ¿Cuál es más frecuente origen de un ataque?

Un “hacker” independiente, un 74%.

Un competidor, un 53%.

Un empleado disgustado, un 86%.

¿Su organización provee servicio de comercio electrónico?

Si, el 29%.

¿Su web-site ha tenido un acceso no autorizado en los últimos 12 meses?

Si, un 18%.

No, un 44%.

No sabe un 38%.

Enseguida damos un reporte que se tiene del tema en Europa, dado a conocer en unos cursos de criptografía industrial en Bélgica en junio de 1997. En donde se mide la frecuencia de incidentes de seguridad de la información relacionada con sus causas.

Frecuencia	Razón
50-60%	Errores debido a la inexperiencia, reacciones de pánico, mal uso,...
15-20%	Empleados disgustados, accidentes de mantenimiento,...
10-15%	Desastres naturales como inundaciones, incendios,...
3-5%	Causas externas: “hackers”

Otro aspecto importante a considerar es el crecimiento enorme que ha tenido la red internet, algunos datos importantes son los siguientes, proporcionados por Paul Van Oorschot de Entrust Technologies en una conferencia del ciclo The Mathematics of Public Key Cryptography en junio de 1999:

Se duplica el trafico de internet cada 100 días.

En enero de 1999 hubo 150 millones de personas en línea, 75 de ellas en USA.

El comercio sobre internet se duplica cada año.

Podría llegar a \$1 trillón de dólares lo comercializado en internet en el año 2002.

A la radio le tomo 40 años, a la televisión 10 años para alcanzar 50 millones de usuarios a la red le ha tomado menos de 5.

Estos datos sólo son algunos de los que frecuentemente son dados a conocer por algún medio, y aunque algunos obedecen a intereses comerciales, lo que sí es verdadero es el enorme cambio que han tenido gran cantidad de actividades a raíz del uso de internet que incluso se ha considerado como el invento más importante de fin de siglo y de ahí lo primordial de todo lo relacionado con su seguridad.

Siempre podremos encontrar razones para reafirmar la trascendencia que tiene la seguridad en los sistemas computarizados, enseguida nos dedicamos a dar una introducción de cómo podemos atacar este problema.

El diseñar una estrategia de seguridad depende en general mucho de la actividad que se este desarrollando, sin embargo se pueden considerar los siguientes tres pasos generales: el **primero** crear una política global de seguridad, el **segundo** realizar un análisis de riesgos y el **tercero** aplicar las medidas correspondientes.

Política global de seguridad: se debe de establecer el estatus de la información para la empresa o la organización, debe de contener un objetivo general, la importancia de la tecnología de la información para la empresa, el periodo de tiempo de validez de la política, los recursos con que se cuenta, objetivos específicos de la empresa.

Debe de establecerse la calidad de la información que se maneja según su objetivo, la calidad que debe tener la información quiere decir que se establezca cuando o para quien la información debe ser confidencial, cuando debe verificarse su integridad y cuando debe de verificarse su autenticidad tanto de la información como de los usuarios.

Análisis de riesgos: consiste en enumerar todo tipo de riesgos a los cuales esta expuesta la información y cuales son las consecuencias, los posibles atacantes entre persona empresas y dependencias de inteligencia, las posibles amenazas etc., enumerar todo tipo de posible perdida desde perdidas directas como dinero, clientes, tiempo etc., así como indirectas: créditos, perdida de imagen, implicación en un litigio, perdida de imagen, perdida de confianza etcétera.

El riesgo se puede calcular por la formula $\text{riesgo} = \text{probabilidad} \times \text{perdida}$, por ejemplo el riesgo de perder un contrato por robo de información confidencial es igual a la probabilidad de que ocurra el robo multiplicado por la perdida total en pesos de no hacer el contrato. El riesgo de fraude en transacciones financieras es igual a la probabilidad de que ocurra el fraude por la perdida en pesos de que llegara ocurrir ese fraude. Si la probabilidad es muy pequeña el riesgo es menor, pero si la probabilidad es casi uno, el riesgo puede ser casi igual a la perdida total. Si por otro lado la perdida es menor aunque la probabilidad de que ocurra el evento sea muy grande tenemos un riesgo menor. Por ejemplo la perdida de una transacción de 300 pesos con una probabilidad muy grande de que ocurra al usar criptografía débil, el riesgo llega a ser menor.

En el análisis de riesgo debe también incluirse los posibles ataques que puedan existir y su posible efectos.

Medidas de seguridad: esta parte la podemos plantear como la terminación de la toda la estructura de seguridad de la información. Una vez planteada una política de seguridad, decir cuanto vale la información, un análisis de riesgo, decir que tanto pierdo si le ocurre algo a mi información o que tanto se gana si se protege, debemos de establecer las medidas para que cumpliendo con la política de seguridad, las perdidas

sean las menores posibles y que esto se transforme en ganancias ya sean materiales o de imagen.

Las posibles medidas que se pueden establecer se pueden dividir según la siguiente tabla:

tipos	Protección Física	Medidas Técnicas	Medidas de Organización
Preventivas	PF	PT	PO
Detectivas	DF	DT	DO
Correctiva	CF	CT	CO

PF: guardias a la entrada del edificio, control en el acceso de entrada, protección al hardware, respaldo de datos, ...

DF: monitor de vigilancia, detector de metales, detector de movimiento, ...

CF: respaldo de fuente de poder, ...

PT: firewalls, criptografía, bitácora, ...

DT: control de acceso lógico, sesión de autenticación, ...

CT: programa antivirus, ...

PO: cursos de actualización, organización de las claves, ...

DO: monitoreo de auditoria, ...

CO: respaldos automáticos, plan de incidentes (sanciones), ...

En resumen debemos de mencionar que no existe un sistema computarizado que garantice al 100% la seguridad de la información debido a la inmensa mayoría de formas diferentes con que se puede romper la seguridad de un sistema. Sin embargo una buena planeación de la estrategia para dar seguridad a la información puede resultar desde la salvación de una empresa hasta la obtención de grandes ganancias directas en pesos, o como ganancias indirectas mejorando la imagen y la seguridad de la empresa. Uno de los objetivos principales de establecer una política de seguridad es de reducir al mínimo los riesgos posibles, implementando adecuadamente las diferentes medidas de seguridad.

Enseguida repasamos algunas de las técnicas de seguridad que pertenecen a la criptografía, se pretende exponer de una forma simple algunas de las partes mas conocidas de este amplio campo.

Introducción

La palabra criptografía proviene del griego *kryptos*, que significa esconder y *gráphein*, escribir, es decir, escritura escondida. La criptografía ha sido usada a través de los años para mandar mensajes confidenciales cuyo propósito es que sólo las personas autorizadas lo puedan entender el mensaje.

Alguien que quiere mandar información confidencial aplica técnicas criptográficas para poder “esconder” el mensaje (lo llamaremos cifrar o encriptar), manda el mensaje por una línea de comunicación que se supone insegura y después solo el receptor autorizado pueda leer el mensaje “escondido” (lo llamamos descifrar o desencriptar)

CIFRAR



ENVIAR



DESCIFRAR



Desde sus inicios, la criptografía llegó a ser una herramienta muy usada en el ambiente militar, en la segunda gran guerra tuvo un papel determinante, una de las máquinas de cifrado y que tubo gran popularidad se llamó **ENIGMA**. Al terminar la guerra las agencias de seguridad de las grandes potencias invirtieron muchos recursos para su investigación. La criptografía como la conocemos hoy surgió con la invención de la computadora.

La Criptografía actual se inicia en la segunda mitad de la década de los años 70. No es hasta la invención del sistema conocido como **DES (Data Encryption Standard)** en 1976 que se da a conocer mas ampliamente, principalmente en el mundo industrial y comercial. Posteriormente con el sistema **RSA (Rivest, Shamir, Adleman)** en 1978, se abre el comienzo de la criptografía en un gran rango de aplicaciones: en transmisiones militares, en transacciones financieras, en comunicación de satélite, en redes de computadoras, en líneas telefónicas, en transmisiones de televisión etcétera.

La criptografía se divide en dos grandes ramas, la criptografía de clave privada o simétrica y la criptografía de clave pública o asimétrica, **DES** pertenece al primer grupo y **RSA** al segundo.

Para poder entender un poco de la criptografía, es tiempo de plantear que tipo de problemas resuelve ésta. Los principales problemas de seguridad que resuelve la criptografía son: la privacidad, la integridad, la autenticación y el no rechazo.

La privacidad, se refiere a que la información sólo pueda ser leída por personas autorizadas.

Ejemplos: Si la comunicación se establece por teléfono y alguien intercepta la comunicación o escucha la conversación por otra línea podemos afirmar que no existe privacidad. Si mandamos una carta y por alguna razón alguien rompe el sobre para leer la carta, podemos decir que se ha violado la privacidad. En la comunicación por Internet es muy difícil estar seguros que la comunicación es privada, ya que no se tiene control de la línea de comunicación. Por lo tanto si ciframos (escondemos) la información cualquier interceptación no autorizada no podrá entender la información confidencial. Esto es posible si se usan técnicas criptográficas, en particular la privacidad se logra si se cifra el mensaje con un método simétrico.

La integridad, se refiere a que la información no pueda ser alterada en el transcurso de ser enviada.

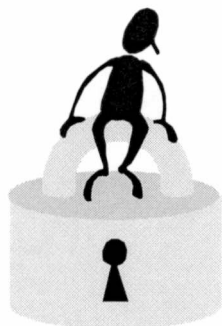
Ejemplos: Cuando compramos un boleto de avión es muy prudente verificar que los datos son los correctos antes de terminar la operación, en un proceso común esto se puede realizar al mismo tiempo de la compra, por internet como la compra se puede hacer desde dos ciudades muy distantes y la información tiene necesariamente que viajar por una línea de transmisión de la cual no se tiene control. Es muy importante estar seguros que la información transmitida no ha sido modificada (en tal caso se dice que hay integridad). Esto también se puede solucionar con técnicas criptográficas particularmente con procesos simétricos o asimétricos. La integridad es muy importante por ejemplo en las transmisiones militares ya que un cambio de información puede causar graves problemas.

La autenticidad, se refiere a que se pueda confirmar que el mensaje recibido haya sido mandado por quien dice lo mando o que el mensaje recibido es el que se esperaba.

Ejemplos: las técnicas necesarias para poder verificar la autenticidad tanto de personas como de mensajes usando quizá la más conocida aplicación de la criptografía asimétrica que es la firma digital, y de algún modo reemplaza a la firma autógrafa que se usa comúnmente, para autenticar mensajes se usa criptografía asimétrica.

Por Internet es muy fácil engañar a una persona con quien se tiene comunicación respecto a la identidad, resolver este problema es por lo tanto muy importante para efectuar comunicación confiable.

El no rechazo, se refiere a que no se pueda negar la autoría de un mensaje enviado.



Información Segura Autorizada

Cuando se diseña un sistema de seguridad una gran cantidad de problemas pueden ser evitados si se ponen en función de comprobar autenticidad, de garantizar privacidad, de asegurar integridad y evitar el no-rechazo.

La criptografía simétrica y asimétrica conjuntamente con otras técnicas, como el buen manejo de las claves y la legislación adecuada resuelven satisfactoriamente los anteriormente problemas planteados, como lo veremos en los capítulos posteriores.



Persona

Criptografía Simétrica

La criptografía simétrica se refiere al conjunto de métodos que permiten tener comunicación segura entre las partes siempre y cuando anteriormente se hayan intercambiado la clave correspondiente que llamaremos clave simétrica. La simetría se refiere a que las partes tienen la misma llave tanto para cifrar como para descifrar.



Este tipo de criptografía es conocida también como criptografía de clave privada o criptografía de llave privada.

Existe una clasificación de este tipo de criptografía en tres familias, la criptografía simétrica de bloques (block cipher), la criptografía simétrica de lluvia (stream cipher) y la criptografía simétrica de resumen (hash functions). Aunque con ligeras modificaciones un sistema de criptografía simétrica de bloques puede modificarse para convertirse en alguna de las otras dos formas, e inversamente, sin embargo es importante verlas por separado dado que se usan en diferentes aplicaciones.

La criptografía simétrica ha sido la más usada en toda la historia, ésta a podido ser implementada en diferentes dispositivos, manuales, mecánicos, eléctricos, hasta los algoritmos actuales que son programables en cualquier computadora. La idea general es aplicar diferentes funciones al mensaje que se quiere cifrar de tal modo que solo conociendo una clave pueda aplicarse de forma inversa para poder así descifrar.

Aunque no existe un tipo de diseño estándar, quizá el más popular es el de Feistel, que consiste esencialmente en aplicar un número finito de interacciones de cierta forma, que finalmente da como resultado el mensaje cifrado. Este es el caso del sistema criptográfico simétrico más conocido, **DES**.

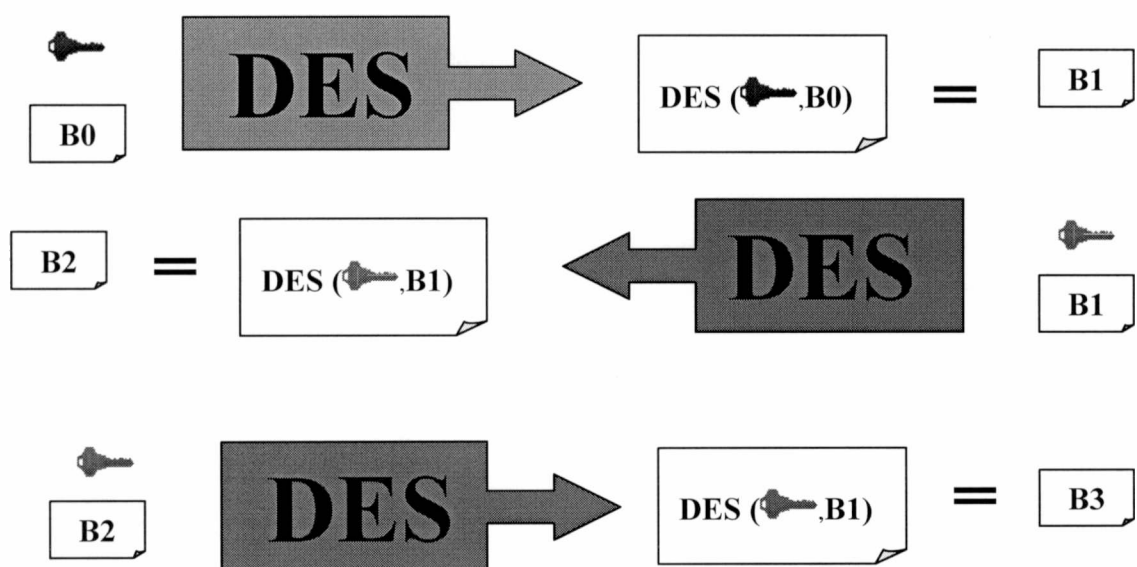
DES es un sistema criptográfico que toma como entrada un bloque de 64 bits del mensaje y este se somete a 16 interacciones, con una clave de 56 bits. Este sistema fue tomado como estándar y ha sido uno de los más conocidos, usados y estudiados.

DES opera con una llave de longitud de 56 bits, en la práctica el bloque de la clave tiene 64 bits, ya que a cada conjunto de 7 bits se le agrega un bit que puede ser usada como de paridad, pero en si la clave solo tiene 56 bits de longitud. Dependiendo de la naturaleza de la aplicación DES tiene 4 modos de operación para poder implementarse: **ECB** (**E**lectronic **C**odebook **M**ode) para mensajes cortos, de menos de 64 bits, **CBC** (**C**ipher **B**lock **C**haining **M**ode) para mensajes largos, **CFB** (**C**ipher **B**lock **F**eedback) para cifrar bit por bit ó byte por byte y el **OFB** (**O**utput **F**eedback **M**ode) el mismo uso pero evitando propagación de error.

En la actualidad no se ha podido romper el sistema **DES** desde la perspectiva de poder deducir la clave simétrica a partir de la información interceptada, sin embargo con un método a fuerza bruta, es decir, probando todas las 2^{56} posibles claves se ha podido romper **DES** en Enero de 1999. Lo anterior quiere decir que, es posible verificar todas las claves posibles en el sistema **DES** en un tiempo corto, lo que lo hace inseguro para propósitos de alta seguridad. La opción que se ha tomado para poder suplantar a **DES** ha sido usar lo que se conoce como cifrado múltiple, es decir aplicar varias veces el mismo algoritmo para fortalecer la longitud de la clave, esto a tomado la forma de un nuevo sistema de cifrado que se conoce actualmente como **triple-DES** o **TDES**.

TDES El funcionamiento de **TDES** consiste en aplicar 3 veces **DES** de la siguiente manera: la primera vez se usa una clave **K1**(azul) junto con el bloque **B0**, de forma ordinaria **E** (de Encryption), obteniendo el bloque **B1**. La segunda ves se toma a **B1** con la clave **K2** (roja), diferente a **K1** de forma inversa, llamada **D** (de Desencryption) y la tercera vez a **B2** con una clave **K3** (verde) diferente a **K1** y **K2**, de forma ordinaria **E** (de Encryption), es decir, aplica de la interacción 1 a la 16 a **B0** con la clave **K1**, después aplica de la 16 a la 1, a **B1** con la clave **K2**, finalmente aplica una vez mas de la 1 a la 16 a **B3** usando la clave **K3**, obteniendo finalmente a **B3**. En cada una de estas tres veces aplica el modo de operación más adecuado.

El proceso del cifrado con **TDES** se puede apreciar en las siguientes figuras:



Este sistema **TDES** usa entonces una clave de 168 bits, aunque se ha podido mostrar que los ataques actualmente pueden romper a **TDES** con una complejidad de 2^{112} , es decir efectuar al menos 2^{112} operaciones para obtener la clave a fuerza bruta, además de la memoria requerida.

Se optó por **TDES** ya que es muy fácil interoperar con **DES** y proporciona seguridad a mediano plazo.

En los últimos 20 años se han diseñado una gran cantidad de sistemas criptográficos simétricos, entre algunos de ellos están: **RC-5**, **IDEA**, **FEAL**, **LOKI'91**, **DESX**, **Blowfish**, **CAST**, **GOST**, etcétera. Sin embargo no han tenido el alcance de **DES**, a pesar de que algunos de ellos tienen mejores propiedades.

Podemos decir que el estado actual de la criptografía simétrica es la búsqueda de un nuevo sistema que pueda reemplazar a **DES** en la mayor parte de aplicaciones. Es así como se ha optado por convocar a un concurso de sistemas criptográficos simétricos y que este decida quien será el nuevo estándar al menos para los próximos 20 años.

AES El **NIST** (National Institute of Standards Technology) convocó a un concurso para poder tener un sistema simétrico que sea seguro y pueda usarse al menos en los próximos 20 años como estándar. En la mitad del año de 1998 se aceptaron 15 candidatos, estos se han sometido a pruebas públicas y por parte del **NIST**. Actualmente se cuentan con 5 finalistas que son: **MARS**, **RC6**, **Rijndael**, **Serpent**, y **Twofish**, se espera que el candidato elegido se tenga a mediados del año 2000.

Las principales características que se pide a **AES** son que al menos sea tan seguro y rápido como **TDES**, es decir, que al menos evite los ataques conocidos. Además de que pueda ser implementado en una gran parte de aplicaciones. Una vez designado **AES** este podrá ser usado tanto como cifrador de bloques (block cipher), como cifrador de lluvia (stream cipher), como función resumen (hash function), y en el generador de números pseudoaleatorios.

Los cifradores de lluvia o stream ciphers, son usados donde se cuenta con un ancho de banda restringido (el número de bits que se transmiten a la vez), además de que se requiere independencia en los bloques transmitidos, entonces la mejor opción es cifrar bit por bit o byte por byte, este tipo de cifradores tiene la característica además de ser muy rápido. Los algoritmos más conocidos de este tipo están **RC-4**, **SEAL** y **WAKE**.

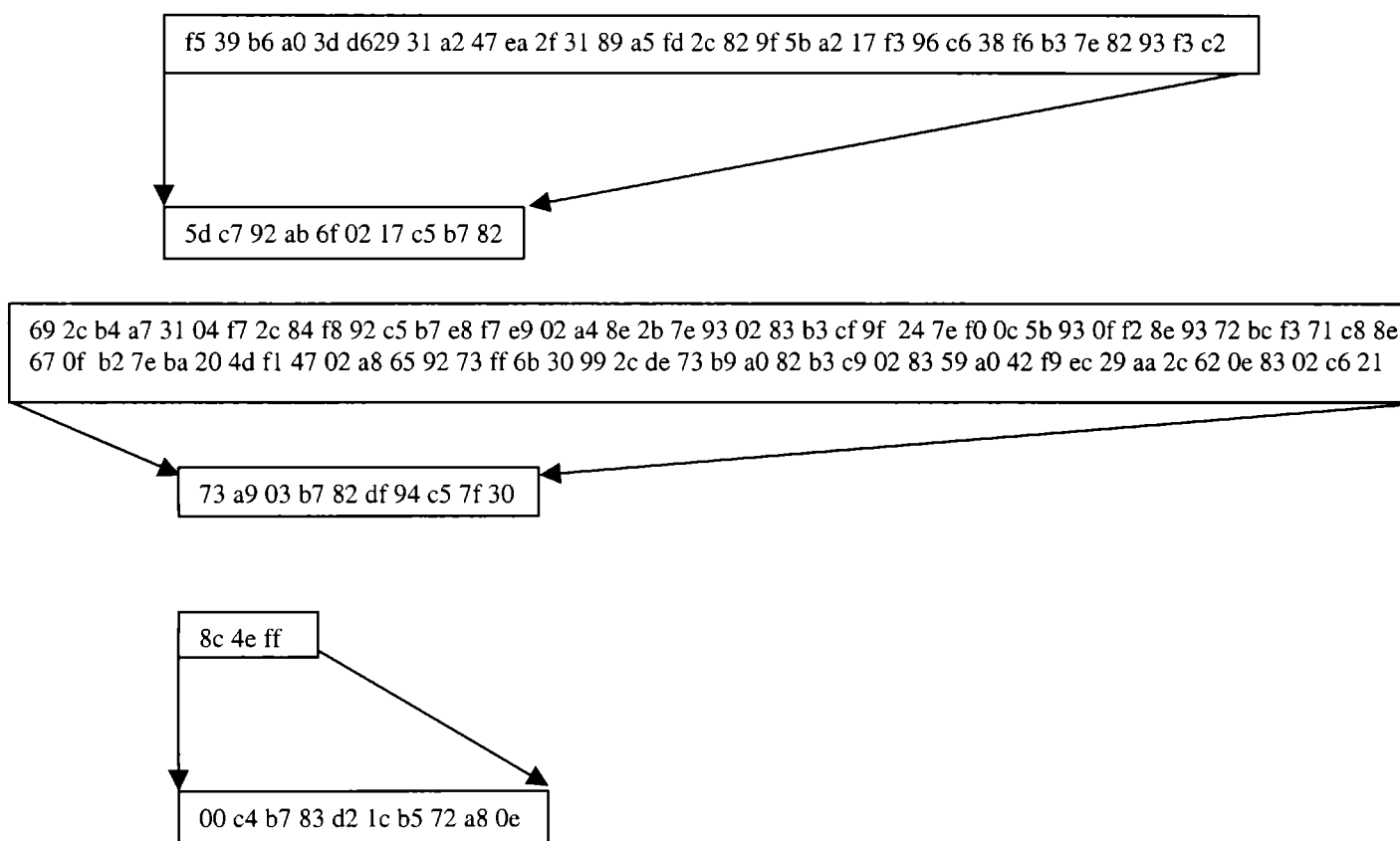
Entre los ataques más potentes a la criptografía simétrica están el criptoanálisis diferencial y lineal, sin embargo no han podido ser muy eficientes en la práctica por lo tanto, por el momento después de que un sistema criptográfico es publicado y se muestra inmune a estos dos tipos de ataques (y algunos otros más) la mayor preocupación es la longitud de las claves.

Funciones Hash

Una herramienta fundamental en la criptografía son las funciones hash, son usadas principalmente para resolver el problema de la integridad de los mensajes, así como la autenticidad de mensajes y de su origen.

Una función hash es también ampliamente usada para la firma digital, ya que los documentos a firmar pueden ser en general demasiado grandes la función hash les asocia una cadena de longitud 160 bits que son mas manejables para el propósito de firma digital.

De forma gráfica la función hash efectúa lo siguiente:



Esto es, un mensaje de longitud arbitraria lo transforma de forma “única” a un mensaje de longitud constante.

¿Cómo hace esto?

La idea general es la siguiente:

La función hash toma como entrada una cadena de longitud arbitraria, digamos 5259 bits,

luego divide este mensaje en pedazos iguales, digamos de 160bits, como en este caso y en general el mensaje original no será un múltiplo de 160, entonces para completar un número entero de pedazos de 160 bits al último se le agrega un relleno, digamos de puros ceros. En nuestro caso en 5259 caben 32 pedazos de 160 bits y sobran 139, entonces se agregarán 21 ceros más.

Entonces el mensaje toma la forma $x=x_1, x_2, x_3, \dots, x_t$ donde cada x_i tiene igual longitud (160bits por ejemplo).

Posteriormente se asocia un valor constante a un vector de inicialización IV , y se efectúan las siguientes interacciones

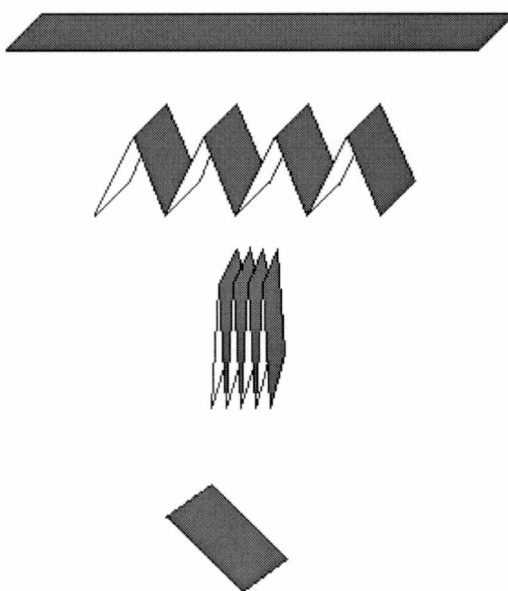
$$H_0 = IV$$

$$H_i = f(H_{i-1}, x_i) \quad 1 \leq i \leq t$$

$$h(x) = g(H_t)$$

donde f es una función que combina a dos cadenas de bits de longitud igual y fija, y g es una función de salida.

De alguna forma lo que se hace es tomar el mensaje partirlo en pedazos de longitud constante y combinar de alguna forma pedazo por pedazo hasta obtener un solo mensaje de longitud fija como muestra la figura siguiente:



Las funciones hash (o primitivas hash) pueden operar como: **MDC** (**M**odification **D**etection **C**odes) ó **MAC** (**M**essage **A**uthentication **C**odes).

Los **MDC** sirven para resolver el problema de la integridad de la información, al mensaje se le aplica un **MDC** (una función hash) y se manda junto con el propio mensaje, al recibirlo el receptor aplica la función hash al mensaje y comprueba que sea igual al hash que se envió antes.

Los **MDCs** son usados principalmente para resolver el problema de la integridad y lo hacen tomando el razonamiento siguiente:

Se aplica un hash $h(\mathbf{M})$ al mensaje \mathbf{M} y se envía con el mensaje, cuando se recibe $(\mathbf{M}, h(\mathbf{m}))$ se le aplica una vez más el hash (que es público a \mathbf{M}) obteniendo $h'(\mathbf{m})$, si $h(\mathbf{M})=h'(\mathbf{M})$, entonces se puede aceptar que el mensaje se transmitió sin alteración.

Los **MAC** sirven para autenticar el origen de los mensajes (junto con la integridad), un **MAC** es un mensaje junto con una clave simétrica que se les aplica un hash y se manda, al llegar la autenticidad del origen del mensaje se demuestra si la clave del receptor corresponde a la que se creó en el origen del mensaje.

Los **MACs** son usados para resolver el problema de autenticar el origen del mensaje y tiene el siguiente argumento:

Se combina el mensaje \mathbf{M} con una clave privada \mathbf{K} y se les aplica un hash $h(\mathbf{M}, \mathbf{K})$, si al llegar a su destino $h(\mathbf{M}, \mathbf{K})$ se comprueba de integridad de la clave privada \mathbf{K} , entonces se demuestra que el origen es solo el que tiene la misma clave \mathbf{K} , probando así la autenticidad del origen del mensaje.

Las propiedades que deben de tener las primitivas hash son:

- 1) **Resistencia a la preimagen:** significa que dada cualquier imagen y , es computacionalmente imposible encontrar un mensaje x , tal que $h(x)=y$. Otra forma como se conoce esta propiedad es que h sea de un solo sentido.
- 2) **Resistencia a una 2° preimagen:** significa que dado x , es computacionalmente imposible encontrar una x' tal que $h(x)=h(x')$. Otra forma de conocer esta propiedad es que h sea resistente a una colisión suave.
- 3) **Resistencia a colisión:** significa que es computacionalmente imposible encontrar dos diferentes mensajes x, x' tal que $h(x)=h(x')$. Esta propiedad también se conoce como resistencia a colisión fuerte.

Para ilustrar la necesidad de estas propiedades veamos los siguientes ejemplos:

Consideremos un esquema de firma digital con apéndice, entonces la firma se aplica a $h(x)$, en este caso h debe ser un **MDC** con resistencia a una 2° preimagen, ya que de lo contrario un atacante \mathbf{C} que conozca la firma sobre $h(x)$, puede encontrar otro mensaje x' tal que $h(x) = h(x')$ y reclamar que la firma es del documento x' .

Si el atacante **C** puede hacer que el usuario firme un mensaje, entonces el atacante puede encontrar una colisión (x, x') (en lugar de lo más difícil que es encontrar una segunda preimagen de x) y hacer firmar al usuario a x diciendo que firmo x' . En este caso es necesaria la propiedad de resistencia a colisión.

Por último si (e,n) es la clave pública **RSA** de **A**, **C** puede elegir aleatoriamente un y y calcular $z = y^e \bmod n$, y reclamar que y es la firma de z , si **C** puede encontrar una preimagen x tal que $z = h(x)$, donde x es importante para **A**. Esto es evitable si h es resistente a preimagen.

Las funciones hash más conocidas son las siguientes: las que se crean a partir de un block cipher como **DES**, **MD5**, **SHA-1**, y **RIPEMD**.

Actualmente se ha podido encontrar debilidades en las funciones hash que tienen como salida una cadena de 128 bits, por lo que se ha recomendado usar salidas de 160bits.

Así mismo se han encontrado ataques a **MD5** y **SHA-0** (antecesora de **SHA-1**), esto ha dado lugar que se dirija la atención sobre la función has **RIPEMD**.

Criptografía Asimétrica

La criptografía asimétrica es por definición aquella que utiliza dos claves diferentes para cada usuario, una para cifrar que se le llama clave pública y otra para descifrar que es la clave privada. El nacimiento de la criptografía asimétrica se dio al estar buscando un modo más práctico de intercambiar las llaves simétricas Diffie y Hellman, proponen una forma para hacer esto, sin embargo no fue hasta que el popular método de Rivest Shamir y Adleman **RSA** publicado en 1978, cuándo toma forma la criptografía asimétrica, su funcionamiento esta basado en la imposibilidad computacional de factorizar números enteros grandes.

Actualmente la Criptografía asimétrica es muy usada, sus dos principales aplicaciones son precisamente el intercambio de claves privadas y la firma digital, una firma digital se puede definir como una cadena de caracteres que se agrega a un archivo digital que hace el mismo papel que la firma convencional que se escribe en un documento de papel ordinario.

En la actualidad la criptografía asimétrica o de clave pública se divide en tres familias, según el problema matemático del cual basan su seguridad. La primera familias la que basa su seguridad en el Problema de Factorización Entera **PFE**, los sistemas que pertenecen a esta familia son, el sistema **RSA**, y el de Rabin Williams **RW**. La segunda familia es la que basa su seguridad en el Problema del Logaritmo Discreto **PLD**, a esta familia pertenece el sistema de Diffie Hellman **DH** de intercambio de claves y el sistema **DSA** de firma digital. La tercera familia es la que basa su seguridad en el Problema del Logaritmo Discreto Elíptico **PLDE**, en este caso hay varios esquemas tanto de intercambio de claves como de firma digital que existen como el **DHE** (Diffie Hellman Elíptico), **DSAE**, (Nyberg-Rueppel) **NRE**, (Menezes, Qu, Vanstone) **MQV**, etcétera.

Aunque a las familias anteriores pertenecen los sistemas asimétricos más conocidos, existen otro tipo de sistemas que basan su seguridad en otro tipo de problema como por ejemplo en el Problema del Logaritmo Discreto Hiperelíptico, sobre problemas de retículas y sobre subconjuntos de clases de campos numéricos reales y complejos.

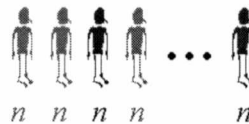
RSA, en el caso de **RSA** el problema matemático es el de la factorización de un número entero n grande (1024 bits), este número entero se sabe es producto de dos números primos p, q de la misma longitud, entonces la clave pública es el número n y la privada es p, q . El razonamiento del funcionamiento de **RSA** es el siguiente:

a) a cada usuario se le asigna un número entero n , que funciona como su clave pública

- b) solo el usuario respectivo conoce la factorización de n (o sea p, q), que mantiene en secreto y es la clave privada



- c) existe un directorio de claves públicas



- d) si alguien quiere mandar un mensaje m a algún usuario entonces elige su clave pública n y con información adicional también pública puede mandar el mensaje cifrado c , que solo podrá descifrar el usuario correspondiente, el mensaje m convertido a número (codificación) se somete a la siguiente operación.

$$c = m^e \bmod n$$

- e) Entonces el mensaje c puede viajar sin problema por cualquier canal inseguro

```
68a9bc498ff034e0572fd5d267193f2a
e12b7fa8d735cd927a7166bc3f4e5b82
a6bc937ade8ba4073e25ca9e7f48b26f
```

- f) cuando la información cifrada llega a su destino el receptor procede a descifrar el mensaje con la siguiente fórmula

$$m = c^d \bmod n$$

- g) Se puede mostrar que estas formulas son inversas y por lo tanto dan el resultado deseado, (m, e) son públicos y se pueden considerar como la clave pública, la clave privada es la pareja (p, q) o equivalentemente el número d . La relación que existe

entre d y e es que uno es el inverso multiplicativo del otro módulo $\lambda(n)$ donde $\lambda(n)$ es el mínimo común múltiplo de $p-1$ y $q-1$, esto significa que la clave privada o el la pareja p, q o es el número d .

En términos muy generales es así como funciona el sistema **RSA**. Sin embargo en la realidad existen dos formas que son las más comunes, estas formas depende de la aplicación y se llaman el esquema de firma y el esquema de cifrado, cada una de estas dos diferentes aplicaciones consiste en una serie de pasos que a continuación se describen

Esquema de cifrado

Uso: este esquema se usa principalmente en cifrar claves de sistemas simétricos (claves de 128 bits aprox.)

- 1) Se toma el mensaje **M** (por ejemplo una clave simétrica de 128 bits), como en la practica actual es recomendable usar arreglos de longitud de 1024 bits, los complementa esos 128 bits con una serie de técnicas para obtener un arreglo de 1024 bits, que la computadora entiende como un número entero m , este proceso se llama codificación.
- 2) Se le aplica la formula de cifrado de **RSA** al entero m
- 3) Se envía el número entero c
- 4) Al recibir este número se aplica la formula de descifrado al entero c para obtener el entero m
- 5) Se decodifica m para obtener el mensaje **M**

Esquema de Firma Digital

Existen dos tipos de esquemas sobre firma digital, el que se denomina esquema de firma digital con apéndice y el esquema de firma digital con mensaje recuperable. También cualquier esquema de firma cuenta con dos partes la primera parte se denomina proceso de firma (similar al cifrado) y la segunda parte proceso de verificación de la firma (similar al descifrado).

El esquema más usado y conocido es el esquema de firma con apéndice y consiste en los siguientes puntos:

Proceso de Firma

- 1) El mensaje a firmar es M , se le aplica una función hash que reduce su longitud de forma única a un mensaje $H(M)$ de longitud de 128 o 160 bits, lo que permite ver cualquier mensaje de cualquier longitud como una cadena de caracteres de longitud constante.
- 2) $H(m)$ se somete también a un proceso de codificación, por lo tanto se obtiene un número $h(m)$, al que se le aplica la fórmula con la potencia d , equivalentemente con la clave privada del firmante para obtener

$$s = h(m)^d \bmod n$$

- 3) Se envía entonces el mensaje firmado s

Proceso de Verificación

- 1) El que recibe s , se supone conoce el mensaje m , aplica la función para obtener con la clave pública del que dice ser

$$h'(m) = s^e \bmod n$$

- 2) Aplica la función hash al mensaje m y si $h(m) = h'(m)$ entonces acepta la firma

En un esquema con mensaje recuperable no es necesario saber el mensaje, después de que la firma es aceptada el mensaje puede recuperarse a partir de la firma.

Aspectos Importantes

1) La longitud de las claves

Existe una gran discusión, sobre este aspecto pero sin duda en la actualidad se acepta que es recomendable usar claves de longitud 768 para actividades personales, 1024 bits para corporaciones y 2048 para actividades de alto riesgo. La longitud de las claves tiene que ver con la seguridad del sistema si el número n pudiese ser factorizado entonces sin mucha dificultad puede calcular a d a partir de e , p , y q por lo tanto descifrar cualquier mensaje. El último récord conocido sobre factorización de números enteros producto de dos primos es de 155 (512 bits) dígitos alcanzado en Jul de 1999.

2) La aleatoriedad de las claves

La generación de las claves **RSA** es muy importante, muchos ataques son evitados si las claves son elegidas de forma aleatoria, esto incrementara la seguridad del sistema.

3) método de codificación

El método que actualmente es usado para aplicaciones en el esquema de cifrado es el **OAEP**, este resiste a los ataques que actualmente se conocen y el estándar más conocido sobre **RSA** es el **PKCS#1 v.2** de la RSA Data Security.

En el caso de Esquemas de firma digital el método de codificación recomendable es **PSS**, que esta descrito en **PKCS#1 v 2.1**

4) Elección de parámetros

La elección adecuada de los parámetros que se usan aumenta la seguridad del sistema asi como su fácil y rápida implementación. Como elegir a $e=65537$, que es el número 4 de Fermat. Esto implica que d , la clave privada sea de una longitud considerable, evitando el ataque de Wiener. Por otro lado usar el método de descifrado por el teorema chino del residuo aumenta la rapidez de descifrado.

CCE otro tipo de criptografía de clave pública es el que usa curvas elípticas definidas en un campo finito. La diferencia que existe entre este sistema y **RSA** es el problema del cual basan su seguridad, mientras **RSA** razona de la siguiente manera: te doy el número 15 y te reta a encontrar los factores primos. El problema del cual están basados los sistemas que usan curvas elípticas que denotaremos como **CCE** es el problema del logaritmo discreto elíptico, en este caso su razonamiento con números sería algo como: te doy el número 15 y el 3 y te reta a encontrar cuantas veces tienes que sumar el mismo 3 para obtener 15.

En lo que sigue nos dedicaremos a explicar un poco mas lo más importante de los **CCE**

- 1) entenderemos como una curva elíptica a un conjunto finito de puntos P, Q, \dots, S donde cada punto en una pareja $P = (x, y)$ y las coordenadas x, y satisfacen una ecuación de la siguiente forma:

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

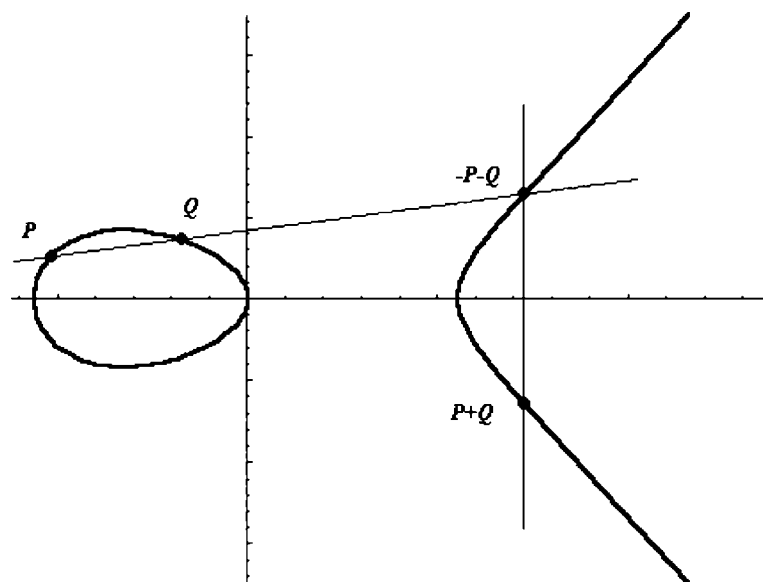
Donde las constantes a, b, c, d y e pertenecen a cierto conjunto llamado campo **F**, que para propósitos de la criptografía o es un campo primo (\mathbb{Z}_p) o un campo de característica 2, o sea donde los elementos son n-adas de ceros y unos (\mathbb{F}_2^n)

- 2) El conjunto de puntos que satisfacen a una ecuación similar a la de 1) lo podemos representar como

$$E : O, P_1, P_2, P_3, \dots, P_n$$

Este conjunto de puntos puede sumarse y tiene las mismas propiedades que la suma de los números enteros, es decir lo que se conoce como un grupo abeliano, hay que hacer notar que en este caso el que hace el papel de cero (identidad aditiva) es un punto especial que no tiene coordenadas y se representa como O llamado punto al infinito

- 3) La suma de estos puntos tiene una explicación geométrica muy simple, en este caso la gráfica representa a todos los puntos que satisfacen la ecuación de 1), si suponemos que queremos sumar a P y Q , trazamos una línea recta que pase por P y Q , la ecuación de 1) es de grado 3 y la línea de grado 1, entonces existe siempre tres soluciones, en este caso la tercera solución esta dibujada como el punto $-P-Q$, enseguida se procede a dibujar una línea recta paralela al eje Y que pase por $-P-Q$, esta línea vertical también intercepta tres veces a la recta, todas las líneas verticales interceptan al punto especial llamado infinito y que geoméricamente esta en el horizonte del plano, el tercer punto es por definición $P+Q$, como se muestra en la figura



- 4) La anterior forma de sumar puntos de una curva elíptica es un poco extraña sin embargo, es esta extrañeza lo que permite que sea un poco más difícil romper los CCE. En el área de las matemáticas conocida como teoría de grupos se sabe que estos grupos son muy simples llamados grupo finitos abelianos lo que permite también que los CCE sean fáciles de implementar, llamaremos al número de puntos racionales de la curva como el orden de la curva
- 5) Los CCE basan su seguridad en el Problema del Logaritmo Discreto Elíptico (PLDE), esto quiere decir que dados P, Q puntos de la curva hay que encontrar un número entero x tal que $xP = Q$ ($xP = P+P+\dots+P$, x veces). Obsérvese que a diferencia del PFE (Problema de Factorización Entera) el PLDE no maneja completamente números, lo que hace más complicada su solución.
- 6) La creación de un protocolo con criptografía de curvas elípticas requiere fundamentalmente una alta seguridad y una buena implementación, para el primer punto se requiere que la elección de la curva sea adecuada, principalmente que sea

no-supersingular y que el orden del grupo de puntos racionales tenga un factor primo de al menos 160 bits, además de que este orden no divida al orden de un número adecuado de extensiones del campo finito, para que no pueda ser sumergido en él, si el campo es Z_p , se pide que la curva no sea anómala. Todo esto con el fin de evitar los ataques conocidos.

Para el caso de la implementación hay que contar con buenos programas que realicen la aritmética del campo finito, además de buenos algoritmos que sumen puntos racionales, si el campo es Z_p existen varios y si el campo es F_{2^n} entonces se toma una base polinomial que tenga el mínimo de términos por ejemplo un trinomio para generar los elementos del campo finito esto si la implementación es en software y se toma una base normal si es en hardware. Además de contemplar que las operaciones de puntos racionales pueden hacerse en el espacio proyectivo esto elimina el hacer divisiones ahorrando tiempo.

- 7) Lo anterior se ve reflejado en las ventajas que ofrecen los **CCE** en comparación con **RSA**, la principal es la longitud de la clave secreta. Se puede mostrar que mientras en **RSA** se tiene que usar una clave de 1024 para ofrecer una considerable seguridad, los **CCE** solo usan 163 bits para ofrecer la misma seguridad, así también las claves **RSA** de 2048 son equivalentes en seguridad a 210 de **CCE**. Esto se debe a que para resolver el **PLDE** el único algoritmo conocido toma tiempo de ejecución totalmente exponencial, mientras que el algoritmo que resuelve **PFE** incluso también el **PLD** en Z_p toman tiempo subexponencial.
- 8) Otra buena noticia sobre los **CCE** es que los elementos de los puntos racionales pueden ser elementos de un campo finito de característica 2, es decir pueden ser arreglos de ceros y unos de longitud finita (01001101110010010111), en este caso es posible construir una aritmética que optimice la rapidez y construir un circuito especial para esa aritmética, a esto se le conoce como Base Normal Optima.
- 9) Lo anterior permite con mucho que los **CCE** sean idóneos para ser implementados en donde el poder de computo y el espacio del circuito sea reducido, donde sea requerida una alta velocidad de procesamiento o grandes volúmenes de transacciones, donde el espacio de almacenamiento, la memoria o el ancho de banda sea limitado. Lo que permite su uso en Smart Cards, Teléfonos celulares, Fax, Organizadores de Palma, PCs, etcétera.
- 10) En la actualidad existen varios estándares que permiten el uso adecuado y óptimo de los **CCE**, entre los cuales se encuentran: **IEEE P1363** (Institute of Electrical and Electronics Engineers), el **ANSI X9.62**, **ANSI X9.63**, **ANSI TG-17**, **ANSI X12** (American National Standards Institute), **UN/EDIFACT**, **ISO/IEC 14888**, **ISO/IEC 9796-4**, **ISO/IEC 14946** (International Standards Organization), **ATM Forum** (Asynchronous Transport Mode), **WAP** (Wireless Application Protocol). En comercio electrónico: **FSTC** (Financial Services Technology Consortium), **OTP 0.9** (Open Trading Protocol), **SET** (Secure Electronic Transactions). En internet **IETF** (The Internet Engineering Task Force), **IPSec** (Internet Protocol Security Protocol)
- 11) Los **CCE** están reemplazando a las aplicaciones que tienen implementado **RSA**, estas definen también esquemas de firma digital, Intercambio de claves simétricas y otros.

Otras Herramientas criptográficas

En esta sección me dedicare principalmente a enumerar otro tipo de herramientas o técnicas que son usadas en criptografía, cada una de ellas tiene una gran aplicación y tienen un propósito muy específico dentro del ámbito de la criptografía.

A) Compartición de Secretos

La compartición de secretos, como su nombre lo dice es una técnica criptográfica que se dedica a partir un secreto, que puede ser una clave secreta, en la responsabilidad de varias personas y que solo con el número mínimo de personas se podrá reconstruir el secreto compartido. Por ejemplo si el secreto es el número 100 y este debe ser compartido por tres personas A1, A2 y A3 una forma de poder hacerlo es generar un número aleatorio menor a 100, digamos el 33 posteriormente se genera otro número aleatorio menor a $100-33$, digamos el 27, y finalmente la tercera parte será $100-(27+33)=46$. Así el secreto 100 esta compartido por A1(33), A2(27) y A3(46) cada quien con su parte correspondiente. Como ninguno de ellos sabe las otras partes, solo los tres juntos podrán reconstruir el mensaje sumando sus partes. Claro esta este es solo un ejemplo para explicar el concepto.

La comparación de secretos puede ser usada para compartir digamos la combinación de una caja fuerte, la clave de lanzamiento de algún proyectil, la clave secreta de una autoridad certificadora, la clave de activación de algún dispositivo de alto riesgo, etc.,

Uno de los mejores métodos de comparación de secretos y mas conocido es el esquema (n,k) límite de Shamir. Este método consiste en partir una clave K en n partes, y se tiene como mínimo (límite) el número k de partes para reconstruir la clave, es decir cualquiera k de los n custodios pueden reconstruir la clave K , pero ningún subgrupo de $k-1$ custodios podrá hacerlo.

Para ampliar este tema:

A. Shamir, How to share a secret, Communications of the ACM V. 22 1979, 612-613
<http://www.cacr.math.uwaterloo.ca/~dstinson/ssbib.html>

B) Criptografía Visual

Una idea ingeniosa de usar un método de comparación de secretos con un esquema límite (n,k) es la criptografía visual, esto consiste en lo siguiente: una imagen es partida en n partes, y si se sobreponen al menos k de estas partes se puede reconstruir la imagen. Veamos en ejemplo de un esquema $(2,2)$, esto trabaja considerando que si la imagen es de blanco y negro, entonces la imagen podrá ser un conjunto de cuadros completamente blancos y completamente negros, por ejemplo la siguiente imagen



Ahora cada cuadro de la imagen podrá ser considerado como blanco o negro, equivalentemente con valores 0 y 1. Para partir esta imagen en dos partes $n=2$ y considerando el límite con $k=2$, se procede como sigue:

Cada cuadro que es completamente negro podrá ser partido en dos partes de la siguiente forma:

$$\begin{array}{c}
 \blacksquare \quad \blacksquare \quad \blacksquare \quad \text{ó} \quad \blacksquare \quad \blacksquare \quad \blacksquare \\
 \mathbf{11} = \mathbf{10} + \mathbf{01} \qquad \qquad \qquad \mathbf{11} = \mathbf{01} + \mathbf{10}
 \end{array}$$

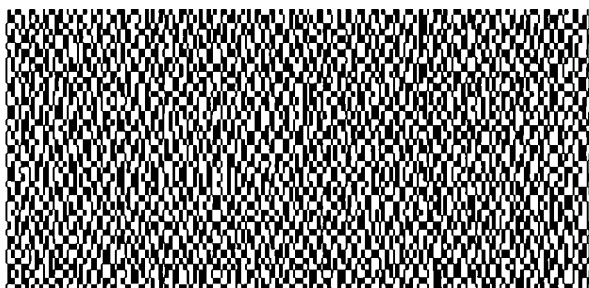
Y un cuadro completamente blanco podrá ser partido en dos de la forma siguiente:

$$\begin{array}{c}
 \square \quad \square \quad \square \quad \text{ó} \quad \square \quad \square \quad \square \\
 \mathbf{00} = \mathbf{10} + \mathbf{10} \qquad \qquad \qquad \mathbf{00} = \mathbf{01} + \mathbf{01}
 \end{array}$$

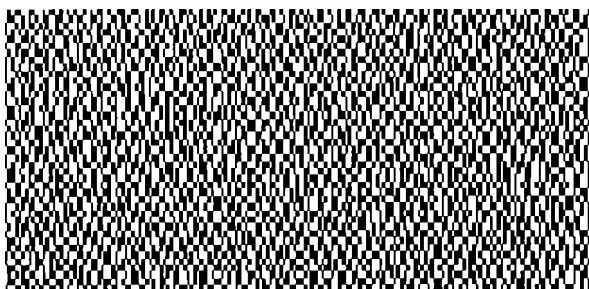
Que significa suma módulo 2, es decir $1+0=1$, $0+1=1$, $0+0=0$ pero también $1+1=0$, de este modo se pueden tomar cualquiera de las dos particiones de los cuadros de color blanco.

Para formar las dos partes de la figura en un acetato se elige aleatoriamente una de las combinaciones anteriores según se parta un cuadro blanco o uno negro

En el caso de nuestra figura una vez elegidas las partes, la figura partida en un esquema limite (2,2) queda así:



Parte 1



Parte 2

De esta forma se tiene partida la figura en dos partes y se recuperara solo sobreponiendo una sobre la otra.

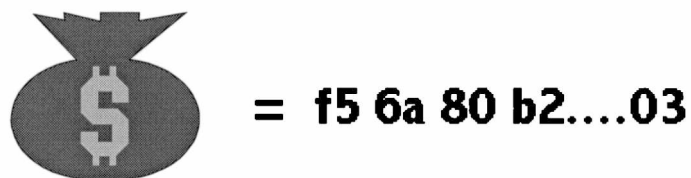
Al sobreponer las dos partes se recupera la figura, de la siguiente forma:



En el caso general se parte los cuadros blancos y negros en n pedazos y hasta no tener k pedazos negros el cuadro reconstruido será siendo blanco, a partir de k pedazos negros hasta n el cuadro reconstruido será negro. En nuestro caso, un cuadro con solo la mitad negra será considerado blanco, es necesario que tenga dos mitades negras para que el cuadro reconstruido se considere negro, que es el caso del esquema (2,2).

C) Dinero Electrónico

Una aplicación más, que puede ser realidad gracias a la criptografía de clave pública es conocida como dinero electrónico, en términos sencillos el dinero electrónico es otra representación de lo que conocemos como dinero o valor, por ejemplo tenemos dinero en billetes emitidos por algún país, podemos tener cheques pagaderos en un banco, bonos, pagares pagaderos en algún plazo, en fin. El dinero electrónico es físicamente un número que se genera aleatoriamente se le asigna un valor, se cifra y firma y se envía al banco, ahí el banco valida el número y certifica el valor, y lo regresa al usuario firmado por el banco, entonces el usuario puede efectuar alguna transacción con ese billete electrónico.



Las principales propiedades del dinero electrónico son las siguientes:

- 1) **Independencia:** la seguridad del dinero digital no debe depender de la el lugar físico donde se encuentre, por ejemplo en el disco duro de una PC
- 2) **Seguridad:** el dinero digital (el número) no debe de ser usado en dos diferentes transacciones
- 3) **Privacidad:** el dinero electrónico debe de proteger la privacidad de su usuario, de esta forma cuando se haga una transacción debe de poder cambiarse el número a otro usuario sin que el banco sepa que dueños tuvo antes.
- 4) **Pagos fuera de línea:** el dinero electrónico no debe de depender de la conexión de la red, así un usuario puede transferir dinero electrónico que tenga en una “smart card” a una computadora, el dinero digital debe ser independiente al medio de transporte que use.

- 5) **Transferibilidad:** el dinero electrónico debe de ser transferible, cuando un usuario transfiere dinero electrónico a otro usuario debe de borrarse la identidad del primero.
- 6) **Divisibilidad:** el dinero electrónico debe de poder dividirse en valores fraccionarios según sea el uso que se da, por ejemplo en valor de 100, 50 y 25

La serie de pasos que puede seguir una transacción que se realiza con dinero electrónico en un escenario simple es la siguiente:

Supóngase que el usuario **A** quiere mandar un cheque a **B**, usando ahora dinero electrónico.

- 1) **A** genera un número aleatorio grande **N** de digamos 100 dígitos y le da un valor digamos 1000 pesos
- 2) **A** cifra este número junto a su valor con su clave secreta asimétrica.
- 3) **A** firma este número y lo transmite a su banco.
- 4) El banco de **A** usa, la clave pública de **A** para descifrar el número y verificar la firma, así recibe la orden y sabe que es de **A**. El banco borra la firma de **A** del documento electrónico.
- 5) El banco revisa que **A** tenga en sus cuentas la cantidad pedida 1000 pesos y la debita de alguna de estas cuentas.
- 6) El banco firma el número que mando **A**, con el valor asignado de 1000 pesos
- 7) El banco regresa el número que ya es dinero a, **A**
- 8) **A** envía este dinero a **B**
- 9) **B** verifica la firma del banco de **A**, que esta en **N**
- 10) **B** envía **N** a su banco
- 11) El banco de **B** re-verifica la firma del banco de **A** en **N**
- 12) El banco de **B** verifica que **N** no este en la lista de números “ya usados”
- 13) El banco de **B** acredita la cantidad de 1000 pesos a la cuenta de **B**
- 14) El banco de **B** pone a **N** en la lista de números “ya usados”
- 15) Finalmente el banco de **B** envía un recibo firmado donde establece que tiene 1000 pesos más en su cuenta

En el mundo comercial existen varias empresas privadas que proveen el servicio de dinero electrónico en diferentes modalidades entre ellas están: CheckFree, CyberCash, DigiCash, First Virtual, Open Market, NetBill y Netscape.

En <http://www.ecashtechologies.com/> pueden encontrarse algunos ejemplos interactivos de cómo trabaja el dinero electrónico en la práctica

Certificados digitales

Los certificados digitales, tienen una similitud con las licencias de conducir, las primeras permiten viajar por las carreteras, los certificados digitales permiten navegar por la red Internet, la principal característica es que da identidad al usuario y puede navegar con seguridad. De igual forma que la sola licencia de conducir o un pasaporte sirve para dar identidad a quien la porta en ciertos casos, el certificado digital da identidad a una clave pública y se comparte como una persona en el espacio cibernético.

El nacimiento del certificado digital fue a raíz de resolver el problema de administrar las claves públicas y que la identidad del dueño pudiera ser falsa. La idea es que una tercera entidad intervenga en la administración de las claves públicas y asegure que las claves públicas tengan asociado un usuario claramente identificado.

Las tres partes más importantes de un certificado digital son:

- 1) Una clave pública
- 2) La identidad del implicado: nombre y datos generales,
- 3) La firma privada de una tercera entidad llamada autoridad certificadora que todos reconocen como tal y que válida la asociación de la clave pública en cuestión con el tipo que dice ser.

En la actualidad casi todas las aplicaciones de comercio electrónico y transacciones seguras requieren un certificado digital, se ah propagado tanto su uso que se tiene ya un formato estándar de certificado digital, este es conocido como X509 v. 3.

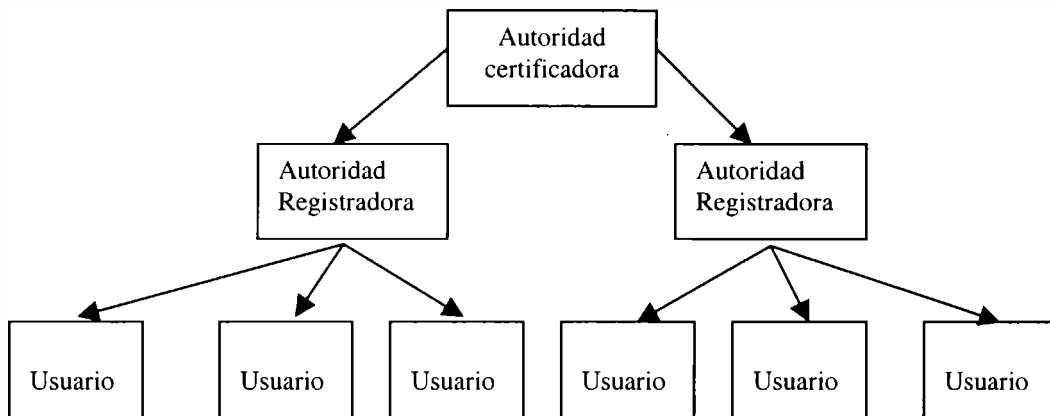
Algunos de los datos mas importantes de este formato son los siguientes:

Versión: 1,2 o 3
Número de Serie:
Emisor del Certificado: VeriMex
Identificador del Algoritmo usado en la firma: RSA, DSA o CE
Periodo de Validez: De Enero 2000 a Dic 2000
Sujeto: Damian Romano
Información de la clave pública del sujeto: la clave, longitud, y demás parámetros
Algunos datos opcionales, extensiones que permite la v3
Firma de la Autoridad Certificadora

Un certificado digital entonces se reduce a un archivo de uno o dos k de tamaño, que autentica a un usuario de la red.

Infraestructura de claves públicas

Teniendo ya un certificado digital que es generado con la ayuda de un algoritmo de clave pública ahora el problema es como administración todos estos, la estructura más básica es la siguiente:



El papel de la Autoridad certificadora (AC) es de firmar los certificados digitales de los usuarios, generar los certificados, mantener el status correcto de los certificados, esto cumple el siguiente ciclo:

- 1) La generación del certificado se hace primero por una solicitud de un usuario, el usuario genera sus claves pública y privada y manda junto con los requerimientos de la solicitud su clave pública para que esta sea certificada por la AC.
- 2) Una vez que la AR (es la AC regional) verifica la autenticidad del usuario, la AC vía la AR firma el certificado digital y es mandado al usuario
- 3) El status del usuario puede estar en: activo, inactivo o revocado. Si es activo el usuario puede hacer uso del certificado digital durante todo su periodo válido
- 4) Cuando termina el período de activación del certificado el usuario puede solicitar su renovación.



Entre las operaciones que pudiera realizar una **AC** están:

- Generar certificados
- Revocar certificados
- Suspender certificados
- Renovar certificados
- Mantener un respaldo de certificados.....

Entre las que pudiera realizar una **AR** están:

- Recibir las solicitudes de certificación
- Proceso de la autenticación de usuarios
- Generar las claves
- Respaldo de las claves
- Proceso de Recobrar las claves
- Reportar las revocaciones....

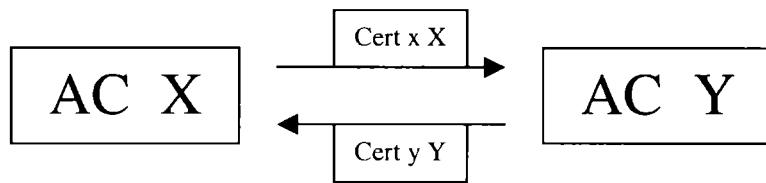
Y las actividades de los usuarios:

- Solicitar el certificado
- Solicitar la revocación del certificado
- Solicitar la renovación del certificado....

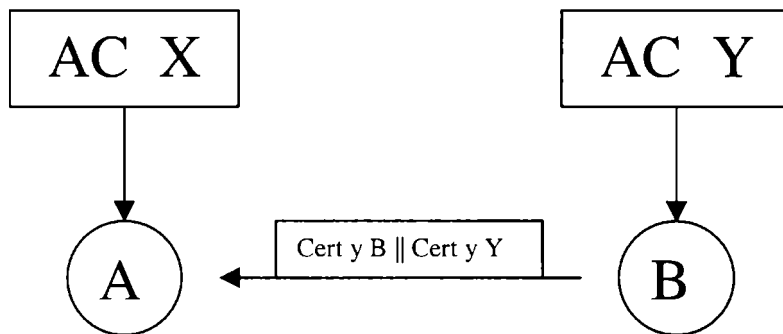
Una vez que algún usuario tiene un certificado digital este puede usarlo para poder navegar por la red con nombre y apellido en forma de bits, esto permite entrar al mundo del comercio electrónico, al mundo de las finanzas electrónicas y en general a la vida cibernética con personalidad certificada. El usuario dueño de un certificado digital tiene la potencialidad de poder autenticarse con cualquier otra entidad usuaria, también puede intercambiar información de forma confidencial y estar seguro de que esta es íntegra, así estar seguro que contactos vía el certificado digital no serán rechazados. Los primeros usuarios de certificados digitales fueron los servidores, actualmente son quienes más los usan, sin embargo también se ha incrementado el número de personas que los usan.

Si suponemos que algún tipo de aplicación funciona ya con certificados digitales, esta tendrá una **AC** y las correspondientes **AR**, sin embargo es común que haya más autoridades certificadoras y que sus usuarios puedan interoperar con sus respectivos certificados, a esto se le conoce como certificación cruzada y opera de la siguiente forma:

1) Las diferentes **AC** pueden estar certificadas enviándose una a otra sus respectivos certificados que ellas mismas generan



- 2) Entonces la **AC X** tendrá el certificado de la **AC Y** y viceversa, pudiendo generar un certificado para **Y** que genera **X** y otro para **X** que genera **Y**
- 3) Ahora como un usuario **A** de la **AC X** puede comunicarse con un usuario **B** de la **AC Y**



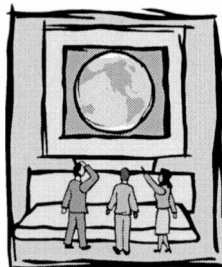
- 4) El usuario **B** envía a **A** el certificado de **B** que genera **Y** (**Cert y B**) junto con el certificado de **Y** que el mismo se genera (**Cert y Y**)
- 5) Ahora **A** puede validar a **B** (**Cert y B**) usando el certificado de **Y** que genera **X**

En la práctica se ha demostrado que el estatus de un certificado cambia con gran frecuencia, entonces la cantidad de certificados digitales revocados crece considerablemente, el problema está en que cada vez que se piensa realizar una comunicación y es necesario validar un certificado se debe de comprobar que este no está revocado. La solución que se ha venido usando es la de crear una lista de certificados revocados **LCR** y así verificar que el certificado no está en esa lista, para poder iniciar la comunicación. El manejo de las listas de certificados revocados ha llegado a tener un gran costo que sin embargo aún no se ha reemplazado por otra técnica a pesar que se han propuesto ya salidas al problema.

Las operaciones de la administración de los certificados digitales puede cambiar de acuerdo a las leyes particulares de cada país o entidad.

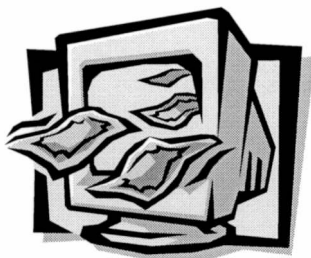
Comercio electrónico

Hoy en día, gran parte de la actividad comercial ha podido transformarse gracias a redes de conexión por computadoras como Internet, esta transformación facilita hacer transacciones en cualquier momento de cualquier lugar del mundo. Todo lo que esta alrededor de esta nueva forma de hacer negocios es lo que se ha llamado comercio electrónico, sin duda la gran variedad de actividades que giraban alrededor del quehacer comercial se han tenido que conjuntar con las nuevas técnicas cibernéticas. Así hoy tanto un comerciante, un banquero, un abogado o una matemático puede hablar de comercio electrónico enfocándose a la parte que le corresponde.



Existen diferentes niveles de hacer comercio electrónico, y su clasificación aún esta por formarse, sin embargo, la parte más visible es la que cualquier usuario en una computadora personal puede ver, esto es hacer comercio electrónico se convierte a comprar o vender usando una conexión por internet en lugar de ir a la tienda. La forma de hacer esto es muy similar a lo que tradicionalmente se hace, por ejemplo: en la tienda uno entra al establecimiento, de forma electrónica se prende la computadora y una vez conectado a internet entra a la página del negocio, enseguida un comprador revisa los productos que posiblemente compre y los coloca en una carrito, de la misma forma en la computadora se navega por la página del negocio y con el browser se revisan los productos que éste vende, al escoger éstos se colocan en un carrito virtual, que no es nada mas que un archivo del usuario. Una vez elegido bien los productos de compra se pasa a la caja, donde se elige un sistema de pago y se facturan los productos al comprador. De forma similar en la computadora se pueden borrar productos que no se quieren comprar o añadir nuevos, una vez elegidos éstos se procede a una parte de la pagina que toma los datos y solicita el método de pago, generalmente se lleva a cabo con tarjeta de crédito.

En la parte tradicional de comprar al pagar en la caja termina el proceso, en la parte por computadora aún tiene que esperarse que sean enviados los productos. A pesar de esto las ventajas que ofrece el comercio electrónico son magníficas, ya que es posible comprar en un relativo corto tiempo una gran cantidad de productos sin necesidad de moverse de lugar, es decir al mismo tiempo se puede comprar una computadora, un libro, un regalo, una pizza, hacer una transacción bancaria etc., de la forma tradicional se llevaría al menos un día completo y eso si los negocios esta en la misma ciudad, si no, el ahorro de tiempo que representa comprar por internet es incalculable.



Al efectuar una operación comercial por internet se presentan nuevos problemas, por ejemplo cómo saber que la tienda virtual existe verdaderamente, una vez hecho el pedido cómo saber que no se cambia la información, cuando se envía el número de tarjeta de crédito cómo saber si este permanecerá privado, en fin, para el comerciante también se presentan problemas similares, cómo saber que el cliente es honesto y no envía información falsa, etc. Todos estos problemas pueden ser resueltos de manera satisfactoria si se implementan protocolos de comunicación segura usando criptografía. En la siguiente sección nos dedicamos a describir como es que estos protocolos resuelven los problemas planteados.

Protocolos de seguridad

Un protocolo de seguridad es la parte visible de una aplicación, es el conjunto de programas y actividades programadas que cumplen con un objetivo específico y que usan esquemas de seguridad criptográfica.

El ejemplo más común es **SSL** (Secure Sockets Layer) (que vemos integrado en el Browser de Netscape y hace su aparición cuando el candado de la barra de herramientas se cierra y también si la dirección de internet cambia de http a https, otro ejemplo es **PGP** que es un protocolo libre ampliamente usado de intercambio de correo electrónico seguro, uno más es el conocido y muy publicitado **SET** que es un protocolo que permite dar seguridad en las transacciones por internet usando tarjeta de crédito, **IPsec** que proporciona seguridad en la conexión de internet a un nivel más bajo.

Estos y cualquier protocolo de seguridad procura resolver algunos de los problemas de la seguridad como la integridad, la confidencialidad, la autenticación y el no rechazo, mediante sus diferentes características

Las características de los protocolos se derivan de las múltiples posibilidades con que se puede romper un sistema, es decir, robar información, cambiar información, leer información no autorizada, y todo lo que se considere no autorizado por los usuarios de una comunicación por red.

Enseguida vemos un escenario donde puede ocurrir algo de esto:

Por ejemplo sobre la seguridad por Internet se deben de considerar las siguientes tres partes: seguridad en el browser (Netscape o Explorer), la seguridad en el Web server (el servidor al cual nos conectamos) y la seguridad de la conexión.

Un ejemplo de protocolo es **SET**, objetivo efectuar transacciones seguras con tarjeta de crédito, usa certificados digitales, criptografía de clave pública y criptografía clave privada.

SSL Es el protocolo de comunicación segura mas conocido y usado actualmente, **SSL** actúa en la capa de comunicación y es como un túnel que protege a toda la información enviada y recibida.

Con **SSL** se pueden usar diferentes algoritmos para las diferentes aplicaciones, por ejemplo usa **DES, TDES, RC2, RC4, MD5, SHA-1, DH** y **RSA**, cuando una comunicación esta bajo **SSL** la información que es cifrada es:

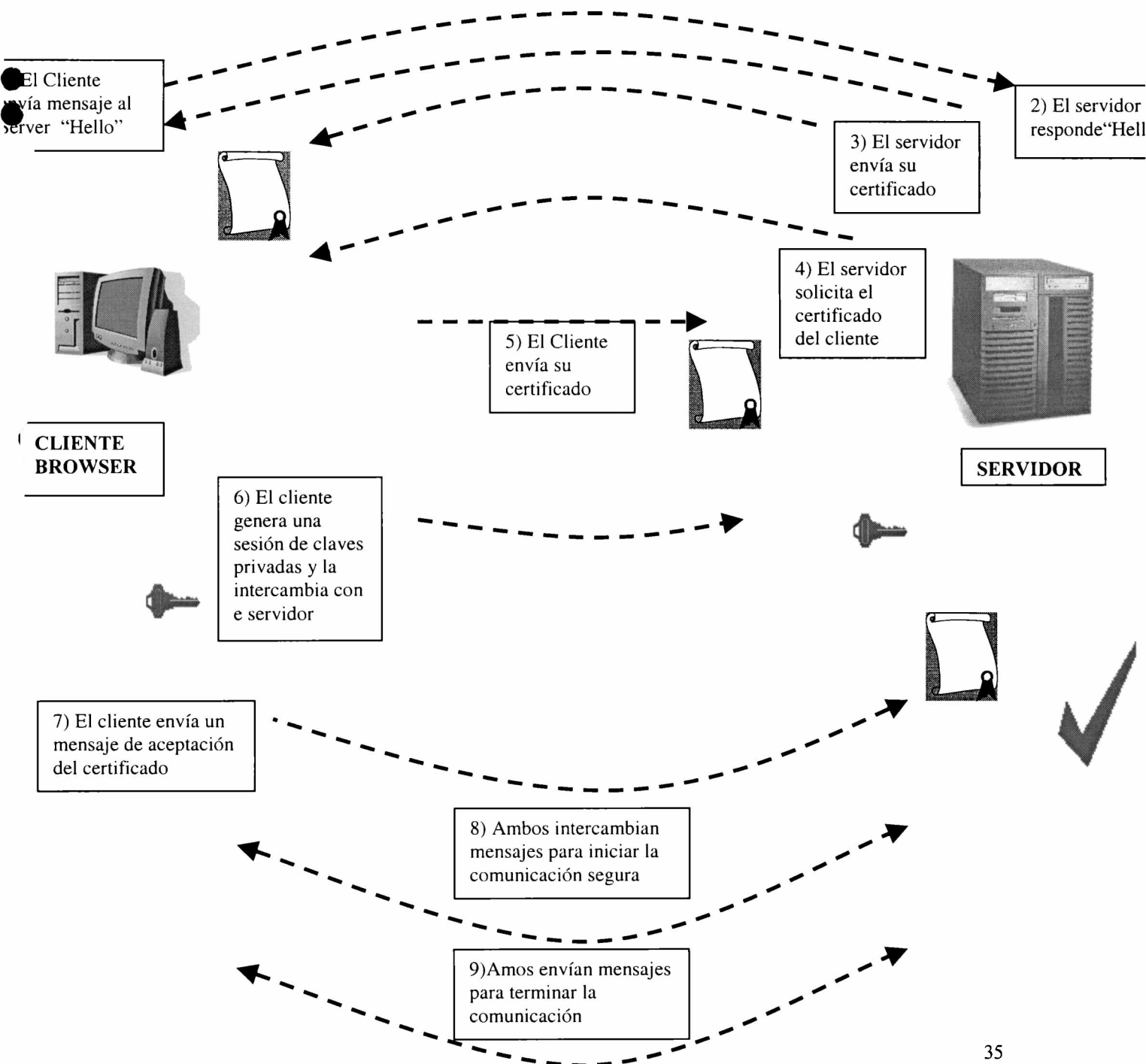
El URL del documento requerido
El contenido del documento requerido
El contenido de cualquier forma requerida
Los “cookies” enviados del browser al server
Los “cookies” enviados del server al browser
El contenido de las cabeceras de los http

El procedimiento que se lleva acabo para establecer una comunicación segura con **SSL** es el siguiente:

- 1) El cliente (browser) envía un mensaje de saludo al Server “ClientHello”
- 2) El server responde con un mensaje “ServerHello”
- 3) El server envía su certificado
- 4) El server solicita el certificado del cliente
- 5) El cliente envía su certificado: si es válido continua la comunicación si no para o sigue la comunicación sin certificado del cliente
- 6) El cliente envía un mensaje “ClientKeyExchange” solicitando un intercambio de claves simétricas si es el caso
- 7) El cliente envía un mensaje “CertificateVerify” si se ha verificado el certificado del server, en caso de que el cliente este en estado de autenticado
- 8) Ambos cliente y server envían un mensaje “ChangeCipherSpec” que significa el comienzo de la comunicación segura.
- 9) Al término de la comunicación ambos envían el mensaje “finished” con lo que termina la comunicación segura, este mensaje consiste en un intercambio del hash

de toda la conversación, de manera que ambos están seguros que los mensajes fueron recibidos intactos

La versión más actual de **SSL** es la v3, existen otro protocolo parecido a **SSL** solo que es desarrollado por **IETF** que se denomina **TLS** (Transport Layer Security Protocol) y difiere en que usa un conjunto un poco mas amplio de algoritmos criptográficos. Por otra parte existe también **SSL plus**, un protocolo que extiende las capacidades de **SSL** y tiene por mayor característica que es interoperable con **RSA**, **DSA/DH** y **CE** (Criptografía Elíptica).



El protocolo SSL

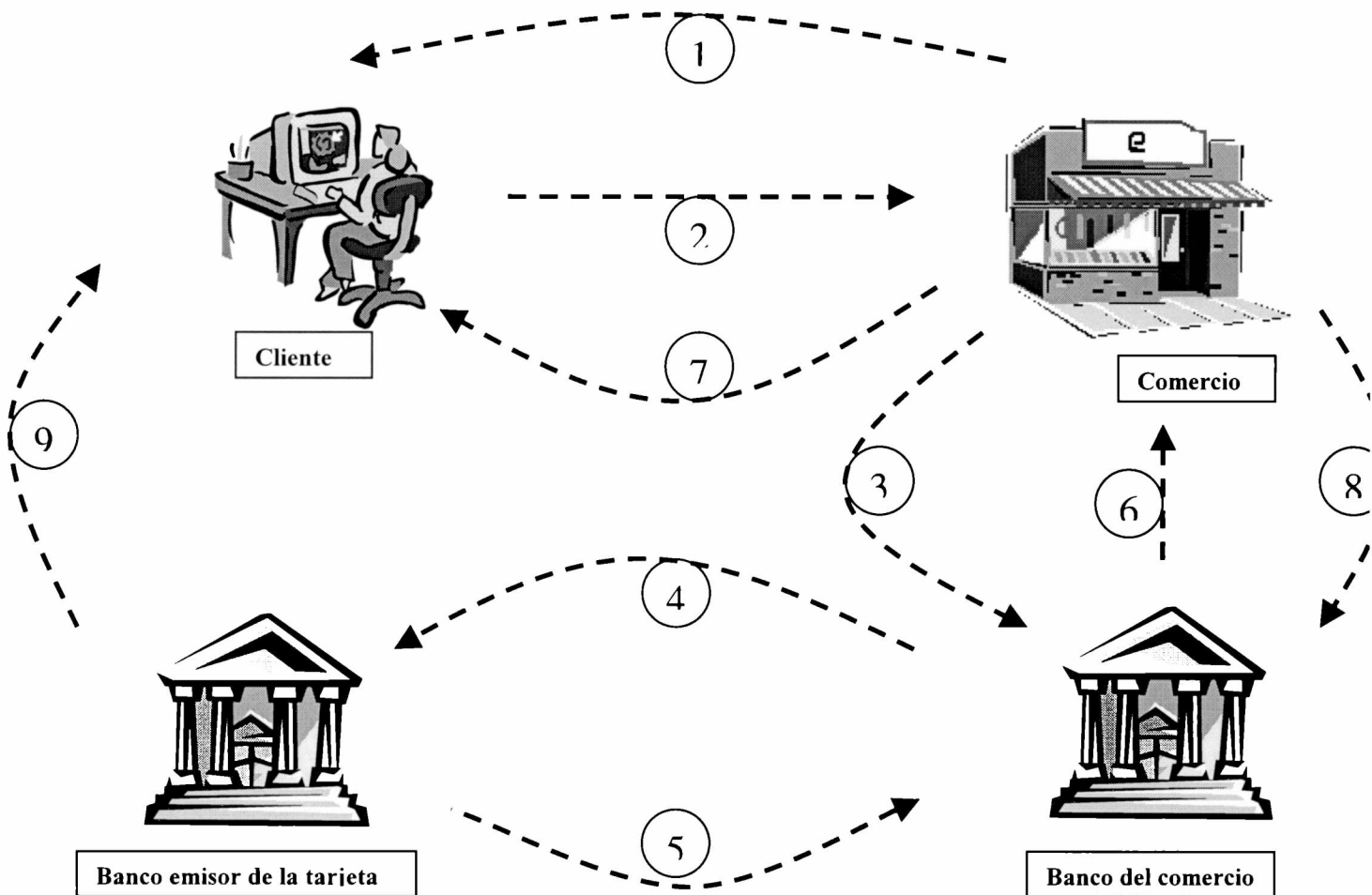
SET este protocolo esta especialmente diseñado para asegurar las transacciones por internet que se pagan con tarjeta de crédito. Esto es debido a que una gran cantidad de transacciones de compra por internet son efectuadas con tarjeta de crédito, por otro lado SSL deja descubierto alguna información sensible cuando se usa para lo mismo. La principal característica de **SET**, es que cubre estos huecos en la seguridad que deja **SSL**.

Por ejemplo con **SSL** solo protege el número de tarjeta cuando se envía del cliente al comerciante, sin embargo no hace nada para la validación del número de tarjeta, para chequear si el cliente esta autorizado a usar ese número de tarjeta, para ver la autorización de la transacción del banco del comerciante etc., Además que el comerciante puede fácilmente guardar el número de tarjeta del cliente. En fin todas estas debilidades son cubiertas por **SET**, éste permite dar seguridad tanto al cliente, al comerciante como al banco emisor de la tarjeta y al banco del comerciante.

El proceso de **SET** es mas o menos el siguiente:

- 1) **El cliente inicializa la compra:** consiste en que el cliente usa el browser para seleccionar los productos a comprar y llena la forma de orden correspondiente. **SET** comienza cuando el cliente hace clic en “pagar” y se envía un mensaje de iniciar **SET**.
- 2) **El cliente usando SET envía la orden y la información de pago al comerciante:** el software **SET** del cliente crea dos mensajes uno conteniendo la información de la orden de compra, el total de la compra y el número de orden. El segundo mensaje contiene la información de pago, es decir, el número de la tarjeta de crédito del cliente y la información del banco emisor de la tarjeta. El primer mensaje es cifrado usando un sistema simétrico y es empaquetada en un sobre digital que se cifra usando la clave pública del comerciante. El segundo mensaje también es cifrado pero usando la clave pública del banco (esto previene que el comerciante tenga acceso a los números de tarjetas de los clientes). Finalmente el cliente firma ambos mensajes.
- 3) **El comerciante pasa la información de pago al banco:** el software **SET** del comerciante genera un requerimiento de autorización, éste es comprimido (con un hash) y firmado por el comerciante para probar su identidad al banco del comerciante, además de ser cifrado con un sistema simétrico y guardado en un sobre digital que es cifrado con la clave pública del banco.
- 4) **El banco verifica la validez del requerimiento:** el banco descifra el sobre digital y verifica la identidad del comerciante, en el caso de aceptarla descifra la información de pago del cliente y verifica su identidad. En tal caso genera una requerimiento de autorización lo firma y envía al banco que genero la tarjeta del cliente.
- 5) **El emisor de la tarjeta autoriza la transacción:** el banco del cliente (emisor de la tarjeta) confirma la identidad del cliente, descifra la información recibida y verifica la cuenta del cliente en caso de que no haya problemas, aprueba el requerimiento de autorización, lo firma y lo regresa al banco del comerciante.

- 6) **El banco del comerciante autoriza la transacción:** una vez recibida la autorización del banco emisor, el banco del comerciante autoriza la transacción la firma y la envía al servidor del comerciante.
- 7) **El servidor del comerciante complementa la transacción:** el servidor del comerciante da a conocer que la transacción que la tarjeta fue aprobada y muestra al cliente la conformidad de pago, y procesa la orden que pide el cliente terminado la compra cuando se le son enviados los bienes que compró el cliente.
- 8) **El comerciante captura la transacción:** en la fase final de SET el comerciante envía un mensaje de “captura” a su banco, esto confirma la compra y genera el cargo a la cuenta del cliente, así como acreditar el monto a la cuenta del comerciante.
- 9) **El generador de la tarjeta envía el aviso de crédito al cliente:** el cargo de SET aparece en estado de cuenta del cliente que se le envía mensualmente.



SET requiere un certificado digital en cada paso de autenticación y usa dos pares de claves, una para el cifrado del sobre digital y otra para la firma, (**SSL** solo usa un par de claves), actualmente **SET** usa la función hash **SHA-1**, **DES** y **RSA** de 1024 bits, estos parámetros fueron tomados para ser compatible con los certificados existentes, aunque el piloto de **SET** usó el sistema asimétrico de cifrado con curvas elípticas y se piensa que soporte también curvas elípticas en la próxima versión de **SET**.

Conceptos de Firma Digital

(Presentación Power Point)

Conceptos de Firma Digital



- Damian Romano
- E-Mail: dromano@bigfoot.com

Temario

- Criptografía, Criptografía de Clave Simétrica y Criptografía de Clave Asimétrica
- Firma Digital: Requisitos de Certificación de Autor, Contenido y Fecha
- Estándares Criptográficos Internacionales
- El Comercio Electrónico en el Internet
- Normativa de Firma Digital del Colegio de Abogados de los Estados Unidos
- Ley de Firma Digital de Utah

La Firma Digital NO es:

- La impresión del dígito pulgar derecho
- La imagen escaneada de una firma quirografaria
- Porqué no? Porque son fácilmente duplicables!

La Firma Digital es:

- Un proceso que permite **asegurar** la
 - **IDENTIDAD** del autor del documento, y la
 - **INALTERABILIDAD** del contenido del documento luego de haber sido firmado.
 - **FECHA** y **HORA** de la firma.
- Mediante métodos **CRIPTOG@#?~\$***

Criptografía: Qué es?

- Es un proceso matemático que convierte información (el texto plano) en algo aparentemente ininteligible (el texto cifrado), en base a una clave secreta (password).
- El proceso es reversible, o sea que se puede volver a obtener el texto plano en base al texto cifrado, pero únicamente si se posee la clave secreta correspondiente.

Repaso: Encriptar vs. Desencriptar

- **Encriptar:**

Texto Plano + Clave de Encriptado --->

---> Texto Cifrado

- **Desencriptar:**

Texto Cifrado + Clave de Desencriptado --->

---> Texto Plano Recuperado

Criptografía de Clave Simétrica

- Utiliza la **MISMA** clave para encriptar que como para desencriptar
- Ejemplos: DES, IDEA.

Repudio: El Problema de la Criptografía de Clave

Simétrica

- Como utiliza la misma clave para encriptar que como para desencriptar, el remitente de la información necesariamente tiene que divulgar su clave secreta para que el recipiente de la información la pueda leer.
- Al verse forzado a divulgar su clave secreta, el remitente luego puede **REPUDIAR** la autoría de la información originalmente enviada.

Repudio: El Problema de la Criptografía de Clave

Simétrica

- Al conocer la clave secreta, el destinatario de la información bien puede modificar el texto plano y volver a re-criptarlo utilizando la misma clave secreta!
- Por ello la criptografía de clave simétrica sirve para garantizar la inalterabilidad del contenido del documento pero no para garantizar la identidad de su autor.

La Criptografía de Clave Pública: Porqué?

- Fue diseñada específicamente para solucionar el problema del repudio, es decir, para garantizar la identidad del autor de un documento.
- También se denomina “Criptografía de Clave ASIMETRICA”.

La Criptografía de Clave Pública: Como Funciona?

- Utiliza una clave **DISTINTA** para encriptar que para desencriptar.
- La clave para **encriptar** se denomina **CLAVE SECRETA o CLAVE PRIVADA, y NUNCA NUNCA NUNCA NUNCA NUNCA NUNCA** se divulga! A nadie!
- La clave para **desencriptar** se denomina **CLAVE PUBLICA** y se publica.

La Clave Secreta y la Clave Pública

- La clave secreta y la clave pública están **INTIMAMENTE RELACIONADAS**
- Un texto plano se puede encriptar tanto con una clave secreta como con una clave pública, pero...

La Clave Secreta y la Clave Pública

- Si se encripta el texto plano con la clave **SECRETA** se debe desencriptar necesariamente con la correspondiente clave **PÚBLICA**.
- Si se encripta el texto plano con la clave **PÚBLICA** se debe desencriptar necesariamente con la correspondiente clave **SECRETA**.

Procedimiento de Firma Digital (Parte 1 de 2)

- El autor del documento plano lo encripta con su clave secreta que sólo él conoce.
- El destinatario lo desencripta con la clave pública del autor (no la propia!) y recupera el texto plano.

Procedimiento de Firma Digital (Parte 2 de 2)

- El texto plano no es recuperable si:
 - El texto cifrado no fue generado en base a la clave secreta del AUTOR.
 - El texto cifrado fue ALTERADO, aunque sólo sea en un bit.

El Problema de la Distribución de Claves Públicas

- Para verificar una firma digital es necesario obtener la clave pública del firmante.
- Esto crea problemas si no se tiene contacto directo con el autor del documento como para obtener su clave pública

Solución: Los Certificados de Clave Pública

- Los Certificados de Clave Pública son documentos que contienen la clave pública de un tercero.
- Los Certificados llevan la firma digital de una Autoridad Certificante, la cual es de confianza a la persona que necesita verificar la firma digital de un documento.

Las Autoridades Certificantes

- Certifican la autenticidad de claves públicas
- Pueden ser empresas respecto de las claves públicas de sus empleados, los bancos respecto de sus clientes, los escribanos, o las Autoridades certificantes per se que serían empresas de objeto social único.

Estructura Jerárquica de los Certificados

- Una Autoridad Certificante puede a su vez tener su propia clave pública certificada por otra autoridad certificante.
- Por ejemplo, la clave pública de un banco podría ser certificada por el Banco Central, la de una empresa en la oferta pública por la Comisión Nacional de Valores, la de un escribano por el Colegio de Escribanos.

La Revocación de Claves Públicas

- Cuando el titular de una clave sospecha que su correspondiente clave secreta ya no lo es, puede revocar su clave pública mediante notificación a la Autoridad Certificante.
- La revocación de la clave pública contiene fecha y hora, permitiéndole así al titular de la clave revocada repudiar cualquier documento firmado con su clave secreta en fecha posterior a la revocación.

La Lista de Certificados Revocados

- La Autoridad Certificante a su vez mantiene una lista de Certificados de claves públicas revocadas, la cual necesariamente debe ser consultada como parte de la verificación de una firma digital.
- Por ejemplo, una empresa mantendría una lista de claves públicas revocadas de los empleados que ya no pertenecen a la empresa.

Servicios de Fechado (Time-Stamping)

- El fechado de un documento es imprescindible para establecer un claro orden cronológico en las transacciones que así lo requieran.
- Las Autoridades Certificantes generalmente también ofrecen servicios de fechado (“time-stamping”) para determinar fehacientemente el momento en que ha sido firmado un documento.

La Criptografía de Clave Pública “RSA”

- De los algoritmos de clave pública disponibles, el que ampliamente es el más popular es el RSA, denominado así por sus tres inventores (Ron Rivest, A. Shamir y L. Adleman) que lo descubrieron en MIT (Massachusetts Institute of Technology) en 1977.

Robustez de la Criptografía de Clave Pública “RSA”

- Con claves de 2048 bits, una computadora capaz de ejecutar 1,000,000 de instrucciones por segundo tardaría 300,000,000,000,000,000,000 años en hallar la clave secreta con la que fuera firmado un texto plano.

RSA en el Comercio Electrónico de Internet (1 de 2)

- SET: "Secure Electronic Transactions Protocol", con la participación de Visa, Mastercard, American Express, Microsoft, Netscape, GTE, Terisa Systems, Verisign e IBM.
- SSL: "Secure Sockets Layer", de Netscape.

RSA en el Comercio Electrónico de Internet (2 de 2)

- **EPP:** "Electronic Commerce Payment Protocol", de Netscape, de evolución posterior a SSL.
- **HTTPS:** "Secure HTTP de **NETSCAPE**" lenguaje de Internet Web para trabajar con estos protocolos de seguridad.

Es Común el Uso de RSA?

- Sí.
- Es tan común como el uso del Netscape Navigator (y otros).
- Todos los que “navegan” el Internet lo tienen,
- y la mayoría ni lo saben!

RSA en Dinero Electrónico

- Tarjetas Inteligentes (“Smart cards”)
- e-Cash, DigiCash, otros.

Bancos en Internet:

- <http://www.sfnb.com/>
- SFNB: "Security First Network Bank", un verdadero ejemplo de comercio electrónico, que opera UNICAMENTE por el Internet y ofrece cuenta corriente, caja de ahorro, chequera, tarjeta de débito y tarjeta de cajero automático.

Habilitación del SFNB

- El SFNB fue habilitado por el Banco Federal de Reserva de los EE.UU. específicamente para operar en el Internet, de acuerdo a los resultados de una auditoría de sistemas llevada a cabo por la NSA ("National Security Agency" - Agencia de Seguridad Nacional), el ente rector de la criptografía en los EE.UU.

Acceso a SFNB

- El acceso por Internet al SFNB se realiza por medio de Web browsers que implementen los protocolos de seguridad HTTPS y SSL, los cuales a su vez utilizan la criptografía de clave pública para establecer la identidad de las partes.

La Normativa de Firma Digital de los EE.UU

- El Colegio de Abogados de los EE.UU. (“ABA- American Bar Association”) publica sus recomendaciones de firma digital para que la misma pueda tener fuerza de LEY.
- Es el resultado de un Grupo de Trabajo integrado por abogados, criptógrafos y profesionales de informática.

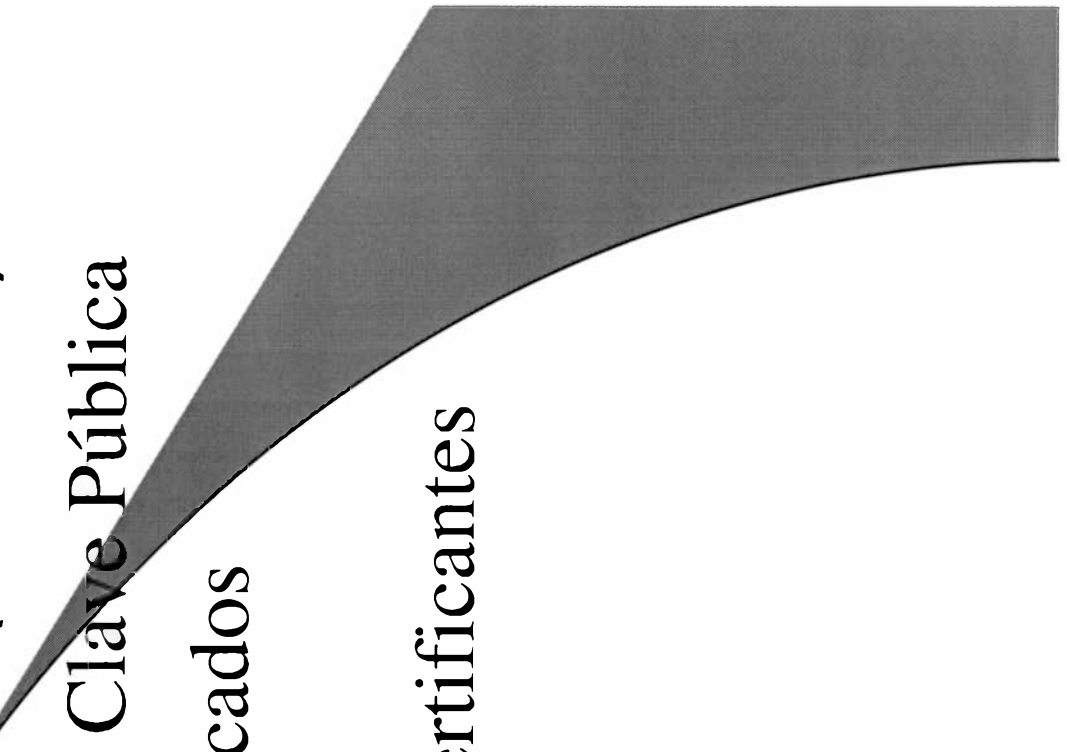
La LEY de Firma Digital del Estado de Utah

- Primera Ley de Firma Digital en el Mundo!
- Promulgada en Mayo de 1995
- Desarrollada en base a la Normativa de Firma Digital del Colegio de Abogados de los EE.UU.
- Ver: <http://www.state.ut.us/ccjj/digsig/default.htm>

La Normativa y la Ley: Qué Tienen en Común? (1 de 2)

- 1. **REQUIEREN** implementar la firma digital por medio de la criptografía de clave pública, como **UNICA** alternativa.
- 2. **REQUIEREN** implementar la firma digital por medio de la criptografía de clave pública, como **UNICA** alternativa.
- 3. **REQUIEREN** implementar la firma digital por medio de la criptografía de clave pública, como **UNICA** alternativa.

La Normativa y la Ley: Qué Tienen en Común? (2 de 2)

- Preveen los Certificados de Clave Pública
 - Preveen las listas de Certificados Revocados
 - Preveen las Autoridades Certificantes
- 

La Ley de Firma Digital de Utah

- En el Estado de Utah HOY...
- La firma digital tiene IDENTICA validez a la firma de puño y letra!

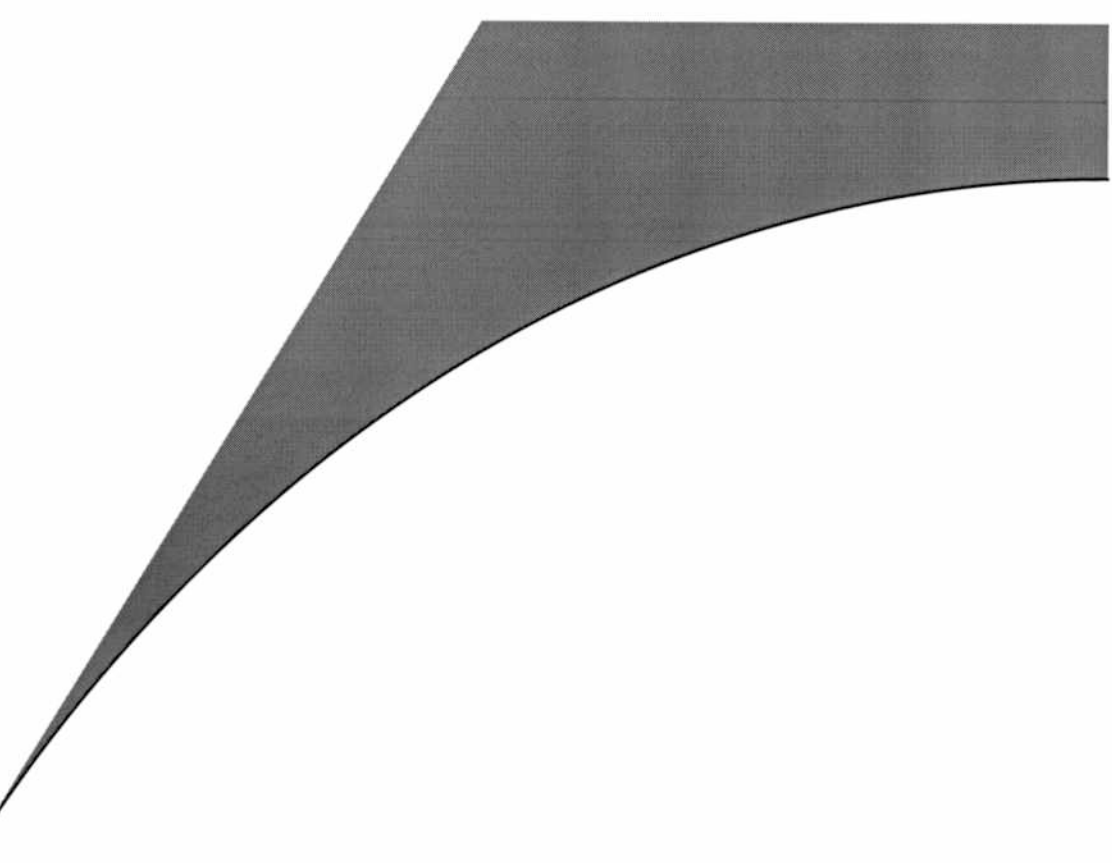
La OLA que se Viene...



- Comercio Electrónico
- Telecommuting
- Tele-Learning
- e-Cash
- HOY!

Ya está en...


- Los EE.UU.
- Inglaterra
- Brasil
- Chile
- Venezuela





Llegará a la Argentina?

- Depende de...



NOSOTROS!

Trabajo de Grado Licenciatura en Informática
Romano Damián A.

Firma Digital y Autoridades Certificantes

(Presentación Power Point)

•
•
•

Certificación de claves públicas Autoridades Certificadoras

Damián Romano

dromano@bigfoot.com

• • • • • • • • • •

•
•
•

Par de claves de una persona

- **Una persona, para poder firmar, necesita un *par de claves*.**
- **Este par de claves es individual**
- **La persona obtiene su par de claves utilizando un programa en su computadora**

• • • • • • • • • •

•
•
•

Qué sucede con las claves?

- **La clave pública debe hacerse disponible a quienquiera la solicite**
- **La clave privada debe mantenerse secreta, y no debe olvidarse**

• • • • • •

•
•
•

Clave privada

- **Para mantenerla secreta, previamente a almacenarla se cifra utilizando un *password*.**
- **El resultado del cifrado se almacena, en lo posible en un medio seguro:
Ej.: smart card, etc.**

• • • • •

Clave privada del firmante

-----BEGIN RSA PRIVATE KEY-----

Proc-Type: 4, ENCRYPTED

DEK-Info: DES-EDE3-CBC, 93832DFCC1628BCB

K2uyyzgaN0CZcebWuHd7jp0lRPa89Yu03MwZ1RPjLUQBxjnQC0jVgX7U00NwE+Tf
nHTGW78Go/h081z4d2k/NItW+sLJctH/HsqGCL7UeeKCrfezz2+X/JbDznw0I/W
+WPhiw45JNax0t/c5D3R+EukJdQ2LsiCMXsPCFbs7fzdsM/dNtw5GsGhRVRsheh/
T2polwDKy1VIYJhT7N2+Py1erSLjdb75m4XT3xuc8pgmENryBFEK8nFRPtp56Eft
8AG1RzzKi/7rydq6TT7c9ZJfCeDCu04WF8N+A3BuWriQnO7mYoIgaBNMbqfPyJff
BAiQXVF1CUMYYNHuLHO11VFbnfTpucPI7RU62/jW8wERRbmmZu12t5jgdUYtYUp9
WGQ1+Vd21KmlJQSWXNrJW1VUs05vezNM/osPUpOCRdzjtjSKkHwjJr9bK8CbKUJ
RP7o3jHgMkjDC9MdbfkwcgTi3mMwWzKQa4b3L8A4CS3t4sdsvzz4NYXx1KuyN+JG
f8dfi1UpGSewS/YNwGZyxfTvoVqzikNrt5n5ajFK0uVMPm1nIjIQkLABmTxVOGWP
74XccrK3m3dujucM5HkCBRRBJJpc2tHfkqLTS2iQF+gW8ofVmIodqYgOga420mKGm
montBaBaQ3WzgJ7t4ppaCAMRD75TJpWahxffF0o49YR6NQ70i7LmENkxBDRR7kxx
o4cmtyT0AQHXk8BKGNrWc1wWmS1y9MQO1PH9m0ydd0iEV2ETOq+Sn1bjCz5f+Oyh
BbBiIGYetz57bX8Eow0WWWJbjlJCjLkzTPswsSGWulaW/OKXc0tJfQ==

-----END RSA PRIVATE KEY-----

•
•
•

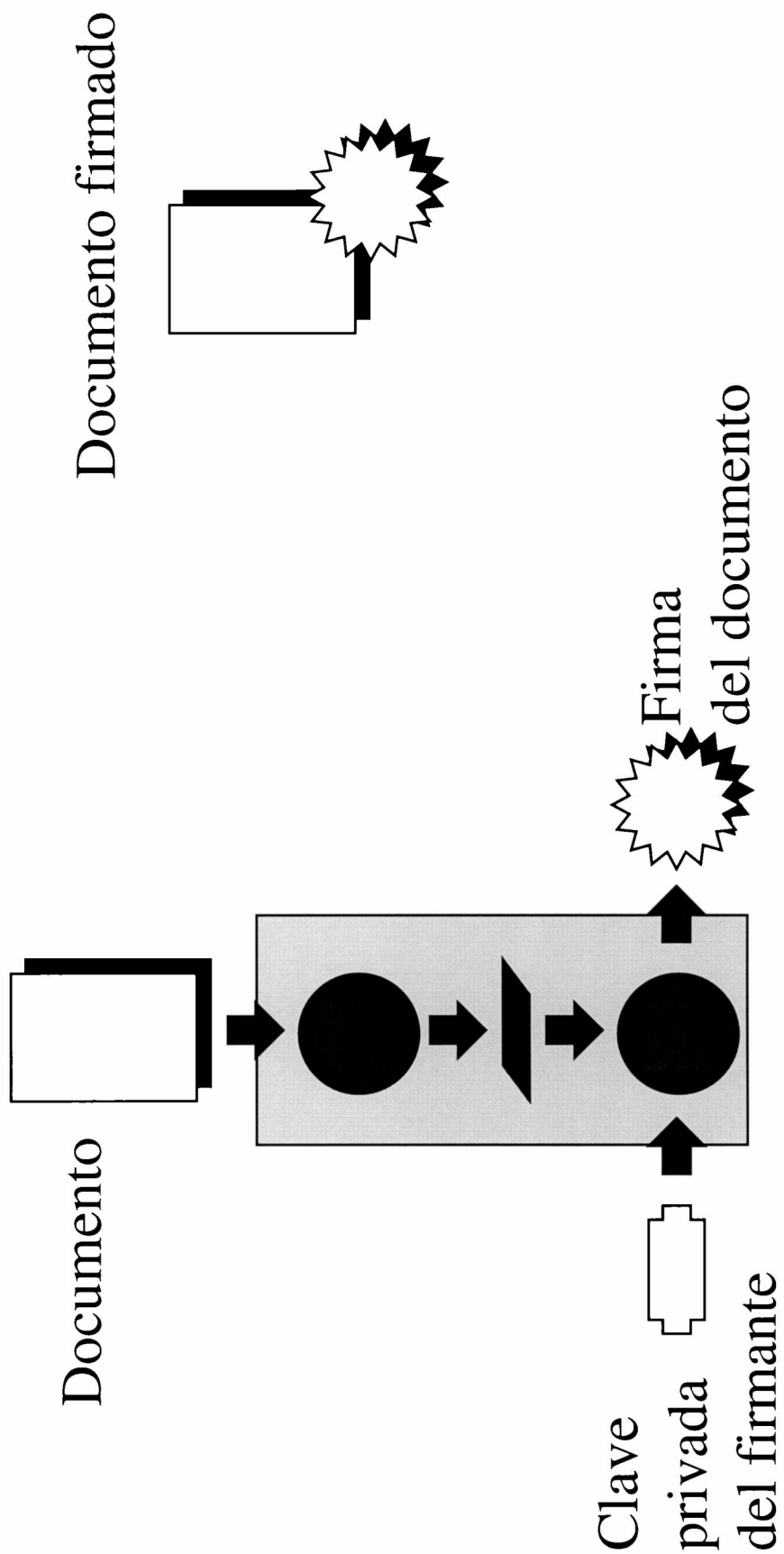
Clave pública

- **Se incorpora a un servicio de directorio público, donde otras personas pueden obtenerla cuando lo deseen**

• • • • • • • • • •

•
•
•

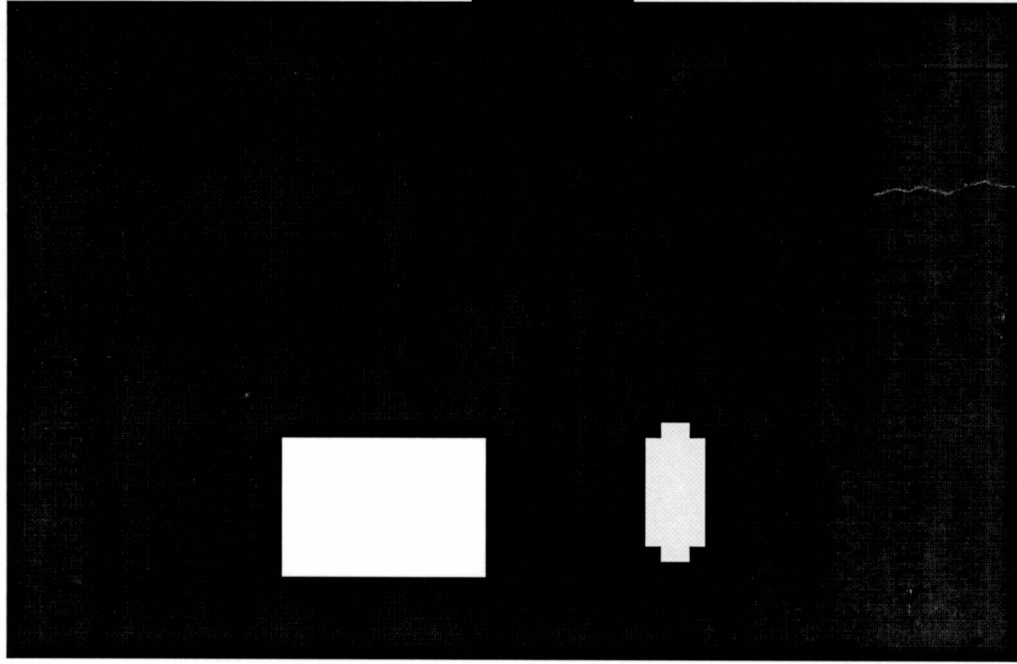
Firma Digital



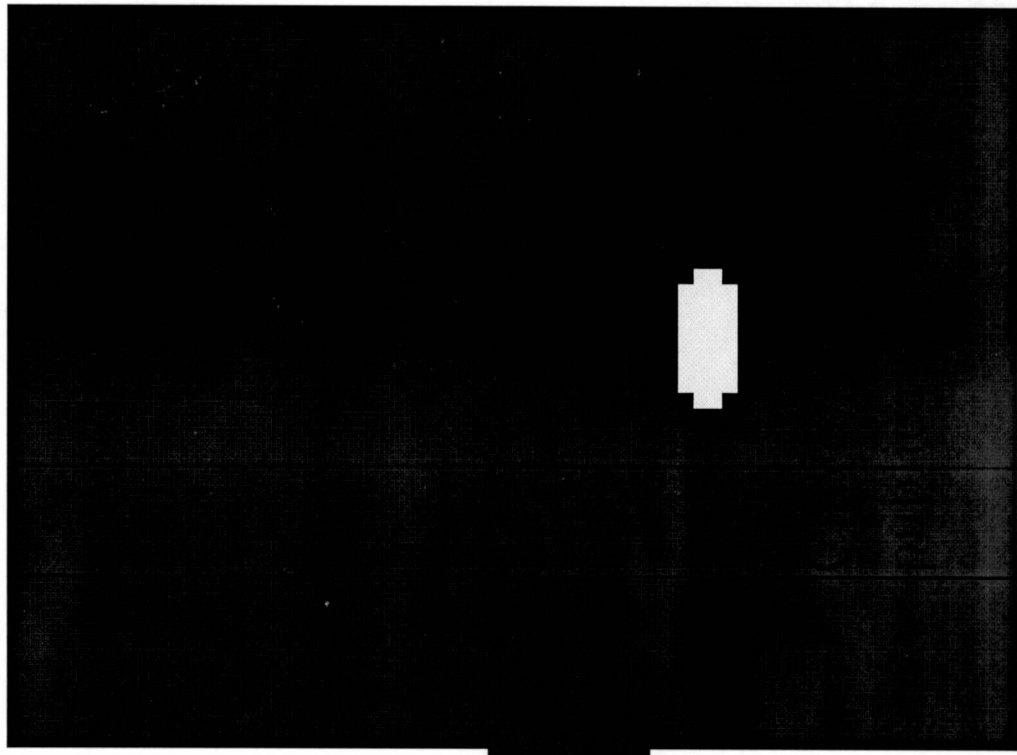
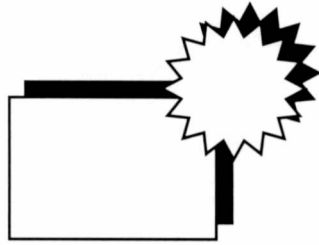
• • • • •

...

Firma y Verificación



Documento
firmado



...

•
•
•

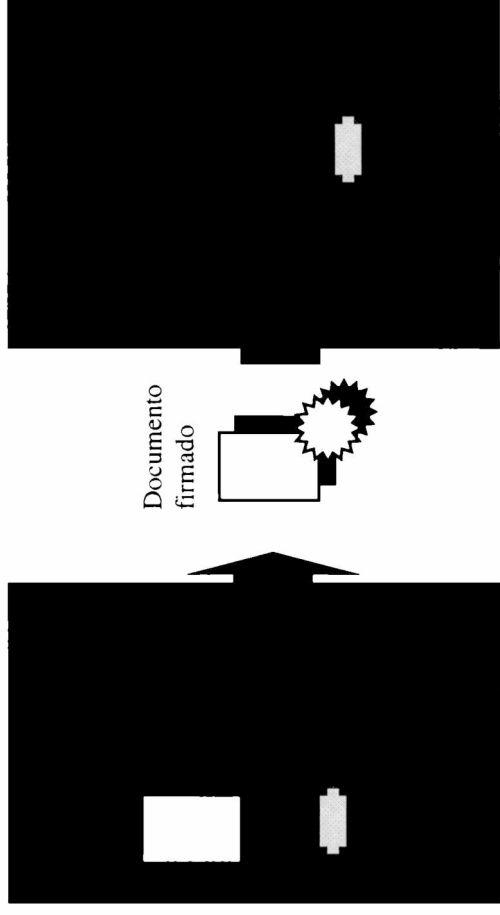
- **El sistema permite asegurar que el autor del documento posee la clave privada asociada con la clave pública utilizada para verificar la firma.**
- **... pero cómo asegurar que ambas claves no son las de un impostor?**

• • • • • • • • • • •

•
•
•

Directorio de claves públicas

- **Es importante notar que el sistema resulta vulnerable a ataques donde se reemplace la clave pública del firmante en el directorio**



- **El directorio debería ser seguro, pero...**

• • • • •

-
-
-

Autoridad Certificadora (AC)

- **Concepto básico:** acreditar la veracidad de una afirmación a través de la confianza en un certificador.
- **En particular:** acreditar una relación entre una identidad (o sus atributos) y una clave pública.
- **La AC es una entidad confiable que garantiza que una determinada clave pública está asociada a una determinada persona.**
Lo hace firmando un *certificado* donde consta que un determinado individuo es poseedor de un determinado par de claves

-
-
-

Autenticación mediante Certificados

- **Un certificado es una estructura de datos que contiene**
 - el nombre de un sujeto (el “*subject*”),
 - su clave pública, y
 - el nombre de una entidad (“*issuer*”, AC) que garantiza que la clave pública está asociada al nombre.
- **Estos datos son firmados criptográficamente usando la clave privada del “*issuer*”**

Certificado emitido por una AC

Certificate:

Data:

Version: 0 (0x0)
Serial Number: 1 (0x1)
Signature Algorithm: md5withRSAEncryption
Issuer: C=AR, SP=Neuquen, L=Piedra del Aguila,
O=Trooch Certificados, Ltd.,
OU=Depto. Certificaciones, CN=Yo-Yo Ma,
Email=yoyo@trooch.com.ar

Validity

Not Before: Oct 9 02:47:53 1996 GMT
Not After : Oct 9 02:47:53 1997 GMT
Subject: C=AR, SP=Buenos Aires, L=Cap. Federal,
O=Consultora San Gabriel S. A.,
OU=Depto. Contable, CN=Juan Perez,
Email=jperez@csg.com.ar

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Modulus:

00:b7:96:f9:23:50:66:cf:ff:a1:3d:f9:91:e3:e3:

...

Exponent: 65537 (0x10001)

Signature Algorithm: md5withRSAEncryption

8b:8e:20:1e:32:02:67:c7:ae:df:50:e9:21:17:48:7b:80:d5:

b7:9d:6d:07:c8:0f:ac:cd:8e:4e:e5:57:59:42:fe:42:04:df:

...

•
•
•

Certificado

-----BEGIN CERTIFICATE-----

MIICmTCCAkMCAQEwDQYJKoZIhvcNAQEEBQAwgbYxCzAJBgNVBAYTAkFzMR4wDgYD
VQqIEwdOZXVxdWVudMR0GAYDVoQHEXfQaWVkcmeGZGVsIEFndWl sYTEiMCAGA1UE
ChMZVHJvb2NoIEN1cnRpb2NoIEN1cnRpb2NoIEN1cnRpb2NoIEN1cnRpb2NoIEN1
cnRpb2NoIEN1cnRpb2NoIEN1cnRpb2NoIEN1cnRpb2NoIEN1cnRpb2NoIEN1cnRpb2
NoIEN1cnRpb2NoIEN1cnRpb2NoIEN1cnRpb2NoIEN1cnRpb2NoIEN1cnRpb2NoIEN1
En1veW9AdHJvb2NoLmNvbS5hcjAeFw05NjEwMDkxMjQ3NTNaFw05NjEwMDkxMjQ3
NTNaMIGzMQswCQYDVQQGEwJBUjEwMDkxMjQ3NTNaFw05NjEwMDkxMjQ3NTNaMIGzMQ
swCQYDVQQGEwJBUjEwMDkxMjQ3NTNaFw05NjEwMDkxMjQ3NTNaMIGzMQswCQYDVQ
QHEwXDYXAuIEZlZGVyYWw5dGVyYWw5dGVyYWw5dGVyYWw5dGVyYWw5dGVyYWw5dGVy
ZWwgUy4gQS4xGDAwBgNVBAsTD0RlcHRvLiBDb250YWJsZS50ZW50ZW50ZW50ZW50
biBQZXJleJEGMB4GCSSqGSIB3DQEJARYRanB1cmV6QGNzZy5jb20uYXlwgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBALeW+SNQzs//oT35kePjcmHxSMkNXdhys/wY
ogIAqALWGHoVYi8nqbrHsk+e3Ldpk6UmiuDz6hR3I8j7C50J4vfJ9KngpMjKbFza
fHEkDufF5DOaUVF/08v5qm7+cbUC8xgXFDHvytSk42GYonE0eAbs+bTNXI9LwJfi
rCr2I8UNAgMBAEAEdQYJKoZIhvcNAQEEBQADQQLjiAeMgJnx67fUOkhF0h7gNW3
nW0HyA+szY505VdZQv5CBN9E/sm7FWpcwWX4FTPL4WCRUF0uP/zohE7SRe2

-----END CERTIFICATE-----

• • • • •

-
-
-

Identidad: Nombres distinguidos (DN)

- **Rec. ITU-T serie X.500**
X.500, X.501, X.509, X.511, X.518, X.519, X.520, X.521
- **Country name**
 - C = "AR".
- **State or Province Name**
 - S = "Misiones".
- **Locality Name**
 - L = "Posadas"
- **Common name**
 - CN = "Dr. Robin Lachlan McLeod BSc CEng MIEE"

X.520 - Tipos de atributos

ATTRIBUTE TYPES

•	A	Aliased Object Name *
•		Authority Revocation List
•	B	Business Category
•	C	CA Certificate
•		Certificate Revocation List
•		Common Name
•		Country Name
•		Cross Certificate Pair
•	D	Description
•		Destination Indicator
•	F	Facsimile Telephone Number
•	I	International ISDN Number
•	K	Knowledge Information
•	L	Locality Name
•	M	Member
•	O	Object Class *
•		Organization Name
•		Organizational Unit Name
•		Owner

ATTRIBUTE TYPES

•	P	Physical Delivery Office Name
•		Post Office Box
•		Postal Address
•		Postal Code
•		Preferred Delivery Method
•		Presentation Address
•		Registered Address
•	R	Role Occupant
•		Search Guide
•	S	See Also
•		Serial Number
•		State or Province Name
•		Street Address
•		Supported Application Context
•		Surname
•		Telephone Number
•	T	Teletex Terminal Identifier
•		Telex Number
•		Title
•		User Certificate
•	U	User Password
•		X.121 Address

-
-
-

Si confiamos en la C.A., podemos extender el ámbito de confianza:

podemos relacionar a un documento con una *identidad*; veamos:

- **la C.A. es confiable, y asegura la relación entre una clave pública y una identidad;**
- **el sistema de firma asegura que quien generó el documento conocía la clave privada correspondiente a la pública =>**
- **entonces es posible verificar la identidad del autor.**

-
-
-

Pero..

- **Estos mecanismos sólo “garantizan” la integridad del mensaje, y la relación entre un DN y el poseedor de una clave privada.**
- **La confianza se translada a la autoridad certificante**
- **El sistema requiere la consulta *online* de CRLs**
- **La filtración inadvertida de una clave privada no puede distinguirse de una firma verídica.**
- **El firmante puede negar la firma denunciando fraudulentamente su pérdida. - Timestamping**

•
•
•

Ejemplo

- **Transacciones comerciales seguras a través de la Internet usando el World Wide Web**
 - requiere autenticación del server (y cifrado)
 - el server presenta un certificado al cliente

• • • • • • • • • •

•
•
•

Cómo obtengo un certificado?

- **Solicitándolo a una AC**
- **vía WWW/SSL; e-mail; correo común**
- **en persona**
- **podría intervenir un escribano que certifica la solicitud, en papel, que luego se envía a la AC para que genere el certificado (en este caso la AC confía en el escribano)**

• • • • •

Obtención del certificado: Paso 1

- El interesado obtiene el software (*)
- Con él, genera su par de claves.
- El software cifra y almacena la clave privada;
- El software genera una *solicitud de certificado* en la que incluye el DN de la persona, y su clave pública.
- La solicitud se firma (dig.) y se guarda en un diskette. El interesado concurre a una AC con la solicitud.

•
•
•

Paso 1bis (optativo)

- **El interesado imprime la solicitud, y concurre a un escribano. Este certifica que la identidad del portador es la correspondiente al DN que figura en la solicitud, firma y sella el papel.**
- **El interesado envía por correo la solicitud certificada a una AC.**

• • • • • • • • • •

•
•
•

Paso 2

- **La AC verifica la autenticidad de la solicitud recibida, ya sea por la presencia del interesado o confiando en la firma del escribano.**
- **La AC genera un certificado, y lo firma digitalmente.**
- **La AC incorpora el certificado en el directorio público y devuelve una copia al interesado.**

• • • • •

•
•
•

Paso 3

- **El usuario recibe la copia del certificado, y la instala en su software.**
- **Una vez instalado, el software puede presentar el certificado a otros sistemas, cuando sea necesario informar la clave pública del interesado.**

• • • • •

•
•
•

Paso 4

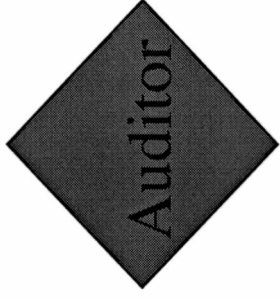
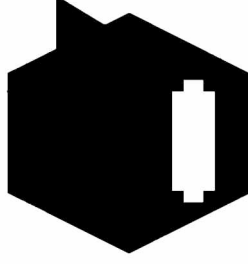
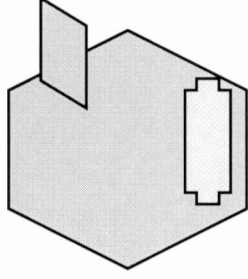
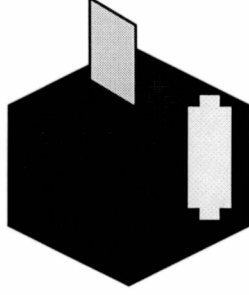
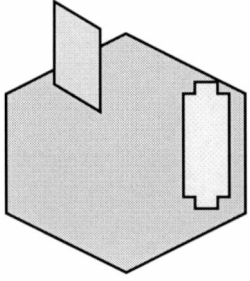
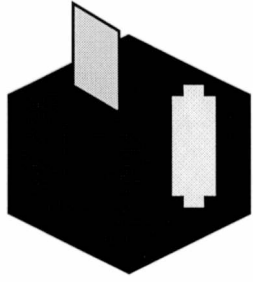
- **El usuario puede ahora firmar documentos con su clave privada, y adjuntar al documento firmado el certificado, para permitir al destinatario la verificación de su firma.**
- **El certificado permite utilizar el sistema de firma digital en múltiples aplicaciones interoperantes**

• • • • •

•
•
•

ACs?

- Las necesitamos?
- Cuántas?
- Cuándo?



•
•
•
•
•
•
•
•
•
•
•

•
•
•

Gracias!

- **Damián Romano**
- **dromano@bigfoot.com**

•
•
•
•
•
•
•
•
•
•

Extractos de los Boletines de Kriptopolis

(www.kriptopolis.com)

EXTRACTOS DE LOS BOLETINES DE KRIPTÓPOLIS	4
INTRODUCCIÓN:	4
DEL BOLETIN #106:	5
<i>INTIMIDAD EN LA RED: ¿UNA BATALLA PERDIDA?</i>	5
<i>DESASTRE ANUNCIADO: AGUJEROS DE SEGURIDAD DEL EXPLORER 5.0</i>	6
<i>ANONIMATO: NUEVA SECCIÓN EN KRIPTÓPOLIS</i>	7
DEL BOLETIN #107:	8
<i>LLEGAN LAS REDES PRIVADAS VIRTUALES GRATUITAS</i>	8
<i>PRESENTADO PGP 6.5: LA SAGA CONTINUA</i>	9
<i>Caballo de troya</i>	10
DEL BOLETIN #108:	11
<i>EXPLORER 5: A SUSTO POR SEMANA</i>	11
<i>FRONTERAS ELECTRÓNICAS ESPAÑA PIDE UN DEBATE PÚBLICO SOBRE ENFOPOL</i>	12
DEL BOLETIN #110:	14
<i>FILETOPIA: Una nueva forma de intercambio seguro de información</i>	14
<i>Censura</i>	16
<i>KRIPTOPOLIS ALCANZA LOS 10.000 SUSCRIPTORES</i>	17
DEL BOLETIN #118:	20
<i>GRAVÍSIMO NUEVO AGUJERO DE SEGURIDAD EN LOS SERVIDORES WEB MICROSOFT</i>	20
<i>"COMERCIO ELECTRÓNICO EN INTERNET: ¿EL FUTURO TENDRÁ QUE ESPERAR?"</i>	22
DEL BOLETIN #121:	25
<i>UN PROFESOR ESPAÑOL PUBLICA INTEGRO EN LA RED SU LIBRO "CRIPTOGRAFÍA Y SEGURIDAD EN COMPUTADORES"</i>	25
<i>UNA NUEVA DISPOSICIÓN LEGAL REGULA LA EXPORTACIÓN DE CIFRADO DESDE ESPAÑA</i>	29
<i>ALEPH</i>	30
<i>BUZÓN KRIPTÓPOLIS</i>	31
DEL BOLETIN #122:	32
<i>NUESTRO TEMA DE DEBATE DE LA SEMANA: "EFECTO 2000, ¿REALIDAD O FANTASÍA"</i>	32
<i>NUEVA EDICIÓN DE "APLICACIONES CRIPTOGRÁFICAS"</i>	34
<i>PRIMERA FASE DE LA CREACIÓN DE UNA RED TEMÁTICA HISPANOAMERICANA EN CRIPTOGRAFÍA</i>	35
DEL BOLETIN #123:	35
<i>BACK ORIFICE 2000 Y OTRAS SERPIENTES DE VERANO</i>	36
<i>"Responsabilidad en la información sobre seguridad. ¿Dónde están los límites?"</i>	38
DEL BOLETIN #125:	40
<i>MAS SOBRE EL FRAUDE POR E-MAIL</i>	40
DEL BOLETIN #127:	43
DEL BOLETIN #128:	44
<i>"CON MICROSOFT NO PUEDE HABER VACACIONES"</i>	44
DEL BOLETIN #130:	53
<i>CUIDADO CON EL PELIGROSO "BUG" DEL ETNOCENTRISMO</i>	53
<i>TAMBIÉN FALLA LA MAQUINA VIRTUAL JAVA DE MICROSOFT</i>	55
<i>OTRO GRAVE FALLO DE SEGURIDAD EN HOTMAIL</i>	56
<i>EXCEL Y EL EFECTO 2000</i>	56
<i>FACTORIZADA CLAVE PÚBLICA DE 512 BIT</i>	57
DEL BOLETIN #131:	57
<i>¿DÓNDE ESTÁN LAS LLAVES?</i>	57
DEL BOLETIN #132:	63
<i>DECIDIDA APUESTA DE KRIPTÓPOLIS POR EL CÓDIGO ABIERTO</i>	63
<i>DISPONIBLE -POR FIN- LA VERSIÓN 1.0 DE GNUPG</i>	66
DEL BOLETIN #133:	67
<i>NUEVA EDICIÓN DEL LIBRO ELECTRÓNICO GRATUITO "CRIPTOGRAFÍA Y SEGURIDAD EN COMPUTADORES"</i>	67
DEL BOLETIN #134:	68
<i>UN JUEGO NO TAN INOCENTE</i>	68

<i>LA VULNERABILIDAD</i>	69
<i>VIAS DE ATAQUE</i>	69
<i>PISTAS</i>	70
<i>¡A JUGAR!</i>	71
DEL BOLETIN #136:.....	73
<i>"LIBERTAD, LIBERTAD, LIBERTAD."</i>	73
<i>DISPONIBLES NUEVOS DOCUMENTOS ELECTRÓNICOS</i>	76
DEL BOLETIN #138:.....	77
<i>UN ESPAÑOL DESCUBRE UNA NUEVA VULNERABILIDAD EN WINDOWS</i>	77
<i>UNDERCON III: ALTO NIVEL EN I+D</i>	81
DEL CRIPTOGRAMA #17:.....	82
1. <i>CÓDIGO ABIERTO Y SEGURIDAD</i>	82
2. <i>¿CLAVE DE LA NSA EN MICROSOFT CRYPTO API?</i>	87
3. <i>COUNTERPANE SYSTEMS: INVESTIGACIÓN DOCUMENTADA</i>	88
4. <i>NOTICIAS</i>	89
5. <i>NOTICIAS EXTREMEDAMENTE PREOCUPANTE</i>	92
6. <i>NOTICIAS DE COUNTERPANE</i>	93
7. <i>EN LA RATONERA: E-TRADE</i>	93
8. <i>FACTORIZAR UN NÚMERO DE 512 BIT</i>	93
9. <i>COMENTARIOS DE LOS LECTORES</i>	95
DEL BOLETIN #139:.....	102
<i>SOBRE EL USO DEL ANONIMATO EN INTERNET</i>	102
DEL CRIPTOGRAMA #18:.....	105
1. <i>PARA QUIENES HAN DECIDIDO HACERSE CRIPTÓLOGOS</i>	105
3. <i>COUNTERPANE SYSTEMS: INVESTIGACIÓN DOCUMENTADA</i>	111
4. <i>NOTICIAS</i>	111
5. <i>NOTICIAS SOBRE SEGURIDAD EN INTERNET</i>	115
6. <i>EN LA RATONERA: AMD</i>	116
7. <i>EMPRESAS PKI Y SUS ESLÓGANES</i>	116
8. <i>LONGITUD DE CLAVE Y SEGURIDAD</i>	117
DEL BOLETIN #141:.....	122
<i>PEOR IMPOSIBLE: GRAVE DEFICIENCIA DE SEGURIDAD EN MICROSOFT ACCESS</i>	122
DEL BOLETIN #142:.....	124
<i>PROTECCIÓN DE MS ACCESS 2000</i>	124
DEL BOLETIN #145:.....	126
<i>"HACKERS": ¡OTRO LIBRO GRATIS PARA TODOS NUESTROS LECTORES!</i>	126
CRIPTOGRAMA #19:.....	127
1. <i>POR QUÉ SON INSEGUROS LOS ORDENADORES</i>	127
2. <i>COUNTERPANE: INVESTIGACION DOCUMENTADA</i>	130
3. <i>NOTICIAS</i>	130
4. <i>ROTO EL CIFRADO DE DVD</i>	133
5. <i>EN LA RATONERA: MICROSOFT WINDOWS CE</i>	137
6. <i>PUESTA AL DÍA DE LA LEGISLACIÓN EN ESTADOS UNIDOS</i>	137
7. <i>CRIPTOGRAFÍA DE CLAVE PÚBLICA DE CURVAS ELÍPTICAS</i>	138
DEL BOLETIN #147:.....	142
<i>CRIPTO SISTEMAS INEXPUGNABLES (I)</i>	142
DEL BOLETIN #148:.....	145
<i>CRIPTO SISTEMAS INEXPUGNABLES (y II)</i>	145
DEL BOLETIN # 151:.....	149
<i>GriYo: CREADOR DE VIDA ARTIFICIAL</i>	149
DEL CRIPTOGRAMA #20:.....	152
1. <i>LA SEGURIDAD NO ES UN PRODUCTO; ES UN PROCESO</i>	153
2. <i>El algoritmo de clave pública de Sarah Flannery</i>	154
Del Boletín #155:.....	155
<i>LA LOTERÍA CHINA</i>	155

Extractos de los Boletines de Kriptópolis

Introducción:

Mucho antes de comenzar mi tesis, cuando todavía estaba por decidir hacia que área de la informática apuntar mi Trabajo de Grado, me encontré en la Web con la gente de Kriptópolis.

Kriptópolis es un Site español dedicado a todo lo concerniente con Criptografía, PGP y Seguridad Informática.

Además de estar presentes en www.kriptopolis.com envían un boletín semanal vía correo electrónico con todas las novedades del área. Este boletín llega a sus suscriptores firmado electrónicamente para verificar la autenticidad del mismo.

La gente de Kriptópolis me ha ayudado durante este tiempo, respondiendo mis consultas y enviándome no solo su boletín, el cual llega a mas de 25.000 personas, sino también material específico para hacer mi tesis.

En este momento la gente de Kriptópolis posee una copia de mi Tesis que será evaluada y luego incorporada al Site para que cualquier persona la pueda bajar e instalar en su propia maquina.

En esta sección quiero mostrar los extractos mas interesantes, según mi criterio, que obtuve de estos Boletines.

Todos los Boletines, que ya son mas de 170, se encuentran en el Site de Kriptópolis para consultar.

Del Boletín #106:

INTIMIDAD EN LA RED: ¿UNA BATALLA PERDIDA?

De nada serviría intentar ocultar ahora que la lucha por mantener a salvo la privacidad de los internautas no atraviesa precisamente por su mejor momento. Múltiples intereses de todo signo parecen conspirar para que todos y cada uno de nuestros datos se conviertan en una simple moneda de cambio, siempre a merced del mejor postor. Las tentaciones para el usuario de la Red son cada vez más abundantes: algunas empresas nos ofrecen llamadas telefónicas más baratas o gratuitas, otras nos regalan cuentas de correo, espacio web, servicios y productos de todo tipo (¡incluso ordenadores!) a cambio de que les cedamos alguna parcela (siquiera ínfima) de nuestra intimidad.

Algunos arrojan públicamente la toalla y consideran la batalla perdida de antemano. Al fin y al cabo -se nos dice interesadamente- en la vida "real" hace mucho que no disfrutamos de intimidad alguna. En efecto; hace tiempo que nuestra personalidad se diluye en un montón de números: documento de identidad, identificador fiscal, número de afiliación a la seguridad social, los códigos asociados a nuestras tarjetas de plástico... Todos fácilmente computables y susceptibles de ser monitorizados, vendidos, comprados e intercambiados. Incluso el hipermercado nuestro de cada día nos tienta con fabulosos sorteos de coches y viajes, a cambio -eso sí- de que dejemos constancia de nuestros datos personales en un simple cupón (sólo para poder avisarnos de que podemos pasar a recoger nuestro ansiado premio, por supuesto).

Así las cosas, a nadie debe extrañar el correlato que tienen en la Red estas actividades invasivas (pero asumidas) de la vida "real". Los sitios web nos sobrecargan el disco duro de 'cookies', para poder monitorizar a su antojo dónde vamos, en qué nos detenemos más y durante cuánto tiempo. Lógicamente, a nadie alarma ya que el navegador más ultramoderno tenga más agujeros que un colador, o que el procesador de última hornada nos coloque una placa de matrícula sin pedirnos permiso. Todo controlado, todo medido, todo monitorizado. ¡Hagan juego, señores!... ¿quién da más? El primitivo ideal de las democracias ("un hombre, un voto") parece haber dejado paso al nuevo lema del conformismo electrónico postmoderno: "un cibernauta, un número de identificación". Pero, ¿no habíamos quedado en que la Red iba a ser "otra cosa"?

Ánimo, prietas las filas y que no decaigan las fuerzas. Todavía no está todo perdido: en Internet aún puede quedar sitio para la utopía.

DESASTRE ANUNCIADO: AGUJEROS DE SEGURIDAD DEL EXPLORER 5.0

Lo que sigue es una recopilación de todos los fallos de seguridad descubiertos en Microsoft Internet Explorer 5.0 sólo en sus primeros días de vida. Como detalle agravante, citar que casi todos ellos ya fueron detectados y corregidos (!?) en la versión anterior del navegador.

Los usuarios más desconfiados deberían probar todas y cada una de las demostraciones que se apuntan, para comprobar hasta qué punto peligran sus datos.

21-03-99 _____

Cómo hacerse impunemente con el contenido del portapapeles del usuario (J.C.G. Cuartango).

* Demo:

<http://pages.whowhere.com/computers/cuartangojc/cb5.html>

21-03-99 _____

Cómo firmar una amenaza de muerte a la Casa Blanca (por ejemplo) mientras se cree estar rellenando un inofensivo formulario (J.C.G. Cuartango).

* Demo:

<http://pages.whowhere.com/computers/cuartangojc/dhtml3.html>

30-03-99 _____

Cómo robar un fichero de su ordenador desde un sitio web. (G. Guninski).

* Demo:

<http://www.nat.bg/~joro/fr.html>

09-04-99 _____

Otra forma de robar un fichero de su ordenador desde un sitio web o

mediante un simple e-mail. (G. Guninski).

* Demo:

<http://www.nat.bg/~joro/scriptlet.html>

09-04-99

Cómo engañar a Explorer 5.0 para que el usuario crea que visita cierto sitio web, cuando en realidad se encuentra en uno totalmente diferente. (G. Guninski).

* Demo:

<http://www.nat.bg/~joro/scrspoof.html>

ANONIMATO: NUEVA SECCIÓN EN KRIPTÓPOLIS

A pesar de llevar algunos años en la brecha, Kriptópolis dista mucho de ser un sitio acabado y podemos vaticinar que no lo será nunca. La Red es un organismo en constante evolución que muestra -además- una vitalidad admirable. Las reglas del "darwinismo digital" está muy claras: quien no sepa adaptarse a sus rotundas exigencias simplemente sucumbirá.

Lo cierto es que aún nos quedan por abarcar muchas herramientas de defensa de la privacidad electrónica, y el anonimato es una de ellas, lo que nos ha llevado a inaugurar una nueva sección sobre el tema.

Al igual que otros instrumentos potentes (la criptografía, por ejemplo), el anonimato es también un arma de doble filo, que puede ser aprovechado por algunos seres, de valentía escasa y naturaleza psicológica dudosa, para insultar y calumniar sin ningún temor a tener que enfrentarse después con las consecuencias. A pesar de ello (y al igual también que ocurre con la criptografía), el anonimato no puede convertirse sin más en una práctica condenable en sí misma, pues también puede proporcionar la protección de la intimidad que sería deseable en muchos casos. Pensemos por ejemplo en la persona que decide buscar ayuda para su problema de alcoholismo en un grupo de news de alcohólicos anónimos ¿no parecería descabellado postear los mensajes desde una dirección de correo real y plenamente identificable?

La nueva sección también trata otros aspectos de la cuestión, como la navegación web anónima. No obstante, nadie debe llamarse aquí a engaño: el anonimato completo sencillamente NO EXISTE. Interponiendo

un 'proxy' (o un 'remailer') entre usted y la Red, tan sólo está causando dificultades para su identificación, pero no debe olvidarse nunca que si sus actividades son abiertamente ilegales y merece la pena el esfuerzo requerido, usted acabará siendo plenamente identificado y será enfrentado a la responsabilidad que se derive de sus actos. Una placa del FBI o Interpol, o un tricomio bien puesto pueden imponer bastante a su intermediario, llegado el caso. Piense - -por ejemplo- en la preocupante situación legal que puede pesar sobre el presunto autor del virus Melissa, quien podría verse abocado a ocupar la celda que Mitnick deje vacante. ¿Acaso no le pareció a él lo bastante fácil enviar un simple fichero contaminado y sustraerse a las perniciosas consecuencias de su acción?

A pesar de todo lo dicho, creemos que esta nueva sección de Kriptópolis está más que justificada. Cualquier lector respetuoso de las leyes y celoso de su intimidad, encontrará en ella sitios desde donde descargar programas que combinan correo anónimo y PGP, interfaces web para navegar o postear anónimamente, abundantes enlaces a sitios y documentos en inglés y en español, etc, etc.

Por supuesto (y como toda página web que se precie), la nueva sección estará permanentemente abierta a sugerencias, críticas y nuevas aportaciones de enlaces, programas y documentos de interés.

* Anonimato en Kriptópolis:

<http://www.kriptopolis.com/anon.html>

Del Boletín #107:

LLEGAN LAS REDES PRIVADAS VIRTUALES GRATUITAS

Acaba de anunciarse la disponibilidad para libre descarga de FreeS/WAN 1.0, un nuevo software para Linux que permite construirse un túnel seguro de comunicaciones a partir de ordenadores de gama baja, que actúen como pasarelas. A diferencia del software habitual de alto coste necesitado hasta ahora para construirse estas "Redes Privadas Virtuales" (VPN), FreeS/WAN es totalmente gratuito y puede descargarse incluso su código fuente.

El proyecto de elaboración de este programa comenzó tres años atrás, impulsado por defensores de la privacidad en la Red de la talla de John Gilmore (cofundador de la Electronic Frontier Foundation) y

algunos otros. Por otra parte, se ha tenido especial cuidado en que todo el código fuese desarrollado fuera de los Estados Unidos, gracias a lo cual ahora puede obtenerse legalmente desde cualquier país, por evitarse las fuertes restricciones norteamericanas a la exportación de criptografía.

El programa permite establecer un canal totalmente seguro en cualquier red de ordenadores. Así, la sede central de una empresa puede quedar conectada a oficinas distantes u ordenadores portátiles por un "túnel seguro", mediante una simple pasarela que cifra la información a la entrada y otra que la descifra a la salida (además de proporcionarse autenticación de los paquetes enviados).

FreeS/WAN (cuyo nombre deriva de S/WAN, el producto patentado y comercializado por RSA), utiliza los protocolos IPSEC (Internet Protocol Security), presentes ya -en mayor o menor medida- en muchos cortafuegos y programas de seguridad, y que formarán parte importante de la versión 6 del protocolo IP, en el que se basa la propia red Internet.

En concreto, FreeS/WAN utiliza intercambio de claves Diffie-Hellman de 1024 bits y cifra cada paquete con claves Triple-DES (3DES) de 168 bits. Al parecer, un Pentium moderno de gama baja es capaz de establecer un túnel seguro en menos de un segundo y puede cifrar 6 megabits de paquetes por segundo. En las pruebas, FreeS/WAN ha interactuado perfectamente con programas similares de OpenBSD, PGP, SSH, Cisco, Raptor y Xedia.

Por otro lado, no debe perderse de vista la ventaja que supone que su código fuente sea público, por cuanto FreeS/WAN puede ser sometido a escrutinio por los expertos, lo que reducirá previsiblemente la posibilidad de aparición posterior de debilidades o fallos no detectados.

* FreeS/WAN:

<http://www.xs4all.nl/~freeswan>

PRESENTADO PGP 6.5: LA SAGA CONTINUA

Y hablando de productos basados en IPsec, hace ahora una semana que se presentó en Estados Unidos PGPNet 6.5, una implementación de IPsec para Windows y Mac, capaz de interactuar con productos similares de otros fabricantes e incluso con certificados X.509, según se afirma en su nota de prensa. De momento, el producto sólo está disponible en la

versión para Windows NT 4.0.

Al igual que otros productos basados en IPSec, en PGPNet 6.5 el cifrado y autenticación ocurren en la capa IP, lo que permite cifrar todo el tráfico (y no sólo el tráfico web, como con SSL, por ejemplo).

Algunas características destacadas son:

* Ficheros autodescifrables (SDA), que podrán ser enviados incluso a destinatarios que no utilicen PGP.

* Plena interoperabilidad con los certificados X.509 de Verisign, tan extendidos en los navegadores web. Naturalmente, esto conlleva el soporte de las autoridades de certificación al uso.

Por último, destacar que se anuncia la próxima disponibilidad de versiones gratuitas del producto y la publicación impresa del código fuente, conforme a la política habitual de los productos PGP.

http://www.nai.com/about/news/press/1999/april/040599_m.asp

Caballo de troya

Programa que se ofrece como capaz de realizar una determinada función (ser un divertido juego, comprimir ficheros, dar información sobre un determinado tema) en lugar de (o además de ello) realiza algún acto de sabotaje contra el ordenador que lo ejecuta, normalmente formatear el disco duro. Los más conocidos caballos de troya son una supuesta versión mejorada del Pkzip, un antivirus para Mac o un programa de información sobre el SIDA. Parece que la primera persona en utilizar esta expresión fue el hacker Dan Edwards y, desde luego, el nombre hace referencia a la famosa trampa ideada por Ulises de regalar a los troyanos un gigantesco caballo de madera con medio ejército griego dentro que acabó de forma expeditiva con el asedio a la ciudad de Troya. Suponiendo que tal evento hubiera tenido lugar alguna vez, el autor de estas líneas comparte completamente la opinión del Lister de "Red Dwarf" de que si la anécdota demuestra algo no es la inteligencia de los griegos, sino la estupidez absoluta de los troyanos.

De todas formas, es sabio seguir en Internet el adagio clásico de "Desconfía de los griegos que te ofrecen regalos". Delante de cualquier programa desconocido hay que aplicar siempre un buen anti-virus antes de ejecutarlo.

Del Boletín #108:

EXPLORER 5: A SUSTO POR SEMANA

El nuevo navegador de Microsoft no se cansa de darnos sorpresas. La última de la saga data tan sólo de hace un par de días, cuando la revista Wired descubrió que una nueva característica de este navegador puede ser aprovechada por los sitios web para mantener un seguimiento de los usuarios que crean marcas (favoritos o bookmarks) en sus navegadores, algo que podría considerarse intrusivo en la privacidad de las personas.

Verán; el asunto funciona como sigue. Cuando usted visita un sitio web que le interesa y decide colocar una referencia en su libro de sitios favoritos, Explorer 5 busca en el directorio raíz de ese web si existe un fichero llamado FAVICON.ICO. Si así es, Explorer descarga ese fichero y lo utiliza para mostrar el pequeño icono -junto al nombre del sitio- en su lista de favoritos.

Lógicamente, la descarga del icono queda perfectamente grabada en los ficheros de registro (logs) del sitio web, que pueden así saber quién lo incluyó entre sus favoritos y cuándo. Quizás tampoco se tarde demasiado en que un sitio web pueda monitorizar cuáles son sus sitios favoritos, con consecuencias terribles para su intimidad (quizás este sea el susto que Explorer puede tenernos reservado para alguna próxima semana...)

Lo más preocupante de este tipo de descubrimientos radica en que revelan la irrefrenable tendencia hacia una programación invasiva de la intimidad, sin que se consulte al usuario en absoluto sobre lo que quiere o no revelar (como han manifestado diversos expertos). Dado que nadie avisó sobre la existencia de esta nueva característica en Explorer 5, tampoco se proporcionó al usuario ninguna posibilidad de desactivarla (si así lo deseara). El fabricante decidió -una vez más- el nivel de intimidad que el usuario puede disfrutar.

Curiosamente, esta característica sí se menciona en un sitio web para desarrolladores de sitios web, como puede verse en:

<http://msdn.microsoft.com/workshop/essentials/versions/ICPIE5.asp>

Pero casi ningún usuario particular tuvo conocimiento de ello hasta que Wired descubrió casualmente el hecho cuando efectuaba una revisión

rutinaria de los registros de su propio servidor web, y difundió al mundo esta curiosa particularidad que, por cierto, no comparte (al menos de momento) el navegador de Netscape.

<http://www.wired.com/news/news/email/explode-infobeat/technology/story/19160.html>

FRONTERAS ELECTRÓNICAS ESPAÑA PIDE UN DEBATE PÚBLICO SOBRE ENFOPOL

(Comunicado de Fronteras Electrónicas, FrEE: <http://www.arnal.es/free>)

"En diciembre de 1991, durante el encuentro de Trevi de ministros comunitarios, se decidió iniciar un estudio para la creación de un sistema de escucha y vigilancia en toda la Unión Europea, bajo el nombre de ENFOPOL. Sin embargo, esta entidad fue un secreto absoluto para los ciudadanos europeos hasta que en febrero de 1997 el grupo anti-escuchas británico Statewatch hizo público un documento sobre ENFOPOL datado en 1995.

El pasado noviembre de 1998 los periodistas de Telepolis.de (una publicación electrónica alemana) Christiane Schulzki-Haddouti y Erich Moechel iniciaron una serie de artículos sobre el tema, detallando los planes para la creación de una masiva red de escuchas en la Unión Europea, y publicando en su integridad diversos documentos relacionados con la creación de ENFOPOL. Los documentos fueron obtenidos por más de una fuente del Parlamento Europeo y contrastados entre sí para asegurar su validez. Sin embargo, hasta la fecha, nadie de la Unión Europea ha desmentido o aceptado la existencia de esta organización, y el resultado final es que el debate público, tan necesario en una democracia, es -de momento- imposible.

Si las informaciones de Telepolis son ciertas, la creación de ENFOPOL supondría un duro golpe a la privacidad de los ciudadanos de la Unión Europea. En los documentos liberados por Telepolis puede observarse como las demandas de ENFOPOL refieren a todo tipo de comunicación: llamadas telefónicas locales y de larga distancia, buzones de voz, correo electrónico, chats, teléfonos móviles y satélite. Aún más preocupante: los documentos de ENFOPOL exigen una actuación rápida y sin barreras, y la posibilidad de monitorizar continuamente. Si a ello unimos el secretismo que rodea a esta organización -aún no oficialmente admitida en la Unión Europea- no hay que ser un genio

para deducir que las escuchas de ENFOPOL se llevarán a cabo sin ningún tipo de garantía legal o autorización judicial. Si este plan se lleva finalmente a cabo, todos los ciudadanos de la Unión Europea se convertirían en posibles criminales a los que hay que espiar.

No contentos con ello, ENFOPOL exige también puertas traseras a proveedores de Internet para poder monitorizar conversaciones en la red en cualquier momento necesario, sin necesidad de pedir permiso previo al proveedor.

Aún más inquietantes son las exigencias del documento ENFOPOL sobre criptografía. Así se pide que sólo se permitan servicios criptográficos regulados desde un "tercero de confianza". Esta tercera persona de confianza (de confianza para ENFOPOL, suponemos) debería entregar automáticamente a un agente de ENFOPOL la identificación completa del usuario de una clave, los servicios técnicos que usa y "los parámetros técnicos del método usado para implementar el servicio criptográfico".

EL documento también habla de poder descifrar, en caso de que sea necesario, un mensaje en unas pocas horas, o incluso en minutos. No es difícil colegir que ENFOPOL quiere un acceso directo, sin ni siquiera pasar por el juzgado, a las claves privadas de los ciudadanos de la Unión Europea. En pocas palabras: la muerte de la criptografía segura.

También hay problemas económicos: si las exigencias de ENFOPOL se aprueban, las compañías que ofrecen telefonía móvil y de satélite tendrán que efectuar costosas inversiones en nuevo equipo por varios millones de Euros, pues las infraestructuras actuales no están preparadas para ofrecer este tipo de "servicios".

De todas formas, lo más preocupante es la falta absoluta de transparencia. El documento que daba luz verde a la creación de ENFOPOL -ENFOPOL 112 10037/95- está firmado por todos y cada uno de los miembros de la Unión Europea. Sin embargo, ningún estado ha admitido todavía la existencia de negociaciones y reuniones para crear este organismo. Sus funciones, prioridades y facultades son desconocidas para el público y la oposición y no hay ningún debate para decidir si estamos de acuerdo con la creación de ENFOPOL y qué poderes queremos concederle realmente.

Y hay otros acontecimientos sospechosos: las recientes leyes aprobadas en Austria y en Alemania para regular la actividad de los proveedores de acceso a Internet se parecen demasiado a los documentos oficiales de ENFOPOL como para ser una coincidencia. Lo normal sería que las cuestiones acerca de telecomunicaciones se discutieran en el

Parlamento Europeo, que tiene competencias en estos asuntos, pero se prefiere la vía indirecta de reuniones uno a uno en los departamentos de telecomunicaciones y fomento, pues un debate en el Parlamento implicaría hacer público toda una serie de documentos clasificados, cosa que no es necesaria en reuniones a puerta cerrada.

No somos tan ingenuos como para pedir la desaparición de los servicios secretos del mundo. El crimen informático existe, y es necesario luchar contra él. Pero no podemos hacer de la seguridad nacional el único baluarte de la democracia. La privacidad, la libertad de expresión y la posibilidad de que los ciudadanos participen en la vida pública también son pilares importantes. El control parlamentario es condición de posibilidad de cualquier democracia, y alguien quiere escamotearnoslo en el asunto ENFOPOL. Es razonable pedir el secretismo de operaciones concretas en las que el éxito policial esté en peligro. No lo es que las prerrogativas, poderes y la misma existencia de una organización permanezca en las tinieblas.

Por ello, Fronteras Electrónicas pide a los gobiernos europeos y al español en particular que los documentos sobre ENFOPOL dejen de ser secretos, que se debata sobre ellos públicamente, que las comisiones en las que se efectúa su construcción sean públicas. También pedimos la movilización de todo tipo de organizaciones pro libertades civiles, así como a los partidos de la oposición, para conseguir que la transparencia llegue finalmente al asunto ENFOPOL."

Del Boletín #110:

FILETOPIA: Una nueva forma de intercambio seguro de información

Por Enrique Martín
webmaster@filetopia.com

Motivados por la falta de seguridad y privacidad de los protocolos existentes en Internet para el intercambio de información en tiempo real, nos hemos decidido a crear un nuevo servicio que permita solventar todos estos problemas mediante el uso de tecnología criptográfica.

Filetopia consiste en un Gestor de Colecciones acompañado de un Servidor de Ficheros y de un cliente de Chat. Usa cifrado fuerte -256

bits- además de un mecanismo de Clave Pública para garantizar la seguridad tanto en el servidor de ficheros como en el chat. Es muy fácil de usar; el Wizard de Iniciación ayudará a los usuarios novatos a funcionar con Filetopia en minutos, automatizando todas las tareas de configuración. El Gestor de Colecciones permite crear listas con todos tus ficheros, compararlas con listas de otros usuarios y coger sólo aquellos ficheros que no están ya en tu colección. El cliente de Chat contiene mecanismos para proteger tu identidad y para prevenir ataques por parte de otros usuarios (que se ven con tanta frecuencia en IRC).

Por un lado, las direcciones IP de los usuarios no aparecen en parte alguna en Filetopia, el Nick está protegido con password para prevenir usurpaciones de identidad y los canales, además de operadores, introducen el concepto de "dueño", la persona que creó el canal por primera vez y que funciona como árbitro entre los operadores, lo que permite impedir una toma del canal. El Chat no es programable con scripts, lo que dificulta mucho los posibles ataques desde dentro. Para evitar los ataques desde fuera, además de ocultar la dirección IP, Filetopia utiliza puertos aleatorios tanto para el servidor de ficheros como para el Chat. Además, a diferencia del IRC, todos los mensajes de Chat, tanto públicos como privados y todos los ficheros intercambiados, no pasan nunca por el servidor de Filetopia, sino directamente entre los usuarios. El servidor de Filetopia se limita a servir de punto de encuentro gestionando los canales e implementando los protocolos de intercambio de la forma más segura y automática posible.

Para el protocolo de clave pública, se emplea el algoritmo de la Curva Elíptica y los componentes de cifrado simétrico están basados en el algoritmo RC6 de Ron Rivest, un firme candidato al AES (American Encryption Standard), el sucesor del DES y futuro estándar de cifrado para el siglo XXI. En Filetopia no hemos inventado nuevos métodos de encriptación, sino que empleamos aquellos algoritmos considerados como "estado del arte" en la comunidad criptográfica, adelantándonos un poco a lo que es probable que pronto sea el estándar.

Además de privado y seguro, Filetopia pretende ser fácil de usar y muy cómodo. Para intercambiar ficheros con un usuario no es necesario conocer su dirección IP, puerto, login y password, tan sólo basta con seleccionarlo en la lista de usuarios de un canal y presionar un botón. Si queremos saber primero lo que este usuario ofrece, al presionar otro botón recuperamos automáticamente su lista de ficheros, que se compara con nuestra lista y nos muestra los resultados, indicando tanto los ficheros que nos faltan como los que podemos aportar. En resumen, se ha pretendido crear un sistema auto-suficiente e integrado para permitir el intercambio de ficheros de forma segura.

El cliente de Filetopia es totalmente gratuito sin ninguna limitación ni restricción en su uso. Actualmente se encuentra en fase beta, aunque ya está disponible públicamente. Para más información y para bajarte el software, visita:

<http://www.filetopia.com>

Censura

Acto mediante el cual se impide que se haga pública una determinada información. La censura puede aplicarse a textos, fotografías, dibujos, sonidos, etc. La censura es casi tan vieja como la humanidad: de 1275 data una ordenanza del rey francés Felipe el Atrevido para tener consignados a todos los libreros, pero no es hasta el siglo XIX, con los sistemas de comunicación de masas, en que ésta se convierte en una verdadera preocupación de los gobiernos, llegando al sùmmum en este siglo XX.

La censura puede presentar múltiples grados y justificaciones. Tenemos actos censores simplemente insensatos, que ni siquiera aprovechan a los censores, como la introducción de El Quijote en la lista de libros subversivos de determinadas dictaduras latinoamericanas; luego hay actos de censuras maquiavélicamente justificadas, que atentan contra los derechos humanos, como la censura de una dictadura impidiendo que se hable mal de su gestión; también está la censura moral por la que un determinado grupo se erige en portavoz del buen gusto o la ética y decide qué pueden ver, oír, o leer los ciudadanos de un país...

Pero también existe la censura justificada. El concepto en sí es problemático, sobre todo en una Red donde la divisa "la información quiere ser libre" campa por sus respetos, pero sí parece claro que hay situaciones como la registrada en la entrada "amenazas" de este diccionario, en que parece razonable censurar lo que afirma una persona si ésta, al ejercer su libertad de expresión, pone en peligro un derecho más básico de otra persona, como su derecho a la vida o al honor. De todas formas, repetimos lo que se dijo en "amenazas": la violación del otro derecho más básico ha de estar perfectamente clara. En caso de duda es mucho más razonable permitir que el discurso fluya libremente. Es demasiado fácil malinterpretar (consciente o inconscientemente) a una persona y convertir todo lo que dice en una amenaza o un libelo.

Otro aspecto importante de la censura es que ésta no ha de ser

necesariamente gubernamental. Efectivamente, los gobiernos, por su misma estructura, son los que más fácilmente pueden censurar, pero no son los únicos. Existe también la "censura social" tan vieja como la gubernamental, en la que grupos de presión pueden conseguir hacer callar a personas o medios de comunicación. De hecho, buena parte de los casos de censura gubernamental en Internet son de hecho casos de censura social, pues no es el gobierno per se el que decide cerrar un website, sino el gobierno que, por miedo a perder votos o sufrir desgaste, decide seguir a la opinión mayoritaria del país. Un ejemplo perfecto es lo sucedido con el website portugués Terravista.

La existencia de instrumentos tecnológicos como Floodnet o el mailbombing hacen que la censura social pueda ponerse en práctica por grupos pequeños sin necesidad de la intervención de ningún estado u organismo, como ejemplifica perfectamente el caso del EuskalHerria Journal.

Finalmente, mencionar que la censura en Internet se puede practicar de dos formas básicas: una de forma directa, impidiendo que una determinada página o website continúe en la Red, y otra de forma indirecta, utilizando software de filtrado, bloqueo o etiquetado. La existencia de fenómenos como los mirrors pone de manifiesto que la censura directa es poco efectiva, pues siempre hay alguien dispuesto a hacer copias de la página original en algún país lejano. Por el contrario, la censura indirecta casi ni se nota, no genera titulares en los diarios y por ello es mucho más insidiosa.

KRIPTOPOLIS ALCANZA LOS 10.000 SUSCRIPTORES

"How do you have 10,000 subscribers?

Where did they come from?"

(Bruce Schneier, a Kriptópolis)

Si hace unas semanas celebrábamos el emblemático número 100 de este boletín, en esta ocasión deseamos festejar otra cifra 'redonda': este ejemplar (el 110) llegará -por primera vez- nada menos que a 10.000 suscriptores.

Hace unos días, el propio Bruce Schneier, gurú mundial de la criptografía, se mostraba sorprendido por la elevada difusión de nuestro boletín. La pregunta de Bruce estuvo resonando en nuestra mente al menos un par de días más, ya que responderla no parecía demasiado fácil. La dedicación, el esfuerzo, la paciencia son

cualidades necesarias, pero que hay que presuponer a cualquiera que decide publicar en la Web; por tanto, no sirven de explicación suficiente. Afortunadamente (y como acostumbra a suceder en este medio cuasi-milagroso), la respuesta adecuada nos llegó cuando menos la esperábamos, y gracias al mensaje de crítica de un suscriptor.

Nuestro comunicante manifestaba su disgusto por la aparición en nuestro web y en algunos de nuestros boletines, de la firma de un determinado personaje de la Red española que -evidentemente- no era santo de su devoción. Se preguntaba cómo era posible que un sitio como el nuestro, tan claramente implicado en la defensa de los ciberderechos, contara entre sus colaboradores habituales con esa cierta persona que -a su juicio- tanto daño había hecho a la Internet española más libre y combativa.

Esta crítica nos trajo a la luz una de las características más distintivas de Kriptópolis y que (quizás de puro asumida) había desaparecido ya de nuestra consciencia más inmediata: **KRIPTÓPOLIS ESTÁ ABIERTO A *TODOS***.

De verdad: ¿cuántos boletines electrónicos conoce donde se pueda leer (¡y a veces en un mismo ejemplar!) a un respetable profesor de universidad y a un recalcitrante 'hacker'? ¿Cuántos sitios web exhiben -sin prejuicios ni autojustificaciones- enlaces a sitios 'bienpensantes' junto a los que conectan con "el-lado-más-oscuro"?

Reflexionando sobre estas obviedades, nos vinieron a la mente algunas otras peculiaridades de Kriptópolis, que le confieren su particular sello distintivo a la hora de tratar la seguridad en Internet:

* Aún a estas alturas (más de tres años de presencia, más de 10.000 suscriptores, medio millón de impactos/mes), **KRIPTÓPOLIS NO ESCRIBE AL DICTADO** de nadie, ya se trate de editoriales, organismos oficiales o empresas de cualquier tipo.

* La creciente popularidad de Kriptópolis tiene otro signo característico: **JAMÁS** (insisto: **JAMÁS**), hemos pagado un duro por anunciarnos. Kriptópolis ha llegado hasta aquí gracias a la **PUBLICIDAD BOCA A BOCA**.

* Con todo mi respeto y agradecimiento a quienes ocasional o continuamente me ayudan, Kriptópolis continúa siendo un **SITIO PERSONAL**, por mucho que su popularidad, difusión o volumen de visitas puedan inducir a pensar otra cosa. En la época de los magaportales y las macroinversiones, Kriptópolis es el ejemplo vivo de cómo la Red sigue permitiendo a un mero **PARTICULAR** llevar lo bastante lejos sus

opiniones o informaciones. Por ello siempre nos halaga (aunque no sea demasiado justo) que se compare nuestro modesto trabajo al que realizan redacciones de muchas mayores dimensiones y -no digamos- presupuesto.

* Como consecuencia de las dos afirmaciones anteriores, el creador, editor y único responsable de Kriptópolis siempre decide -según su exclusivo criterio PERSONAL- lo que aquí se trata o no, así como la importancia de cada asunto o el tratamiento que merece. La INDEPENDENCIA es -pues- TOTAL Y ABSOLUTA. Incluso cuando la información trata de un nuevo producto o servicio, cuya existencia Kriptópolis considera interesante o necesario difundir, jamás median - - hasta la fecha- implicaciones económicas o publicitarias.

No obstante, nuestros detractores habituales (que también los tenemos) y los impenitentes envidiosos (quién no los tiene) pueden descansar tranquilos: el éxito de Kriptópolis se refiere sólo a su popularidad y al respaldo que le proporciona su respetable número de seguidores incondicionales. En lo que respecta a su utilidad como fuente de ingresos ésta es absolutamente inexistente. Parece claro que nos "vendemos" muy mal. Nuestros ingresos con *CERO* absoluto y respecto a los gastos... ¡prefiero no mencionarlos por pura vergüenza! Así pues, que nadie ose contratarnos como gestores económicos; nadie es capaz de sacar tan poco beneficio a tanto volumen de visitas y tanta riqueza potencial. Como anunciamos públicamente unos boletines atrás, tenemos un acuerdo de colaboración con Anonymizer y ni siquiera lo recomendamos en nuestras páginas porque somos bastante críticos con respecto a su eficacia real. En cuanto a nuestro acuerdo respecto a Freedom, tenemos grandes esperanzas puestas en ese programa, pero mal podemos recomendar algo que aún sólo es una promesa. ¿A dónde podríamos llegar así?

Grandeza y miseria de Kriptópolis. Nuestra independencia tiene un coste muy alto. Las empresas no ven perspectivas de doblegar nuestro criterio y -por tanto- no invierten. Sin ingresos, sólo quedan los gastos (que crecen -por cierto- al mismo ritmo que lo hacen nuestras visitas).

No obstante, Kriptópolis es un ser vivo que sigue aspirando a su supervivencia. No tenemos vocación de mártires, ni de 'out-siders' ni de marginales. Aspiramos a seguir vivos e incluso (¡que atrevimiento!) a seguir creciendo. Por tanto, seguimos ofreciendo nuestro sitio como medio potencial de comunicar la existencia de productos o servicios (es decir, como soporte publicitario del mejor nivel), pero dejando claro que nunca podrán contar con nosotros para ser "la voz de su amo". Nuestra libertad de criterio -sencillamente- no está en venta.

Por tanto, permítasenos presumir de lo único que tenemos: libertad, independencia y... ¡nada menos que 10.000 suscriptores!

Gracias a todos por seguir apreciando como hasta ahora las virtudes (y miserias) que caracterizan nuestro sitio y lo diferencian de los demás.

Del Boletín #118:

GRAVÍSIMO NUEVO AGUJERO DE SEGURIDAD EN LOS SERVIDORES WEB MICROSOFT

El "Equipo de Seguridad Digital Ojo Electrónico" (eEye) soltó por fin la bomba: más del 90% de los servidores web bajo Windows NT pueden ser controlados por cualquier intruso sin más que explotar un grave defecto (recién descubierto por ellos) en el Internet Information Server 4.0 de Microsoft (IIS4).

A la gente de 'eEye' le bastó poner en marcha su propio scanner (denominado 'Retina', aún en fase beta) para buscar algo -en sí- no demasiado original: nuevos desbordamientos de buffer; es decir, defectos de programación que permiten que zonas de memoria intermedia se saturen ante un exceso de datos que no pueden asimilar, provocando la caída del sistema y/o su control remoto por parte de intrusos.

Como es sabido, el servidor IIS4 de Microsoft, reconoce unas determinadas extensiones de fichero que requieren ser procesadas en el propio servidor. Cuando un usuario demanda un fichero de uno de estos tipos (ASP, IDC, HTR...) se filtra hacia una biblioteca dinámica (un fichero DLL) que se encarga de manejarlos. Si la URL es suficientemente larga (y dado que IIS la pasa completa a la DLL), es muy probable que se logre sobresaturar algún buffer y que ocurra lo peor.

Pues bien; cuando 'Retina' llevaba una hora intentando lograr ese efecto con unas diez diferentes extensiones de fichero, una URL de unos 3000 caracteres de longitud hizo caer el servidor. La extensión testada en ese momento era HTR.

¿A qué responde esa extensión? Pues nada más y nada menos que a una funcionalidad de IIS4 que permite modificar las contraseñas de usuario de forma remota. Para complicar aún más las cosas, el filtro de las extensiones HTR (ISM.DLL) se instala siempre por defecto y

- -según afirma eEye- sitios como el Ejército USA, el NASDAQ y la propia Microsoft utilizan estos servidores. El desastre estaba -pues- servido.

Al parecer, el equipo de 'eEye' comunicó el hallazgo a Microsoft el pasado 8 de Junio, y la empresa de Redmond confirmó el problema y comenzó a trabajar en la solución. Microsoft no emitió un aviso público hasta el día 15 (justo una semana después), pero el parche seguía sin estar disponible en el momento de elaborar este artículo. Según la nota de Microsoft, el problema puede conducir a la caída de IIS4 o incluso permitir la ejecución de código arbitrario en el servidor si se utiliza una determinada cadena específicamente construida para el ataque. Según Microsoft, en ningún caso resulta manipulable la administración remota de contraseñas en sí misma.

Quizás para hacer más evidente la gravedad del problema y en pro - -afirman- de la libertad de información, 'eEye' adoptó (tras algún debate, según confiesan) la arriesgada decisión de hacer público también el código que permite aprovechar el agujero (IISHack).

¿Soluciones? Hasta que Microsoft elabore y difunda el parche correspondiente (algo que puede resultar extremadamente urgente en esta situación) no parece haber otra solución que eliminar la extensión HTR de la lista de extensiones a filtrar. Este es el método:

1) Arrancar IIS

2) Seleccionar Nombre Equipo - Propiedades - Servicio WWW - Editar - Directorio raíz - Configuración - seleccionar .HTR - Borrar - Confirmar.

La prestigiosa lista de seguridad NTBugtraq que dirige Russ Meyer, anunció hoy mismo la disponibilidad de un pequeño programa en VisualBasic, desarrollado por Nelson Bunker <nelson_bunker@wvm.net>, que automatiza el proceso de eliminar los filtros implicados en redes con gran número de máquinas. Una ayuda hasta la aparición del parche, considerada inminente.

* Descripción detallada:

<http://www.eeye.com/database/advisories/ad06081999/ad06081999.html>

* Información de Microsoft:

<http://www.microsoft.com/security/bulletins/ms99-019.asp>

* Aviso del CERT:

<http://www.cert.org/advisories/CA-99-07-IIS-Buffer-Overflow.html>

"COMERCIO ELECTRÓNICO EN INTERNET: ¿EL FUTURO TENDRÁ QUE ESPERAR?"

Por José Manuel Gómez
<jmg@kriptopolis.com>

Con independencia de si el crecimiento de la población internauta española se ha estancado o no, algo parece estar muy claro: a la mayoría de los recién incorporados parece que les ha traído a Internet el irresistible gancho de una especie de neo-televisión, postmoderna e hipertecnologizada, en la que abundan la música, el sexo y -cómo no- los anuncios de neón; una especie de gran hipermercado virtual donde los pantallazos de publicidad son algo inevitable, las promesas de privacidad un simple señuelo engañoso y la navegación arranca siempre de las (le pese a quien le pese) siempre estrechas estancias de cualquier megaportal al uso.

No hará falta recordar a los internautas más veteranos que hace menos de cuatro años, Microsoft lanzó su -entonces- sistema operativo estrella (Windows 95) sin apenas utilidades para Internet, que se veía por aquella época como una confusa y bucólica tierra de nadie, donde cuatro idealistas intercambiaban información, la mayoría de las veces (¿cabe mayor despropósito?) por el simple placer de hacerlo. Incluso el preclaro talento del profeta Gates no supo ver el filón que tenía justo delante de sus narices. Por la misma época (antediluviana, lo sé), el máximo dirigente en España de una multinacional cuyos productos ostentan (entonces como hoy) su breve anagrama en la mesa de la mayoría de nosotros (no de la mía, por fortuna), afirmaba en público -y con una media sonrisa irónica- que "Internet le recordaba al rastro de Madrid, un sitio ideal para perder el tiempo." Sobra decir que Microsoft parece hoy el creador de Internet y la multinacional que entonces dirigía tan brillante cerebrín, hoy comercializa sus productos por la Red sin ningún recato y considera Internet "un medio imprescindible" en sus planes estratégicos.

¿Qué pudo pasar para que Internet, cuyo uso con fines comerciales estuvo expresamente prohibido hasta principios de la década de los 90, se convirtiera -casi de repente- en el objetivo prioritario de los modernos 'buscadores de oro'? El origen de este cambio repentino de actitud coincide en el tiempo con el nacimiento de un nuevo mito denominado "comercio electrónico".

Este nuevo becerro de oro empezó pronto a exigir el sacrificio de

algunas cosas "sin importancia": primero cayó el placer de compartir por compartir (claramente incompatible con el nuevo culto, donde es fundamental que todo, absolutamente todo, tenga un precio); luego, y como toda nueva religión que se precie, empezó a exigir más adeptos con nombre y rostro ('cookies' para todo el mundo, ¡la casa invita!); por fin, y convenientemente revestido de abundantes regalos envenenados (buzones e-mail, espacio web, incluso ordenador o conexión gratis), el monstruo se atrevió a exigir el sacrificio máximo: el fin de la privacidad (dáme tus datos, díme tus gustos y yo decidiré lo que más te conviene comprar).

Hace ya casi un año, cuando el ciberpelotazo no estaba aún tan de moda, Kriptópolis recibió su primera oferta seria de compra, proveniente de una megaempresa cuyo interés en nuestro web nunca hubiéramos podido sospechar, de no ser porque uno de sus principales ejecutivos era un viejo amigo nuestro. Ante nuestra reticencia, se nos aseguró con convicción que "en muy poco tiempo, quien no esté dentro de un portal sencillamente no existirá". Es la misma proclama que esgrime cualquier secta: "quienes vengan a mí, serán salvos; el resto, al fuego eterno". Lo malo es que el tiempo está dando la razón a nuestro amigo. Muchos emblemáticos sitios web "de la vieja guardia", con una acreditada buena labor detrás, sólo aspiran a subsistir y no ven más solución de futuro que vender al mejor postor. Pero -en primer lugar- ¿por qué existen postores? ¿por qué se invierte tanto en Internet? ¿de dónde esperan sacar los futuros (que no actuales) beneficios? La respuesta es sencilla, ¿no la adivina? Pues del comercio electrónico que -cual esquivo mesías- algún día llegará. Pero... ¿de verdad llegará? ¿cuándo?

El caso es que, según sus muchos voceros y capitostes, ya tendría que haber llegado (hasta en eso se parecen a las sectas que anticipan el fin del mundo, una vez tras otra, error tras error, ridículo tras ridículo). Se nos anticiparon previsiones apabullantes y cifras astronómicas para antes de este fin de siglo-milenio, pero la fecha señalada se acerca y nada de lo anunciado ha ocurrido. ¿Dónde están ahora los profetas de antaño? ¿Qué pasa con SET, el protocolo-panacea? ¿Quién puede instalar hoy en España una tienda virtual con garantías? ¿Por qué se siguen esgrimiendo en los congresos estadísticas que casi nadie se cree? ¿Qué ha sido de tantos negocios electrónicos que se anunciaron a bombo y platillo y de los que nunca más se supo? ¿Cuántos españoles compran con cierta regularidad (o incluso una sola vez) en Internet? ¿Qué bancos pueden prestar los servicios que anuncian? ¿Han bajado acaso las abultadas comisiones con que los medios de pago castigan a los comerciantes? ¿Algún comerciante nos ofrece algún descuento significativo por comprar vía Internet? ¿Con qué medios de reparto se cuenta, cuál es su coste añadido y su eficiencia real? ¿Se fía el cliente de una tienda virtual? ¿y la tienda del cliente? ¿Nos

podemos fiar todos de las míseras claves de 40 bits de nuestros navegadores? ¿A dónde fueron los brillantes sistemas de dinero digital? ¿Puede despegar el comercio electrónico con tarifas telefónicas y "calidades" de conexión como las que soportamos? ¿Tienen algo que ver los hábitos de compra del consumidor español con los del norteamericano? ¿Cuántos años se necesitarán para que la aldea global se convierta en el hipermercado global? ¿Se imagina usted algún día haciendo las compras más habituales vía Internet?

Son demasiadas preguntas, que poca gente se atreve a formular (es incómodo e impopular) y mucho menos a responder (es imposible). Entre las empresas que aspiran a vender se palpan el desánimo y la frustración; entre los posibles clientes, el escepticismo. ¿Somos nosotros demasiado pesimistas? Bien; para que no se nos tache de agoreros, acudiremos ahora a la opinión de una tercera parte muy poco sospechosa: la de 105 ejecutivos norteamericanos del máximo nivel, ejerciendo en empresas informáticas de primera línea (un alto porcentaje de ellos como presidente o vicepresidente). El trabajo acaba de publicarse (ver dirección más abajo) y ha sido elaborado por ITAA (Information Technology Association of America), en colaboración con Ernst&Young. En la encuesta, se investigan los posibles factores de riesgo que pueden frenar el esperado desarrollo del comercio electrónico como factor vitalizador de la vida económica de los Estados Unidos y del resto del mundo:

<http://www.ita.org/software/research/indpulse/bartext.htm>

Pues bien; la falta de confianza fue elegida como el primer obstáculo nada menos que por el 63% de los ejecutivos. El 45% señaló la falta de referencias claras a la hora de abordar el tema (por falta de acuerdo, educación o conocimiento de las empresas). El 36% señaló obstáculos relativos a los procedimientos de negocio ahora existentes. Diferentes porcentajes de ejecutivos apuntaron a otros factores diversos. Destacar que tan sólo un 1% no encontró ningún obstáculo a la prometida expansión ilimitada del comercio electrónico (ojo: tan sólo un 1%, en el país de las llamadas locales gratuitas y la pasión compulsiva por comprar).

Al profundizar en detalle en el principal obstáculo apuntado (la falta de confianza), un 60% señaló la preocupación por la pérdida de privacidad, un 56% a dificultades de autenticación entre clientes y empresas y otro 56% se inclinó por el miedo a la falta de seguridad de las transacciones (temor a que la infraestructura técnica no sea aún suficientemente robusta como para impedir ataques de intrusos). En general, todos estos factores le parecían muy importantes al 44%, moderadamente importantes al 42% y poco o nada importantes al 13%. Más

adelante en el trabajo, al considerar las posibles barreras de tipo técnico, vuelve aparecer la preocupación por la seguridad en un 57% de encuestados. Curiosamente, el 45% de los ejecutivos considera un impedimento grave las restricciones al uso de cifrado que impone el gobierno de EEUU.

Las conclusiones del estudio son bastante claras:

- 1) El comercio electrónico se encuentra aún en estado relativamente naciente.
- 2) Existen aún muchos obstáculos que vencer para que el prometido boom del comercio electrónico sea -por fin- una realidad.
- 3) La mayoría de los inconvenientes afectan a cuestiones de confianza y seguridad.
- 4) Se requiere un abordaje pluridisciplinar de estos problemas, donde consumidores, empresas, industria y gobierno tendrán por delante muchos puntos que trabajar.

Y no conviene perder la perspectiva: el estudio procede de un entorno social muy diferente al nuestro, donde el comercio electrónico disfruta ya de un respetable auge. En nuestro medio es de prever que las perspectivas serían bastante peores, con el agravante de que no se toman medidas valientes para abordar la cuestión en casi ninguno de sus frentes.

Ahora es usted, amable lector, quien tiene la palabra. Le esperamos en nuestro foro con el siguiente tema de debate:

¿QUÉ IMPIDE QUE EL COMERCIO ELECTRÓNICO SE CONVIERTA EN UNA REALIDAD COTIDIANA EN NUESTRO ENTORNO?

<http://www.kriptopolis.com/wwwboard/wwwboard.html>

Del Boletín #121:

UN PROFESOR ESPAÑOL PUBLICA INTEGRO EN LA RED SU LIBRO "CRİPTOGRAFÍA Y SEGURIDAD EN COMPUTADORES"

INTERNET SIGUE VIVA

Andábamos en Kriptópolis preguntándonos qué queda aún del espíritu original de Internet (frente la "fiebre del oro" de los portales y el ciberpelotazo), cuando un sorprendente correo vino a sacarnos de nuestras cuitas.

Verán; Manolo Lucena es un joven y activo profesor de Informática, que ejerce la docencia en la Universidad de Jaén (España), donde es además miembro del Grupo de Análisis de Imágenes Digitales y más que prometedor investigador en el campo de la Visión Artificial. Ya desde niño (y como tantos de nosotros), Manolo se divertía elaborando códigos secretos, pero esta precoz afición fue -en su caso- bastante más allá. Con tan sólo quince años de edad, un Commodore 64 le permitió elaborar su primer algoritmo criptográfico "serio". Desde entonces, ha sido necesario bastante tiempo de estudio e investigación hasta convertirse en un competente profesor de la asignatura de "Criptografía y Seguridad" en la Escuela Politécnica Superior de Jaén.

Y es que, por encima de todo, lo que le gusta a Manolo es compartir sus conocimientos con los demás. Así, ya en 1995 (y junto a otros compañeros), elaboró una colección de apuntes de criptografía que es posible encontrar aún en la biblioteca de muchas universidades. No obstante (y como dice Manolo en la introducción de su nueva obra, que hoy presentamos a nuestros lectores), "como cabía esperar en una disciplina de tan rápida evolución, las cosas han cambiado."

Precisamente por ello, Manolo ha decidido escribir un nuevo libro, completamente actualizado (y que pretende remozar además cada año). Pero lo más sorprendente de esta nueva obra es su forma de distribución: el libro puede distribuirse **LIBRE Y GRATUITAMENTE** bajo cualquier soporte, siempre que se respete su integridad.

La generosa actitud de Manolo Lucena es la muestra palpable de que (oculta bajo las toneladas de basura de la insípida Red de neón que algunos pretenden construirnos a nuestro pesar), continúa viva e ímpolita el alma original de este medio: la voluntad de compartir el conocimiento con nuestros semejantes sin trabas ni cortapisas, y que sin duda constituye el elemento más revolucionario de una Internet que nunca se convertirá (al menos por completo) en otro banal electrodoméstico al uso.

LA OBRA

El libro "Criptografía y Seguridad en Computadores" tiene 119 páginas y se distribuye gratis en formato electrónico PDF (quien aún

no tenga lector, puede obtenerlo también gratis en la web de Adobe).

El libro se ha concebido como texto de apoyo para estudiantes de la asignatura en tercer curso de Ingeniería Técnica Informática de Gestión en la Universidad de Jaén. No obstante, nos permitimos aconsejarlo a cualquier tipo de lector con ciertos conocimientos técnicos e interés por la criptografía. Incluso nos atrevemos a afirmar que no es necesario disponer de ningún bagaje científico para poder disfrutar de bastantes capítulos y hacerse una idea general del mundo de la Criptografía.

Un índice muy abreviado de capítulos podría ser el siguiente:

I. Preliminares

1. Introducción
2. Conceptos básicos sobre Criptografía

II. Fundamentos teóricos de la Criptografía

3. Teoría de la Información
4. Fundamentos de Aritmética Modular
5. Aritmética Entera de Múltiple Precisión
6. Criptografía y Números Aleatorios

III. Criptografía de Llave Privada

7. Criptografía Clásica
8. Algoritmos Simétricos de Cifrado

IV. Criptografía de Llave Pública

9. Algoritmos Asimétricos de Cifrado
10. Métodos de Autenticación
11. PGP

V. Seguridad en Redes de Computadores

12. Seguridad en Redes

UN LIBRO VIVO

En el mejor espíritu de la Red, el libro pretende no ser nunca una obra definitiva y seguir evolucionando gracias a las aportaciones de todos. El propio autor nos dice: "Mi intención exacta con el libro es sacar una edición cada año, con todas las correcciones, ampliaciones y modificaciones que estime oportunas en función de las opiniones y posibles colaboraciones de los lectores. Por supuesto que todas aquellas personas que aporten algo importante serán incluidas en un apartado de colaboraciones. La única condición es que esas aportaciones sean sin ánimo de lucro."

Dada la envergadura de esta obra, los planes previstos para su incesante mejora y su generosa (aunque inteligente) política de distribución, así como la notable carencia de obras tan accesibles en español, desde Kriptópolis nos atrevemos a conjeturar que este libro va a experimentar una difusión sin precedentes en toda España y Latinoamérica, y a nosotros nos ha correspondido esta vez el honor de alumbrarla y darla a conocer.

Gracias a Manuel Lucena en nombre de todos nuestros lectores y de la comunidad de habla española en general.

OBTENER EL LIBRO

El libro completo, sin ningún tipo de restricciones o cortapisas, puede ser descargado **TOTALMENTE GRATIS** desde:

<http://www.kriptopolis.com/criptografia.zip>

El fichero **CRIPTOGRAFIA.ZIP** (473 KB) incluye los cinco ficheros siguientes:

* **CRIPTOGRAFIA.PDF**:

El libro completo, en formato PDF de Adobe.

* **CRIPTOGRAFIA_DSS.PDF.SIG** y **CRIPTOGRAFIA_RSA.PDF.SIG**:

Las firmas RSA y DSS de su autor (que permiten comprobar la autenticidad del libro).

* **MANUEL LUCENA DSS.ASC** Y **MANUEL LUCENA RSA.ASC**:

Las claves públicas RSA y DH/DSS del autor (que permiten verificar la autenticidad de los ficheros de firmas), firmadas

además con las propias claves de Kriptópolis como medida adicional de seguridad.

Las **PERSONAS QUE *NO* UTILIZAN PGP** pueden descomprimir el fichero, quedarse con el fichero **CRIPTOGRAFIA.PDF** y borrar los otros cuatro, pero no podrán nunca verificar que se trata del ejemplar original, sin ninguna manipulación posterior. Este aspecto puede parecer absurdo ahora que el fichero tiene un claro origen en el web de Kriptópolis, pero empezará a cobrar importancia cuando comiencen a proliferar copias (unas auténticas, otras quizás no) en la Red. Tienen aquí un buen ejemplo de la ventaja que supone usar PGP a la hora de poder comprobar firmas digitales.

INSTRUCCIONES ADICIONALES PARA LECTORES QUE UTILIZAN PGP

Los usuarios de PGP disponen de la ventaja adicional que supone verificar las firmas digitales del fichero y poder así comprobar que se trata del libro original, tal como lo editó su autor. Para ello deberán:

- 1) Importar los ficheros ASC desde PGP y verificar la existencia de las firmas de Kriptópolis en ellos. Comprobar la validez de la firma de Kriptópolis utilizando nuestras propias claves (disponibles en nuestro web).
- 2) Una vez seguros de lo anterior, firmar ambas claves ASC de Manuel Lucena con la propia clave de cada uno, para otorgarles validez.
- 3) A continuación, pulsar dos veces sobre los ficheros SIG y seleccionar después el fichero CRIPTOGRAFIA.PDF. Si el fichero verifica (firma válida) el usuario puede estar seguro de que se trata del libro original.

UNA NUEVA DISPOSICIÓN LEGAL REGULA LA EXPORTACIÓN DE CIFRADO DESDE ESPAÑA

Por Xavier Ribas (jribas@ibm.net)

Desde el mes de abril de 1999 está en vigor un nuevo régimen para la exportación de tecnologías de doble uso. El apartado que afecta a la criptografía deja fuera del control estatal los productos que cumplan los siguientes requisitos:

- a. Que se hallen generalmente a disposición del público por estar a la venta, sin restricciones, en puntos de venta al por menor por cualquiera de los medios siguientes:
 1. Transacciones en mostrador;
 2. Transacciones por correo;
 3. Transacciones electrónicas; o
 4. Transacciones por teléfono.
- b. Que la función de cifrado no pueda ser modificada fácilmente por el usuario;
- c. Que estén concebidos para que el usuario los instale sin asistencia

ulterior importante del proveedor;

d. Que no contengan un "algoritmo simétrico" que utilice una longitud de clave superior a 64 bits; y

e. Que, en caso necesario, pueda disponerse de información detallada sobre los productos y se facilite ésta, cuando se solicite, a las autoridades competentes del Estado miembro en el que esté establecido el exportador, con el fin de verificar el cumplimiento de las condiciones descritas en las letras a. a d. anteriores.

El texto completo de esta Resolución puede ser consultado en:
<http://www.onnet.es/03005009.htm>

ALEPH

Por Carlos Sánchez Almeida (shooting@intercom.es)

No existe una conciencia real de qué es exactamente lo que se está construyendo a diario, gracias a Internet. Sólo podemos intuirlo si observamos cómo se está generando un orden a partir del caos. La Red fue pensada para que la comunicación fluyese libremente, incluso en las condiciones más adversas. Ello hace ardua e infantil cualquier acción dirigida a constreñir sus límites y la libertad que crece en sus páginas.

Los gobiernos y los grandes monopolios de la información y la comunicación han conseguido imponer en todos los restantes medios su mensaje unidireccional. En la Red es especialmente difícil establecer jerarquías, pese a la obsesión de algunos por crear portales en un espacio donde no hay paredes. Cuando se ha leído lo suficiente para escoger lo que uno quiere seguir leyendo, llega un momento que las personas inteligentes deciden apagar el televisor y las radios, y acaban por cerrar los periódicos y los libros donde sólo escriben aquellos que han pasado las pruebas de iniciación.

Las élites de la nueva generación saben ya que la verdad está en la Red, no ahí fuera. Y eso da mucho miedo a aquellos que han diseñado una estructura para controlar las ideas. Nadie puede controlar el caos, ni tan siquiera entenderlo: hay que empezar a asumirlo como algo consustancial a nuestra existencia.

Hablando del comercio electrónico, reflexionaba que no es posible la libertad ni el crecimiento económico dentro de la Red si no se

consolidan primero las libertades políticas, los ciberderechos, que incluyen necesariamente la libertad de expresión, de comunicación y el derecho a la privacidad mediante la criptografía. Hoy añado que sólo será posible eliminar las barreras al libre comercio internacional, dentro de la Red, si primero eliminamos cualquier frontera electrónica.

Para posibilitar el acceso sin pasaporte a la Polis Global, debemos suprimir las trabas al libre intercambio de las ideas. Las económicas serán las más difíciles, porque se hace imposible poder expresarse si no se ha podido acceder previamente a la educación y a los medios técnicos de acceso a la Red. Pero hay una frontera que ha de caer mucho antes: el idioma. Cuando las herramientas de traducción en tiempo real comiencen a implantarse, Babel habrá dejado de ser una maldición bíblica. Es posible un futuro donde todos se expresen en el idioma que libremente escojan, porque los receptores del mensaje lo leerán en su propia lengua.

Quien haya leído El Aleph puede tener una pequeña intuición de lo que representaría para la Humanidad el acceso a una ventana desde la que se puede contemplar todo nuestro Universo. Sólo son ciegos los que no quieren ver, y pretenden cegarnos a todos. Es curioso que fuese precisamente un ciego el que nos diese la clave para entender nuestro pequeño Cosmos. Les dejo con Borges.

"No pertenezco a ningún partido político y no he hecho política activa. Quizá yo sea un tranquilo, silencioso anarquista, que sueña en su casa con que desaparezcan los gobiernos. Descreo de las fronteras, y también de los países, ese mito tan peligroso. Sé que existen y espero que desaparezcan las diferencias angustiosas en el reparto de la riqueza. Ojalá alguna vez tengamos un mundo sin fronteras y sin injusticias".

<http://www.bufetalmeida.com>

BUZÓN KRIPTÓPOLIS

* "Comercio Electrónico en Argentina"
Victor Hugo Vergel (vicverg@coop-tortu.com.ar):

"Hace muchos años que me dedico a esto (la computación), y tengo contacto con cientos de personas (por desgracia no todos clientes), que le tienen terror a Internet, más precisamente al comercio

electrónico. Y no por histeria o historias colectivas, sino que por hechos concretos. Vaya como ejemplo la aparición en sus resúmenes de tarjetas de crédito de las temidas iniciales "DMR" más un valor "agregado" por algo nunca consumido y no inferior a los u\$s 29,99.

Concretamente, quienes reciben estos "débitos" son internautas primerizos (y no tanto), que han ingresado su tarjeta para entrar, por lo general, a algún sitio porno. Esos donde con un "Join Now Free Trial", "Membership Free Today" o algún otro sucio gancho por el estilo, nos invitan a ver fotos, dibujos o videos horrorosos y entrecortados de gente dándonos lecciones de posiciones para reproducir más gente. Por supuesto que el gancho muestra hermosas mujeres jamás soñadas (algo así como en los Mc Donalds, donde la hamburguesa de la foto se parece tanto a la que te dan, como el huevo a la gallina). Se podría decir que esto ya configura una cierta estafa, pero lo peor todavía no llega hasta que el resumen de la tarjeta muestra la debilidad (debilidad ?) del "secure server". Pero ya es tarde; los cargos vienen duplicados: uno cierto y reconocido por el cliente y otro duplicado -de vaya a saber qué origen- por un monto (ruegan los usuarios) no menor al mencionado. Cientos de llamadas al administrador de la tarjeta para anular el pago, tiempo perdido en reclamos, e-mails al bendito "DMR" y sin resultados. Final: pedido al banco emisor de la tarjeta para que la anule y vuelta a empezar para gestionar otra. Debo aclarar que pongo como ejemplo a "DMR" porque es sobre el que más me han consultado pero supongo que existirán otros cientos de "DMRs". Menos casos se detectan en compras de libros, hardware y varios para el hogar. Leí recientemente que alguien que tenía un sitio porno en la red, goza de un depósito de u\$s 49,4 millones en un banco de las islas Caimán, gracias a este método."

Del Boletin #122:

NUESTRO TEMA DE DEBATE DE LA SEMANA: "EFECTO 2000, ¿REALIDAD O FANTASÍA"

Por Víctor Hugo Vergel (vicverg@coop-tortu.com.ar)

A estas alturas, ya no puedo decir que no estoy informado sobre el asunto, ni que la falta de preparación apura mis conclusiones. Lo cierto es que todo este rollo sobre el Efecto 2000 (Y2K) me tiene

bastante harto.

Este fin de semana, aprovechando las horas "baratas" de Telefónica, me bajé todos los testeadores share y "oficiales" de Intel (NSTL), AMI, Award (hoy Phoenix)..., más la documentación con que avalan las pruebas.

Y aquí está el problema, toda la documentación leída no hizo más que reforzar mi teoría sobre el gran circo armado en torno a este tema. No digo que alguna máquina vieja y/o programa antiguo no vaya a fallar, pero de ahí a afirmar que el 1/1/2000 todo va a quedar patas arriba, me parece una enormidad fabulera, vaya a saber con qué oscuros designios.

Lo publicado en los sitios parece calcado de uno a otro; son casi copias textuales, con una vaguedad tal que hasta da vergüenza ajena pensar en que estos tipos son los que manejan el segundo corazón de las máquinas y toneladas de dólares.

Los programas (programitas), de testeo hacen lo mismo que nosotros podríamos hacer adelantando y atrasando el reloj de la máquina desde el Setup. AMI dice desde su página de Y2K que todos sus BIOS son "Year 2000 compliant" desde el año 1995; Award (Phoenix), le mata el tanto afirmando que los suyos lo son desde el año 1994, Intel desde el 486 o te manda a NSTL para bajar el tester y confirmarlo. Cyrix e IBM tampoco se quedan atrás. Encima, no falta el aprovechado en río revuelto, que te ofrece un test shareware y aunque pruebes una Pentium III 550 con un QDI Brilliant de última generación o un BX Intel, te da error en algún punto, y termina ofreciéndote un "Fix" por un valor de X dólares.

Hasta el momento llevo probadas 320 máquinas, desde 486 en adelante, y ninguna falló; ni tampoco los programas. Tengo funcionando una 486 DX 4/100 de AMD hace ya un mes y medio con el reloj en el 2000; lo paso algunos días al 2038 (ahora inventaron esto por si les falla el primero), y sigue sin errores. Pero Intel dice que hasta los ABS de los autos podrían fallar...(????)

Hay encuestas que dicen que el 80% de las personas que habitan este mundo son taradas y si no fuera por la publicidad no sabrían qué comer ni con qué vestirse, que el 80% de los médicos aprenden medicina de los visitantes médicos y que el 80% (casualidad?) opina exactamente lo que opinan sus diarios, revistas y programas de TV preferidos. ¿Pertenece al 20% restante?

----- Participe en nuestro Debate sobre este tema -----

<http://www.kriptopolis.com/wwwboard/wwwboard.html>

NUEVA EDICIÓN DE "APLICACIONES CRIPTOGRÁFICAS"

Acaba de salir a la venta en el Departamento de Publicaciones de la Escuela Universitaria de Informática de la UPM (Universidad Politécnica de Madrid), la segunda edición del libro Aplicaciones Criptográficas del profesor Dr. Jorge Ramió Aguirre.

Creado con el objeto de ser un Texto Guía para los alumnos de la asignatura de Seguridad Informática de la EUI-UPM -y también para cualquier lector interesado en estos temas sin formación universitaria, dado el enfoque del mismo- esta segunda edición corregida, aumentada y actualizada de la primera de enero de 1998, contempla los siguientes capítulos:

- Capítulo 1. Introducción a la Seguridad Informática
- Capítulo 2. Fundamentos de la Seguridad Informática ·
- Capítulo 3. Criptosistemas Clásicos
- Capítulo 4. Introducción a los Criptosistemas Modernos
- Capítulo 5. Cifrado en Flujo con Clave Secreta
- Capítulo 6. Cifrado en Bloque con Clave Secreta
- Capítulo 7. Cifrado con Mochilas de Clave Pública
- Capítulo 8. Cifrado Exponencial
- Capítulo 9. Integridad y Autenticidad
- Capítulo 10. Protocolos Criptográficos
- Capítulo 11. Correo Electrónico Seguro

Se incluye un Anexo con tablas comunes para la resolución de ejercicios y un apartado con una amplia bibliografía de consulta, destacando los diez títulos más importantes a juicio del autor con una breve reseña de cada texto en particular, entre ellos cuatro españoles. Además, entre sus 11 capítulos escritos con un lenguaje claro y ameno (y en algunos casos desenfadado para dar una mayor frescura a su lectura), el lector encontrará 225 ejemplos con sus correspondientes soluciones.

Aplicaciones Criptográficas. Segunda edición (junio de 1999)

437 páginas

I.S.B.N.: 84-87238-57-2

Departamento de Publicaciones de la Escuela Universitaria de Informática de Madrid, bajo el patrocinio de la Fundación General de la Universidad Politécnica de Madrid

PRIMERA FASE DE LA CREACIÓN DE UNA RED TEMÁTICA HISPANOAMERICANA EN CRIPTOGRAFÍA

Durante este mes de julio se encuentra en Chile el profesor de la Universidad Politécnica de Madrid Dr. Jorge Ramió Aguirre con dos objetivos básicos. Primero, dar a conocer en distintas universidades de ese país los resultados de la encuesta sobre la Enseñanza de la Criptografía realizada recientemente en España, recogidos en un primer Informe publicado en el número de abril de este año de la Revista SIC, Seguridad en Informática y Comunicaciones, y que ya hemos comentado en un número anterior de Kriptópolis. Segundo, echar a andar una Red Temática en Criptografía con un objetivo claramente docente, que permita e incentive el intercambio de información de temas relacionados con la seguridad informática entre profesores universitarios e investigadores de todos los países latinoamericanos y España.

El primer paso es precisamente este contacto con profesores de la Universidad de Chile, la Universidad de Santiago de Chile, la Universidad Diego Portales y la Universidad de Ciencias de la Informática, las cuatro en la capital de dicho país, y la Universidad de Tarapacá en Arica, ciudad limítrofe con Perú y Bolivia. Tras estas entrevistas, la primera tarea que se pretende desarrollar simultáneamente con la creación de dicha red, es la realización de un estudio sobre la Enseñanza de la Criptografía en las universidades de estos países, comenzando en este caso con el país más alejado geográficamente de España como es Chile, similar al que entregó hace algunos meses unos interesantes datos sobre el espectacular crecimiento de la oferta de este tipo de asignaturas en las universidades de nuestro país en los últimos tres años.

Durante su estadía en Chile, del 8 al 23 de julio, aquellos profesores del cono sur interesados en la enseñanza de la criptografía, o bien específicamente en la Red Temática, y que deseen participar en este interesante proyecto docente, pueden comunicarse con el profesor Ramió en la dirección de correo jramio@visviri.electa.uta.cl y, en todo caso, en su dirección habitual en el Departamento de Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática de la UPM (jramio@eui.upm.es).

Del Boletín #123:

BACK ORIFICE 2000 Y OTRAS SERPIENTES DE VERANO

Tal y como anunciamos la semana pasada, y en medio de una inusitada expectación mediática, "El Culto de la Vaca Muerta" (ver entrada correspondiente de nuestro diccionario en este mismo boletín), presentó el sábado en Las Vegas su nuevo Back Orifice 2000 (BO2K), la segunda generación de su conocido troyano.

Al igual que el primitivo Back Orifice, BO2K permite a un extraño tomar el control total de una máquina remota pero, con respecto a su antecesor, BO2K incorpora novedades importantes; por decirlo en pocas palabras: son posibles más maldades y éstas pueden configurarse y encubrirse mejor. Pero, sobre todo, BO2K amplía su campo de operaciones a ordenadores gobernados por Windows NT, que eran inmunes al primitivo BO.

No parece necesario insistir en lo grave que puede resultar una inoculación de BO2K en el ordenador de la víctima: éste se convierte en una marioneta en manos del intruso. Sus ficheros pueden ser borrados o robados, pueden ejecutarse o detenerse procesos, cambiar el registro, robar contraseñas, e incluso tomarle el pelo a través de sus propios altavoces.

¿Cuáles son -pues- las buenas noticias? Pues fundamentalmente que **BO2K SIGUE SIENDO UN TROYANO**, y un troyano no puede instalarse sin la ayuda (involuntaria, claro) de su propia víctima. Para poner las cosas en su sitio exacto, hay que decir bien claro que BO2K no aprovecha ninguna debilidad intrínseca del sistema operativo, ni ningún otro fallo desconocido hasta ahora. Por tanto, BO2K no introduce un "antes" y un "después" en el campo de la seguridad informática. Es un troyano más y punto.

Por tanto, si usted sigue las más elementales normas de seguridad (fundamentalmente no ejecutar ficheros de origen desconocido o dudoso, aunque vengan atacheados a un e-mail), y si usted utiliza un **ANTIVIRUS TOTAL Y PERMANENTEMENTE ACTUALIZADO**, sencillamente no debe temer a BO2K. De hecho, aunque los creadores anunciaron desde el principio la publicación del código fuente de BO2K (hecho que acaba de consumarse y supone un riesgo añadido, puesto que facilitará la creación de variantes), las principales casas de antivirus ya habían actualizado sus ficheros de firmas frente a BO2K antes de que ese código se hiciera público.

En resumen: no deje que le asusten. Todos los medios informativos suelen andar faltos de noticias en verano, y el tema virus siempre vende, porque se presta como ningún otro al amarillismo y la paranoia al uso.

Afortunadamente, siempre hay excepciones. Ante la solicitud de información por parte de clientes aterrorizados por la alarma creada, AVP-España (<http://www.avp-es.com> , distribuidores del famoso antivirus AVP) decidió emitir un boletín informativo especial, del que sólo entresacaremos algunos párrafos, aunque nos cuesta resistir la tentación de reproducirlo en su totalidad, como ejemplo palmario de una actitud responsable y alejada del sensacionalismo que tantos otros debieran aprender e imitar. En ese boletín se decía (entre otras cosas) lo siguiente:

----- Boletin AVP-España -----<

[BackOrifice 2000. A bombo y platillo]

"Es curioso lo que ocurre, cada día más y más, en el mundillo de los virus. Se aprovecha cualquier ocasión para publicitar un producto, para lanzar alarmas que no siempre son consecuentes con la determinada peligrosidad de un virus, pero solo para "acojonar" a la gente y que compren mas y mas programas antivirus, no ya por protección, sino por puro miedo."

(---)

"Mensaje interno de Eugene Kaspersky, el creador y máximo representante del AVP, a sus distribuidores, acerca de este virus:

"Bueno, pues ya ha aparecido una nueva versión del BackOrifice. Están haciendo mucho ruido acerca de esto. Por lo que yo he visto, podría ser solamente una línea más en el listado de virus de las actualizaciones semanales (hay varios virus "BackDoor" similares a este cada semana). Pero concretamente este está siendo "bien anunciado". Por tanto, si necesitais actualización urgente para los clientes paranoicos, aquí os envío la actualización correspondiente. Eugene Kaspersky."

(---)

"La próxima vez que anunciemos un virus, será porque tenga verdadero peligro, no para hacer publicidad ni del AVP, ni de la lista."

[NOTA: Para quienes acostumbran a ver fantasmas por todas partes, nada liga a AVP (ni a ninguna otra firma) a Kriptópolis. Este ejemplo ha sido traído a colación porque nos ha parecido muy ilustrativo de

cómo pueden abordarse las cosas de otra manera. La lectura de ese boletín completo es altamente recomendable.]

-----<

No hay mucho más que decir al respecto; TOME SUS PRECAUCIONES (eso sí) y disfrute del verano sin hacer demasiado caso a los agoreros, que parece gozar agitadamente interesadamente las aguas con la varita de su irresponsabilidad.

"Responsabilidad en la información sobre seguridad. ¿Dónde están los límites?"

Es muy fácil sentirse tentado a proclamar -sin más- que la información no debe tener ningún límite, pero en este artículo intentaremos demostrar que tales límites pueden (y quizás también deben) existir. Tampoco esta vez aspiramos a sentar cátedra, pero sí que hablamos desde nuestra experiencia de varios años en este difícil y confuso mundo de la seguridad, donde casi nada es lo que parece y uno se juega su reputación (y quizás algo más) en cada palabra que dice o escribe.

Si la información sobre seguridad ha de ser responsable (y en Kriptópolis pensamos que siempre debe serlo), puede haber dos tipos de frenos a la hora de decidir publicar o no una determinada información (damos aquí por supuesto que la información de que hablamos es CIERTA, DEMOSTRABLE Y VERIFICABLE; no hay dudas en eso). En primer lugar, puede existir un FRENO ÉTICO (por ejemplo, cuando la publicación del hecho pudiera tener consecuencias negativas sobre los usuarios más desprevenidos); en segundo lugar, puede haber un FRENO LEGAL (el hecho puede tener consecuencias o sugerir conductas contrarias a las leyes).

Como cualquiera puede suponer, pocas veces las cosas son claras en estos aspectos. De hecho, cuando el freno legal o ético es evidente se opta siempre por no publicar la noticia. ¿Pero qué hacer cuando nos movemos en el terreno de la duda? (algo que -por cierto- ocurre con demasiada frecuencia en temas tan delicados).

Veamos; en lugar de teorizar, plantearemos tres supuestos concretos, tres ejemplos más o menos utópicos, pero suficientemente paradigmáticos, de las difíciles decisiones que en Kriptópolis debemos tomar casi a diario.

* SUPUESTO 1)

"Todo el mundo conoce que Netscape Navigator puede modificarse (mediante Fortify) para utilizar claves de 128 bits en los servidores seguros, infinitamente más seguras que las habituales de 40 bits. Sin embargo, la inmensa mayoría de la gente no sabe que otro tanto puede hacerse (y más fácilmente) con Internet Explorer. ¿Debe divulgarse cómo?"

Observemos nuestros dos frenos:

1) ÉTICO: Ninguna duda al respecto. ¿A quién se perjudica haciendo los navegadores más seguros? ¡Únicamente a quienes aspiren a robarnos los datos de nuestra tarjeta de crédito!.

2) LEGAL: Dudas. ¿Es lícito modificar un programa mediante otro que se supone no debiera haber salido de Estados Unidos?

* SUPUESTO 2)

"Una nueva versión de PGP sale en Estados Unidos. A pesar de la prohibición de exportación desde su país de origen, lo cierto es que ese mismo día ya está disponible en servidores europeos. ¿Debe informarse dónde?."

1) ÉTICO: Si el programa es freeware no se haría mal a nadie revelando su ubicación, pero habitualmente en el mismo directorio se halla también la versión comercial completa, lo que conduciría a vulnerar los derechos de los propietarios del software.

2) LEGAL: No hemos logrado jamás un consejo legal concluyente sobre la situación de programas de criptografía freeware exportados ilegalmente. Las peculiaridades de las leyes norteamericanas parecen más difíciles de "exportar" a nuestro medio que los propios programas. Es evidente que ha existido un delito en la exportación, pero ha sido contra las leyes norteamericanas, y con grandes dificultades para detectar al infractor y procesarlo (en función de que sea norteamericano o no). Parece un complicado asunto de Derecho Internacional en el que preferimos no intentar poner el cascabel al gato.

* SUPUESTO 3)

"Supongamos que Kriptópolis tiene conocimiento de un grave defecto de diseño de los navegadores que deja a los usuarios expuestos a gravísimos riesgos contra su honor. Supongamos que los fabricantes han

sido informados y han reconocido el hecho, pero no tienen ninguna prisa en repararlo hasta la próxima versión prevista. ¿Deben publicarse los detalles del problema?."

1) ÉTICO: Por un lado, el usuario tiene derecho a conocer cuanto antes el grave riesgo que corre y lo poco que parece preocupar esa circunstancia a los fabricantes. Por otro, nadie en el mundo conoce aún cómo explotar el problema y el hecho de publicarlo puede poner sobre la pista a todo tipo de maleantes, que podrían proceder de inmediato a perpetrar todo tipo de fechorías.

2) LEGAL: Decir la verdad y estar en condiciones de poder demostrarla no debería conducir a intentar matar al mensajero, pero lo cierto es que algunos podrían pedir nuestras cabezas. Hay un precedente reciente que levantó una gran polémica: el caso de 'eEye' versus 'IIS 4' (ver boletín 118 de Kriptópolis).

Con estos tres ejemplos creemos haber planteado la envergadura del tema que queremos proponer para el debate de esta semana. Se trata de un tema que atañe a la propia singularidad de Kriptópolis, dado que nuestro sitio existe -precisamente- por el estímulo que nos proporciona esta constante tensión dialéctica y creativa. Parece obvio que si estuviéramos plenamente conformes con el tratamiento que otros medios dispensan a las noticias sobre seguridad informática, sencillamente no existiríamos. Es esa constante búsqueda de una información atinada y responsable la que determina nuestra propia existencia y muchas de nuestras actitudes (por ejemplo: esta delicada libertad de criterio nos parece mucho más difícil de mantener cuando se está al servicio de según qué "amos").

En definitiva, amable lector: le convocamos a nuestro foro de esta semana para que nos proporcione su opinión sobre este apasionante tema, que nos toca -además- tan de cerca:

"Responsabilidad en la información sobre seguridad. ¿Dónde están los límites?"

Le esperamos en <http://www.kriptopolis.com/wwwboard/wwwboard.html>

Del Boletín #125:

MAS SOBRE EL FRAUDE POR E-MAIL

Kriptópolis destapó otra vez la caja de los truenos. Nunca antes hubiéramos pensado que nuestra labor en pro de la seguridad de los internautas requiriera algún día que tuviéramos que intentar abortar una estafa en marcha, pero al fin y al cabo la seguridad en Internet comienza por la seguridad del propio bolsillo del usuario.

El caso es que, desde que en la noche del martes 20 revelamos la operación de 'spam' fraudulento que parecía haberse iniciado ese mismo día, empezaron a llegarnos mensajes confirmando nuestras sospechas.

Así, algunos suscriptores afirmaron haber recibido hasta cuatro copias del mensaje en cuestión a lo largo del día, lo que les hizo sospechar de inmediato, y les previno de hacer la llamada fatal que consumaba el timo; sin embargo, y por desgracia, para algunos otros nuestro aviso llegó demasiado tarde y deberán afrontar en sus próximas facturas el cargo por una llamada innecesaria a Chile. Parece que incluso hubo personas que decidieron cancelar todas sus tarjetas de crédito, en un intento de evitar el desastre. También tuvimos que tranquilizar a quienes pensaban que el mero hecho de haber abierto el mensaje ya les perjudicaba de alguna forma. Otros comunicantes aún parecían estar bajo el 'síndrome Melissa/Happy99' y afirmaban que la infausta dirección de marras se había colocado inadvertidamente en su libreta de direcciones. No parece el caso...

Los timadores no tienen muchos motivos para sentirse satisfechos de su acción. Aparte de topar con el aviso urgente de Kriptópolis como primer escollo en su camino, a la mañana siguiente muchos medios digitales recogían nuestra información y contribuían a difundirla aún más. En todo caso, el camino se les ponía ya muy cuesta arriba a los defraudadores, apenas unas pocas horas después de poner en marcha su acción. Durante los días siguientes prensa, radio y TV (¡la fugaz aparición del boletín 124 de Kriptópolis en los informativos de Tele 5 de las 20:30 resultó impactante!), se encargaron de prevenir a los pocos internautas que aún no se hubieran enterado.

Pero es que además estos sujetos cometieron otros errores imperdonables. Por ejemplo, resulta muy poco "profesional" confundir una lista de correo con una dirección particular (algunas personas recibieron la "factura" a través de listas de correo sobre Linux), o enviar una confirmación de un pedido a un grupo de noticias como <soc.culture.galiza . En cuanto a la originalidad del método es totalmente nula, aunque parece ser la primera vez que se utiliza en España. El único "mérito" de estos angelitos consiste en haber logrado hacerse con una base de direcciones de correo en la que parece estaban incluidos casi todos los internautas del país. Un detalle a tener en

cuenta por los que se muestran excesivamente tolerantes o complacientes ante el tráfico desvergonzado con estas bases de datos. A partir de ahora, podemos decir que los internautas que no cuiden su privacidad se exponen no sólo al correo basura, sino incluso a tentativas de estafa masiva como la que ahora nos ocupa.

Bien; pero... ¿qué pueden hacer a partir de ahora los afectados? Sin duda alguna, la pelota está ahora claramente en el lado de la Justicia. Quienes se consideren perjudicados pueden encontrar algún consuelo en la extraordinaria rapidez de reflejos demostrada en esta ocasión por el Grupo de Delitos Informáticos de la Policía Judicial. A las pocas horas de saltar la alarma, nos consta que ya estaban trabajando en este asunto. La recomendación de este grupo a quienes se consideren perjudicados es muy clara: **DENUNCIAR LOS HECHOS EN LA COMISARIA DE POLICIA MAS CERCANA**. Aparte de esa medida recomendada, el Grupo de Delitos Informáticos agradecerá cualquier colaboración que puedan prestar los afectados. Cualquier persona que desee colaborar en el esclarecimiento de estos hechos puede dirigirse por correo electrónico al propio Grupo de Delitos Informáticos (din@dgp.mir.es), pudiendo incluso solicitar la clave PGP del grupo para poder proteger adecuadamente la información que se les remita. También están a disposición de los perjudicados los teléfonos de Madrid 91-5822358 y 91-5822491 y el número de fax 91-5822380.

El trabajo judicial para descubrir y castigar a los culpables se preve laborioso, dado que el teléfono en que se consumaba la estafa radica en suelo de otra nación. Algunas personas se han dirigido ya a Kriptópolis preguntando por qué no puede lograrse -al menos- bloquear los ingresos telefónicos de los timadores, tal y como se logró en Estados Unidos hace un par de años. La respuesta es clara: tan sólo en el proveedor America Online se recibieron entonces más de 20.000 denuncias de afectados. En el caso que nos ocupa, también parece claro que cuantas más denuncias existan, más fáciles resultarán las cosas. Por otro lado, parece interesante atajar cuanto antes (y con la máxima dureza) este tipo de riesgos, para prevenir futuros episodios similares que convertirían la Red en un caos y podrían dinamitar para siempre la escasa confianza existente en la actualidad y que tanto está costando instaurar.

Por último, y dadas las circunstancias, Kriptópolis desea servir al menos de punto de encuentro común para quienes ahora mismo están preocupados por aspectos tan concretos como el coste que puedan tener al final las llamadas realizadas. A tal fin, hemos habilitado a estos efectos nuestra habitual sección de debate, para que pueda ser utilizada libremente por quienes lo necesiten o deseen:

<http://www.kriptopolis.com/wwwboard/wwwboard.html>

Del Boletin #127:

- LA CARTA DE JANET RENO (Fiscal General USA) al ministro alemán de Justicia (publicada recientemente por la revista digital alemana Telepolis), ha dejado bien claro cuáles son las últimas intenciones de la Administración Clinton con respecto a la criptografía en el mundo: por decirlo lisa y llanamente, su prohibición más total y absoluta. Veamos tan sólo un significativo párrafo de la misma:

"Queda mucho por hacer. En particular, pienso que debemos centrarnos cuanto antes en los riesgos que supone la distribución electrónica de programas de cifrado. Aunque los países de Wassenaar han alcanzado ya un acuerdo para controlar la distribución de productos de cifrado de cierta potencia dirigidos al mercado masivo, algunas de estas naciones continúan sin controlar los programas distribuidos a través de Internet, bien sea por tratarse de programas de dominio público, bien porque esos países no controlan la distribución de bienes intangibles. Aunque somos conscientes de que se trata de un tema polémico, a menos que sepamos manejar esta situación, el uso de Internet para distribuir productos de cifrado convertirá en inútiles los controles de Wassenaar."

Sólo nos resta agradecer a Mrs. Reno que nos haya proporcionado la respuesta a la pregunta que hace tiempo veníamos haciéndonos: ¿Cómo combatir las restricciones sobre la libre distribución de criptografía potente? Pues -tal y como sospechábamos- gracias al medio de distribución más libre y revolucionario que la mente humana haya podido nunca concebir: Internet.

Deseamos fervientemente que los gobiernos europeos tengan el valor necesario para recordar a la todopoderosa Administración USA que la democracia es un invento netamente europeo (como la Web, por cierto), y que el fascismo tecnológico de nuevo cuño que parecen querer imponernos desde aquellas latitudes, entra en abierta contradicción con las constituciones y leyes bajo las que hemos acordado vivir aquí.

* Traducción de la carta al inglés: <http://jya.com/reno-ban.htm>

- MICROSOFT DEBERÁ PARCHEAR SU PROPIO PARCHE: pocas horas después de anunciar un parche dirigido a solventar un nuevo fallo de seguridad descubierto en sus servidores web IIS, la empresa lo retiró de sus

sitios de descarga y emitió un comunicado en que desaconsejaba su instalación, debido a los errores detectados en el mismo.

- LAS RECIENTES REFERENCIAS DE DIVERSOS MEDIOS DE COMUNICACIÓN al excelente libro de Criptografía de Manuel Lucena (Radio 5 el miércoles, Ciberpais ayer jueves), parecen haber multiplicado las descargas del mismo desde Kriptópolis, habiéndose alcanzado las 5.500 en estos últimos días. Si a ese hecho añadimos que el libro dispone cada vez de más servidores alternativos (desde Argentina hasta Londres, gentilmente ofrecidos para aliviar nuestro tráfico), resulta cada vez más difícil valorar la tirada real que está alcanzando esta obra, pero todo nos hace pensar en un éxito rotundo. Por otra parte, Manuel Lucena nos anticipa la disponibilidad, en el próximo mes de septiembre, de una nueva edición que -entre otras cosas- mejorará la actual legibilidad del libro.

<http://www.kriptopolis.com/libro.html>

- QUIZÁS YA NO SEA NOTICIA para la mayoría, pero la página web de la presidencia del gobierno español fue jaqueada ayer. Una excelente crónica de esta jugada puede leerse en:

<http://labrujula.net/1999/agosto/jueves,12,1.htm>

- MICROSOFT SIGUE SIN DAR CON EL PARCHE "DEFINITIVO" para la reciente vulnerabilidad descubierta en Office 97 por el español Juan Carlos García Cuartango. La solución se anuncia ahora para la próxima semana. Mientras tanto, Microsoft ha puesto a disposición de sus clientes un pequeño programa que se encarga de avisar al usuario cada vez que se abre, desde Explorer, un documento Office. Como la empresa reconoce, no solventa el problema, pero al menos alerta al usuario del tremendo riesgo existente. Puede bajarse desde:

<http://www.microsoft.com/security/Issues/OfficeDocOpenTool.asp>

Del Boletín #128:

"CON MICROSOFT NO PUEDE HABER VACACIONES"

Ya no tenemos ninguna duda de que Kriptópolis pecó de ingenuidad al anunciar (boletín 126, 29 de julio) que se tomaría vacaciones durante el mes de agosto. Sencillamente, no ha sido posible. Y aún más: mucho nos tememos que mientras Microsoft siga produciendo software como el que produce (y todo parece indicar que aún le queda cuerda para rato), ningún sitio dedicado a la seguridad podrá relajarse una sola semana.

Recordarán también nuestros lectores que en aquel boletín de hace tres semanas, Kriptópolis anunciaba también el mayor agujero jamás descubierto por nadie (y no sólo por Cuartango) en un programa de Microsoft. Quienes nos siguen desde hace años saben que Kriptópolis procura huir del alarmismo como de la peste, por lo que cuando decimos "EL MAYOR" queremos decir que la infinita saga de agujeros detectados por Cuartango en Internet Explorer palidece a la sombra de este tremendo butrón, por el que millones de usuarios pueden dar entrada a un virus en sus sistemas sin necesidad de hacer absolutamente nada (aparte -claro- de haber adquirido Office, con su grave fallo de diseño incorporado).

REACCIONES PARA TODOS LOS GUSTOS

Como era de esperar, algunos comprendieron al vuelo la tremenda importancia de nuestro aviso y otros no. Entre los primeros, citaremos (en España) a Panda Software (que solicitó de inmediato a Cuartango una ampliación de detalles) y al informativo económico "Cinco Días" (que sacó la noticia el 3 de agosto en primera plana y con gran alarde tipográfico). Por respeto, y porque ya importa poco, silenciaremos los nombres de los medios que en primera instancia temieron abordar el asunto en toda su magnitud, para después intentar subirse al carro de la forma que fuese (incluso manipulando fechas con descaro y sin ningún temor al consiguiente ridículo). Pobre papel también el de algunos otros medios y empresas que intentaron restar importancia al asunto, pretendiendo incluirlo en la categoría de 'hoaxes' (o casi). Vaya en descargo de estos últimos el hecho de que la profusión de agujeros detectados por Cuartango en los productos de Microsoft (a los que invariablemente sigue un parche oficial en pocos días), han producido una pernicioso insensibilidad en la mayoría de nosotros, demasiado acostumbrados a contemplar con resignación la constante sucesión "ad nauseam" de agujeros y parches. Tampoco citaremos aquí a esos "expertos" en seguridad que en algún foro público desautorizaban a Cuartango (y a Kriptópolis), por presentar como novedad algo que estos preclaros varones - -supuestamente- ya habrían anunciado hacía mucho tiempo (¿alguien sabe dónde?), hecho harto dudoso cuando de todos es conocida la obsesión de estos sujetos por aparecer algún día en su adorada NTBugTraq, siquiera

fuese por reseñar un error de traducción en un manual (es un suponer), dada su -al menos hasta ahora- manifiesta incapacidad para firmar hallazgos de tanta envidia como el que ahora nos ocupa, que afecta -y de forma grave- a nada menos que 100 millones de licencias.

MICROSOFT LO SABÍA, PERO... ¿TODO?

Pero, lejos de amainar, la tormenta levantada por el tremendo butrón de Office 97 (y sus consecuencias) no dejó de producirnos sorpresas y sobresaltos durante todo el mes. Si repasan ustedes nuestra hemeroteca, verán que el boletín 126 de Kriptópolis incidió en un aspecto particular (pero fundamental) de la noticia: Microsoft *conocía* desde mucho antes el fallo, pero la empresa se lo calló. Curiosamente, ningún medio más reflejó este detalle, quizás porque carecían de las pruebas que nosotros sí teníamos delante. Pero es que aún fue peor: en un primer momento, la propia Microsoft se atrevió a calificar públicamente como falsa esa afirmación (lo que equivale a llamar mentirosos a Cuartango y -por ende- a Kriptópolis):

NTBugTraq, 30 de Julio 1999:

(...)

Finally, I need to dispel some incorrect information. It is not true that Microsoft knew about this vulnerability for some time but did not alert customers until the author posted to NTBugTraq

(...)

Secure@microsoft.com

Ante esta acusación, y en ejercicio de su derecho de réplica, Cuartango procedió a enviar a NTBugTraq los mensajes intercambiados con los responsables de seguridad de Microsoft, en los que éstos reconocían abiertamente tener conocimiento previo de este asunto.

Sorprendentemente, Russ Cooper decidió NO publicar la réplica de Cuartango, lo que despejó cualquier duda (a quien aún la tuviera) acerca de qué opinión es la que prevalece en NTBugTraq. Dado que ese señor censuró la réplica de Cuartango, y éste deseaba hacerla pública, Kriptópolis se siente autorizado para publicar a continuación uno de los fragmentos más esclarecedores de la misma, y deja al lector que sea él quien decida si Microsoft conocía o no previamente el fallo denunciado.

From: Richie Lai
To: 'Juan Carlos Garcia Cuartango'
Sent: miércoles 28 de julio de 1999 1:12
Subject: RE: ODBC Vulnerability

Hey Juan,

(-----)

As for the bug you described, it is a known issue and has been fixed in the latest Jet release which you can get at:
<http://www.microsoft.com/data>

The issue lives in the Jet 3.51 module, which shipped with Office 97. We fixed this security hole in Jet 4.0. User can download the new Jet 4.0 ODBC drivers as part MDAC 2.1 SP2 from the Data Access web page. After installing this upgrade, it should fix the problem.

This bug was shipped with office 97 but has since been corrected in office 2000

Richie

Por tanto, Microsoft (o -al menos- uno de sus máximos responsables de seguridad) conocía el asunto aunque... quizás no *todo* el asunto. Afirmaciones como que Jet 4.0 solventa el problema (que llevó a muchos usuarios a descargar 6 MB, algo que luego Microsoft reconoció como inútil y cuya instalación otros clasificaron -incluso- como contraproducente), o que Office 2000 no está afectado (lo que no es cierto, como se ha demostrado luego), nos llevan a sospechar que Microsoft conocía el riesgo, pero no su auténtica gravedad hasta que Cuartango se la colocó ante sus ojos en forma de fichero-demostración.

CUESTION DE IMAGEN

En todo caso, Microsoft (según Richie Lai) reconoce saber de la existencia de un agujero de seguridad (cualquiera que fuese) y -a pesar de ello- no alerta a los usuarios registrados de Office (un simple boletín de seguridad quizás hubiera bastado), ni les recomienda instalar la actualización correspondiente. Curioso. Pero como ustedes bien saben, la reticencia a reconocer abiertamente problemas de seguridad no es exclusiva de Microsoft, sino que son muchas las empresas que prefieren no comprometer su imagen (así lo entienden) y

esperar hasta ver si alguien da la voz de alarma. Resulta espeluznante imaginar la cantidad de programas que pueden tener un agujero de tal calibre sin reparar, a la espera de que algún día alguien ponga el dedo en la llaga.

¿No sería preferible que las empresas alertaran a sus usuarios de inmediato y les conminaran a instalar soluciones antes de que ocurra un desastre? Personalmente, consideramos que este tipo de empresas (que las hay) no manchan su reputación al actuar así, sino más bien al contrario.

Pero no sólo es la imagen lo que induce a algunas empresas a ocultar estos fallos hasta que alguien los descubre y se los comunica.

¿Alguien puede imaginar el coste que ha podido suponer para Microsoft poner a sus ingenieros a trabajar durante tres semanas, bajo la presión de los medios y -probablemente- de sus principales clientes, con el único objetivo de corregir un fallo de diseño tan profundo como este? Seguro que Microsoft hubiera preferido ahorrarse el coste de toda esta operación.

EL FACTOR RIESGO

Imagen, coste... ¿y qué pasa con los usuarios? Los usuarios corren siempre un evidente peligro en el período que media entre el anuncio de un fallo y su solución definitiva. Afortunadamente, la mayoría de los descubridores de este tipo de agujeros observan una conducta escrupulosamente ética, informando antes a la empresa afectada y renunciando a satisfacer las frecuentes peticiones que reciben (algunas de origen dudoso) acerca de más detalles sobre cómo explotarlos. La información técnica detallada, las demostraciones que pueden resultar reveladoras para ojos expertos, quedan relegadas al secreto mientras la empresa trabaja en la solución. Lo contrario significaría el desastre; en este caso, millones de ordenadores podrían haber sido inoculados con virus, sufrir destrucciones de datos y todo género de ataques.

En principio, parece fácil mantener contra viento y marea esa posición tan escrupulosamente ética, pero en la práctica no siempre ocurre así. Cuando la empresa no reacciona con prontitud y además se muestra prepotente (o incluso desdeñosa), el descubridor puede ver tentada su paciencia y decidir aumentar por su cuenta la presión. Generalmente, el anuncio de la inminente publicación de los datos técnicos obra milagros, pero no siempre es así. Baste recordar el reciente episodio en que eEye decidió hacer público un grave defecto en el servidor web IIS de Microsoft antes de que la solución estuviera a punto.

En el caso que nos ocupa, los detalles técnicos estuvieron desde un primer momento en poder de Microsoft y luego sólo al alcance de unos cuantos privilegiados de su confianza (NTBugTraq, Kriptópolis y algún otro). No obstante, en este caso las tres semanas de espera resultaron demasiado largas y tentadoras para la vanidad de algunos, por lo que los riesgos se multiplicaron innecesariamente.

RESPONSABILIDAD FRENTE A AFÁN DE NOTORIEDAD

Así, mientras Microsoft y Cuartango seguían manteniendo los detalles a buen recaudo, irrumpió un factor inesperado que desequilibró el frágil "status quo". Russ Cooper -otra vez- se descolgó en NTBugTraq (18/08/99) con un mensaje en el que invitaba a los suscriptores de la lista a responder a una encuesta, en el sentido de si era conveniente publicar o no una demostración una vez que Microsoft publicase el parche (cuestión que Cooper consideraba entonces inminente, aunque luego se demoró). El propio Cooper reconocía en su mensaje el fortísimo interés mediático que atraería la publicación del parche, con lo que resultaba fácil sospechar el gran interés publicitario que podía esconder su maniobra de intentar publicar -justo en ese momento- la tan ansiada demostración. Por otro lado, su idea planteaba algunos inconvenientes serios: 1) aun estando disponible el parche oficial, usuarios de otros programas no-Microsoft, pero que también emplean ODBC, podrían ser atacados si se desvelaban los detalles. 2) por otro lado, es fácil comprender que la publicación de la demostración sólo podría ser autorizada por su legítimo descubridor y creador (Cuartango) y por la empresa afectada (Microsoft), sin que Russ Cooper ni ningún otro suscriptor de NTBugTraq tuvieran nada que decidir al respecto.

En este sentido, Cuartango envió de inmediato un mensaje público a NTbugTraq desautorizando la publicación de los detalles en esa lista. En ese mensaje (que ha sido malinterpretado recientemente por algunos medios), Cuartango citaba como ejemplo a Kriptópolis, "la principal lista de seguridad en español, que también fue informada y nunca solicitó publicar los detalles." Cuartango finalizaba haciendo notar que la eventual publicación de esos detalles sin autorización suya (o de Microsoft) supondría que "NTBugTraq estaría revelando información confidencial". Poco después, Cuartango recibió un mensaje de agradecimiento del propio equipo de seguridad de Microsoft en el que le manifestaban compartir su opinión y haber convencido a Russ Cooper de que lo mejor para todos era que este señor permaneciese callado.

EL TERCER HOMBRE ENCIENDE LA MECHA

No obstante, el presunto intento autopublicitario de Russ Cooper no había pasado desapercibido para otra mucha gente, entre ellos Elias Levy de SecurityFocus.com (albergue de la lista BugTraq para Unix) quien, alegando que también disponía de una demostración del fallo, decidió anticiparse a todos y publicarlo por su cuenta y riesgo. El fichero de demostración de Levy no pasaba de ser una especie de burda imitación (de origen dudoso) del 'exploit' original de Cuartango, pero aún así cumplió a la perfección su cometido: Cooper renunció definitivamente a publicar su "propia" demostración y los usuarios quedaron expuestos a un grave peligro durante las horas que Microsoft tardó aún en sacar el parche oficial.

APARECE EL PARCHE

Por fin, el 20 de agosto Microsoft publicó su boletín de seguridad número 30 del año en curso, conteniendo la información oficial y anunciando el parche correspondiente:

<http://www.microsoft.com/security/bulletins/MS99-030faq.asp>

A lo largo de ese boletín, Microsoft admite la existencia de dos problemas diferentes subyacentes a este fallo y -por primera vez- la empresa reconoce (de pasada, eso es cierto) lo que hasta ese mismo momento había venido persistentemente negando, a pesar de que Cuartango se lo comunicó desde el primer momento: Office 2000 también es susceptible. Antes de transcurridas 24 horas de la salida del parche, Cuartango insistió en mensaje a NTBugTraq en que los usuarios de Office 2000 son tan vulnerables a un ataque a través de Excel o Word como los de Office 97.

En el momento actual (pueden llamarlo ustedes paranoia si quieren) en Kriptópolis esperamos a saber si Cuartango "autoriza" (o al menos "válida" de alguna manera) ese parche. Para nosotros (y a la vista de lo sucedido) será sólo él quien tenga la última palabra en todo este enojoso culebrón.

UN VERANO PARA OLVIDAR

Aunque nunca "perdonaremos" a Microsoft haber tenido que sacrificar nuestras vacaciones para asistir en directo a todo este despliegue (en beneficio final -eso sí- de nuestros suscriptores, a los que suponemos bastante confundidos en este tema), la empresa de Redmond tiene más motivos que nosotros para intentar olvidar este aciago mes de agosto de 1999, en el que algunas de las chapuzas de seguridad que subyacen a

muchos de sus productos han decidido salir a la luz, y todas ellas al mismo tiempo. Vayan algunos ejemplos, todos -insistimos- de *este* mismo mes:

1) El pasado 29 de Junio, "The New York Post" informaba de una conversación entre un dirigente de la DEA (agencia USA antidroga) y Bill Gates, en la que el primero se quejaba de que los sistemas de cifrado incorporados a los productos de Microsoft (y -al parecer- también utilizados por los capos de la droga), eran tan potentes que los cuerpos de seguridad no podían descifrarlos. A esta argumentación, el hombre más rico del mundo replicó: "Bien; eso significa que necesitan ustedes ordenadores más grandes."

Tan sólo un mes después, parece claro que ambos sólo representaban una pantomima más de ese grotesco circo en que se ha convertido todo lo que tenga que ver con la criptografía en los Estados Unidos, donde nadie dice una sola verdad y absolutamente nada es lo que parece.

Para muestra un botón: el sistema de cifrado de ficheros de Windows 2000 parece haber sido roto, como menciona Bruce Schneier en el último CriptoGramma (<http://www.kriptopolis.com/criptograma/cg0016.html#3>).

Microsoft asegura que no, que nada puede romperse si el usuario no comete antes un imperdonable error (dejar la clave de recuperación en el ordenador), pero el descubridor del fallo no está conforme y el tema no parece aún demasiado claro.

2) El producto MSN Messenger con el que Microsoft pretende arrebatar a AOL el mercado de la mensajería instantánea (terreno dominado aún por ICQ), presenta un fallo (ya reconocido por Microsoft) gracias al cual cualquier persona con acceso físico al ordenador de un usuario de MSN Messenger puede obtener el nombre de usuario y contraseña de éste en Hotmail, dado que ambos sistemas trabajan en conjunción y comparten esos datos vitales. Según algunos expertos, ya habrían existido problemas de suplantación de usuarios de Hotmail, en cuyo nombre se habrían enviado mensajes sin su conocimiento.

<http://www.techweb.com/wire/story/TWB19990820S0006>

3) Antes de que Microsoft hiciera público el pasado martes su flamante Windows Media Audio (su nuevo formato para poner música protegida en la Red), ya circulaba en medios hackers una herramienta que -se asegura- es capaz de vencer esa protección.

<http://www.zdnet.com/zdnn/filters/bursts/0,3422,2317256,00.html>

4) Georgi Guninski hace pública en BugTraq-Unix, el día 21 de Agosto, una nueva y grave vulnerabilidad en Explorer 5. Según Guninski, existe un problema en un control ActiveX que permite que una página web o un correo HTML que contengan un código malicioso, puedan escribir en la carpeta de inicio de Windows. Al reiniciar el ordenador, ese código será ejecutado.

(MUY IMPORTANTE: Si desea visitar esta demostración con Explorer 5, cambie antes la configuración de seguridad de la Zona Internet a "Alta" o desactive la opción "Activar la secuencia de comandos de los controles ActiveX marcados como seguros". Ello le permitirá leer los detalles sin riesgos de que el exploit (inofensivo, por otra parte) se ejecute en su ordenador. Para ello seleccione Herramientas - Opciones - - - Seguridad - Personalizar nivel). Una vez hecho lo anterior, se puede visitar la demostración original y leer los detalles en: <http://www.nat.bg/~joro/scrtlb.html>

MICROSOFT TIENE LA PALABRA

Nadie comprende muy bien por qué la empresa de software más poderosa del mundo no pone orden de una vez en su departamento de seguridad. Sabemos de sobra que los fallos de seguridad son difíciles de prever y resulta mucho más fácil detectarlos cuando el programa ya está en la calle y los auténticos expertos pueden comprobar a fondo sus características de funcionamiento. Por otra parte, si Microsoft tiene una presencia tan abrumadora en todos los campos de la informática, también resulta lógico que la mayoría de fallos aquejen a sus productos. Pero -aún así- aquí hay algo que no acaba de cuadrar...

Parece claro que la empresa informática más potente del planeta no cuenta con el mejor equipo humano para prever y solventar sus abundantes fallos de seguridad. Definitivamente, los mejores expertos están fuera y no dentro. La facilidad con que un ingeniero español como Cuartango descubre (por afición y en su tiempo libre, no se olvide) gravísimos fallos en productos de Microsoft, cuya evidencia parece escaparse a equipos dedicados profesionalmente a investigarlos (que cuentan con todos los medios, incluido el código de los programas), es cuando menos sorprendente. Y lo peor es que no se trata de meros descubrimientos ocasionales y aislados, sino de una saga interminable que se repite casi hasta el aburrimiento.

Volvemos aquí al principio de este artículo: nos hemos acostumbrado de tal forma a esta inseguridad permanente, que cada vez que se anuncia

un nuevo fallo poco nos interesan ya los detalles; sólo importa cuándo y dónde publican el parche.

¿Debe ser así para siempre? ¿Debería Microsoft intentar contratar a Cuartango? En Kriptópolis estamos aburridos de recibir mensajes serios, pero también chistes y chascarrillos ("¿Cuál es el colmo de Microsoft?"), sobre este particular: casi todo el mundo parece clamar porque Microsoft despida de una vez a su equipo de seguridad y encomiende al español la dirección de esa tarea. ¿Le parece a usted algo disparatado o resultaría la solución más seria? O, por el contrario: ¿considera usted inevitable este perpetuo sobresalto y esa continua danza de fallo-parche-fallo-parche....?

Esperamos su opinión en nuestro foro de debate:

¿Cómo podría Microsoft mejorar la seguridad de sus programas?
<http://www.kriptopolis.com/wwwboard/wwwboard.html>

Como es obvio, dudamos mucho de que a Bill Gates le importe lo más mínimo conocer la opinión de nuestros suscriptores al respecto, pero nuestros consejos serán "freeware" incluso para él, tan hábil a la hora de exprimir nuestros bolsillos y que, no contento con ello,...
¡¡además nos deja sin vacaciones!!)

Del Boletín #130:

CUIDADO CON EL PELIGROSO "BUG" DEL ETNOCENTRISMO

Por favor: tengan un poco de paciencia conmigo y permítanme contarles una pequeña anécdota personal. Como compensación, les aseguro que al final (y aunque no lo parezca), todo acabará guardando relación con la temática habitual de este boletín.

Verán; hace casi una década, cuando temporalmente residía en California (USA), fui invitado -con un grupo de amigos- a una pequeña fiesta particular en Santa Barbara. Nuestro anfitrión parecía un preclaro ejemplo del "yupismo" aún rampante por aquella época y latitudes. Dedicado a actividades financieras casi incomprensibles para mí, este individuo disfrutaba (a sus veintipocos años) de una preciosa y amplia casa, en cuyo garaje reposaba un espectacular deportivo italiano ("Cavallino" incluido), aparte de otros evidentes signos de riqueza, que venían a dar fe de que entre aquellos muros no

se conocía ninguna escasez material. Como consecuencia (o causa quizás) de aquel impresionante entorno en que el angelito había sabido encaramarse con precoz destreza, este sujeto tendía a lucir una especie de sonrisilla estúpida y autocomplacida, que progresivamente amenazaba con fastidiarme la velada. Por otra parte (y dada mi condición de extranjero), la conversación no tardó en derivar hacia nuestras respectivas nacionalidades y sus cada vez más grandilocuentes y espinosas exaltaciones acerca de las evidentes bellezas naturales de California, en claro y ostentoso detrimento -además- de las (no menos evidentes) de mi Cantabria natal, que aquel paleta no tenía siquiera el gusto de conocer. La verdad es que me costó lo mío hacer comprender a aquel energúmeno que la belleza salvaje del mar Cántabro no tiene nada que envidiar a la del Océano Pacífico, que cualquiera de nuestras playas es infinitamente mejor que Avila Beach, y que Cantabria (mucho más pequeña que California) dispone también de nieve y montañas de 2.500 metros de altitud. De hecho, sólo tuve que claudicar en cuanto al desierto del Mohave, aunque maldita la falta que nos hace...

Al final de la charla, y con forzada cortesía, tuve a bien invitar al individuo a darse una vuelta por España (el coste del viaje no debería suponer ningún problema para él) y comprobar por sí mismo cuanto le decía. El cafre sólo se dignó responder: "¿Y para qué querría yo conocer cualquier otro país, si en este tenemos de todo?".

He traído esta anécdota a colación porque es lo primero que me vino a la cabeza cuando un amable lector de Kriptópolis me comentó la mala jugada que nos habían hecho otros zoquetes del mismo estilo que el referido, es decir, de los parecen considerar Estados Unidos el ombligo del mundo y actúan en consecuencia.

Este lector nos hizo ver que en el boletín 116 (4 de Junio de 1999) nos habían colado un clamoroso gol. Se trata del artículo "Los humildes ficheros GIF tampoco se libran del Efecto 2000", firmado por nuestro colaborador (y experto en Mac) Daniel Comín. Este artículo se basaba (y así se hacía constar) en un hallazgo de BoxTop Software, del que se hacía eco la revista TidBits. Pues bien; cuando repasamos el artículo original y los enlaces correspondientes (como siempre hacemos), se nos pasó por alto un "insignificante" detalle: al final de la noticia aparecía una especie de gorrito de bufón, acompañando una letra pequeña en la que se venía a decir: "si algo de este artículo le resulta extraño, compruebe su fecha de publicación." Bien; pues la fecha no era otra que el 1 de Abril, el "Fool's Day" que para los anglosajones es el equivalente del Día de los Inocentes (28 de diciembre en España) y quién sabe qué fecha y qué nombre en otros lugares de la tierra.

Resulta increíble que una revista como TidBits, con varios años en la

Web y convertida en referencia de culto para los usuarios de Mac, se permita jugar de tal forma con sus lectores y suscriptores, máxime cuando manifiesta ser consciente de que estos pertenecen a 106 países diferentes y tiene además en marcha traducciones a varios idiomas.

Bien; pues Kriptópolis mordió al anzuelo. ¿Significa ello que Kriptópolis es un medio poco serio? No para nosotros. Cuando uno acude a fuentes de supuesto prestigio, no tiene por qué pensar -al menos a priori- que esos medios puedan jugar con la buena fe de sus lectores, presuponiendo un conocimiento universal de costumbres que son exclusivamente anglosajones. ¿Se imaginan a Kriptópolis publicando el 28 de diciembre (el día equivalente en España) una bromita donde se diga que Estados Unidos ha decidido liberalizar la exportación de criptografía ó que PGP ha sido definitivamente vencido? ¿Captarían el chiste nuestros suscriptores de Caracas, Montreal, Londres o Sidney? No se preocupen, porque nunca será el caso: Kriptópolis es (o intenta ser, a pesar de algunos) un medio serio.

Más allá de la pura anécdota, el incidente manifiesta que existe un peligro de dominación cultural del que no somos demasiado conscientes. Por un lado, la implantación de nuestro idioma en el mundo real es importantísima (incluso pronto habrá un 30% de hispanos en USA). Y si, aún así, el colonialismo cultural de los Estados Unidos es cada día más abrumador en nuestras vidas, ¿qué podemos esperar en la Red, donde las páginas en nuestro idioma apenas representan el 1% del total?.

No; no creemos que -al menos todavía- sea necesario un decreto-ley que adapte nuestro santoral y calendario festivo a la cultura sajona dominante, para así poder estar prevenidos ante sus eventuales gracias y reimos con ellos. De momento bastará con que no leamos (ni por tanto citemos) nunca más a algunos medios a los que se les llena pomposamente la boca con la dichosa "aldea global", pero que con sus actitudes demuestran bien a las claras que, para ellos, ese espacio utópico limita al este con los Apalaches y al oeste con las Montañas Rocosas (o, como mucho, con la preciosa -¡faltaría más!- Malibú).

TAMBIÉN FALLA LA MAQUINA VIRTUAL JAVA DE MICROSOFT

En nuestra relación de fallos de seguridad detectados en Agosto en productos Microsoft, no pudimos reseñar otro nuevo que la empresa acaba de reconocer y solucionar hace sólo unos días:

<http://www.microsoft.com/security/bulletins/MS99-031.asp>

Este nuevo fallo permite que 'applets' Java maliciosos albergados en

sitios web, superen las limitaciones propias de la 'sandbox' y puedan ejecutar todo tipo de acciones en el ordenador de la víctima que visite esas páginas.

Son susceptibles Internet Explorer 4 y 5, y aunque desactivar la ejecución de Java debería ser suficiente, Microsoft insiste en que se instale el parche correspondiente. Para Russ Cooper, responsable de NTBugTraq, este fallo puede ser considerado tan peligroso como el descubierto por Cuartango en Office.

OTRO GRAVE FALLO DE SEGURIDAD EN HOTMAIL

El correo de 50 millones de usuarios quedó este fin de semana al alcance de cualquiera, en el fallo más grave -hasta ahora- de la larga saga que viene sufriendo Hotmail.

* Español: <http://labrujula.net/1999/agosto/martes,31,1.htm>

* Inglés:

<http://cgi.zdnet.com/slink?/adeska/adt0831ba/3801:30467>

<http://www.news.com/News/Item/0%2c4%2c41069%2c00.html?dd.ne.txt.0830.01>

<http://www.wired.com/news/news/email/explode-infobeat/business/story/21490.html>

EXCEL Y EL EFECTO 2000

Muchos usuarios de Excel (5.0, 95, 97 y 2000) podrían tener una desagradable sorpresa el próximo uno de enero si han utilizado fechas de dos dígitos para el año en alguna hoja de cálculo y no visitan antes la página informativa de Microsoft sobre el Efecto 2000. Desafortunadamente, este problema pasa desapercibido para la mayoría de las utilidades que analizan el riesgo del Efecto 2000. Dice Microsoft: "La función DATE() no está diseñada para admitir años expresados con dos dígitos, ya que recibe parámetros numéricos. La función DATE() calcula los números menores que 1900 como años que se deben contar a partir de 1900. Por tanto, en una fórmula como DATE(15,1,1), la fecha resultante sería el 1 de enero de 1915, no del 2015."

* Excel 5.0:

http://www.microsoft.com/Spain/2000/Guia/user_view68331ES.htm

* Excel 95:

http://www.microsoft.com/Spain/2000/Guia/user_view68281ES.htm

* Excel 97:

http://www.microsoft.com/Spain/2000/Guia/user_view68318ES.htm

* Excel 2000:

http://www.microsoft.com/Spain/2000/Guia/user_view68842ES.htm

FACTORIZADA CLAVE PÚBLICA DE 512 BIT

El pasado 22 de agosto, un equipo de investigadores de seis países logró hallar los factores primos de un número de 512 bits, es decir, una clave pública RSA, idéntica a las que aún protegen un gran número de transacciones electrónicas vía SSL.

<http://www.rsa.com/pressbox/html/990826.html>

Del Boletín #131:

¿DÓNDE ESTÁN LAS LLAVES?

- Un criptógrafo levanta una gran polémica al afirmar que todos los sistemas operativos Windows incorporan una puerta trasera para permitir (según él) el acceso de la agencia de espionaje norteamericana.

*** LA NOTICIA**

Andrew Fernandes (director científico de la empresa de seguridad Cryptonym en Morrisville, Carolina del Norte) hizo públicos el mes pasado los resultados de su investigación sobre Microsoft Crypto-API.

Ocurrió en una de las sesiones finales de la conferencia criptográfica "Advances in Cryptology, Crypto'99" celebrada en Santa Barbara (California, EEUU). Los asistentes a la reunión (no más de doce, según algunas fuentes) se quedaron perplejos cuando Fernandes anunció que -según sus investigaciones- todos los Windows 95/98/NT incorporarían una puerta trasera destinada (según él) a permitir a la todopoderosa NSA (National Security Agency, la agencia norteamericana de espionaje electrónico) acceder a millones de ordenadores de todo el mundo. El pasado viernes 3, Fernandes decidió emitir una nota de prensa dando total publicidad a sus hallazgos y el revuelo organizado desde entonces está siendo monumental.

* FUNDAMENTOS TÉCNICOS

Crypto-API es el bloque fundamental para construir los servicios criptográficos en los sistemas operativos de Microsoft. En principio, cualquiera podría instalar un determinado componente criptográfico en Windows, de no ser porque Crypto-API exige que ese componente venga firmado con *una* determinada clave, perteneciente a la propia Microsoft. De esta forma, puede verificarse también la validez de cualquier CSP externo (Cryptographic Service Provider: proveedor de servicios criptográficos). En pocas palabras: Windows sólo puede incorporar módulos criptográficos que vengan firmados por la propia clave de Microsoft. Punto. ¿Y qué sentido tiene todo esto? Uno muy claro: los usuarios Windows de fuera de Estados Unidos no pueden incorporar criptografía potente en sus sistemas operativos, porque ningún CSP capaz de proporcionársela será firmado por Microsoft, en aplicación de las conocidas restricciones a la importación vigentes en EEUU.

Fernandes llevaba algún tiempo buscando posibles vulnerabilidades en Crypto-API, basándose precisamente en los trabajos anteriores de Nicko van Someren (de NCipher), que fueron presentados en Crypto'98 y donde este investigador británico, a base de desensamblar el driver ADVAPI.DLL, aseguraba haber descubierto que existían dos claves y no sólo una (como aseguraba la documentación oficial), aunque no podía averiguar nada más sobre la segunda: ni para qué servía, ni quién era su propietario. La pista definitiva que Fernandes necesitaba llegó de la mano del Service Pack 5 para Windows NT, donde los programadores de Microsoft al parecer "olvidaron" retirar la descripción simbólica de los componentes, que se utiliza en la fase de depuración del código y luego es regularmente borrado. Esta circunstancia permitió a Fernandes descubrir que la etiqueta identificativa de esta segunda clave es -nada menos- que "_NSAKEY".

* PONERLE EL CASCABEL AL GATO

Hasta aquí los hechos; ahora comienzan las interpretaciones...

¿A quién puede pertenecer esa segunda y misteriosa clave? Este es quizás el principal aspecto del asunto, por lo que las opiniones al respecto varían sustancialmente. Según Fernandes, el nombre lo dice todo: a la NSA; según Microsoft, a la propia Microsoft. La empresa de Redmond ha negado desde el primer momento que las conclusiones de Fernandes tengan el menor fundamento. En cualquier caso, lo que está claro es que el poseedor de la segunda clave estaría en condiciones de instalar componentes criptográficos y servicios de seguridad en cualquier ordenador bajo Windows de todo el mundo, lo que le permitiría controlarlo a su antojo (por ejemplo, sustituyendo el módulo de cifrado por otro que no cifre en absoluto). Y todo ello -por supuesto- sin conocimiento ni autorización de los usuarios. Por otro lado, si la NSA fuera realmente la propietaria de la segunda clave, y la utilizara para espiar, serían las empresas y ciudadanos extranjeros quienes afrontarían el riesgo, puesto que la NSA tiene prohibido espiar a ciudadanos norteamericanos. No obstante, un experto de la talla de Bruce Schneier resta importancia al descubrimiento de Fernandes, ya que -según él- la NSA no tendría ninguna necesidad de utilizar este mecanismo para tomar el control, puesto que ya existen herramientas como Back Orifice que permiten lograrlo. De idéntica opinión es Alec Muffett, consultor de seguridad de Sun Microsystems. Para Muffett "resultaría muy poco práctico que la NSA utilizase esa clave (caso de existir) para espiar a empresas extranjeras; sería demasiado trabajo." También Russ Cooper (moderador de NTBugTraq), considera muy poco creíble todo el asunto.

En cualquier caso, ¿en qué se basa Microsoft para decir que la clave identificada como "_NSAKEY" no pertenece a quien Fernandes cree? Según Microsoft, esa clave sería una simple clave de resguardo que permitiría a Microsoft revocar la primera clave (en caso de una eventual pérdida o compromiso) y sustituirla por otra nueva. No obstante (afirman otros), Microsoft no tendría ninguna necesidad de revocar su clave ante tal eventualidad, pues les bastaría publicar un nuevo Service Pack y/o otra versión de Explorer. ¿Y por qué denominarla con el prefijo "NSA"? Simplemente para reflejar -según Microsoft- que esa clave había pasado los controles que la propia NSA exige a los productos de seguridad destinados a la exportación. Pero, en todo caso, ¿por qué ocultar la existencia de una clave supuestamente tan inofensiva?

Por otro lado, y como el propio Fernandes reconoce, aunque la clave misteriosa perteneciese efectivamente a la NSA, ello no demostraría al 100% que su finalidad fuese el espionaje; bien pudiera ser que la NSA

no confiase totalmente en Microsoft a la hora de incorporar sus propios módulos criptográficos, que quizás no desearía someter siquiera a la supervisión y firma por parte de esa empresa.

Incluso alguien podría elucubrar con la idea de que el "olvido" de los programadores de Microsoft a la hora de retirar la descripción simbólica de la clave NSA pudo no ser casual. El nombre pudo haberse mantenido para que señalase inequívocamente al responsable de su implantación el día que pudiera ser descubierta. Probablemente nunca lo sabremos.

Por último, si usted ya tiene muy claro en su fuero interno a quién pertenece la segunda clave y para qué puede servir, vaya pensando en el siguiente reto: las últimas investigaciones de Nicko van Someren revelaron que en el driver ADVAPI.DLL de Windows 2000 existe además una *TERCERA* clave, cuyo desconocido propietario algunos ya se precipitan a afirmar que pudiera ser el mismísimo FBI. Al parecer, incluso los propios expertos criptográficos de Microsoft que asistían a la conferencia (encabezados por Brian LaMachia, director de desarrollo de CAPI), se mostraron sorprendidos por la revelación de la existencia de una tercera clave en W2000 por parte de alguien ajeno a la empresa. La tercera clave fue descubierta por Van Someren a base de avanzados métodos que estudian la entropía del código de programación. Microsoft ha asegurado que la tercera clave es una clave de prueba, que será eliminada en la versión definitiva.

*** PRIMERO LAS MALAS NOTICIAS...**

Con independencia de que algún día se confirmen o no tantas sospechas, lo cierto es que la resignada pasividad de los usuarios de productos Microsoft ante los reiterados fallos de seguridad en sus productos (Explorer, Office...) y servicios (Hotmail...), comienza a convertirse en una indignación cuyas últimas consecuencias aún están por ver.

La mera *posibilidad* de que terceras partes desconocidas se permitan instalar y ejecutar a sus anchas (y a nuestras espaldas), códigos desconocidos que pudieran revelar a organismos extranjeros de espionaje nuestra información más confidencial, parece lo suficientemente grave como para exigir -al menos- una investigación formal inmediata a los organismos competentes.

*** Y AHORA LAS BUENAS NOTICIAS...**

Andrew Fernandes ha sabido cómo hacer una tortilla con los huevos rotos. El razonamiento es sencillo: es posible eliminar la clave NSA

sin que el funcionamiento del sistema operativo se resienta, puesto que la clave original de Microsoft sigue presente y continúa validando el código criptográfico autorizado por la empresa. De esta forma, nos liberamos de la posibilidad de que nuestro Windows instale y ejecute código proveniente de esa tercera parte sospechosa. Pero aún hay más: la clave NSA eliminada de nuestro sistema puede reemplazarse por cualquier otra (incluso la nuestra), mientras se mantenga la clave original de Microsoft. ¿Con qué objeto? con el de poder incorporar a nuestro Windows el CSP que se nos antoje, incluido cualquier componente que utilice criptografía potente, y que la clave original por sí sola nunca autorizaría. Por decirlo en palabras de Andrew: "El control de exportación está definitivamente muerto para Windows". No obstante, las cosas no parecen tan sencillas, por cuanto la manipulación del software seguiría siendo tan ilegal como la propia exportación no autorizada. Por otra parte, Crypto-API constituye una auténtica caja negra, cuyo funcionamiento es desconocido por la inmensa mayoría de los mortales y cuyo simple desensamblado tampoco permitiría aprehender todos sus detalles.

Aún así, y a modo de demostración, Fernandes ha elaborado un programa (sólo para Windows NT y Windows 2000) que elimina la clave NSA y la reemplaza por una clave de prueba. El código fuente sólo está disponible bajo previo acuerdo de no revelación con Cryptonym:

<http://www.cryptonym.com/hottopics/msft-nsa/ReplaceNsaKey.zip>

*** MEDIDAS PREVENTIVAS**

Episodios tan preocupantes como los que venimos presenciando en los últimos tiempos, probablemente nunca hubieran tenido lugar si disfrutáramos de sistemas operativos de código abierto, en lugar de vernos obligados a depender para siempre de sistemas cerrados como los actuales. Tendremos que ir pensando seriamente en deshacernos de una vez de ese permanente yugo.

En la misma línea, se hace también inaplazable disponer de código criptográfico abierto, tal y como pretende el proyecto emprendido recientemente por Kriptópolis desde nuestra lista de programación. El propio Alec Muffett (a quien citábamos antes) comparte nuestra opinión cuando dice: "Cualquier empresa que se precie exigiría utilizar código criptográfico abierto, en lugar de productos empaquetados como el que ahora nos ocupa."

*** OPINIONES DIVERSAS**

Al final, la pregunta clave sigue siendo: ¿incorpora Windows puertas traseras? Pues bien; de momento se trata de un asunto de fe, donde todas las posibilidades están abiertas.

Veamos; si usted confía a priori en Microsoft, seguro que se sentirá plenamente satisfecho con la explicación de la empresa, o al menos encontrará alguna excusa para disculpar -otra vez- sus quizás medias verdades; si usted es más proclive a pensar que el nombre de la clave no es casual y que el gobierno de EEUU pretende ejercer -a cualquier precio- un imperialismo tecnológico evidente sobre el resto del mundo, y que para ello no duda en servirse de la colaboración más o menos gustosa de las grandes corporaciones allí radicadas (la célebre "Teoría de la Conspiración"), está usted también en su derecho. Incluso puede imaginarse una especie de tercera vía: Microsoft, ya bastante cercada por el gobierno norteamericano y los pleitos que éste le tiene abiertos, se ve forzada a plegarse a los deseos controladores de la NSA, al tiempo que deja una pista (una especie de "huevo de pascua" acusador) en el propio código. Todas estas opciones pudieran ser posibles... ¡o quizás ninguna de ellas! Queda aún mucho por investigar...

Para Kriptópolis, no deja de resultar significativo que acreditados expertos independientes (como Bruce Schneier o Richard Smith) estén apoyando abiertamente las tesis de Microsoft. El español Juan Carlos García Cuartango tampoco toma partido en este tema. Según Cuartango "averiguar el uso real de dicha clave requeriría un análisis del código fuente del Crypto API de MS, y éste no está disponible. Otra posibilidad sería su desensamblado y esta operación es ilegal." Coincide también Cuartango con nosotros al afirmar que "la única conclusión válida de este asunto es que ninguna organización seria que requiera la utilización de criptografía debería utilizar el software de Microsoft, ya que éste no es público y nadie sabe lo que hace realmente." Kriptópolis ha recabado también la opinión del abogado español Carlos Sánchez Almeida. Para él, "si Microsoft quiere probar su inocencia, y dejar su imagen a salvo de toda duda, la empresa de Redmond sólo tiene un camino: publicar su código fuente. No perdería, todo lo contrario: quizás a la larga ganase a Linux con sus mismas armas." Pero en la práctica, y por lo que respecta a este caso concreto, ni siquiera publicar el código fuente daría la seguridad absoluta. Lo más importante aquí sería saber quién posee la segunda clave.

Cualquiera que sea la opinión del lector, Kriptópolis desea conocerla y le ofrece -como siempre- los medios para expresarla. Nuestro foro de debate tratará esta semana del siguiente tema:

"Puertas traseras en Windows: ¿Fantasía o realidad?"

<http://www.kriptopolis.com/wwwboard/wwwboard.html>

*** TODAS LAS FUENTES DE LA NOTICIA**

La información del propio descubridor está en su página web:

<http://www.cryptonym.com/hottopics/msft-nsa.html>

Pueden ampliar información en cualquiera de las siguientes fuentes:

* Un artículo anticipativo de Schneier, que ahora cobra aún mayor relevancia:

<http://www.kriptopolis.com/criptograma/cg0012.html#5>

* La noticia en la prensa:

<http://www.elpais.es/p/d/19990905/sociedad/gates.htm>

<http://cnnenespanol.com/tec/1999/09/03/clave/index.html>

<http://www.el-mundo.es/navegante/diario/99/septiembre/05/microsoft.html>

Del Boletín #132:

DECIDIDA APUESTA DE KRIPTÓPOLIS POR EL CÓDIGO ABIERTO

Los numerosos y preocupantes incidentes de seguridad a que hemos venido asistiendo las pasadas semanas, han venido a confirmar algo que todos sabíamos: nuestra seguridad en la Red no puede confiarse a programas y sistemas operativos elaborados apresuradamente, bajo la implacable presión de los planes de marketing, y contruidos en base a misteriosas e impenetrables "cajas negras".

Sin pretender negar la contribución histórica de tales productos a la tremenda difusión actual de la informática personal, también parece claro que hoy en día necesitan urgentemente ser superados. Los programas basados en código oculto y bajo la perpetua sospecha que impone su dependencia de gobiernos que sostienen una visión muy peculiar respecto a quién debe disfrutar de confidencialidad absoluta

y quién no, ya no parecen estar en condiciones de servir a los ciudadanos de otros ámbitos geográficos, en directa competencia comercial -además- con los Estados Unidos de América.

La solución no parece fácil. No obstante, llevan años en marcha iniciativas muy interesantes, que -curiosamente- tienen además un origen netamente europeo, y que aspiran a construir algún día una realidad informática muy diferente para el usuario de a pie. Nos referimos fundamentalmente a Linux, el cada vez más popular sistema operativo basado en el sólido Unix.

Como en casi todos los campos de la vida, al sistema del pingüino tampoco le faltan arribistas de última hora, que pregonan sus extraordinarias ventajas antes de haber sido capaces de instalar una sola de sus distribuciones. Muchos de ellos, sólo han mirado por encima del hombro de alguien y se han quedado maravillados al ver el impresionante aspecto gráfico que confiere Enlightenment al escritorio (cuando probablemente pensaban que Linux era sólo un árido sistema de caracteres "blanco sobre negro", al más puro estilo MS-DOS).

La verdad es que el entusiasmo por Linux, basado en tan pobres presupuestos, nos sirve de bien poco. Hay que haberse peleado desde hace años con unas cuantas distribuciones diferentes, para saber cuánto se ha avanzado en los últimos años (o meses) al intentar convertir a Linux en una alternativa al alcance de todos. Las últimas distribuciones de Red Hat, Open Linux o SuSE están a punto de lograrlo, pero aún queda un (cada vez más breve) trecho por recorrer. Quien todavía huye espantado ante la necesidad de particionar su disco duro o la interesante posibilidad de compilar un kernel a medida, necesitará aún trabajar y sufrir mucho para obtener hoy de Linux la productividad que obtiene de su Windows habitual.

No obstante, en Kriptópolis estamos convencidos de que merece la pena empezar a intentarlo desde ahora mismo; seguro que las cosas irán bastante más rápidas si se logra pronto una masa crítica de usuarios interesados en la labor. Pero, aún a riesgo de herir la sensibilidad de los más puristas, afirmaremos sin ningún pudor que no es necesario que desinstale Windows de momento. Con las debidas precauciones, Linux puede convivir con él sin problemas. Puede dedicar tan sólo una pequeña partición (o quizás un segundo ordenador o disco duro) a practicar y aprender; ir configurando sin ninguna prisa su acceso a Internet desde Linux. Si cuenta con la debida ayuda (y no hay comunidad más colaboradora y comprensiva que la de los "linuxeros"), no tardará en recibir y enviar su correo, navegar por la Web y realizar las mismas actividades de ahora, pero desde un sistema operativo "de verdad" y cuyo código es transparente. Las dificultades no son pocas, pero cada obstáculo que se vence proporciona una

satisfacción mayor. Cuando instale una suite ofimática gratuita del nivel de Star Office, o descubra el poder de GIMP como programa de tratamiento de imágenes, es posible que se acabe replanteando algunas cosas.

Existen numerosas páginas web y listas de correo en español dedicadas a ayudar a los usuarios (nuevos o no) de Linux. Nosotros (en particular) aprendimos mucho del grupo de news en español sobre este sistema operativo (es.comp.os.linux). Lógicamente, la temática de Kriptópolis es la seguridad y no podemos proporcionarles ayuda detallada sobre cómo se configura en Linux una tarjeta de sonido. Por tanto, como muestra de apoyo a la comunidad Linux en español, y sin ánimo de competir en absoluto con ninguno de los excelentes recursos ya existentes (cuya eficacia y sapiencia nunca podríamos igualar), hemos decidido crear una NUEVA LISTA DE CORREO dedicada a discutir *exclusivamente* aspectos de SEGURIDAD en sistemas bajo Unix/Linux.

Para suscribirse basta enviar un mensaje en blanco a:

kriptopolis-seg-unix-subscribe@onelist.com

y replicar el mensaje de confirmación que se reciba.

A juzgar por las muchas peticiones de información sobre Linux que recibimos en Kriptópolis y las constantes sugerencias sobre aumentar la presencia de información relativa a este sistema en nuestros boletines, esperamos que la nueva lista goce de una extraordinaria acogida, como viene ocurriendo con el resto de las que hemos inaugurado recientemente. De hecho, nos parece bastante significativo que a las 24 horas de anunciarla en nuestra lista de ayuda, ya contase con nada menos que 75 suscriptores.

En la misma línea de apoyo al código abierto, comentar que esta misma semana ha arrancado formalmente el trabajo en nuestra lista de programación, con la creación de grupos específicos para cada lenguaje. También nuestra lista de Teoría contribuirá, desde la faceta de documentación, a la creación de una amplia base de datos de código criptográfico ABIERTO. Nuestro proyecto ya tiene incluso nombre propio: COLOSSUS, aunque desconocemos todavía si el nombre hace alusión a la magnitud de la idea y al tremendo esfuerzo que requerirá, o bien al nombre del primer ordenador electrónico de la historia, construido en secreto por los ingleses durante la II Guerra Mundial para descifrar los mensajes cifrados nazis, y mantenido en secreto hasta 1976, lo que motivó que el famoso ENIAC se llevara todos los honores al respecto (según afirman José Pastor y M.A. Sarasa en su excelente libro "Criptografía Digital").

Esperamos que nuestros lectores sepan valorar estos nuevos esfuerzos realizados desde Kriptópolis y dirigidos a lograr para todos los usuarios de Internet MAYOR SEGURIDAD de la única forma que a largo plazo puede lograrse: a través de la máxima TRANSPARENCIA.

<http://www.kriptopolis.com/listasc.html>

DISPONIBLE -POR FIN- LA VERSIÓN 1.0 DE GNUPG

Otra novedad relativa a programas de código abierto:

El pasado 7 de septiembre fue publicada la primera versión oficial definitiva (1.0) de GnuPG, el nuevo software de cifrado compatible con OpenPGP (RFC 2440), y que se presenta como una alternativa gratuita a PGP que, al no emplear RSA ni IDEA, no sufre restricciones relativas a patentes.

GnuPG se acoge a la licencia GPL y es compatible con las versiones 5.x de PGP, al que añade algunos otros algoritmos soportados (puede utilizar ElGamal, DSA, 3DES, Blowfish, Twofish, CAST5, MD5, SHA-1, RIPE-MD-160 y TIGER), así como soporte de varios idiomas: Español, Francés, Alemán, Italiano, Polaco, Ruso y Portugués).

El programa funciona en plataformas Linux, FreeBSD y OpenBSD. Puede funcionar también sobre: HPUX v9.x y v10.x con HPPA CPU, IRIX v6.3 con MIPS R10000 CPU, OSF1 V4.0 con Alpha CPU, OS/2 versión 2, SCO UnixWare/7.1.0, SunOS, Solaris en Sparc y x86, USL Unixware v1.1.2, Windoze 95 y WNT con CPUs x86, pero en estos sistemas no se garantiza seguridad absoluta, debido a la ausencia de una fuente de entropía bien comprobada. El uso en estos casos ha de ser experimental.

En el aspecto de documentación GnuPG cuenta con su propia página man, así como con un mini-COMO en español:

http://www.dewinter.com/gnupg_howto/spanish/GPGMiniComo.html

También existen varios interfaces gráficos para el programa, algunas de ellas incluidas en los proyectos GNOME y KDE.

* Descarga del programa: <http://www.gnupg.org/download.html>

Del Boletín #133:

NUEVA EDICIÓN DEL LIBRO ELECTRÓNICO GRATUITO "CRIPTOGRAFÍA Y SEGURIDAD EN COMPUTADORES"

Desde hoy está disponible en Kriptópolis la segunda edición del excelente libro electrónico de Manuel Lucena. Las principales novedades respecto al ejemplar publicado en julio pasado son las siguientes:

- * Revisión exhaustiva del texto, con ampliaciones en casi todos los capítulos
- * Aumento sustancial del número de páginas (de 119 a 167) e inclusión de dos nuevos capítulos: "Hackers" y "Virus".
- * Elaborado en formato PDF nativo. No más problemas de visualización e impresión.
- * Revisión profunda del capítulo sobre números aleatorios.
- * Un apéndice con valores de prueba para facilitar la implementación de DES e IDEA.

Con esta nueva edición, Manuel Lucena comienza a dar forma a su inicial propósito, en el sentido de que (gracias a su formato y distribución electrónicos) este libro no sería jamás una obra definitiva, sino un proyecto siempre abierto a su constante ampliación y perfeccionamiento.

Para dar una idea del éxito de esta iniciativa, baste indicar que se registraron más de 10.000 descargas de la primera edición. Por otro lado, diversos medios de comunicación se han hecho eco de este original proyecto:

- Ciberpaís (suplemento de Internet del diario "El País"):
<http://www.ciberpais.elpais.es/c/d/19990812/tecno/tecno2.htm>

- 10-08-99: La noticia en Radio Nacional de España (Radio 5)
(823 KB):
<http://home.worldonline.es/kriptopo/Radio5.mp3>

- 21-08-99: Entrevista a Manuel Lucena en la cadena COPE de radio (1.709 KB):
<http://home.worldonline.es/kriptopo/COPE.mp3>

La nueva edición del libro (831 KB) ya puede descargarse desde
<http://www.kriptopolis.com/libro.html>

Del Boletín #134:

UN JUEGO NO TAN INOCENTE

A primeras horas de la madrugada de hoy, Bernardo Quintero (responsable de Hispasec, <http://www.hispasec.com>) enviaba un mensaje a Kriptópolis para avisarnos de una vulnerabilidad descubierta en WWWBoard, el 'script' que gobierna nuestro foro de debates y el de la mayoría de los web albergados en servidores UNIX (como -por ejemplo- casi todos los albergados en RapidSite, uno de los servicios de 'hosting' más utilizados y que da cabida a sitios tan populares como Kriptópolis o LaBrujula.Net). Por supuesto, no hará falta decir que el gesto de Bernardo le honra, ya que su intención no era otra que avisarnos de esa vulnerabilidad y darnos tiempo suficiente para corregirla. Una conducta muy razonable en un experto en seguridad.

En realidad, se trata de una vulnerabilidad conocida desde hace tiempo (entre otras cosas por ser absolutamente obvia), pero que David E. Weins había reavivado en un mensaje dirigido el pasado día 16 a BugTraq.

Ante estos avisos, lo más lógico hubiera sido corregir de inmediato el problema y a otra cosa. Pero -sorprendentemente- Kriptópolis ha optado por un comportamiento que nos parece más creativo: primero le propusimos la idea al propio Bernardo, pero, ¿por qué negar a todos nuestros lectores esta excelente ocasión de experimentar y aprender? Continúen leyendo...

LA VULNERABILIDAD

Es fácil de comprender. En cada foro de debate existe un administrador, que tiene la potestad de borrar mensajes, cerrar el debate, iniciar uno nuevo, etc, etc... Lógicamente, todo ello se realiza a través de un nombre de usuario y una contraseña.

WWWBoard incorpora un nombre de usuario y una contraseña por defecto, lo que no constituye una buena idea, aunque la documentación del programa insiste en que debe cambiarse cuanto antes. No hará falta decir que quien gestione uno de estos tableros y no haya cambiado la contraseña sí que está en un evidente peligro.

Para el resto de administradores (los que sí cambian la contraseña por otra propia) la vulnerabilidad consiste en que el fichero que contiene nombre y contraseña se almacena -por defecto- en un directorio accesible desde la Web. De esta forma, si usted va a su navegador y teclea:

<http://www.kriptopolis.com/wwwboard/passwd.txt>

obtendrá instantáneamente lo siguiente:

SanMeliton:aedZZ2E8n8I6A

donde "SanMeliton" es el nombre de usuario del administrador del foro de debate de Kriptópolis (¿curioso nombre, verdad?) y "aedZZ2E8n8I6A" es su contraseña (previamente cifrada con crypt, el cifrado estándar de UNIX).

Lógicamente, esto ya constituye en sí mismo una vulnerabilidad, puesto que ese sensible fichero no debería estar expuesto a cualquiera. La solución es sencilla: basta modificar el path de una variable del script (\$passwd_file) para que apunte a un directorio protegido u oculto (ej: "/path/to/non-web/dir/brdpass.txt"), al que se habrá movido el fichero previamente renombrado (passwd.txt - brdpass.txt, en este ejemplo del propio Weins).

VIAS DE ATAQUE

Lo cierto es que (por unas u otras razones) Kriptópolis no ha realizado aún esa sencilla modificación, lo que nos coloca en una situación de vulnerabilidad potencial. ¿Por qué decimos "potencial"? porque aunque nuestro fichero sea accesible a cualquiera (de hecho, se lo hemos dado más arriba, y no vamos a cambiarlo hasta dentro de unos días, luego sabrá por qué...), aún queda cierto trabajo por hacer, antes de poder disfrutar de derechos de administrador sobre nuestro panel de debates. Es sencillo: tiene usted el nombre de usuario (la mitad de lo que necesita); ahora le resta la otra mitad: descifrar nuestra contraseña.

PISTAS

El propio Bernardo Quintero nos dice cómo se cifra la contraseña:

"Recordemos que la función crypt de Unix utiliza la cadena que se pasa de parámetro como clave para realizar el cifrado de un bloque de 64 bits puestos a cero, y repite la operación 25 veces volviendo a cifrar el resultado. Este proceso nos da 64 bits que se representan con 11 caracteres, a los que hay que sumar lo que se conoce como "salt", que es un número de 12 bits que proviene del reloj del sistema, utilizado durante el proceso de cifrado. La "salt" es representada por dos caracteres, y se sitúa delante de la contraseña cifrada, con lo que el resultado final será una cadena de 13 caracteres, como por ejemplo: "GiBzoWz4Y6E1A"."

Y por cierto: no se desanime; la opinión de otros expertos juega en este caso a su favor. Relea lo que dice Bruce Schneier sobre el algoritmo de cifrado Crypt de UNIX, y verá que no da un duro por nosotros:

<http://www.kriptopolis.com/criptograma/cg0016.html#6>

Aún así, no pretendemos retarle a que queme su ordenador (o su cerebro) intentando revertir un proceso como este. Desafortunadamente (para nosotros ;-)) hay métodos de ataque mucho más prácticos, que dejan todo el esfuerzo a programas craqueadores de contraseñas. En nuestro delirio, incluso le propondremos un sitio desde donde puede descargar estas herramientas:

<http://www.vanhackez.com/h/superindex.html>

Simplemente vaya a la sección "Password Crackers" y elija. Eso sí: no nos pida que le proporcionemos -además- ayuda para utilizarlos y reventar nuestra propia contraseña. Eso sería demasiado pedir...

Como última pista, permítame decirle que nuestra contraseña (de nueve caracteres) no es mala, aunque podría haber sido mejor. (Por si acaso: mi perro se llama Yako y nació el 1-8-59).

¡A JUGAR!

Por si aún no lo tiene claro, le estamos proponiendo un juego, consistente en craquear nuestra propia contraseña y tomar el control de nuestro foro de debates:

<http://www.kriptopolis.com/wwwboard/wwwboard.html>

Eso tampoco significa que nos hayamos vuelto rematadamente locos. Nuestra locura es sólo temporal, y acabará el próximo domingo 26 a las 24 horas (00:00 del lunes 27), en que cambiaremos la contraseña y protegeremos adecuadamente el nuevo fichero.

Pero con independencia del aspecto lúdico del asunto, nuestros objetivos son un poco más serios:

- 1) Comprobar hasta qué punto protege una contraseña de cierta calidad.
- 2) Comprobar si las alarmas ante ciertas vulnerabilidades se corresponden siempre con los riesgos reales. No obstante, y como medida de precaución, coincidimos en que los administradores de WWWBoard deberían adoptar cuanto antes las medidas de protección sugeridas, dado lo razonable de las mismas y su indudable eficacia ante contraseñas débiles o inmodificadas tras la instalación.

EL PREMIO

Como es lógico, todo reto tiene un premio. Nos hubiera gustado ser RSA Data Security y poder ofrecerles unos succulentos 10.000 dólares, pero nuestro presupuesto no da para tanto. No obstante, haremos un esfuerzo por darle un premio doble al ganador:

- 1) Por un lado, la satisfacción íntima de vencer y lograr el control ("Show some control", como le gusta decir a la gente de eEye). También tentaremos a la humana vanidad, publicando el nombre del vencedor en

el próximo boletín.

2) Ya, ya sabemos que las recompensas morales son insuficientes, así que hemos pensado regalar al vencedor un premio modesto aunque valioso. Aprovechando la actualización en estos días de nuestro sistema operativo al nuevo Linux SuSE 6.2, regalaremos al vencedor nuestra versión anterior. Se trata de Linux SuSE 5.2, en su caja original, con sus 4 CDs y el manual (en inglés) de casi 450 páginas. Es una versión que tiene algo más de un año, por lo que no parece demasiado indicada para ordenadores mucho más recientes, pero sí correrá muy bien en el resto, aunque debido a la obsolescencia de algunos paquetes creemos que no debería emplearse en funciones delicadas, sin realizar unas cuantas actualizaciones.

En justa correspondencia, tan sólo demandamos un comportamiento noble por parte del posible ganador, comunicándonos cuanto antes su hallazgo y la metodología concreta empleada.

RECAPITULANDO...

1. Kriptópolis convoca oficialmente el reto de lograr controlar nuestro panel de debates, basado en el susceptible WWWBoard y con nuestro fichero de contraseña todavía accesible.
2. El objetivo de este reto es comprobar si una contraseña adecuada puede ser protección suficiente ante cierto tipo de vulnerabilidades.
3. El plazo finaliza el próximo domingo 26 de septiembre a las 12 de la noche. A partir de ese momento no podrán existir ganadores, ya que cambiaremos de contraseña y la protegeremos adecuadamente.
4. El premio obtenido por el ganador consistirá en publicar su nombre en el próximo boletín y en el paquete original y completo de la distribución SuSE Linux 5.2.
5. Con independencia del resultado de nuestro reto y de su carácter lúdico y experimental, la vulnerabilidad debería ser tomada en serio por quienes administren foros de debate basados en WWWBoard.

REFERENCIAS

* Detalles de WWWBoard:

<http://www.worldwidemart.com/scripts/wwwboard.shtml>

* Mensaje original de David Weins en BugTraq:

<http://www.securityfocus.com/templates/archive.pike?list=1&date=1999-09-15&msg=Pine.LNX.4.10.9909170435200.30548-100000@puffer.quadrunner.com>

* La noticia en Hispasec:

<http://www.hispasec.com/unaaldia.asp?id=327>

Del Boletín #136:

"LIBERTAD, LIBERTAD, LIBERTAD."

Nuevas Tecnologías de la Información y Derecho Penal.

Una crónica de Carlos Sánchez Almeida, sanchezalmeida@bufetalmeida.com

Los días 27, 28 y 29 de septiembre tuvo lugar en Barcelona, bajo la dirección de Fermín Morales Prats, catedrático de Derecho Penal de la Universidad Autónoma de Barcelona, un seminario sobre Nuevas Tecnologías de Información y Derecho Penal. El evento fue organizado por el Consorcio Universidad Menéndez Pelayo de Barcelona, en colaboración con la Universitat Oberta de Catalunya y bajo el patrocinio de la Cámara Oficial de Comercio, Industria y Navegación de Barcelona.

Susan F. Wilson, Fiscal de la Oficina de Delitos Informáticos del Departamento de Justicia de los EE.UU., dio algunas pistas de la situación real en la persecución del delito informático. Las mayores preocupaciones de la justicia americana parecen ser la seguridad del sistema financiero y el temor a un crack bursátil forzado desde Internet, riesgos que en todo momento precedieron en su exposición a los sistemas de control de aeropuertos, seguridad nuclear y defensa nacional. Informó de las reuniones de expertos del G-8, que intercambian información en conexión permanente, y abogan por la armonización de todos los códigos penales. De su simpatía por la industria americana -en colaboración permanente con el Departamento de Justicia, según indicó- fue una clara muestra su preferencia por sistemas Windows, que indicó -textualmente- son más seguros que los de

código abierto. Por respeto a la seriedad académica, este cronista tuvo que contener las carcajadas.

El profesor Vincenzo Zeno-Zencovich, catedrático de la Universidad de Sassari, abogó por un estatuto jurídico de los ISP, así como evitar respuestas penales innecesarias, en lo relativo al hacking, apostando por invertir en medidas de seguridad y encriptación.

Que el Real Decreto Ley 14/1999, por el que se regula la firma electrónica, era el disparo de salida de una feroz carrera en la toma de posiciones, lo sabíamos todos. Jordi Masias aprovechó un hueco del programa para publicitar la solución Camerfirma, servicio de certificación digital de las Cámaras de Comercio, uno de cuyos mayores rivales, según las malas lenguas, parece ser FESTE, la autoridad notarial de firma digital. Jordi Buch, de Safelayer, desarrolló una ilustrativa ponencia técnica sobre seguridad en Internet, firewalls, encriptación y firma digital.

El plato fuerte de la primera jornada fue la mesa redonda sobre la experiencia policial y jurisdiccional sobre delito informático, a la que asistieron José María Álvarez Cienfuegos, Magistrado, Emilio Aced, Subdirector General de Protección de Datos, el Fiscal José Ernesto Fernández Pinós y el Capitán Anselmo del Moral, Jefe de la Unidad de Delitos de Alta Tecnología de la Guardia Civil. A los dos últimos, este cronista les tiene un aprecio especial por su actuación en el caso !Hispahack, que fue objeto de un ameno debate.

El Capitán expresó las dificultades con que se encuentra a diario en sus investigaciones, derivadas de la dispersión territorial de los delitos informáticos, que obligan a intervenir a múltiples Juzgados, generando problemas de coordinación. En este sentido, abogó por la creación de una Fiscalía especializada en Delitos Informáticos. No es una mala idea -el subrayado es mío- siempre que esa especialización se extienda también a los Juzgados y al Turno de Oficio.

Las palabras del Fiscal José Ernesto Fernández Pinós sobre la copia de software sin ánimo de lucro, no gustarán en las oficinas de la BSA. En la opinión del ilustre representante del Ministerio Público, no es delictiva en ningún caso la copia para uso privado. Ya sé a quién le debo muchos archivos por falta de denuncia.

La mañana del martes estuvo presidida por la intervención del Director de la Agencia de Protección de Datos, Juan Manuel Fernández López, al que una ciberperiodista gamberra -M&M, no me demandes- tuvo la mala idea de preguntarle por los ficheros de passwords. El encargado de la mayor autoridad española en protección de la privacidad, se vió en el apuro de solicitar auxilio técnico, dado que ignoraba el significado

de las palabras "fichero de passwords". Mayores conocimientos demostró en el ámbito jurídico, realizando una excelente exposición acerca de la LORTAD.

Lorenzo Picotti, de la Universidad de Trento, y Ulrich Sieber, catedrático de Würzburg, coincidieron en la necesidad de una armonización internacional para evitar "paraísos de datos". El profesor alemán, interrogado sobre ENFOPOL, se reveló como un paladín de las libertades, apostando por el necesario control jurisdiccional de la actividad policial, para garantizar los derechos civiles en la investigación del delito informático.

El martes por la tarde estuvo dedicado a la situación en Internet de la propiedad intelectual e industrial. En una docta ponencia, el profesor Oscar Morales García, de la Universitat Oberta de Catalunya, ofreció una visión progresista de la tensión dialéctica entre la libertad de acceso a la información y la necesidad de proteger los derechos de autor. También destacó su defensa de la figura del hacker frente a crackers y vendedores de software ilegal (warez).

La protección civil de la propiedad intelectual en Internet fue ampliamente debatida por el profesor Amadeu Abril, el abogado Sergio Miralles y Borja Adsuara, Jefe de Gabinete de la Secretaria de Estado de Cultura. La discusión derivó al terreno de la filosofía del derecho, coincidiendo los ponentes en que no puede analizarse Internet con los esquemas mentales del pasado. Un mundo nuevo requiere un derecho nuevo, y en este sentido el doctor Miralles alertó del peligro de que sean las grandes corporaciones las que impongan su ley, y se inclinó, antes que por un derecho de la Red, por un sistema de arbitraje alternativo.

Los pesos pesados del Derecho Penal se dieron cita en la mañana del día 27. El profesor Ramón García, de la Universidad de Lleida, dio una clase magistral sobre estafa informática, utilización ilegítima de tarjetas de crédito y falsedades documentales, con una especial mención a la falsificación del documento electrónico. El temido hacking bancario -casi nunca denunciado: el mayor tesoro de un banco es el sigilo y la confianza del cliente- fue estudiado en profundidad, revelándose la insuficiencia de su actual punición: sale más rentable frente al atraco clásico, tanto económicamente como en años de cárcel.

Francesc Baldó, profesor de Derecho Penal de la Universidad de Barcelona, y avezado usuario de Linux, lanzó un valiente desafío a los apóstoles del mercado. Enmarcó el delito informático en su contexto tecnológico, político y económico, concluyendo que no debe perseguirse únicamente a adolescentes rebeldes, verdaderos cabezas de turco. Preguntó a la audiencia si no debería perseguirse a aquellos que

distribuyen productos mal desarrollados, con fallos de seguridad denunciados por betatesters, pero que no se mejoran por razones de marketing. Productos que causan daños informáticos, pérdida de datos y paralización de cadenas productivas. Puso el dedo en la llaga al preguntarse y responder: "¿Por qué son vulnerables los sistemas? Porque si se reforzase la seguridad, los gobiernos no podrían espiar."

Este cronista es un sentimental. Perdonen si soy poco objetivo, pero es que estos días no he parado de opinar; un poco más y no me dan el diploma. Me quedo con tres momentos. El particularísimo sentido del humor, que lo tiene, de Anselmo del Moral: "El hacking, sea blanco, negro o rosa, es intrusismo informático". Su reproche a determinados proveedores de Internet: "Los ISPs entregan datos que la Guardia Civil nunca ha pedido. Yo nunca he solicitado un password sin orden del Juez". Y la vibrante voz de la independencia judicial, encarnada en el Presidente de Sala de la Audiencia Nacional, José María Álvarez Cienfuegos, al citar la Sentencia del Tribunal de Pensilvania sobre la Ley de Decencia en las Telecomunicaciones: "Internet es una conversación universal sin fin: el Gobierno no puede intervenir esa comunicación. La fuerza de Internet está constituida por el Caos. Libertad, libertad, libertad".

Carlos Sánchez Almeida
<http://www.bufetalmeida.com>

DISPONIBLES NUEVOS DOCUMENTOS ELECTRÓNICOS

* La traducción al español del libro "The Hacker Crackdown" ya está disponible en la Red. Este auténtico clásico de la cibercultura, que narra la caza y captura de un grupo de hackers, y cuya versión española pudo elaborarse gracias a la ayuda de un grupo de voluntarios reclutados por David Casacuberta a través de nuestro boletín, ya está disponible desde: <http://www.globaldrome.org/textos/hackercrack/>

* Y entre las novedades en línea, no queremos dejar de reseñar el documento titulado "Cómo sellar y firmar un documento según el Decreto Ley del Gobierno", cuya disponibilidad ha comunicado a Kriptópolis Jorge Dávila, de CriptoLab:

<http://tirmanog.ls.fi.upm.es/CriptoLab/Proyectos/TicTac/FechayFirma.htm>

* También merece una reseña el tutorial en español sobre SSL 3.0 y TLS 1.0 que han elaborado Gabriel Belingueres, Luciana Balbi y Sandra

Serafino:

<http://www.geocities.com/SiliconValley/Byte/4170/articulos/tls/index.htm>

* Actualizados con la segunda edición los sitios de descarga del libro de Criptografía de Manuel Lucena:

<http://www.kriptopolis.com/libro.html>

* También continúa la traducción del documento sobre Tempest y pronto dispondremos de más libros electrónicos totalmente gratuitos en nuestro web. Pronto seremos otra vez noticia...

Del Boletín #138:

UN ESPAÑOL DESCUBRE UNA NUEVA VULNERABILIDAD EN WINDOWS

* AUTOR:

Kput --NGN Team (España)--

* DESCRIPCIÓN

Esta vulnerabilidad responde a una característica de diseño y afecta básicamente al modo en que Windows considera los tipos de archivo a través de su registro (el registro de sistema).

* SISTEMAS AFECTADOS

Por el momento sólo he verificado la vulnerabilidad en Windows 95 y NT 4. No obstante, la estructura del registro de Windows hace casi segura la extensión de esta vulnerabilidad a la casi totalidad del resto de sistemas operativos de la casa Microsoft (Windows 98, NT 3.5 y posteriores). Tan sólo Windows CE parece estar a salvo, por emplear una estructura diferente en el registro de sistema (A pesar de ello, puede ser posible la extensión de la vulnerabilidad adaptándola a la estructura de CE, por su analogía al resto de sistemas operativos Windows).

* IMPLICACIONES

La seguridad de Windows queda comprometida por la posibilidad de la ejecución arbitraria de archivos ejecutables en nuestro sistema, sin Conocimiento del usuario.

* DETALLES TÉCNICOS Y EJEMPLO

Como siempre, un ejemplo es el mejor punto de partida para una explicación. En este caso, un sencillo ejemplo funcional sería:

```
----- Cortar por aquí, guardar como TEST.REG -----
REGEDIT4
[HKEY_CLASSES_ROOT\jjj]
@="jjjfile"
[HKEY_CLASSES_ROOT\jjjfile]
"NeverShowExt"=""
@="Imagen Jpeg"
"EditFlags"=hex:01
[HKEY_CLASSES_ROOT\jjjfile\shell]
[HKEY_CLASSES_ROOT\jjjfile\shell\DefaultIcon]
@="%1"
[HKEY_CLASSES_ROOT\jjjfile\shell\open]
[HKEY_CLASSES_ROOT\jjjfile\shell\open\command]
@="\"%1\" %*"
----- Cortar por aquí -----
```

Bien; ahora vayamos a la parte que más interesa: el "exploit" propiamente dicho. La ejecución del archivo TEST.REG resultante (mediante doble click o llamándolo bajo línea de comandos), hará aparecer una ventana pseudo-informativa con un mensaje que en principio nos puede parecer inofensivo: "La información de TEST.REG ha sido introducida satisfactoriamente en el registro". Sobra decir que, para un usuario medio, esto es poco menos que nada. Aún así, más adelante veremos que incluso esta ventana de atención puede ser suprimida, haciendo todavía más imperceptible nuestro ataque.

De hecho, todavía no hemos hecho nada que se salga del espectro de lo "normal". Tan sólo hemos añadido un nuevo tipo de archivo al registro de Windows. Pero repasemos línea a línea la información introducida y comprenderemos que el compromiso a nuestra seguridad radica en que este nuevo tipo de archivo es en realidad un ejecutable "oculto". Así, podemos renombrar un programa (.com, .bat, .pif o .exe) y dotarlo de la extensión .jjj (por ejemplo: NETBUS.EXE a Marta.jpg.jjj). De este modo, nuestro potencialmente peligroso archivo ejecutable, conteniendo un agresivo troyano, se convierte en un inofensivo archivo de imagen,

que cualquiera nos puede pasar por disquete, en un CD o por IRC. Lógicamente, no esperamos encontrar virus en una foto (¿¿; Virus en una foto!?) así que no vamos a perder el tiempo sometiendo la imagen a nuestro software antivirus. Como mucho, y sólo si somos especialmente curiosos, quizás se nos ocurra comprobar las propiedades del archivo en cuestión. No hay problema; verificamos que se trata de una "Imagen JPEG". ¿Alguien sospecharía de un archivo jpeg?.

No ha costado demasiado proporcionar este camuflaje al fichero. La línea "NeverShowExt"="" de nuestro archivo de ejemplo TEST.REG indica a Windows que oculte el sufijo final de los archivos del nuevo tipo que hemos creado, sufijo que normalmente identifica el tipo de archivo (como los .doc para documentos de Windows, .mp3 para canciones, etc). La clave del tipo de archivo "jjjfile", dentro de \open\command], al añadir la línea @="\"%1\" %*", provoca la ejecución de los archivos de este tipo como la acción predeterminada al hacer doble click sobre el archivo o llamarlo desde la línea de comandos.

Quizás alguien piense que la línea DefaultIcon nos pueda ayudar, pero cualquier programador novel (o usuario medianamente experimentado) sabe lo sencillo que es cambiar en Windows el icono asociado a un archivo ejecutable, escogiéndolo nosotros mismos en caso de haberlo compilado por nuestra cuenta o mediante la ayuda de utilidades (como pueda ser un editor de recursos). He verificado que varios programas antivirus (McAfee ViruScan, Norton y Panda, aunque seguramente todos o casi todos los existentes en el mercado), escanean de manera estándar tan sólo los archivos considerados potencialmente peligrosos, esto es, aquellos en los que es factible la existencia o inclusión de código ejecutable. Es evidente que esa lista por defecto no incluye el nuevo tipo de archivo jjj que hemos creado. De este modo, un escaneo no exhaustivo del sistema, no revelaría la presencia de virus, troyanos u otros en los archivos modificados. Peor aún es el caso del antivirus de McAfee que, por alguna extraña razón, no escanea los tipos de archivo no ejecutable dentro de los archivos comprimidos, a menos que se lo digamos dos veces (la primera vez hará caso omiso de nuestra orden y no escaneará).

* OTRAS VARIANTES DEL ATAQUE

Por si todo lo dicho no fuera suficientemente preocupante, también podemos aprovecharnos de otras características "de diseño" de Windows que pueden camuflar aún mejor nuestro ataque:

- 1) El envío vía Internet (por ejemplo, IRC o cualquier navegador web gráfico) de archivos de nombres muy largos, impide al usuario ver el sufijo de extensión final del archivo.

2) Otra opción interesante, sin duda la más peligrosa y camuflada, es la llamada y modificación del registro a través de un programa, ya que la adición de un nuevo tipo de archivo al registro de sistema no es considerada una acción potencial ni directamente peligrosa por ningún software de seguridad/protección/antivirus. También podemos impedir que aparezca la ventana de información que indica que hemos añadido información al registro de Windows (la mencionada anteriormente al principio de la explicación) mediante el uso de un fichero .bat, realizando una llamada desde línea de comandos al editor de registro de sistema (regedit.exe) con el modificador /S (es decir: regedit /s test.reg). Otra opción es la modificación del archivo objetivo para que haga una llamada al programa asociado a la extensión visible de nuestro archivo (que -recordemos- es falsa), como puedan ser WangImg, IrfanView, AcdSee o Photoshop para la extensión ".jpg" de nuestro ejemplo. Haciendo uso de esta posibilidad, la víctima tendrá aún menos razones para sospechar del archivo objetivo: sólo se ve un archivo llamado Marta.jpg , y al hacer doble click, como es lógico, Photoshop se abre para mostrar la foto que alguien nos ha pasado por el IRC.

* CONSECUENCIAS

El peligro que toda esta información involucra es enorme, puesto que existen cientos de posibilidades y variantes que pueden aprovechar esta vulnerabilidad para comprometer la seguridad de nuestro sistema. Mencionaremos a continuación un par de ellas:

- Un programa shareware/freeware que añade una o varias nuevas claves al registro de sistema es -en apariencia- inocuo. Ningún antivirus hará sonar las alarmas por algo así. Tampoco saltará por la existencia de un archivo .jpg.jpg en el mismo paquete de instalación comprimido del programa. Pero una vez la clave ha sido creada en el registro de sistema, el archivo se vuelve un programa *EJECUTABLE*. No creo que a nadie le agrade la idea, pero cualquier archivo descargado a través de Internet permite, de ahora en adelante, la inclusión de puertas traseras, virus o nuevos tipos ejecutables a ser ejecutados en el sistema sin el conocimiento del usuario.

- Los archivos modificados mediante este procedimiento pueden tener también otra "utilidad": permitir el almacenamiento de archivos ejecutables (u otros), en servidores donde eso no se permite a priori, como puedan ser universidades, servidores de empresas, o servidores que ofrezcan hospedaje gratuito.

* SOLUCIÓN

En principio, no existe otra solución que no sea vigilar nuestro sistema. Considero inviable una modificación en la estructura del registro a la hora de tratar los diferentes tipos de archivo en Windows, ya que ello implicaría grandes dificultades a la hora de mantener la compatibilidad "hacia atrás" con programas anteriores, y el hecho de que Windows 2000, todavía en desarrollo, continúe esta política de uso del registro, hace esta posible modificación aún más improbable. Un punto a tener en cuenta a este respecto es el hecho que el clon GNU de Windows NT en desarrollo, React OS, parece que estará a salvo de esta vulnerabilidad, puesto que empleará una estructura diferente en su registro de sistema, compatible no obstante con su homólogo de pago (Para más información visitar la página oficial de ReactOS, <http://www.reactos.com>).

No obstante, existe un modo de evitar, al menos parcialmente, la posibilidad de compromiso de nuestro sistema con este procedimiento: en la ventana de 'Propiedades' del archivo, el nombre del archivo bajo MS-DOS nos muestra su verdadera extensión. Puede ser una tarea realmente lenta, aburrida y molesta, pero en principio es la única vía que nos permite verificar la autenticidad del archivo.

* ACCIONES TOMADAS

Esta información fue notificada a Microsoft por su descubridor el pasado 29 de Agosto de 1999, sin que por el momento se haya recibido respuesta sobre una posible solución.

UNDERCON III: ALTO NIVEL EN I+D

(Por Carlos Sánchez Almeida, sanchezalmeida@bufetalmeida.com)

Sin patrocinadores, sin cámaras, sin periodistas. El largo puente del Pilar fue aprovechado por lo más granado del underground patrio para citarse en la UnderCon III, y compartir los conocimientos adquiridos desde su última convocatoria. Representantes de 29A, CPNE-organizadores del evento-, !Hispahack, NPT, SET, TCD y TDD se reunieron en Murcia para presentar sus últimos descubrimientos en el campo del underground informático.

Las más avanzadas técnicas de phreaking fueron el tema estrella de la reunión, toda una muestra de I+D en la última tecnología telefónica.

Sobre las mesas de la sala de reuniones podían observarse todo tipos de aparatos diseñados para conectarse a la Red de forma anónima y al mínimo coste. Portátiles de última generación, subnotebooks y PDAs compartían su cableado con ingeniosos sistemas de circuitería. Buen número de las ponencias expuestas denotaban el alto nivel investigador de los grupos españoles.

El phreaking, herramienta imprescindible de todo buen hacker, dio paso a ponencias muy elaboradas sobre criptografía y comercio electrónico, protección de routers y diseño de virus para Win32. Esta última ponencia fue presentada por GriYo, de 29A, posiblemente el mejor programador de virus de la escena, además de excelente contertulio.

También tuvo lugar un animado debate sobre la legislación aplicable a los delitos informáticos, con preguntas que dejarían fuera de juego a más de un catedrático: ¿La localización y posterior uso de una cuenta gratuita, es delito? ¿Se puede considerar revelación de secretos el uso de passwords cedidos por terceros? ¿Si toda la familia usa el pc, cómo identificar a un hacker?

Hicieron acto de presencia en las jornadas representantes de distintas empresas, a la búsqueda de nuevos talentos. Todo un síntoma del creciente interés empresarial por la seguridad informática y la investigación en nuevas tecnologías: la puesta de largo del mal llamado lado oscuro, como valor en alza de un mercado cada vez más competitivo.

Tras largas sesiones de debate, los distintos grupos aprovecharon las excelencias de la gastronomía murciana para confraternizar con personas que, en algún caso, sólo se conocían por IRC. La visita de rigor a los locales de copas sirvió también para romper falsos tópicos sobre el presunto autismo de los hackers. La mascota de las jornadas fue Ossy, el rottweiler de GriYo, que al igual que su amo, cumple a rajatabla con la legislación protectora de especies peligrosas.

Carlos Sanchez Almeida
<http://www.bufetalmeida.com>

Del Criptograma #17:

1. CÓDIGO ABIERTO Y SEGURIDAD

Por Bruce Schneier

Traducción: Angel Galindo Sánchez

Como criptógrafo y experto en seguridad informática, nunca he comprendido el alboroto actual sobre el movimiento de software de código abierto. En el mundo de la criptografía se considera que el código abierto es necesario para tener un buen nivel de seguridad; y se ha considerado así durante décadas. La seguridad pública es siempre más

segura que la seguridad privada. Esto es cierto para los algoritmos criptográficos, para los protocolos de seguridad y para el código fuente de seguridad. Para nosotros, el código abierto no es sólo un modelo de negocio, sino una práctica de ingeniería adecuada.

* Criptografía de código abierto

La criptografía ha adoptado los ideales del código abierto durante décadas, aunque lo llamemos "utilizar algoritmos y protocolos públicos".

La idea es simple: la criptografía es difícil de hacer correctamente, y la única forma de saber si algo está bien hecho es ser capaz de comprobarlo.

Esto es vital en criptografía, porque la criptografía no tiene nada que ver con la funcionalidad. Puedes tener dos algoritmos, uno seguro y el otro inseguro, y ambos pueden funcionar perfectamente. Ambos pueden cifrar y descifrar, son eficientes, tienen un bonito interfaz de usuario y nunca se cuelgan. La única forma de distinguir la buena de la mala criptografía es examinándola.

Aún peor, no se consigue nada teniendo un montón de gente cualquiera

examinando el código; la única forma de distinguir la buena criptografía de la mala es examinándola por expertos. Analizar criptografía es duro, y hay pocas personas en el mundo que puedan hacerlo de forma competente.

Antes de que un algoritmo pueda considerarse realmente seguro, necesita ser examinado por muchos expertos a lo largo de muchos años.

Este es un fuerte argumento en favor de los algoritmos de código abierto. Dado que la única forma de tener confianza en un algoritmo es tener expertos que lo examinen, y la única forma en que ellos emplearán el tiempo necesario para examinarlo adecuadamente es permitirles que

publiquen documentos técnicos de investigación sobre él, el algoritmo debe ser público. Un algoritmo privado, no importa quién lo haya diseñado o a quién se haya pagado para examinarlo, es mucho más arriesgado que un algoritmo público.

El contra-argumento que a veces se oye es que la criptografía secreta es más fuerte porque es secreta, y que los algoritmos públicos son más arriesgados porque son públicos. Esto suena bastante lógico, hasta que te paras a pensar un momento sobre ello. Los algoritmos públicos están diseñados para ser seguros incluso aunque sean públicos; así es como

están hechos. Por tanto, no hay ningún riesgo en hacerlos públicos. Si un algoritmo sólo es seguro si permanece secreto, solamente lo será hasta que alguien utilice ingeniería inversa contra él y lo publique.

Una gran variedad de algoritmos secretos para telefonía móvil digital han sido hechos públicos y rotos rápidamente, ilustrando la futilidad de este argumento.

En lugar de utilizar algoritmos públicos, las compañías norteamericanas de telefonía móvil decidieron crear su propia criptografía privada. Durante los últimos años, diferentes algoritmos de estos se han hecho públicos. (No, las compañías telefónicas no querían que se hicieran públicos; lo que generalmente ocurre es que algún criptógrafo recibe sus especificaciones en un paquete anónimo). Una vez que se ha hecho público, ya está roto. Ahora la industria de telefonía móvil americana está considerando el utilizar algoritmos públicos para reemplazar a sus rotos algoritmos privados.

Por otro lado, el popular programa de encriptación de correo electrónico PGP siempre ha utilizado algoritmos públicos. Y ninguno de esos algoritmos ha sido roto jamás. Esto mismo puede decirse de otros protocolos de Internet: SSL, S/MIME, IPSec, SSH, etcétera.

* La mejor evaluación no se compra con dinero

Actualmente el gobierno norteamericano está eligiendo un algoritmo de encriptación para sustituir a DES, llamado AES (Advanced Encryption Standard, Norma de Cifrado Avanzada). Hay cinco candidatos a la norma, y, antes de elegir el definitivo, los mejores criptógrafos del mundo pasarán horas evaluándolos. Ninguna empresa, no importa lo rica que sea, puede permitirse este tipo de evaluación. Y dado que AES es gratuito para cualquier utilización, no hay ninguna razón para que otra empresa ni siquiera se plantee crear su propio algoritmo. La criptografía abierta no es sólo la mejor, sino que además es la más barata.

El mismo razonamiento que lleva a las empresas inteligentes a

utilizar criptografía pública también les lleva a utilizar protocolos de seguridad públicos: cualquiera que cree su propio protocolo de seguridad, es un genio o un loco. Dado que hay más de los últimos que de los primeros, utilizar protocolos públicos es más inteligente.

Consideremos IPSec, el protocolo de seguridad de Internet. A principios de 1992, fue diseñado abiertamente por un comité y fue objeto de numerosos escrutinios públicos desde sus inicios. Todo el mundo sabía que era un protocolo importante y la gente puso mucho esfuerzo tratando de hacer las cosas bien. Diferentes tecnologías de seguridad fueron propuestas, rotas y después modificadas. Se codificaron y analizaron

varias versiones. El primer boceto de la norma se publicó en 1995. Se debatieron diferentes aspectos de IPSec, como sus méritos en seguridad, comportamiento, facilidad de implementación, capacidad de ampliación y uso.

En noviembre de 1998, el comité publicó una serie de documentos - el primer paso de todo un proceso para hacer de IPSec un estándar en Internet. Y todavía sigue estudiándose. Los criptógrafos del Laboratorio de Investigación Naval recientemente descubrieron un pequeño error de implementación. El trabajo continúa, en público, para y por cualquiera que esté interesado. El resultado, basado en años de análisis público, es un protocolo fuerte en el que confía la mayoría de la gente. Por otro lado, Microsoft desarrolló su propio protocolo PPTP para hacer lo mismo. Inventaron su propio protocolo de autenticación, su propia función hash, y su propio algoritmo de generación de llaves. Todas y cada una de estas implementaciones resultaron ser defectuosas.

Utilizaron su propio algoritmo de cifrado, pero lo utilizaron de una manera que impedía su propia seguridad. Tuvieron errores de implementación que debilitaron todo el sistema aún más. Pero como hicieron todo este trabajo internamente, nadie sabía que PPTP era débil. Microsoft instaló PPTP en Windows NT y Windows 95, y lo utilizaron en los productos de su red privada virtual (VPN). A veces publicaron sus protocolos y, en el verano de 1998, la compañía para la que trabajo,

Counterpane Systems, publicó un documento técnico describiendo los errores que encontramos. Una vez más, el escrutinio público mostró su valor. Microsoft rápidamente sacó una serie de parches que evaluamos ese verano y vimos que eran buenas mejoras, pero aún encontramos fallos. Como en los algoritmos, la única forma de distinguir un protocolo de seguridad bueno de uno que está roto, es tener expertos que lo examinen. Por tanto, si necesita utilizar un protocolo de seguridad, será mucho más inteligente utilizar uno que ya haya sido evaluado. Puede crearse el suyo propio, pero, ¿cuáles son las probabilidades de que sea tan seguro como otro que ya ha sido probado por expertos en los últimos años?.

* Asegurando su código

Exactamente el mismo razonamiento lleva a cualquier ingeniero de seguridad inteligente a demandar código abierto para cualquier aspecto relacionado con la seguridad. Repasemos: la seguridad no tiene nada que ver con la funcionalidad. Por tanto, ningún nivel de pruebas de versiones beta podrá nunca descubrir un fallo de seguridad. La única

forma de encontrar fallos de seguridad en un segmento de código -como un algoritmo criptográfico o un protocolo de seguridad- es evaluarlo. Esto es cierto para cualquier código, tanto si es abierto como privado. Y no puedes hacer que cualquiera te evalúe el código, sino que necesitas que lo hagan expertos en software de seguridad. Necesitas que lo evalúen varias veces y desde diferentes perspectivas, durante el transcurso de varios años. Se puede conseguir contratar a esta clase de expertos, pero es mucho más barato y efectivo dejar que toda una comunidad lo haga. Y la mejor forma de conseguirlo es hacer público el código. Pero si quieres que tu código sea realmente seguro, necesitará hacer algo más que simplemente publicarlo bajo una licencia de código público. Hay dos cuestiones que deberá tener siempre presentes. Primero, la simple publicación del código no significa que la gente lo examinará para buscar sus fallos. Los investigadores de seguridad son personas inconstantes y ocupadas. No tienen tiempo de examinar todas las muestras de código que se publican. Por tanto, aunque hacer público el código es bueno, no es una garantía de seguridad. Podría nombrar una docena de librerías de seguridad de código abierto de las que nunca nadie ha oído hablar jamás o que nunca han sido evaluadas. Por contra, el código de seguridad de Linux ha sido comprobado por un montón de excelentes ingenieros de seguridad.

Segundo, necesita asegurarse de que los fallos de seguridad sean corregidos tan pronto como se hagan públicos. La gente encontrará errores en el código. Esto es bueno. No hay razón para pensar que el código abierto, en el momento de escribirlo, es más seguro que el código privado. El objetivo de hacer que sea abierto es conseguir que mucha, mucha gente analice el código buscando sus fallos, y los encuentre. Rápido. Hay que corregir los errores encontrados. Un código público con dos años de antigüedad tendrá muchos menos fallos de seguridad que un código privado, simplemente porque ya se habrán encontrado y corregido muchos. Los fallos de seguridad también se descubrirán en códigos privados, pero a un ritmo mucho menor.

Comparar la seguridad de Linux con la de Microsoft Windows no es muy instructivo. Microsoft ha hecho un trabajo tan malo con la seguridad que no es en realidad una comparación justa. Pero comparar Linux con Solaris, por ejemplo, es más instructivo. La gente está encontrando

problemas de seguridad más rápidamente en Linux, y se están corrigiendo a un ritmo mucho mayor. El resultado es un sistema operativo que, incluso aunque sólo tiene unos pocos años de antigüedad, es mucho más robusto de lo que Solaris era a su edad.

* Programas seguros

Uno de los grandes beneficios del movimiento de código abierto es el efecto de realimentación positiva que tiene la publicidad. Entre en un gran almacén de ordenadores es estos días, y verá estanterías enteras de productos basados en Linux. La gente los compra, porque el uso de Linux ya no está limitado a los expertos; es una herramienta útil en muchas aplicaciones. El mismo bucle de realimentación funciona en seguridad:

los algoritmos y protocolos públicos ganan credibilidad porque se conocen y se utilizan, y por tanto se convierten en la moda. No es un modelo perfecto, pero es mejor que la alternativa.

2. ¿CLAVE DE LA NSA EN MICROSOFT CRYPTO API?

Por Bruce Schneier

Traducción: Isidre Marques Serret

Hace unos meses, hablé del sistema de Microsoft para firmar digitalmente las librerías criptográficas que se incluyen en su sistema operativo. El detalle importante es que solo podrán utilizarse las librerías criptográficas firmadas, lo cual hace que cosas como el control de la exportación sean más fáciles. Molesto como es, es el mercado actual. Microsoft tiene dos de claves, una primaria y una de reserva. El artículo de CriptoGrama hablaba de ataques basados en el hecho de que una librería criptográfica se considera firmada si está firmada por CUALQUIERA de las claves, y que no hay ningún mecanismo concreto para pasar desde la clave primaria a la de reserva. Es criptografía estúpida, pero es el tipo de cosas que podríamos esperar de Microsoft.

De repente hay un estallido de actividad de la prensa porque alguien se da cuenta que la segunda clave en la API criptográfica de Microsoft en el Service Pack 5 de Windows NT tiene el nombre de "NSAKEY" en el código. ¡Ajá! La NSA puede firmar librerías criptográficas. Puede usar esta capacidad para instalar una librería criptográfica Troyana en nuestro ordenador. Así la teoría de la conspiración esta en marcha.

No me lo trago.

Primero, si la NSA quisiera comprometer la API criptográfica de Microsoft, le resultaría mucho más fácil 1) convencer a MS para que les proporcionara la clave secreta correspondiente a la clave de

firmado, 2)
conseguir que MS firmara un módulo amañado para la NSA, o 3)
instalar un módulo diferente a la API criptográfica para romper el cifrado (ninguna otra de las librerías necesita firma). Siempre es más fácil romper un buen cifrado atacando el generador de números aleatorios que realizando un ataque del tipo fuerza-bruta a la clave.

Segundo, la NSA no necesita una clave para comprometer la seguridad en Windows. Los programas como el Back Orifice puede hacerlo sin ninguna clave. Para atacar la API criptográfica todavía se necesita que la víctima ejecute un programa (o una macro de Word) en su ordenador. Si puede convencerse a la víctima para que ejecute un programa que no sea de confianza, hay un billón de maneras más inteligentes para comprometer seguridad.

Tercero, ¿por qué llamaría alguien a una clave secreta de la NSA "NSAKEY"? Una gran cantidad de gente tiene acceso al código original

dentro de Microsoft; una conspiración como esta solo debería ser conocida por muy poca gente. Alguien con un debugger podría encontrar esta "NSAKEY." Si esto pretende ser un mecanismo secreto, no es muy secreto.

Veo dos posibilidades. Uno, que la clave es lo que Microsoft dice, una clave de reserva. Se llama "NSAKEY" por alguna estúpida razón, y eso es todo.

Dos, que sea realmente un clave de la NSA. Si la NSA pretende utilizar productos Microsoft para sus comunicaciones secretas, van a instalar sus propias librerías criptográficas. No van a querer mostrárselas a nadie, ni siquiera a Microsoft. Querrán firmar sus propias librerías. Así que la clave de reserva podría ser una clave interna de la NSA, para poder instalar sistemas fuertes de criptografía sobre los productos de Microsoft para su propio uso interno.

Pero no es una clave de la NSA para que puedan debilitar en secreto la criptografía de las masas desprevenidas. Simplemente hay demasiadas cosas más inteligentes que pueden hacer a las desprevenidas masas.

3. COUNTERPANE SYSTEMS: INVESTIGACIÓN DOCUMENTADA

Por Bruce Schneier
Traducción: Miguel Camacho

"Criptoanálisis sobre las extensiones Microsoft de autenticación PPTP (MS-CHAPv2)"

Por Bruce Schneier y Mudge, CQRE, Duesseldorf, (en preparación) Oct 1999.

El protocolo de direccionamiento punto a punto [Point-to-Point Tunneling Protocol (PPTP)] es usado para conexiones PPP seguras sobre enlaces basados en TCP/IP. En respuesta a [SM98], Microsoft publicó extensiones al mecanismo de autenticación de PPTP (MS-CHAP), llamadas MS-CHAPv2.

Ofrecemos una visión general de los cambios en la autenticación y generación de claves de MS-CHAPv2, y evaluamos las mejoras y las debilidades que permanecen en la implementación Microsoft del PPTP. Mientras se arreglan algunos de los más notorios errores de MS-CHAPv1, el nuevo protocolo todavía adolece de algunas de las mismas debilidades.

<http://www.counterpane.com/pptpv2-paper.html>

4. NOTICIAS

Por Bruce Schneier
Traducción: Miguel Camacho

* El proyecto de auditoria de Internet. Es realmente interesante. Un

grupo realizó una auditoria de seguridad de bajo nivel sobre 36 millones de hosts en internet. ¿De manera imparcial, cómo de segura es en realidad Internet?

http://www.securityfocus.com/templates/forum_message.html?forum=2&head=32&id

http://www.internetnews.com/intl-news/print/0,1089,6_184381,00.html

Y si eso no es suficiente espeluznante, aqui hay una más detallada auditoría sobre 2200 sitios de Internet:

<http://www.fish.com/survey/>

* Mi siempre favorita declaración de conformidad con el 2000 (Y2K):

<http://www.hartscientific.com/y2k.htm>

* Si necesita más pruebas de que la seguridad de formato propietario no funciona, el formato de seguridad Microsoft de musica digital ha sido reventado durante los dias de su presentación.

<http://www.wired.com/news/news/technology/story/21325.html>

<http://www.news.com/News/Item/0,4,0-40672,00.html?st.ne.lh..ni>

<http://www.msnbc.com/news/302195.asp>

* Chantaje de patentes: Abogados de alguien llamado Leon Stambler han estado enviando cartas amenazantes a compañías de seguridad, reclamando que SSL, PCK, FIPS 196, SET, Microsoft PPTP, Authenticode, etc. infringen su patente. Compruébelo por sí mismo; los números de patente son 5,793,302 y 5,646,998.

* Tras todo lo hablado sobre voto electrónico, es agradable comprobar que algunos reconocen que existen algunos serios problemas de seguridad. El más grave, al menos para mí, es el votante coaccionado. Cuando usted entra en una cabina privada de votación puede votar a quien usted desee. Nadie puede hacer nada contra esto. Si puede votar desde su ordenador, en su propia casa, con algún tipo de medida de seguridad electrónica, entonces es posible para alguien comprar su voto y asegurarse su entrega en su beneficio.

<http://www.nytimes.com/library/tech/99/08/cyber/articles/14vote.html>

* Algunos me preguntan sobre mi comentario en el ultimo número referente a la necesidad para Windows NT de realizar alrededor de 300 cambios de seguridad para convertirlo en seguro. Yo interrogué al grupo de noticias de Usenet comp.os.ms-windows.nt.admin.security preguntándoles si era exagerado o cierto y conseguí varias respuestas. El consenso parece apuntar que el número estaba entre 50 y 3000 y que 300 no fue una estimación disparatada.

Una buena lista de comprobación esta disponible aqui:

<http://people.hp.se/stnor/>

* Las regulaciones de Estados Unidos sobre exportación criptográfica han conducido al desarrollo de excelentes productos por compañías no americanas. Juzguen con este artículo, aunque no sea el único sobre ello.

<http://www.rediff.com/computer/1999/jul/09suri.htm>

* Dos documentos sobre seguridad de Microsoft. No son gran cosa, pero nos esclarecen la línea que sigue Microsoft.

Fundamentos en seguridad:

<http://www.microsoft.com/security/resources/security101wp.asp>

Seguridad de macros en Office 2000:

<http://officeupdate.microsoft.com/2000/downloadDetails/o2ksec.htm>

* Un fallo en Hotmail permite a cualquiera leer el correo de cualquier usuario, sin ninguna clave de acceso. Para mí, lo realmente interesante de esta historia no es que el fallo fuera descubierto, sino que quizás haya sido conocido por la comunidad underground mucho antes de que fuera público.

Algunas historias nuevas que aluden a ello:

<http://www.wired.com/news/news/technology/story/21503.html>

<http://www.msnbc.com:80/news/306093.asp>

http://www.zdnet.com.au:80/zdnn/stories/zdnn_display/0,3440,2324361,00.html

<http://news.excite.com/news/zd/990901/10/the-bug-syndrome>

<http://news.excite.com/news/zd/990901/06/how-hotmail-blew>

http://www.salon.com/tech/log/1999/09/02/hotmail_hack/print.html

* Escultura cifrada en el cuartel general de la CIA en Langley, Virginia.

<http://www.npr.org/programs/atc/990826.kryptos.html>

* Únete a los militares y visita los sótanos de Ft. Meade. La Agencia de Seguridad Nacional ofrece alojamiento, manutención e instrucción académica gratis a los hackers que deseen trabajar para ellos durante cinco años tras su graduación.

<http://www.currents.net/newstoday/99/08/27/news3.html>

http://www.cnn.com/TECH/computing/9908/26/t_t/teen.hacker/index.html

* Ameno artículo de la BBC sobre el debate del cifrado en EEUU:

http://news.bbc.co.uk/hi/english/world/americas/newsid_430000/430384.stm

* Divertido tema: la historia real de Alice y Bob:

<http://www.conceptlabs.co.uk/alicebob.html>

* Este fue realmente un muy buen artículo -- claro, completo, comprensible -- publicado recientemente en *_The Sciences_* sobre cálculo cuántico. Cryptome ha colocado el artículo en línea, con el permiso del autor.

<http://cryptome.org/qc-grover.htm>

5. NOTICIAS EXTREMAMENTE PREOCUPANTE

Por Bruce Schneier

Traducción: Sergio Pozo Hidalgo

El Departamento de Justicia está planeando solicitar al Congreso una licencia para permitir a los agentes federales provistos de órdenes de registro, entrar de forma secreta en casas y oficinas para obtener llaves de descifrado o claves, para implantar "dispositivos de recuperación", o si no, para modificar ordenadores para asegurar que cualquier mensaje o fichero cifrado pueda ser leído por el gobierno. Con esta dramática propuesta, la Administración Clinton está diciendo básicamente: "si no se apresura a darle su llave a un tercero, entraremos de forma encubierta en su casa para conseguirla si tenemos sospechas de una conducta criminal".

El texto completo de la propuesta del Departamento de Justicia, un análisis sección a sección preparado por los abogados del DOJ (Department Of Justice - Departamento de Justicia) y material relacionado está disponible en:

http://www.epic.org/crypto/legislation/cesa_release.html

<http://www.cdt.org/crypto/CESA>

<http://www.washingtonpost.com/wp-srv/business/daily/aug99/encryption20.htm>

<http://www.zdnet.com/zdnn/stories/news/0,4586,2317907,00.html>

<http://www.techweb.com/wire/story/TWB19990820S0012>

6. NOTICIAS DE COUNTERPANE

Por Bruce Schneier

Traducción: Sergio Pozo Hidalgo

Bruce Schneier hablará en SANS Network Security 99, del 3 al 10 de Octubre en Nueva Orleans. Mire en <http://www.sans.org/ns99/ns99.htm> para más detalles sobre la conferencia.

Árboles de ataque: Miércoles, 6 de Octubre, 10:30 - 12:30

Criptografía en Internet: Martes, 5 de Octubre, 9:00 - 5:00

Bruce Schneier escribió la columna "Riesgos internos" de las ediciones de Agosto, Septiembre y Octubre de "Comunicaciones de la ACM".

Biometría: usos y abusos:

<http://www.counterpane.com/insiderisks1.html>

La carrera del caballo de Troya:

<http://www.counterpane.com/insiderisks2.html>

Riesgos de confiar en la Criptografía:

<http://www.counterpane.com/insiderisks3.html>

7. EN LA RATONERA: E-TRADE

Por Bruce Schneier

Traducción: Angel Galindo Sánchez

La seguridad de clave de E*Trade no es tal. Limitan la clave de entrada a un máximo de 6 caracteres, y las únicas elecciones son letras (se distingue entre mayúsculas y minúsculas), números, \$, y _. ¿Con qué cartera quiere negociar hoy?

8. FACTORIZAR UN NÚMERO DE 512 BIT

Por Bruce Schneier

Traducción: Sergio Pozo Hidalgo

Un record de factorización fue roto el 22 de Agosto pasado. Un grupo liderado por Herman te Riele de CWI en Amsterdam factorizó un difícil número de 512-bit (155 dígitos). Con "difícil" quiero decir que era el producto de dos primos de 78 dígitos... el tipo de números usado por el algoritmo RSA.

Alrededor de 300 estaciones de trabajo SGI y PCs Pentium hicieron el trabajo, mayoritariamente en noches y fines de semana, en el transcurso de siete meses. El algoritmo usado fue el de siembra en campo numérico.

Tiene dos partes: la etapa del tamiz y la etapa de reducción de la matriz. La primera fue en la que trabajaron los 300 ordenadores: sobre 8000 MIPS-años en 3,7 meses (esta es la parte que el dispositivo TWINKLE de Shamir puede acelerar). La etapa de reducción de la matriz requirió 224 horas de CPU (y al rededor de 3,2Gb de memoria) en el Cray C916 en el Centro Académico de Cmputación de Amsterdam SARA.

El esfuerzo completo fue 50 veces más fácil que romper DES. Factorizar llaves de comercio electrónico es, definitivamente, muy práctico, y llegará a serlo mucho más en unos pocos años. Ciertamente es razonable esperar que números de 768-bit sean factorizados en unos pocos años, luego los comentarios de los laboratorios RSA sobre que las llaves RSA sean de un mínimo de 768 bits son muy optimistas.

Certicom usó el evento para ganar votos sobre los beneficios de la criptografía de llave pública de curva elíptica. Los algoritmos de curva elíptica, al contrario que los algoritmos como RSA, ElGamal y DSA, no son vulnerables a las técnicas matemáticas que pueden factorizar estos grandes números. Por consiguiente, razonan, los algoritmos de curva elíptica son más seguros que los RSA y demás. Hay algo de cierto aquí, pero sólo si acepta la premisa de que los algoritmos de curva elíptica tienen matemáticas fundamentalmente diferentes. Escribí sobre esto anteriormente; en resumen, debe usar criptografía de curva elíptica si las consideraciones de memoria lo demandan, pero RSA con llaves largas es probablemente más seguro.

Este evento es significativo por dos razones. Una, la mayoría de los protocolos de Internet utiliza RSA de 512-bit. Esto significa que los no criptógrafos tomarán nota de ello y probablemente les entrará un poco de pánico. Y dos, al contrario que otros esfuerzos de factorización, este fue realizado por una organización en secreto. La mayoría de los criptógrafos no supieron que este esfuerzo se estaba llevando a cabo.

Esto demuestra que otras organizaciones podrían estar rompiendo llaves de comercio electrónico regularmente y no contándoselo a nadie.

Como de costumbre, la prensa está tomando este argumento de forma errónea. Dicen cosas como: "las llaves de 512-bit ya no son seguras".

Esto está completamente fuera de lugar. Como muchos de estos argumentos de criptoanálisis, las noticias reales son que no hay noticias. La complejidad del esfuerzo de factorización no fue una sorpresa; no hubo avances matemáticos en el trabajo. Factorizar un número de 512-bit requirió más o menos el mismo poder de computación que la gente predijo.

Si las llaves de 512-bit son inseguras hoy, eran igual de inseguras el mes pasado. Cualquiera que implemente RSA debería haber cambiado a claves de 1024-bit hace algunos años, y debería estar pensando en llaves de 2048-bit hoy. Es agotador comprobar como no se escucha a los criptógrafos cuando dicen que algo es inseguro, esperando en cambio que alguien demuestre palpablemente la inseguridad.

<http://www.cwi.nl/~kik/persb-UK.html>

<http://www.msnbc.com/news/305553.asp>

Análisis de RSA

<http://www.rsa.com/rsalabs/html/rsa155.html>

Refutación de Certicom

<http://www.certicom.com/press/RSA-155.htm>

Webs notables que usan todavía RSA de 512-bit:

Travelocity

Tienda en línea Microsoft

Tienda en línea Compaq

Tienda en línea Godiva

Dr. Koop.com

Flowers N More

Hay muchos más. Puede comprobarlo usted mismo conectando con una web con una versión doméstica (EEUU) segura de Internet Explorer 4.0.

9. COMENTARIOS DE LOS LECTORES

Por Bruce Schneier

Traducción: Juan Cruz Ruiz de Gauna (artículos 1 y 2) y David Gómez (artículos 3, 4, 5 y 6)

De: Gene Spafford spaf@cs.purdue.edu

Asunto: Re: Comentarios sobre la clave "NSA" en Windows NT

Bien, siempre es más fácil creer en una teoría de la conspiración o en diseños oscuros. Sin embargo, puede haber explicaciones alternativas.

Por ejemplo, da la casualidad de que sé que varias agencias de 3 letras usan un montón de máquinas Windows (en cualquier caso, este hecho debiera producir terror por si solo). Supongamos que dichas agencias desean cargar versiones propias de sus rutinas de cifrado altamente clasificadas. ¿Pensáis que enviarían copias de su código a Redmond para que se lo firmen de forma que pueda ser cargado? ¿o lo firmarán ellos mismos, con su propia clave, haciéndolo en la propia empresa, donde es "seguro"? Si van a hacerlo en su propia empresa, entonces o bien Microsoft comparte su clave privada con ellos (mala idea), o el código debe permitir acomodar una segunda clave generada por la agencia. Ummm, esto suena familiar ¿no creéis?.

Otra explicación que leí aquí (este tema se ha discutido en varias listas) es que para obtener la aprobación para la exportación, los chicos de Microsoft necesitaban incluir una clave de "respaldo" en caso de que la primera se viese comprometida de alguna manera. Necesitarían cambiarla para usar la clave alternativa en todos los sistemas. ¿Pero cómo lo harían a menos que la segunda clave ya estuviese instalada, de forma que pudiesen realizar el cambio usando esta segunda clave?. Por lo tanto, si fueseis Microsoft y la NSA os solicitase instalar una clave de respaldo como ésta, ¿cómo la llamaríais?.

Por supuesto, también puede suceder que Microsoft quisiese usar una segunda clave por decisión propia, y que el programador envuelto en el código decidiese nombrarla de una forma bastante tonta. También hay una historia sobre código de Microsoft que se pone en circulación con elementos de código no documentados y cosas que los gestores de Microsoft ignoran que están presentes. Supongamos que el código (nos referimos tan solo a unas pocas líneas de código) fue puesto ahí por un agente de los servicios de inteligencia de algún otro país (no debiera ser tan difícil corromper a algún empleado o incluso introducir uno en Microsoft con buenas capacidades para desarrollar código que pudiese conseguir acceso eventualmente al código apropiado). El/ella nombra las variables introducidas con las siglas "NSA" para prevenir revisiones del código e incluye un bloque de comentarios que dice "La NSA nos ha solicitado que esto esté aquí -- no cambiar o realizar preguntas". El "siniestro propósito" es cierto, pero estamos culpando a la entidad equivocada.

¡Qué diablos!, puede incluso que éste sea un propósito del propio Sr. Gates: Después de todo está teniendo una fuerte disputa con el Departamento de Justicia de los Estados Unidos.

Hay otras posibles razones para el nombre.

Estas posibles explicaciones no quieren decir que la clave extra no tenga efectos laterales (como instalaciones clandestinas y sortear los obstáculos de los controles de exportación). Y, por supuesto, probablemente nunca sepamos cuál es el propósito principal de esta clave ni qué papel juegan estos efectos laterales en la decisión de usar dicha clave, a pesar de las eventuales quejas de la gente.

El pensamiento principal es que puede haber posibles escenarios para el nombre de la clave que no impliquen actividad perjudicial, o que no impliquen dicha actividad a cargo de la NSA. Esa no debiera ser la conclusión inmediata a la que llegue la gente.

Y, aún a riesgo de comenzar una diatriba, permitidme realizar una pregunta (retórica): Incluso si la clave fue puesta ahí con propósitos de monitorización clandestina, ¿qué hay de malo en ello? si se usa para controlar a terroristas, cárteles de droga o laboratorios de armas en Iraq?; ¿no es eso lo que deseamos que suceda?. En este caso, ¿deberíamos ser conscientes de que este control ha sido descubierto y, posiblemente, ya no tiene valor!. La historia de la criptografía muestra -- repetidamente -- que tener ventajas criptográficas supone una diferencia enorme en tiempos de conflicto, y que poner esas ventajas en su sitio y funcionando lleva tiempo. Sería ingenuo creer que estas amenazas no se ciernen sobre nosotros, o que no hay probabilidades de que eso suceda en el futuro.

Debieramos tener claro en nuestras discusiones si el asunto a tratar es la presencia del código o quién puede tener el control de dicho código.

El tema principal es ¿Qué controles se ponen para asegurar que el código no sea usado contra objetivos inapropiados (Por ej., personas respetuosas de la ley, negocios legales y ciudadanos)?.

Desafortunadamente carecemos de garantías seguras en este campo, y ha habido abusos en el pasado (o presuntos abusos). Pero esto debieramos planteárnoslo si el código fue puesto para los oscuros propósitos de algún otro grupo.

De: "Lucky Green" shamrock@cypherpunks.to
Asunto: Más cavilaciones sobre la NSAKEY

Me gustaría comentar alguna de tus opiniones públicas acerca de la NSAKEY. El objetivo de este email es ofrecer algunos datos acerca del modo de pensar de las agencias de inteligencia cuando intentar poner en peligro (desestabilizar) sistemas.

En primer lugar, estoy de acuerdo con tu afirmación de que la NSA no necesita desproteger la CAPI para desproteger a los ordenadores que ejecutan Windows. Lo que no es lo mismo que afirmar que la NSA no busca comprometer la CAPI obligando a Microsoft a instalar la clave NSA. Para los criptógrafos académicos, una vez que un fallo catastrófico ha sido hallado en un cifrado, el trabajo ya está terminado. "Tenemos un ataque 2^{16} . El trabajo ha sido realizado. Vámonos a casa". Las agencias de inteligencia no funcionan de esta manera.

Mi trabajo con GSM ha revelado que las agencias de inteligencia, que como todos sabemos últimamente están detrás de los cifrados GSM, realizan una aproximación muy diferente. Las agencias de inteligencia intentarán comprometer cada componente individual de un sistema de cifrado que permita ser comprometido. Las agencias de inteligencia comprometerán, si tienen la oportunidad, un componente simplemente porque pueden hacerlo, no porque lo necesiten. Esto puede parecer una manifestación extraña de implementar redundancia múltiple en un sistema.

Lo que, estoy seguro de que en esto todos estamos de acuerdo, es generalmente una buena idea.

En el caso del GSM, hemos descubierto las siguientes desestabilizaciones (o compromisos):

o Desestabilización de generación de clave.

Las claves de 64 bits tienen los últimos 10 bits puestos a cero. (He oído rumores acerca de que algunas implementaciones sólo ponen a cero los últimos 8 bits, pero en cualquier caso es innegable que la entropía de la clave queda comprometida).

o Desestabilización del sistema de autenticación y algoritmo de generación de claves.

El GSM MoU fue avisado formalmente en 1989 (o 1990 como muy tarde) sobre los fallos que habíamos descubierto el año anterior en COMP128. Mucho antes de que GSM fuese ampliamente interceptado. El Grupo de Expertos en Algoritmos de Seguridad del MoU (SAGE Security Algorithm Group of Experts), compuesto por personas cuyas identidades son desconocidas hasta ahora, mantuvo este descubrimiento en secreto y no informo sobre él ni siquiera a los propios miembros del MoU. Como resultado, las agencias de inteligencia pudieron clonar teléfonos y calcular las

claves privadas de voz usadas durante una llamada.

o Desestabilización del algoritmo robusto de privacidad de voz A5/1.

Este cifrado de 64 bits tiene numerosos "fallos" de diseño, dando como resultado una resistencia de como mucho 40 bits. Es inconcebible para mí, y virtualmente para todos aquellos con los que he hablado de este tema, que estos fallos obvios fueran pasados por alto por sus diseñadores militares franceses.

o Desestabilización del algoritmo débil de privacidad A5/2.

El MoU admite que la fragilidad fue la meta del diseño del A5/2, incluso sabiendo que el SAGE indicó en sus análisis oficiales que no eran conscientes de ningún fallo criptográfico en el A5/2.

Para permitir interceptación y descifrado en el tráfico GSM, bastaría con comprometer la longitud efectiva de la clave. Bastaría con comprometer la generación de la clave. Habría sido suficiente con comprometer los cifrados. La NSA/GCHQ hizo las tres cosas.

Dados estos hechos, no sería inusual que la NSA instalase por sí misma puertas traseras en el sistema operativo Windows *y* obtuviese una copia de la clave de firmas de Microsoft *y* obligase a Microsoft a instalar la propia clave de la NSA.

Pensemos en ello como un buen diseño de desestabilización redundante.

De: "Kevin F. Quinn" kevq@banana.demon.co.uk

Tema: Criptograma de Abril y el reciente debate sobre la clave de repuesto NSA

En el CriptoGrama del 15 de Abril de 1999, mencionaste el enfoque de las dos claves de Microsoft en referencia a sus claves principales para Authenticode, y que ellos incluían las dos claves "presumiblemente por si una de ellas alguna vez es comprometida". Ahora sabemos que el mismo enfoque fue empleado por los CSP (proveedores de servicios criptográficos). El propio comunicado de Microsoft sobre el asunto es interesante; las dos claves estan presentes "en caso de que la clave principal sea destruida" (literalmente). Creo que en tu CriptoGrama querías decir "destruida" mas que "comprometida" -- Microsoft parece estar intentando protegerse contra la posibilidad de que la clave principal secreta se quemase en un incendio o algo asi; no se estan protegiendo contra copias no autorizadas de la clave hecha con el enfoque de las dos claves. Creo que es una distinción importante a

tener en cuenta.

La única buena razón que puedo ver para tener dos claves, es proporcionar seguridad contra el compromiso -- en cuyo caso necesitaras validar las firmas contra ambas claves (i.e., AND en vez de OR). De esa manera si una clave es comprometida, la validación todavia fallará ya que la segunda firma no será valida. Si ambas claves son almacenadas en lugares seguros separados, el atacante tendrá que romper la seguridad de ambos lugares para obtener ambas claves, esperando tú poder darte cuenta de la primera intrusión antes de que ocurra la segunda. La manera sensata de protegerse contra la posibilidad de destrucción (incendio, catastrofe, etc...) es tener varias copias, cada una almacenada con seguridad y monitorizada (de la misma manera que son controlados los documentos clasificados.

Microsoft reclama que el enfoque de las dos claves fue sugerido por la NSA -- Yo encuentro dificil de creer que la NSA sugiriera incluir dos claves principales, para protegerse contra la destrucción de una de ellas. Mi teoría favorita es que había un problema de comunicación; el consejo de la NSA seguía más o menos las lineas de, "tener dos claves principales protegidas contra pérdidas", queriendo decir compromiso, y Microsoft lo interpretó como destrucción.

De: Greg Guerin glguerin@amug.org
Asunto: ¿Nuevo giro del asunto de la NSA-key/NT?

En tu artículo en CriptoGrama, acabas diciendo: "Este virus no existe todavía, pero podría ser escrito." [Este es un virus que sustituiría la clave de backup en NT con una clave falsa, y podría engañar al usuario para aceptar código malicioso como firmado.]

Después de escribir <http://amug.org/~glguerin/opinion/win-nsa-key.html>, se me ocurrió que el virus ya existe, o al menos todas sus partes existen. Solo necesita "transformarse al Lado Oscuro" y ser ensamblado.

El "kit de construcción" para este virus no es otro que el "programa de reparación" en:

<http://www.cryptonym.com/hottopics/msft-nsa/ReplaceNsaKey.zip>

Todas las partes están ahí. El programa "AddDelCsp.exe" (no se proporcionan las fuentes) es el agente de infección activo. "nsarplce.dll" y otras DLL's son las "toxinas". El kit incluye incluso "TestReplacement.exe" (con las fuentes) para comprobar si algún joven emprendedor constructor de kits ha realizado sus cambios con éxitos o no.

Estoy sólo suponiendo, pero alguien con habilidad en programación sobre Wintel podría probablemente construir un virus o Caballo de Troya con este kit en cuestión de horas. Probablemente la única habilidad que tendrían que pulir es la criptografía, pero hay alguna información buena para comenzar en el mismo informe de Fernandes. Un poco de lectura, un poco de tiempo de generación de claves, quizás unas pocas correcciones, y listo. Se prueba en un sistema NT local, y entonces se publica al mundo haciendo un mirror del informe de Fernandes. O simplemente se envía a algunos "amigos" via Hotmail. Ciertamente parecería auténtica, e incluso como el programa de "reparación" estaba sin firmar, y el informe original no dice nada acerca de autenticar la descarga antes de ejecutarla, podría ser un Caballo de Troya bien preparado.

Si este virulento "programa de reparación" se escribe de forma silenciosa, puede extenderse MUY lejos antes de que nadie se de cuenta.

Podría incluso camuflarse a si mismo y nombrar su clave toxica como "NSAKEY", justo como la original de Microsoft. Es decir, después de "borrarse" a si mismo, esta todavía presente. ¿Con que frecuencia a la gente se le ocurriría pensar en comprobar esa clave?

Si conoces a alguien con experiencia de programación en NT, puede ser interesante darles a leer el informe de Fernandes, bajarse el kit de construcción del virus, ehem, quiero decir, programa de "reparación" y entonces intentar hacer esto. Supongo que no serían necesarias habilidades previas en la escritura de virus, solo habilidades de programación en NT por encima de la media. Apuesto a que tendrías una versión virulenta en menos de una tarde. Un proyecto interesante para la perezosa fiesta del Dia del Trabajo, ¿eh?

De: Sam Kissetner
Asunto: Meganet

Pensé que esto podía distraerte. El boletín de Febrero de CriptoGramma se rie de la pagina web de Meganet por decir:
Claves simétricas de 1 millón de bits -- La oferta del mercado tiene sólo 40-160 bits!!

Visité la pagina hoy. (La URL cambió; está en <http://www.meganet.com/index.htm>). Quizás lean CriptoGramma, porque intentaron arreglar el error gramatical. Pero era parte de un gráfico, asi que simplemente pegaron una pequeña caja blanca sobre el apóstrofe y las, dejando:
Claves simétricas de 1 millón de bits -- El mercado oferta sólo

40-160 bits!!

Vaya, eso está *mucho* mejor.

(N. del T: Las frases originales son:

-1 million bit symmetric keys -- The market offer's [sic] 40-160 bit only!!

-1 million bit symmetric keys -- The market offer 40-160 bit only!!!)

De: Marcus Leech mleech@nortelnetworks.com

Asunto: Descripción de crypt(1) de HP

Siendo sinceros con HP, y crypt(1) -- HP simplemente ha reproducido con fidelidad la pagina MAN original de crypt(1). Crypt(1) apareció por primera vez en Unix V7, volviendo la vista a 1978 -- en un tiempo en que DES estaba comenzando a ser usado en ciertas áreas limitadas. Que un sistema operativo tuviera cualquier tipo de facilidad de cifrado de ficheros era como un milagro en aquella época. Sun obviamente ha modificado ligeramente la documentación para reflejar la realidad actual, mientras HP ha elegido el enfoque de permanecer fiel a la documentación original.

Del Boletin #139:

SOBRE EL USO DEL ANONIMATO EN INTERNET

Por Alvaro Ibáñez <alvy@jazztel.com>

Director de Contenidos Internet - Jazztel Internet_Factory

<http://www.jifactory.com>

Casi todo el mundo sabe que lograr el anonimato total en Internet es prácticamente imposible: por todos lados quedan rastros de direcciones de e-mail, direcciones IP, teléfonos de llamada... pero considerando escenarios de perfil bajo (suponiendo que no has hecho nada grave usando ese anonimato, y que no van a usar todos los medios existentes para perseguirte), es relativamente fácil hacerse pasar por "anónimo".

Algunos ejemplos: puedes publicar mensajes en muchos foros como "invitado", o con nombres y e-mails falsos, porque no se comprueban. También puedes crear una cuenta de correo con datos falsos en

cualquier servicio gratuito (como MixMail o Hotmail), usar "anonimizadores" de correo e incluso de navegación web.

El web de KRIPTOPOLIS tiene una muy buena recopilación sobre todas estas técnicas:

Kriptopolis

<<http://www.kriptopolis.com/anon.html>>

Otra buena página es:

Anonymity: Index

<<http://www.stack.nl/%7egalactus/remailers/index-anon.html>>

y aquí hay un artículo de Wired donde se explican los potenciales problemas de usar estos sistemas, que tampoco son infalibles:

Anonymous Web Surfing? Uh-Uh

<<http://www.wired.com/news/news/technology/story/19091.html>>

Los argumentos tradicionales para hacerse pasar por anónimo en Internet (o en la VidaReal(tm)) son bien conocidos... dentro de lo que podría considerarse "lícito": empleados que temen ser despedidos por sus jefes si se enteran que publican algo contrario a su opinión (o que envían fotos a alt.sex.pictures ;-), envío de "filtraciones" a la prensa saltándose los conductos reglamentarios, o miedo a desvelar la identidad en ambientes que no son los habituales de una persona (por ejemplo, en grupos de auto-ayuda, de minorías o de cualquier otro tipo. Entre los usos considerables "ilícitos", la lista es larga: envío de amenazas, insultos, chantajes, calumnias, mentiras dañinas y un largo etcétera (incluyendo "fastidiar a la competencia").

Hay gente que usa el correo (o la publicación en foros) de forma anónima y que se indigna cuando se le pide que se identifique para dar más validez a sus opiniones. Sobre este hecho, que va más allá de lo puramente técnico del "cómo hacerse pasar por anónimo" (divertido y útil muchas veces), hay bastante información en la Red. Dos artículos interesantes son estos:

The Anonymous Fallacy

<<http://www.stack.nl/%7egalactus/remailers/fallacy.html>>

P.S. to The Anonymous Fallacy

<<http://www.stack.nl/%7egalactus/remailers/fallacy2.html>>

El autor argumenta por qué es una falacia informal (o forma impropia de razonamiento) usar frases como "No acepto tus argumentos porque no

estás usando tu nombre verdadero". El primer artículo expone el caso y el segundo hace algunos comentarios adicionales.

Estos artículos provocaron una respuesta:

Response to the Anonymous Fallacy

<http://www.stack.nl/%7egalactus/remailers/no-fallacy.html>

En donde se contra-argumentan los artículos anteriores.

Sinceramente, me quedo con el primero como ejemplo de razonamiento lógico que persigue un fin, pero con el segundo como opinión más válida sobre el caso -- aunque tal vez hasta los haya escrito la misma persona ;-)

Como se dice en "Response to the Anonymous Fallacy", el uso del anonimato en muchas ocasiones está justificado, pero eso *no* mejora la comunicación. Si bien hay casos en los que el dato de la identidad de la persona no revela nada, en otros es parte del contexto de la comunicación, y muy relevante.

Un ejemplo lo dejaré claro. Si alguien publica un mensaje anónimo diciendo:

"El Madrid ganó ayer al Barcelona por 2-1"

no hay ningún problema en aceptar este mensaje como una afirmación cierta, dado que es un hecho comprobable por todos. También lo sería "Estoy contento de que el Madrid ganara al Barcelona", porque es cierto que el autor puede estar contento con esa situación.

En cambio, si el anónimo publicara:

"El Madrid ganó ayer al Barcelona porque me consta que se pagó al árbitro para que expulsara a Karembeu y además los del Madrid cobraron un prima extra del Betis. Estaba en el hotel de la concentración y lo ví."

.. sí hay problema en aceptarlo como argumento, porque el autor del mensaje es *parte* del mensaje y la comunicación: afirma tener información, conocer cosas que otros no conocen, y haber estado en sitios -- sin que se sepa realmente quién es ni se pueda comprobar. Si el autor firmara con nombres y apellidos, se podría investigar y comprobar si es cierto lo que dice, pero si no lo hace el dato no es más que mera especulación.

Razonamiento más sencillo (método científico conocido como "la navaja

de Occam"... "ese tipo anónimo es un fanático del Barcelona, cabreado porque su equipo ha perdido -- su único objetivo es desprestigiar al Madrid".)

El autor de "Response to the Anonymous Fallacy" lo resume así:

"Conclusión: la idea de que todas las afirmaciones deben considerarse como válidas de por sí, sin que importe quién es el autor, tiene cierto sentido, especialmente cuando esto se hace como precaución para evitar males mayores. Sin embargo, eliminar el contexto de cualquier argumentación la puede hacer menos inteligible -- y saber quién es el autor de una afirmación puede ser parte del contexto. El principal obstáculo (que es consecuencia de este papel esencial del contexto) es del tipo práctico: sin el contexto, muchos mecanismos de selección que son esenciales para la eficiencia [de la comunicación] no funcionan."

Dicho esto, ¿cómo mantener el anonimato en Internet en estas situaciones? Es muy fácil: procurando que los mensajes contengan toda la información, haciendo que los receptores obtengan información indirecta del autor anónimo y generando un marco de confianza. (En el ejemplo anterior, si la persona que envía el mensaje tiene una carrera intachable como autor de mensajes con "mucha señal y poco ruido", nadie puede suplantarle, es alguien que aporta cosas positivas a la comunidad Internet, nunca se le ha pillado en una mentira y cuyas afirmaciones se confirman de forma rutinaria día tras día, muchos considerarían el mensaje auténtico sin necesidad de saber su nombre real).

En todos los demás casos, como hacen los investigadores, búsquese el interés final de la publicación de un mensaje anónimo. ¿A quién beneficia un mensaje Z publicado por un anónimo? ¿Por qué lo hace? ¿Por qué razón realmente lo está escribiendo como anónimo y no identificándose? Un poco de lógica aplicada a estas situaciones sacará probablemente a la luz razones mucho menos complejas (de más bajeza y pobre valor humano) que las que realmente son aplicables a las precauciones que hacen legítimo el uso de los mensajes anónimos.

Del Criptograma #18:

1. PARA QUIENES HAN DECIDIDO HACERSE CRIPTÓLOGOS

Por Bruce Schneier

Traducción: José Manuel Gómez

Una de las preguntas que con más frecuencia se me hacen por correo electrónico es la siguiente: "¿Cómo puedo hacerme criptólogo?". Este artículo representa mi intento de responder a esta cuestión.

Mi respuesta se divide en cuatro partes: para el estudiante de instituto, para el de universidad, para el de post-grado y para la persona que trabaja en un campo relacionado (aunque mucho de lo que tengo que decir es común a todos ellos).

En primer lugar: ¿qué es un criptólogo? En relación a lo que ahora nos interesa, un criptólogo es alguien con actividad en el campo de la criptografía: alguien que se dedica a la investigación, escribe documentos, rompe algoritmos y protocolos y eventualmente escribe sus propios algoritmos y protocolos. Un criptólogo puede trabajar como profesor de universidad, pero algunas grandes empresas (AT&T, IBM) contratan criptólogos a tiempo completo, y existen algunos criptólogos que trabajan como consultores para empresas que no tienen criptólogos a tiempo completo en sus plantillas. Y, por supuesto, la NSA fichará casi a cualquiera que pueda ser entrenado como criptólogo. Pero, no importa dónde, la labor es la misma: diseñar sistemas, romper sistemas, investigar, publicar documentos. La criptografía es un tema de investigación y eso se nota.

Como es obvio, la mayoría de la gente que implementa criptografía en productos de hardware o software no son criptólogos. Son implementadores de criptografía, ingenieros de seguridad. Me parece que la mayor parte de la gente que dice querer ser criptólogos, en realidad quieren ser ingenieros de seguridad. Desean ser una persona que construye sistemas seguros basados en criptografía. Este artículo no va dirigido a ellos, aunque gran parte de los consejos son idénticos. La ingeniería de seguridad requiere entender muy bien la criptografía, pero no es necesario crear nueva criptografía.

La respuesta más breve a "cómo puedo hacerme criptólogo" es: "Graduése en criptología". No es la única forma de convertirse en criptólogo pero sí es -con diferencia- la más fácil. Las habilidades que se aprenden para lograr el título son las que se necesitarán como criptólogo y las puertas se abren con mayor facilidad para quienes ostentan un título. Y aún más: el proceso de obtener el título permitirá responder a la pregunta más importante de todas: "¿quiero ser criptólogo?".

La criptografía puede considerarse una especialidad de las matemáticas. Obtenga donde obtenga su título, las matemáticas y la informática son vitales. Pero existe algo más importante: la criptografía es un modo de

pensar. En algún otro sitio, he escrito que la ingeniería de seguridad es diferente de cualquier otro tipo de ingeniería; requiere cierto tipo de mentalidad para considerar los sistemas desde la perspectiva del atacante. Durante la Segunda Guerra Mundial, los británicos descubrieron que los mejores criptólogos eran los jugadores de ajedrez y los músicos. Para mí, la mejor gente en seguridad son los jugadores de "Dragones y Mazmorras" y la gente muy minuciosa. La habilidad para encontrar agujeros en un sistema (sean matemáticos, inherentes al propio sistema o relativos a su utilización) es vital para un criptólogo.

Al estudiante de instituto: estudia matemáticas e informática. Lee libros de criptografía, tanto históricos (como "The Codebreakers" -Rompedores de códigos-, de David Kahn), como libros modernos (como mi propio "Applied Cryptography" -Criptografía Aplicada-). Lee libros sobre seguridad informática: cortafuegos, seguridad en Internet, seguridad en Windows,... lo que sea. Los campos están muy próximos y puede ser que descubras que prefieres la seguridad informática a la criptografía. Participa en las discusiones del grupo de noticias 'sci.crypt' y la lista de correo de los 'coderpunks'. Si en esos foros, eres capaz de diferenciar quién sabe lo que dice y quién no, estás en el buen camino. Casi seguro que te sentirás impulsado a crear nuevos algoritmos y estarás convencido de que son invencibles. No te resistas al impulso: es una de las partes divertidas. Pero resístete a la convicción; casi seguro que tus creaciones serán fáciles de romper y casi nadie perderá tiempo intentándolo por ti. Las podrás romper tú mismo en cuanto vayas mejorando.

Muchas veces me han preguntado a qué universidad acudir para estudiar criptografía. En principio, no importa. La formación matemática que se requiere puede lograrse en cualquier buen departamento de matemáticas. Nota: "buen departamento de matemáticas" significa un lugar donde se conceda importancia a las demostraciones matemáticas. Hay universidades donde las demostraciones sólo aparecen en el último año o casi. Algunas universidades ofrecen cursos de criptografía o seguridad informática (vea en mi web una lista limitada de cursos universitarios). Pero en realidad, poco importa.

Al estudiante universitario: estudia matemáticas. Obtén un título en matemáticas o en informática, pero estudia matemáticas. Sigue cursos de matemáticas para matemáticos, no cursos de matemáticas para ingenieros. Aprende a pensar sobre matemáticas; aprende a demostrar teoremas. Intenta seguir cursos en Teoría de Números, Teoría de la Complejidad (a menudo ofertado fuera del departamento de informática), Algoritmos, Estadística y Álgebra Abstracto. La Criptografía utiliza teoría de números pero también ideas que provienen de muchas y variadas áreas de las matemáticas. Los criptólogos necesitan amplios conocimientos de matemáticas; esta es la única forma de que se establezcan nuevas

relaciones y se descubran ideas realmente nuevas.

Las disciplinas informáticas fundamentales incluyen diseño de algoritmos, complejidad computacional y teoría de la computación. Algunas universidades ofrecen un curso de criptografía para estudiante; realícelo. Continúe leyendo libros de criptografía: "The Handbook of Applied Cryptography" - Manual de Criptografía aplicada (de Alfred J. Menezes, Paul C. van Oorschot y Scott A. Vanstone), o "Cryptography: Theory and Practice" - Criptografía: teoría y práctica (de Doug Stinson). Todos estos libros contienen muchas, muchas referencias. Si algo le interesa, localice la referencia y léala. Realice cursos de informática; lea libros sobre seguridad informática. Y no olvide consultar las referencias si algo le interesa.

Al elegir una escuela de postgrado, escoja una con experiencia en criptografía. La situación puede ser muy cambiante en el mundo académico, así que prefiero no dar una lista de escuelas (puede empezar con el MIT y Waterloo) pero todas están ahí fuera. Muchas están fuera de los Estados Unidos, así que esté abierto a elegir una escuela de un país diferente al suyo. Una forma de construirse una lista de posibles escuelas es buscar informes de investigación que le interesen. Observe dónde enseñan sus autores. Cuando esté en su escuela de postgrado, su consejero le dará mucha más orientación sobre cómo convertirse en criptólogo de la que yo podré darle nunca.

Y finalmente, me toca aconsejar a la gente que dejó la escuela y ya está trabajando. Existen dos opciones. Una, volver a la escuela de postgrado, a tiempo parcial o completo. Dos, puede imitar el proceso usted mismo, sin contar con la ayuda de una institución investigadora o un consejero. Puede leer mucho y aprender por sí solo. Si tiene un buen curriculum en matemáticas puede aprender criptografía usted solo. Esta opción es mucho más dura, pero posible.

Sin importar en que momento vital se encuentre, debería intentar descubrir lo que significa ser criptólogo. Lea todo lo que pueda para darse una idea de qué tipo de preguntas se hacen los criptólogos, qué hacen para resolverlas y qué tipo de cuestiones todavía aguardan ser respondidas. Localice problemas que pueda comprender e intente resolverlos. No se preocupe sobre si está "reinventando la rueda" y resolviendo cosas que otros ya han resuelto; así es como se aprende. He escrito un "Curso de Autoestudio sobre criptoanálisis de cifrado en bloques" que intenta plantear problemas que un estudiante de criptología pueda abordar; se puede intentar resolver problemas en cualquier área de la criptografía.

Aprender a ser criptólogo no es fácil y merece la pena preguntarse si eso es lo que se desea realmente. Por suerte, en el proceso hay muchos

momentos en que se puede decidir cambiar de orientación. Y como se dijo al principio, mucha gente que dice querer ser criptólogo, en realidad quieren ser ingenieros de seguridad. Aunque los requisitos para un ingeniero en seguridad son en gran parte los mismos (leer libros, informes de investigación, tomar clases, aprender criptografía y cómo utilizarla), no se requiere obtener un título.

Lista de cursos de criptografía:

<http://www.counterpane.com/courses.html>

Curso de Autoestudio sobre criptoanálisis de cifrado en bloques:

<http://www.counterpane.com/self-study.html>

2. NUEVAS LEYES DE EXPORTACIÓN EN ESTADOS UNIDOS Y LEGISLACIÓN CRIPTOGRÁFICA ANTI-PRIVACIDAD

Por Bruce Schneier

Traducción: Oscar Esteban

El 16 de septiembre (el día posterior a la publicación de Crypto-gram... ¿coincidencia?) la administración Clinton anunció cambios a sus reglas de control de la exportación. Aún no ha habido cambios, pero si la administración los lleva finalmente a cabo representarán un giro completo a su ya antigua hostilidad hacia la criptografía fuerte. Pero los detalles no son trigo limpio, y la administración Clinton tiene un largo historial de prometer cambios que implicarían relajación en las exportaciones y no llevarlos a cabo después. Y hay una segunda parte, una propuesta de ley llamada Cyberspace Electronic Security Act (CESA; Acta de Seguridad Electrónica del Ciberespacio), que incluye depósito de claves y tiene algunas desagradables previsiones anti-privacidad.

Exportar primero. La administración propone que se pueda exportar sin licencia hardware y software sin límite de potencia tras una "única revisión técnica" y algún informe sobre a quién se venden los productos. Los productos "a medida" tendrán algunas restricciones en la venta a gobiernos extranjeros y organizaciones terroristas o criminales. Los productos con longitudes de clave inferiores a 64 bits no tendrán ningún control.

De nuevo, estos cambios no están vigentes. La administración dijo que entrarían en vigor el 15 de diciembre. Si siguen adelante con su promesa, estas nuevas regulaciones permitirían que se pudiera exportar prácticamente cualquier producto de forma más o menos libre. Obsérvese que no hay nada acerca de recuperación de clave en estas nuevas regulaciones, ni ningún límite artificial basado en la longitud de las

claves. Por otra parte, aún faltan las regulaciones sobre investigación criptográfica; los casos judiciales Karn, Junger y Bernstein aún son importantes. Y ahora, las malas noticias. La administración también ha propuesto la CESA, que contiene regulaciones para el depósito de claves y el uso de descifrado como arma policial. La propuesta de ley exige a terceros ceder claves a agentes del gobierno con orden judicial. Más importante, afirma que no hay "presunción constitucional de la privacidad"(1) del texto en claro descifrado, y no se habla de "causa legítima suficiente" en la propuesta.

Aún más temible resulta que la propuesta de ley permita al gobierno negarse a comunicar los métodos que emplearon para recuperar el texto en claro en el juzgado. Esto significa que la policía podría presentar texto en claro en el juicio, pero negarse a revelar al defendido cómo se obtuvo ese texto. Esto, por supuesto, significa que el defendido puede verse en apuros para defenderse a sí mismo, y facilita mucho a la policía la aportación de pruebas. El derecho a un juicio justo podría estar en juego. Y para asegurarse de que habrá montones de texto en claro disponible para uso de la policía, la propuesta de ley dispone \$80 millones para un Centro de Soporte Técnico del FBI, pensado para desarrollar herramientas policiales útiles contra la seguridad informática y la criptografía.

La CESA era en origen aún peor. Había una cláusula que hablaba de "búsqueda secreta", con la que la policía podía entrar de forma secreta en las casas de la gente e instalar invisibles "dispositivos de recuperación" en sus ordenadores (del tipo Back Orifice). Había también otras cláusulas para promover la recuperación de claves. Estas han desaparecido en la reformulación final, pero quién sabe si alguna vez volverán a aparecer.

De nuevo, todo esto son propuestas y nada de esto es oficial. Por favor, que nadie piense que la partida ha terminado y que no hemos ganado nada. El debate sobre el uso de la criptografía como herramienta de privacidad aún continúa.

(1) N.T.: "constitutional expectation of privacy" implica el derecho a la privacidad salvo que exista una "probable causa", esto es, indicios fuertes de un delito, por ejemplo.

Artículos de las news:

<http://www.wired.com/news/news/politics/story/21786.html>

<http://www.computerworld.com/home/news.nsf/CWFlash/9909175encrypt>

Documentos del gobierno:

Nota de prensa

<http://www.pub.whitehouse.gov/uri-res/I2R?urn:pdi://oma.eop.gov.us/1999/9/16/15.text.1>

Resumen de la Casa Blanca

<http://www.pub.whitehouse.gov/uri-res/I2R?urn:pdi://oma.eop.gov.us/1999/9/17/2.text.1>

3. COUNTERPANE SYSTEMS: INVESTIGACIÓN DOCUMENTADA

Por Bruce Schneier

Traducción: David Gómez

"Minimizando el Ancho de Banda para el Acceso Remoto a Logs de Auditoria Criptográficamente protegidos"

J. Kelsey y B. Schneier, Segundo Grupo Internacional de Trabajo en Recientes Avances en Detección de Intrusiones (RAID '99), Septiembre 1999, por aparecer.

Los logs de auditoría a prueba de falsificaciones son una herramienta esencial para los forenses de ordenadores. Nosotros mostramos cómo construir uno donde la cantidad de información intercambiada necesaria para verificar las entradas en el log de auditoria, es reducida notablemente. Haciendo que la verificación del log sea más eficiente, este sistema se amolda mejor a una implementación en entornos con un bajo ancho de banda.

<http://www.counterpane.com/auditlog2.html>

4. NOTICIAS

Por Bruce Schneier

Traducción: David Gómez

Ahora el Reino Unido pide puertas traseras para poder acceder al tráfico de Internet a través de los ISPs (proveedores de acceso a Internet).
<http://www.theregister.co.uk/990910-000026.html>

Y lean este resumen sobre la vigilancia electrónica y la privacidad en el Reino Unido.

<http://www.zdnet.com/zdnn/stories/news/0,4586,2342025,00.html>

La gente de Hushmail ha publicado una respuesta a mi comentario sobre los programas de cifrado de correo basado en Web:

http://www.hushmail.com/bruce_comments.htm

Resumen: ellos reconocen lo dicho en mi artículo de CriptoGramma, describen su seguridad física, marcan futuras instrucciones como permitir cambios en la palabra de paso (passphrase), señalan que ellos explican como seleccionar una buena palabra de paso en su sistema de ayuda, y argumentan que no hay manera de que se pueda confiar en ellos tanto como en PGP, puesto que después de todo son nuevos y no ha habido tiempo para la revisión y el refinamiento que PGP ha tenido. Todo esto suena bien, y he comenzado a pensar que podía estar bien después de todo. Entonces, leo un informe de noticias que incluye el siguiente párrafo: "'La normas propuestas por el Presidente Clinton para relajar las restricciones sobre la exportación de criptografía no nos afectan a nosotros', dijo el Vice-Presidente de Marketing y Co-fundador de HushMail Jon Gilliam, 'porque su proposición sólo atañe al cifrado de 128-bits', lo cual es (ocho veces menos potente que la utilizada por HushMail)." Gilliam indica que el cifrado de 128-bits ha sido superado. O el periodista o Mr.Gilliam no tienen la más remota idea acerca de lo que es la criptografía.

http://www.internetnews.com/stocks/vcwatch/article/0,1087,archive_6561_206381,00.html

Un email malicioso, diciendo contener un anexo con un contador para el año 2000 de Microsoft, es de hecho un virus Troyano que busca nombres de usuario, y passwords con el propósito de enviarlas a su autor.

http://www.wired.com/news/print_version/technology/story/21823.html?wnpg=all

<http://www.zdnet.com/zdnn/stories/news/0,4586,1017257,00.html>

Ha habido algunas extrañas noticias acerca de un ordenador-cuántico de mano capaz de romper el RSA casi en tiempo real, supuestamente desarrollado en el Instituto Weitzmann de Israel. Esto se parece bastante a la película *Sneakers*, y hasta donde puedo decir, no hay ninguna verdad en este rumor. Hay un fragmento de noticias de verdad que ha salido de esto: el Instituto Europeo de Computación Cuántica ha sido anunciado.

<http://www.the-times.co.uk/news/pages/tim/99/09/29/timintint02001.html?999>

Un buen artículo sobre la privacidad y las relaciones negocio/consumidor:
http://www.hudson.org/American_Outlook/articles_sm99/sparkman.htm

CÓmo nuestra creciente confianza en los ordenadores esta mermando nuestra privacidad.
<http://www.technologypost.com/internet/WEEKLY/19990920201539262.asp?Section=Main>

Extorsión electrónica. Bancos Alemanes y Británicos estan en el punto de mira de los criminales que amenazan la seguridad de los ordenadores.
<http://www.theregister.co.uk/990920-000034.html>

Richard Smith ha recopilado mas o menos una docena de trucos para evaluar los webs que te permiten el anonimato. Ha encontrado bastantes problemas. Leed esto antes de confiar en estas webs.
<http://www.tiac.net/users/smiths/anon/anonprob.htm>
<http://www.tiac.net/users/smiths/anon/test.htm>

La Asociación Internacional para el Contraterrorismo y los Profesionales de Seguridad (si, realmente existen) tenían una conferencia este mes. Suena a un monton de historias de miedo, y a no mucha información.
<http://www.iacsp.com/>
<http://www.wired.com/news/news/politics/story/22146.html>

En un fabuloso ejemplo de cooperación de una comunidad en la red, mas de 300 practicantes de la seguridad aislaron el comportamiento del difundido troyano RingZero de ataque por proxy, encontraron el Troyano, crearon defensas, y, como resultado, la pagina Rusa que lo estaba utilizando para reunir datos fue desconectada y muchos sitios mejoraron sus defensas contra los ataques de proxy.
<http://www.sans.org/newlook/resources/flashadv.htm>

Un articulo sobre el trafico online de propiedad intelectual robada. Interesante area de crecimiento de crimen.
<http://www.techweb.com/wire/story/reuters/REU19991005S0004>

Esto parece mas un rumor y una insinuación que un hecho, pero hay una reclamación de codigo malicioso añadido dentro del proceso de reparación del año 2000.

<http://news.cnet.com/category/0-1009-200-428804.html>
http://dailynews.yahoo.com/h/nm/19990930/tc/yk_code_1.html
<http://www.zdnet.com/zdnn/stories/news/0,4586,2345508,00.html>

India niega cualquier incorrección.

<http://news.cnet.com/category/0-1009-200-429387.html>
<http://www.wired.com/news/news/politics/story/22041.html>

El noveno tribunal de la Corte de Apelaciones ha aceptado una nueva vista en el caso Bernstein. Este es el reglamento de Mayo de 1999 que dice que los programas de cifrado y las fórmulas matemáticas que les acompañan son expresiones de ideas y no pueden ser suprimidas por el gobierno.

<http://www.techserver.com/noframes/story/0,2294,500040274-500065347-500103132-0,00.html>

Las instituciones financieras tienen su propia red de alerta de seguridad. El Centro de Análisis y Compartición de Información de Servicios Financieros (FS/ISAC) es una organización que permitirá a las instituciones financieras seguir las tendencias, compartir información, y obtener informes de incidentes... todo anonimamente.

<http://www.techserver.com/noframes/story/0,2294,500040417-500065579-500104529-0,00.html>

El gobierno de los EEUU dice: "Decid no al hacking." Esto es realmente sorprendente.

http://www.thestandard.com/articles/article_print/0,1454,6711,00.html

Uno de los mayores problemas con las herramientas de red de detección de intrusos y escaneadores de vulnerabilidades es que tienen diferentes maneras de llamar a las vulnerabilidades. Si utilizas dos herramientas, no hay una manera fácil de comparar los resultados. Para resolver este problema, Mitre y otros han desarrollado un diccionario de acerca de 300 vulnerabilidades conocidas llamado la lista de Vulnerabilidades y Riesgos Comunes (CVE).

http://www.mitre.org/news/articles_99/cve_release.shtml

El gobierno da marcha atrás sobre Fidnet.

<http://www.wired.com/news/news/politics/story/22001.html>

En otro esfuerzo distribuido de cracking usando fuerza bruta, un grupo de 200 personas (usando 740 ordenadores) crackearon una clave criptográfica de curvas elípticas de 97-bits. Certicom indica que este esfuerzo es dos veces el esfuerzo requerido para romper una clave RSA de 512-bits, y es mas evidencia de la superioridad de la criptografía de curvas elípticas sobre el RSA convencional. Hablaré acerca de esto en detalle el proximo mes.

<http://www.certicom.com/press/99/sept2899.htm>

<http://www.computerworld.com/home/news.nsf/all/9909282ellip>

Con la popularidad del DSL, cable, y otras conexiones de Internet en el hogar a alta velocidad, mas ordenadores inseguros estan en Internet que nunca antes. Este sitio te permite ejecutar un rápido test para ver si tu ordenador es vulnerable a ciertos ataques. Pasarlo no significa que tu ordenador sea seguro, pero fallarlo ciertamente significa que es inseguro.

<http://grc.com>

5. NOTICIAS SOBRE SEGURIDAD EN INTERNET

Por Bruce Schneier

Traducción: Miguel Camacho

La division de seguridad en internet de Counterpane esta contratando personal, para mas detalles vea:

<http://www.counterpane.com/jobs.html>

Bruce Schneier impartira durante medio dia un tutorial sobre criptografia durante la conferencia en seguridad de ordenadores y comunicaciones (ACM) que se celebrara en Sinpagore del 1-4 de Noviembre de 1999.

<http://www.isi.edu/ccs99/>

Bruce Schneier impartira tres seminarios en la conferencia de institutos anual sobre seguridad en los ordenadores que se celebrara del 15-17 de Noviembre de 1999 en Washington DC:

Arboles de ataque: Modelando amenazas reales de ataque -- Lunes, 2:00 pm - 3:15 pm

Cómo pensar acerca de la seguridad -- Martes, 11:15 am - 12:30 pm

Fallos en productos Criptograficos -- Martes, 4:00 pm - 5:15 pm

<http://www.gocsi.com/>

Un perfil personal de Bruce Schneier aparecio en el USA Today. Aparentemente no posee un direccion URL permanente, pero puede localizarlo mediante busqueda por "Bruce Schneier" en:
<http://search2.usatoday.com/>

Entrevista con Schneier sobre Back Orifice, aparecida en Computerworld:
<http://www.computerworld.com/home/print.nsf/all/990927C40E>

Tambien un articulo sobre los comentarios de Schneier referentes a la polemica de la llave NSA.
<http://www.computerworld.com/home/news.nsf/all/9909094nsakey>

Por ultimo, un articulo sobre Microsoft y seguridad cita a Schneier:
<http://www.cnn.com/TECH/computing/9909/28/ms.security.idg/index.html>

6. EN LA RATONERA: AMD

Por Bruce Schneier
Traducción: Miguel Camacho

AMD posee una tecnologia denominada "Paquete Magico" que permite a través de una red activar un PC apagado o en modo suspendido. En ningún sitio se hace mención alguna a su seguridad, por lo que puede estar seguro que alguien desarrollará un conjunto de herramientas para hackers que enciendan PC apagados a través de la red.

Ahora, apagar su ordenador no lo hace más seguro; necesita tirar del enchufe de la pared.

<http://www.amd.com/products/npd/overview/20212.html>

7. EMPRESAS PKI Y SUS ESLÓGANES

Por Bruce Schneier
Traducción: Miguel Camacho

He encontrado esto divertido. Aquí están las grandes, y pequeñas, compañías PKI y sus consignas corporativas. Fueron pagados montones de

dinero para crear estas consignas y así impresionar apropiadamente:

ABAecom: "Facilitando la banca electrónica y el comercio en Internet".

Baltimore Technologies: "Seguridad electrónica global".

CertCo: "En la raíz del comercio electrónico".

Digital Signature Trust: "Creando confianza en el comercio electrónico".

Entrust: "Traemos la confianza al negocio electrónico".

GTE Cybertrust: "La seguridad para ser estratégico".

Indentrus: "Confianza en línea".

IBM/Lotus: "Localizado, Conectado, Seguro".

Lockstar: "Enlazando la herencia con el futuro".

Shym: "Abriendo el poder de la llave pública".

Thawte: No disponen de eslogan, pero poseen un declaración de intenciones en verso.

Valicert: "Posibilitando confidencialidad global".

Verisign: "El símbolo de la confianza en la red".

Xcert: "Creando confianza en Internet".

8. LONGITUD DE CLAVE Y SEGURIDAD

Por Bruce Schneier

Traducción: Sergio Pozo Hidalgo, Isidre Marques Serret y Daniel Cabezas

A pesar de lo que todo el mundo trata de decirle, la longitud de la llave de cifrado no tiene casi nada que ver con la seguridad. Una llave corta significa poca seguridad, pero una llave larga no significa mucha seguridad.

La cerradura de la puerta principal de su casa tiene una serie de pasadores. Cada pasador tiene muchas posibles posiciones. Cuando alguien

mete una llave en la cerradura, cada pasador se mueve a una posición específica. Si las posiciones de la llave se corresponden con las que la cerradura necesita para abrirse, se abre; si no, no.

Las cerraduras más habituales tienen cuatro pasadores, cada uno de los cuales puede estar en diez posiciones distintas. Eso quiere decir que hay 10.000 llaves distintas. Un ladrón con un llavero muy grande, puede probar toda llave posible, una tras otra, y eventualmente, entrar. Sería mejor que el ladrón fuera paciente, porque si suponemos que tarda quince segundos en probar una llave, le llevaría un mínimo de 21 horas encontrar la llave correcta --y eso no incluye paradas para ir al baño o comer.

Un día, un vendedor llama a su puerta, y quiere venderle una cerradura. Su cerradura tiene seis pasadores con doce posiciones cada uno. Un ladrón, dice, tendrá que probar diferentes llaves en un período de 259 días sin parar antes de que pueda abrir su puerta. ¿Se siente más seguro con esta cerradura?

Probablemente no. Ningún ladrón estaría ni tan siquiera en su entrada durante 21 horas de ninguna manera. Es más probable que forzara la cerradura, echara la puerta abajo, rompiera una ventana, o se escondiera en un arbusto hasta que usted pasara por la entrada. Una cerradura con más pasadores y más posiciones no hará su casa más segura, porque el ataque específico que crea más dificultad --probar toda llave posible-- no es aquel por el que usted está particularmente preocupado. Tan pronto como haya pasadores suficientes como para hacer el ataque no factible, no tiene que preocuparse por él.

Esto mismo es válido para las llaves de cifrado. Si son suficientemente largas, no se preocupe de ellas. Pero qué longitud es suficiente es más complicado que un simple número; depende de la aplicación y de la cantidad de entropía en las llaves.

Comencemos por el principio. Una llave de cifrado es un valor secreto que modifica a un algoritmo de cifrado. Si Alicia y Bob comparten una llave, pueden usar el algoritmo para comunicarse de forma segura. Si Eve, una fisgona, no conoce la llave, no puede leer los mensajes de Alicia o Bob. Está obligada a probar y "romper" el algoritmo; esto es, intentar averiguar la llave a partir del texto en claro.

Algo obvio que podría hacer es probar toda llave posible, como el hipotético ladrón del principio de este artículo. Si la llave tiene un largo de n bits hay 2^n llaves posibles. Luego si la llave es de 40 bits, hay alrededor de un trillón de llaves posibles. Esto sería imposible, por lo aburrido, para un ladrón; pero los ordenadores sobresalen en estas tareas tan aburridas. Un ordenador tiene que probar

alrededor de la mitad de las llaves posibles como mínimo antes de encontrar la correcta; luego uno capaz de probar un billón de llaves por segundo, tardaría 18 minutos en encontrar la llave correcta de 40-bit. El Deep Crack, la máquina rompedora de DES, probaba 90 billones de llaves por segundo; podría encontrar una llave DES de 56-bit en un mínimo de 4,5 días. El proyecto de búsqueda distribuida de llaves en Internet, distributed.net (que incluía a Deep Crack), probaba 250 billones de llaves por segundo como pico.

Todo esto se escala linealmente. En 1996, un grupo de criptógrafos (incluyéndome a mí), investigamos las tecnologías que se podrían usar para construir máquinas criptoanalíticas de fuerza bruta y recomendamos una longitud mínima de llave de 90-bit para conseguir seguridad hasta el 2016. Triple DES tiene una llave de 112-bit, y los algoritmos más modernos tienen al menos una llave de 128-bit. Incluso una máquina billones de veces más rápida que Deep Crack, tardaría 10^{15} años en probar todas las 2^{112} llaves y recuperar el texto plano. Incluso asumiendo que la ley de Moore siguiera siendo cierta para unos cuantos cientos de años más, ésto seguiría siendo seguro por un largo período de tiempo.

Entonces, ¿qué más hay por ahí para preocuparme? ¿Por qué no hacemos llaves de un zillón de bits y estar seguros hasta el fin de los tiempos? Para responder a esto, necesito explicar la entropía.

La entropía es una medida de la incertidumbre. Cuanto más incierto es algo, más entropía hay en ello. Por ejemplo, si una persona tomada al azar de la población general es hombre o mujer, la variable "sexo" tiene un bit de entropía. Si una persona tomada al azar prefiere a uno de los 4 Beatles, y cada uno de ellos tiene, en principio, las mismas posibilidades, eso corresponde a 2 bits de entropía. En cambio, el sexo de una persona en el equipo olímpico masculino de natación no tiene entropía: todos son hombres. La entropía de la preferencia entre los Beatles en una reunión de un club de fans de John Lennon es que es más probable que una persona tomada al azar prefiera a John. Cuanto más certeza hay en la variable, menor es la entropía.

Lo mismo se aplica a las claves criptográficas. Sólo porque un algoritmo acepte claves de 128 bits no significa que tenga 128 bits de entropía en la clave. O, más exactamente, la mejor manera de romper una implementación de un algoritmo de cifrado de 128 bits puede no ser probar cada una de las posibles claves.

Así, la primera preocupación debe ser la calidad del algoritmo de cifrado. Todos los cálculos arriba expuestos asumían que el algoritmo tomaba las claves que le eran dadas y las usaba perfectamente. Si hay defectos en el algoritmo, ello reduce la entropía en las claves. Por

ejemplo, el algoritmo A5/1, usado en los teléfonos celulares GSM europeos, tiene una clave de 64 bits, pero puede ser roto en el tiempo necesario para forzar una clave de 40 bits. Esto significa que incluso a pesar de que al algoritmo le sea dada una clave criptográfica con 64 bits de entropía, éste solo hace uso de 40 bits de entropía en la clave. Ud. podría usar también un algoritmo que efectivamente use una clave de 40 bits.

Este problema es la explicación de que el proceso de elección AES (American Encryption Standard - Estándar de Encriptación Americano) sea tan largo. El gobierno de Estados Unidos quiere reemplazar DES (Digital Encryption Standard - Estándar de Encriptación Digital), que emplea una clave de 56 bits, con un nuevo algoritmo que acepte claves de hasta 256 bits. Hasta ahora hay 5 semifinalistas para el estándar, pero, ¿Realmente alguno de ellos da la entropía de 256 bits que afirma ofrecer? También por ello productos que anuncian el uso de claves de cientos de bits son difíciles de tomar en serio: no entienden como funcionan las claves y la entropía.

La segunda preocupación es el origen de las claves. Todos los cálculos de longitud de la clave que hice simplemente asumen que cada clave tiene una entropía máxima cuando se genera. En otras palabras, asumí que cada clave es igualmente probable. Esto simplemente no es verdad. Se generan muchas claves a partir de contraseñas, en forma de palabra o frase. Un sistema que acepta contraseñas de 10 caracteres ASCII podría necesitar 80 bits para representarlas, pero tiene mucho menos de 80 bits de entropía. Los caracteres ASCII con bit alto no aparecerán en absoluto, y las contraseñas que son palabras reales (o similares a palabras reales) son mucho más probables que las series de caracteres al azar. He visto estimaciones de la entropía del inglés estándar de 1,3 bits por carácter; las contraseñas probablemente tienen menos de 4 bits de entropía por carácter. Esto significa que una contraseña de 6 caracteres es más o menos igual que una clave de 32 bits, y si quiere una clave de 128 bits va a necesitar una contraseña de 98 caracteres en inglés. Ve, un sistema de forzado de contraseñas por fuerza bruta no va a probar cada posible clave en orden. Va a probar primero las más probables, y luego probará el resto según su orden de probabilidad. Probará las contraseñas comunes como "contraseña" y "1234," luego el diccionario inglés entero, luego variaciones en mayúsculas y números añadidos, y así sucesivamente. L0phtcrack es un programa de forzado de contraseñas que hace eso; puede cotejar un archivo de 200 contraseñas con un diccionario 8 Megabytes de contraseñas populares en menos de un minuto en un Pentium Pro 200. Cotejar el alfabeto entero de 26 caracteres le cuesta 26 horas, y el alfanumérico de 36 caracteres le cuesta aproximadamente 250 horas.

Este es el motivo por el cual es de risa que compañías como Microsoft

presuman de cifrado de 128 bits y luego basen la clave en la contraseña. (Esto describe casi toda la seguridad de Windows NT.) Los algoritmos que usan podrán aceptar una clave de 128 bits, pero la entropía de la contraseña es mucho, mucho menor. De hecho, no importa la calidad de la criptografía o su longitud; las contraseñas débiles seguirán destrozando el sistema.

Algunos han combatido este problema exigiendo contraseñas más y más grandes, pero esto ya no es eficaz. Durante las últimas décadas, la ley de Moore ha hecho posible utilizar la fuerza bruta con claves de entropía más y más grande. Al mismo tiempo hay un límite en la entropía que el usuario medio de ordenador (o incluso el usuario avanzado) está dispuesto a recordar. No se puede esperar que memorice una serie de 32 caracteres hexadecimales al azar, pero eso es lo que tendría que hacer para memorizar una clave de 128 bits. Estas dos cifras se han encontrado; los programas forzadores de contraseñas ya pueden forzar ahora lo que se puede esperar razonablemente que un usuario memorice. Las contraseñas buenas son difíciles memorizar, el usuario se quejará, pero esta dificultad precisamente es la causa por la cual son consideradas buenas.

Las claves generadas al azar no son necesariamente mejores, porque el generador de números al azar debe producir claves con la máxima entropía. Una debilidad en el generador de números al azar es lo que rompió el cifrado en Netscape versión 1.1. Mientras que se usaba el generador de números al azar para generar claves de 128 bits, su entropía máxima estaba alrededor de 20 bits. Así que el algoritmo no era mejor que si usara una clave de 20 bits.

Las soluciones existen, pero requieren compromisos en el diseño. La Biometría puede generar mejores contraseñas, por lo menos porque existe suficiente entropía para hacerlo, por ejemplo, una huella digital (mi suposición es que es equivalente a una clave de unos 60 bits), y además porque no existe nada equivalente a una mala huella digital así como hay malas contraseñas. Las tarjetas inteligentes ofrecen una manera adecuada de manejar una clave de alta entropía, pero tienen las restricciones asociadas con un dispositivo físico. Y para algunos sistemas de comunicaciones, los protocolos de clave pública pueden generar claves privadas con una alta entropía usando únicamente información pública. La verificación en línea de las contraseñas que previene los ataques de diccionario fuera de línea, también funciona en algunas circunstancias. Este es un buen trato. Vemos sistemas de PKI complejos donde la clave privada esta protegida con una contraseña. Casi todos los productos de cifrado de discos duros basan su seguridad en una clave memorizada por el usuario. Casi toda la seguridad de Windows NT se derrumba porque esta basada en contraseñas memorizadas por el usuario. Incluso PGP se cae en pedazos si el usuario elige una mala contraseña. No importa que

algoritmos usen o el tamaño de las claves; los secretos memorizados por el usuario no son seguros.

* Una nota sobre los Algoritmos de Clave Pública

Este ensayo habla sobre los algoritmos simétricos (tanto de cifrado en bloque como no), que toman series de bits arbitrarios como claves. Los sistemas de clave pública que emplean claves matemáticas como el producto de dos primos grandes, son diferentes. Ya hay ataques de fuerza bruta contra los sistemas de clave pública, pero implican resolver problemas matemáticos como la factorización. El grupo que acaba de factorizar una clave RSA de 512 bits dijo que el cálculo requirió aproximadamente el 2% del esfuerzo necesario para buscar una clave de 56 bits. Las estimaciones sobre la futura seguridad de las claves RSA son mucho más difíciles, porque tenemos que considerar los adelantos en las matemáticas de la factorización, así como los adelantos en la velocidad de los ordenadores y su conexión en red.

Las estimaciones conservadoras indican que las claves de 1024 bits son suficientemente buenas para los próximos años, y las claves de 2048 bits deberían ser bastante buenas durante aproximadamente diez años. Pero como nadie conoce la “dificultad” de la factorización, es ciertamente posible que pueda aparecer un inteligente matemático convertir incluso estas importantes longitudes en inseguras.

Las claves RSA tienen, por supuesto, demasiada entropía para memorizarse. Se cifran con una contraseña cualquiera y se guardan en la unidad de disco duro, o en un dispositivo como una tarjeta inteligente. A veces incluso se dejan sin cifrar.

Documento sobre la longitud de las claves:
<http://www.counterpane.com/keylength.html>

Del Boletín #141:

PEOR IMPOSIBLE: GRAVE DEFICIENCIA DE SEGURIDAD EN MICROSOFT ACCESS

Artículo elaborado para KRIPTOPOLIS por Juan Carlos García Cuartango <cuartango@teleline.es>, en base a la información proporcionada por Rafael Santos rsantosf@retemail.es

El informático cacereño Rafael Santos, lector de KRIPTOPOLIS y practicante del criptoanálisis (como a continuación se demuestra), nos ha informado en primicia de una grave deficiencia en la protección de las populares bases de datos de Microsoft Access 97. Dichas bases de datos permiten la protección de las mismas mediante una palabra de paso, que es almacenada en el propio archivo de base de datos (.MDB) tras ser protegida mediante un algoritmo de cifrado.

¿Es segura la implementación de este mecanismo?. Rafael ha demostrado que el sistema de cifrado de dicha clave dentro del archivo que contiene la base de datos es de una inocencia infantil. El descubridor ha tenido además la gentileza de explicarnos la metodología utilizada para el descubrimiento tanto del algoritmo como de la clave.

Metodología utilizada

Rafael comienza comparando el contenido de un archivo (.MDB) de base datos sin protección, con el de la misma base de datos con protección por una palabra clave, y concluye que existen ciertas diferencias a partir de la posición 42H (hexadecimal y considerando 0 el primer octeto del archivo). Repitiendo el experimento observa que la clave (aunque admita 14 caracteres) siempre se almacena en los 13 octetos que comienzan en la posición 42H. Lo que para nuestro criptoanalista resulta increíble es que la palabra clave se almacena tras haber realizado una sencilla operación XOR (OR exclusiva) de la contraseña con una clave constante de 13 octetos.

¿Cuál es la clave de cifrado que se utiliza para el XOR?

Os invito, lectores de KRIPTOPOLIS a practicar el método que acabamos de describir para encontrar la clave de cifrado de MS Access 97. Yo lo he hecho y me siento criptoanalista en ciernes...

Si alguien tiene prisa por encontrar dicha clave, la cosa es aún más fácil. No hay nada que calcular, pues basta con crearse una base de datos sin proteger y mirar el contenido de los octetos 42H a 4EH, que contendrán la clave utilizada para la operación XOR.

Puede que esto sea tan sólo un ejemplo de una nueva tecnología criptográfica inventada por el fabricante de Access 97, que podría pasar a los manuales de criptografía con el curioso nombre de "Criptografía XOR de clave pública".

Aparte de felicitar a Rafael Santos por su descubrimiento, quiero agradecer su desinteresado gesto de hacerlo público por primera vez pues, aunque su hallazgo no es nuevo, las gentes que anteriormente realizaron el mismo descubrimiento se lo han callado, para explotarlo

mediante programas comerciales, cuyo precio va desde los 35\$ hasta los 150\$ (al final del artículo mostramos una relación de algunos de estos sitios, obtenida tras escasos minutos de búsqueda).

Rafael nos ha remitido también, como demostración de su hallazgo, un programa en versión Visual Basic y Delphi que, tras unas veinte líneas de código, nos muestra directamente en pantalla la contraseña de cifrado de cualquier base de datos Access 97. Lo bueno si breve...

¿Tiene solución (parche) el problema?

Creo que es demasiado tarde. Existen demasiados millones de bases de datos Access protegidas con el algoritmo que hemos descrito. Por cuestiones de compatibilidad, un hipotético parche debiera contener la posibilidad de abrir dichas bases de datos con el algoritmo actual, con lo cual el problema de seguridad me parece irresoluble.

Coincido con Rafael Santos en que al menos, a partir de ahora, los usuarios de MS Access 97 sean conscientes de que la protección de una base de datos con una palabra de paso es como poner puertas al campo.

¿Será Access 2000 más seguro?

Rafael Santos nos asegura que no. Lo sabremos en primicia en el próximo boletín de KRIPTOPOLIS.

* Algunos crackeadores *COMERCIALES* de passwords para Office:

http://soft4you.com/vitas/downl_mso.htm

<http://www.crak.com/downsoft.htm>

<http://www.accessdata.com/passwdprd.html>

Del Boletín #142:

PROTECCIÓN DE MS ACCESS 2000

(Por Juan Carlos García Cuartango y Rafael Santos)

En el anterior boletín explicábamos el paupérrimo mecanismo con el que MS Access 97 almacena la palabra de paso con la que se protege una base de datos.

Rafael Santos, no contento con su descubrimiento, también ha estudiado la protección que utiliza la nueva versión del producto: MS Access 2000. ¿Es el mecanismo utilizado por la nueva versión más fiable que el de la anterior? Yo diría que si el método utilizado en Access 97 parece haber sido inventado por un niño de 10 años, el de Access 2000 parece haberlo sido por un niño de unos doce años.

EL MECANISMO

En MS Access 2000 se admiten palabras clave de hasta 20 caracteres (antes sólo de 13); a primera vista, parece que el asunto mejora algo.

Sin embargo, esta palabra se almacena -cifrada- en los octetos 40H, 42H, 44H... y siguientes del archivo de base de datos. Los octetos impares no utilizados podrían estar relacionados con el uso del alfabeto UNICODE de 16 bits por carácter.

¿Cómo se cifra la palabra de paso? De nuevo mediante una "sofisticada" operación XOR aunque -eso sí- esta vez la clave no es fija y universal, sino función del día en que se creó la base de datos. Es decir: todas las bases de datos creadas el 10 de Agosto de 1999 van a tener la misma clave de protección de la palabra de paso. Además, cuando se crea la base de datos sin proteger, los octetos 40H, 42H... contienen la clave para el cifrado XOR.

IDEAS PARA LA DESPROTECCIÓN

Dicho esto, es bastante fácil imaginar un mecanismo de desprotección de cualquier base de datos MS Access 2000.

1- Créense bases de datos sin proteger, digamos desde el 1-1-1999 hasta 31-12-2000. Hacer esto a mano resultaría un poco pesado, pero hacerlo mediante un programa Visual Basic o Delphi es cuestión de poco tiempo. Leyendo el contenido de las famosas posiciones 40H, 42H... de los archivos .MDB se tienen las 630 claves posibles para esos dos años.

2- Dada una base de datos de la que queremos averiguar su palabra de paso, hay que escribir un programa que abra la base de datos utilizando las claves obtenidas en el paso uno. Es de suponer que una de las 630 claves nos valdrá, siempre y cuando la base de datos haya sido creada en el intervalo para el que se han calculado las claves.

Por supuesto que sería posible buscar la relación entre la fecha y la

clave generada, pero sinceramente pienso que no merece la pena el esfuerzo.

CÓMO PROTEGERSE

Por último, un consejo práctico para proteger su base de datos creada con Access 2000:

Antes de abrir su programa Access 2000 vaya al reloj del sistema y ponga una fecha difícil de averiguar (pasada o futura), por ejemplo: el 7 de Julio (San Fermin) del año 26.897 antes de Cristo. Cree a continuación su base de datos. Ya puede volver a poner la fecha real en su ordenador. De este modo, si a usted le roban la base de datos, el ladrón nunca sospechará que la base de datos fue creada en la fecha que usted utilizó, y el programa craqueador del intruso fallará.

Del Boletín #145:

"HACKERS": ¡OTRO LIBRO GRATIS PARA TODOS NUESTROS LECTORES!

Como recordarán, hace meses anticipamos que el "efecto Lucena" tendría continuación. Pues bien; esta semana podemos anunciar a nuestros lectores que otro autor español, Claudio Hernández, ha decidido regalar desde KRIPTÓPOLIS su última obra, completa y sin restricciones, que lleva el atractivo título "Hackers".

Se trata de una obra sencilla que contribuirá a introducir a muchos de nuestros lectores a temas tan fascinantes como algunos de los que figuran en el índice del libro:

- * Internet y su funcionamiento
- * Virus, gusanos, bombas lógicas...
- * Hackers, crackers, gurus, lamers, newbies, lamers, phreakers...
- * Preguntas y respuestas sobre hackers
- * El manual del hacker
- * El software del hacker
- * Historias de hackers y crackers
- * Software gratis en Internet: trucos
- * Criptografía y criptoanálisis
- * Echelon: espías en el cielo,
- * etc, etc.

El libro consta de 107 páginas, está en formato PDF y se presenta en dos versiones:

* Una para Windows, con programa de autoinstalación que genera también los iconos de escritorio y menú Inicio para acceder al libro:
<http://www.kriptopolis.com/hackers-exe.zip> (296 KB)

* Otra que incluye sólo el propio libro, sin utilidad de instalación. Apropiada para otras plataformas (Mac, Unix...) o usuarios Windows que no quieren autoinstalación.
<http://www.kriptopolis.com/hackers-pdf.zip> (240 KB)

Ambas direcciones de descarga son PROVISIONALES. Información complementaria sobre requisitos de instalación, índice abreviado y nuevos sitios de descarga para ambas versiones se irán añadiendo a la página web que hemos creado a tal efecto:
<http://www.kriptopolis.com/hackers.html>

Además de a nuestra sección de "Publicaciones", donde también figuran enlaces a otros libros de interés (entre ellos, el de Manuel Lucena):
<http://www.kriptopolis.com/pubs.html>

Sólo nos resta agradecer a Claudio Hernández su generoso gesto y recordar a nuestros lectores que, como es habitual en nuestras publicaciones electrónicas, el autor está abierto a todo género de correcciones y sugerencias para mejorar la obra en sucesivas ediciones.

Criptograma #19:

1. POR QUÉ SON INSEGUROS LOS ORDENADORES

Por Bruce Schneier
Traducción: Isidre Marques Serret

Casi cada semana la prensa informática habla de otro fallo de seguridad: un virus que se aprovecha del Office de Microsoft, una vulnerabilidad en Windows o UNIX, un problema de Java, un agujero de seguridad en una de las páginas principales de Internet, un ataque contra un popular cortafuegos. ¿Por qué no pueden conseguir los diseñadores que esto

funcione?, nos preguntamos. ¿Cuándo mejorará?

Yo no creo que lo haga nunca. He aquí el por qué:

La ingeniería de la seguridad es diferente de cualquier otro tipo de ingeniería. La mayoría de los productos, como procesadores de textos o teléfonos celulares, son útiles por lo que hacen. Los productos de seguridad, o las características de seguridad ofrecidas dentro de estos productos, son útiles precisamente por lo que no permiten que se haga. La mayor parte del diseño implica hacer que las cosas funcionen. Piense en la definición original de un hacker: alguien que se las ingenia para hacer que pasen cosas "diferentes". El diseño de seguridad implica evitar que sucedan estas cosas. Implica deducir cómo fallan las cosas, y a partir de ahí prevenir estos fallos.

En muchos aspectos esto es similar a la ingeniería de seguridad. La seguridad es otro requisito de la ingeniería que no es simplemente una "característica". Pero el diseño de la seguridad implica asegurarse de que las cosas no fallarán en presencia de fallos aleatorios: es programar el ordenador de Murphy, si usted quiere. El diseño de la seguridad implica lograr que las cosas seguras no fallen en presencia de un adversario inteligente y malévolo que fuerza estos fallos precisamente en el peor momento y de la peor manera. El diseño de la seguridad implica programar el ordenador de Satanás.

Y el ordenador de Satanás es difícil de probar.

Prácticamente todo el software se desarrolla a través de un sistema de "prueba-y-error". Se programan pequeños fragmentos, se prueban, se arreglan, y se prueban de nuevo. Varios de estos pequeños fragmentos se combinan en un módulo, y entonces este módulo se prueba, se arregla, y se prueba de nuevo. Luego se combinan varios de estos pequeños módulos en módulos más grandes, y así sucesivamente. El resultado final es un software que más o menos funciona como se esperaba, aunque en programas complejos siempre se escapa algún error.

El método de "prueba-y-error" simplemente no sirve para comprobar la seguridad. Ninguna cantidad de pruebas es suficiente para poder descubrir un fallo de seguridad, por lo que el proceso de la comprobación no descubrirá nada. Recuerde que seguridad no tiene nada que ver con funcionalidad. Si usted tiene un teléfono cifrado, puede probarlo. Puede hacer y puede recibir llamadas. Puede intentar, y no lograr, interceptarlo. Pero no tiene ninguna forma de averiguar si el teléfono es seguro o no.

La única manera razonable de "probar" la seguridad es realizar revisiones de seguridad. Esto es un proceso manual caro y largo. No es

suficiente comprobar los protocolos de seguridad y los algoritmos de cifrado. Una revisión debe cubrir especificación, diseño, aplicación, código fuente, funcionamiento, y todo lo demás. Así como la prueba funcional no puede demostrar la ausencia de errores, una revisión de seguridad no puede demostrar que el producto es realmente seguro. Y todavía peor. Una revisión de seguridad de la versión 1.0 dice poco sobre la seguridad de versión 1.1. Una revisión de seguridad de un producto del software aislado no sirve necesariamente para el mismo producto en un ambiente operacional. Y cuanto más complejo es el sistema, más dura se vuelve una evaluación de seguridad y más fallos de seguridad habrá.

Supongamos que un producto de software se desarrolla sin ninguna comprobación funcional en absoluto. Ninguna comprobación funcional de las versiones alfa o beta. Se escribe el código, se compila, y se envía. Las posibilidades de que este programa simplemente funcione –incluso dejando de lado que esté libre de errores- es cero. A medida que aumenta la complejidad del producto, aumentará el número de errores. Todos sabemos que las pruebas son esenciales.

Desgraciadamente, este es el estado actual, en la práctica, de la seguridad. Están distribuyéndose productos sin ninguna, o con mínimas comprobaciones de seguridad. No me sorprende que los errores de seguridad se presenten una y otra vez. No puedo creer que nadie espere otra cosa.

Aún peor; los productos se vuelven más complejos cada año: los sistemas operativos son más grandes, tiene más características, más interacciones entre los diferentes programas en Internet. Windows NT ha estado funcionando durante algunos años, y todavía se descubren errores de seguridad. Podemos esperar muchos más errores en Windows 2000; el código es significativamente más grande. Podemos suponer que lo mismo será cierto para cualquier otro producto de software.

Esto no cambiará. El incremento en el uso de los ordenadores y la convergencia en Internet, están aumentando a un ritmo creciente. Los sistemas se están convirtiendo en más y más complejos, y necesariamente más inseguros, más rápido de lo que podemos arreglarlos (e incluso más rápidamente de lo que podemos aprender a arreglarlos).

Reconocimientos: La frase "programando el ordenador de Satanás" era originalmente de Ross Anderson. Pero es demasiado buena como para evitar usarla. Una versión acortada de este ensayo aparecía originalmente en número del 15 de noviembre de la revista *_Computerworld_*.

2. COUNTERPANE: INVESTIGACION DOCUMENTADA

Por Bruce Schneier

Traducción: David Gómez

"Autenticar items seguros utilizando acceso lento a memoria"

John Kelsey y Bruce Scheneier, Primer Simposio en USENIX sobre las tarjetas inteligentes, USENIX Press, por aparecer.

Presentamos un protocolo de autenticación que permite que un item, como una tarjeta inteligente, se autentifique en un sistema informático seguro en el otro extremo a traves de un lector inseguro. Este protocolo se basa en el hecho de que el item respondera a las consultas de forma lenta, y que el propietario del objeto no se sentara pacientemente mientras el lector parece que no funciona. Este protocolo puede ser usado de forma independiente, con items de memoria "tontos", o con items basados en procesadores.

<http://www.counterpane.com/slow-memory.html>

3. NOTICIAS

Por Bruce Schneier

Traducción: David Gómez

Una compañía ha desarrollado un paquete de cifrado de correo que permite al remitente elegir durante cuánto estará disponible la clave de descifrado, convirtiendo cualquier copia de ese correo en ilegible después de la fecha elegida. Es un buen sistema para prevenir que la gente accidentalmente olvide borrar su correo (a la vista del juicio de Microsoft, muchas compañías tienen ya políticas de borrar todo el correo antiguo a partir de una cierta fecha.) No es nada bueno, sin embargo, como una medida de seguridad para *prevenir* a alguien de grabar su correo pasada esa fecha. Alguien puede grabar siempre el correo descifrado en un fichero, incluir el correo en un mensaje saliente sin la restricción, o incluso hacer una captura de pantalla y grabar una copia del correo de esa manera. Este puede ser un buen producto, pero no se engañe a si mismo pensando que sirve como una medida de seguridad: <http://www.technologypost.com/internet/DAILY/19991011102015158.asp?Section=Main>

El idioma como criptografía. Estas historias hablan acerca de un lenguaje secreto utilizado en China. La segunda historia menciona que el

castigo en la China Imperial por inventar un lenguaje secreto era la muerte:

<http://www.smh.com.au:80/news/9910/12/text/world5.html>

http://www.foxnews.com/scitech/101899/chinese_lingo.sml

Arjen Lenstra y Eric Verheul han escrito un excelente documento sobre las longitudes de las claves: simétricas, claves públicas (incluyendo curvas elípticas), etc. Comparan las longitudes de las claves de diferentes sistemas, y efectúan predicciones sobre el futuro. Bájese esto:

<http://www.cryptosavvy.com>

El Departamento de Policía de Los Angeles ha sido acusado de efectuar cientos de escuchas ilegales durante varios años. Tenga esta historia a mano la próxima vez que alguien le diga que las escuchas de la policía son algo bueno, y que por supuesto se puede confiar en la policía:

<http://www.zdnet.com/zdnn/stories/news/0,4586,2378149,00.html?chkpt=hpqsnewst>

Falsa alarma: Jaws Technologies está aprovechando una ocasión para sus adquisiciones:

<http://www.nationalpost.com/network.asp?f=991103/117402>

Departamento de malas ideas: la Internet Engineering Task Force (IETF) estaba considerando introducir posibilidades de escucha en futuras versiones de Internet. Cualquiera que tenga voz debería presentarse en los encuentros del IETF para oponerse a esto:

<http://www.wired.com/news/print/1,1294,31895,00.html>

El Congresista Bob Barr se pronuncia contra esta idea:

<http://www.wired.com/news/print/1,1294,32100,00.html>

El sentido común prevaleció, y en el reciente encuentro de la IETF en D.C, decidieron no llevarlo a efecto.

<http://www.wired.com/news/politics/0,1283,32455,00.html>

Una compañía llamada SecureLogix está construyendo un firewall para los PBXs. Esto es una buena idea.

<http://www.securelogix.com>

La administración Clinton está pensando en relajar los controles de exportación sobre el código fuente criptográfico:

<http://www.wired.com/news/print/1,1294,32006,00.html>
<http://www.computerworld.com/home/news.nsf/all/9910225source>

Los analistas del grupo Gartner están haciendo mucho ruido acerca de los virus del año 2000:

<http://www.computerworld.com/home/news.nsf/all/9910122gartner2>

Y el FBI dice que ha recogido 30,000 amenazas de virus relacionados con el año 2000. Aunque esto se parece un poco a la lista McCarthy's de los 500 comunistas conocidos, tiene sentido el ser extremadamente cuidadoso al abrir adjuntos de los correos y bajarse nuevo software hacia el cambio de año:

<http://www.zdnet.com/zdnn/stories/news/0,4586,2386686,00.html?chkpt=zdnsttop>

Microsoft ha respondido ofreciendo software anti-virus gratuito, olvidando aparentemente que el software obtenido hoy no ayudará contra virus que no darán la cara hasta el año que viene:

<http://www.computerworld.com/home/news.nsf/all/9911011msy2k>
<http://www.infoworld.com/cgi-bin/displayStory.pl?99111.hnmsy2k.htm>

La Agencia Nacional de Seguridad valorará ahora la seguridad de las redes para agencias de defensa y civiles. Intentarán incluso romper los sistemas de las agencias para señalar vulnerabilidades:

<http://www.fcw.com/pubs/fcw/1999/1101/fcw-agnsa-11-01-99.html>

Excelente entrevista con Ross Anderson de la revista New Scientist:

<http://www.newscientist.com/ns/19991106/confidenti.html>

TEMPEST es el nombre en código de la NSA para la habilidad de espiar los equipos electrónicos interceptando y decodificando sus señales electromagnéticas. Un archivero piensa publicar en su sitio Web documentos, obtenidos de la NSA a través del Acta para la Libertad de Información, que nos proporcionan más detalles:

<http://www.wired.com/news/print/0,1294,32097,00.html>

Vea también:

<http://www.newscientist.com/ns/19991106/newsstory6.html>

La información sobre Echelon sale lentamente a la luz:

<http://www.wired.com/news/print/0,1294,32302,00.html>

<http://washingtonpost.com/wp-srv/WPcap/1999-11/14/019r-111499-idx.html>
<http://www.independent.co.uk/news/Digital/Features/spies151199.shtml>

Lo inevitable ha sucedido finalmente. Alguien inventó un virus de correo que puede infectar su sistema cuando usted lea su correo; no tiene que ejecutar nada. El virus Bubbleboy necesita que el usuario esté ejecutando los programas de correo Microsoft Outlook o Outlook Express; Windows 95, 98, o 2000; y Internet Explorer 5.0 o superior. Su blanco es un agujero de seguridad para el cual Microsoft ya ha creado un parche, pero que muchos usuarios todavía tienen que actualizar:

<http://www.wired.com/news/story/0,1240,32434,00.html>

<http://www.wired.com/news/technology/0,1282,32529,00.html>

El parche de Microsoft:

<http://www.microsoft.com/security/Bulletins/ms99-032.asp>

Mi comentario en Comunicaciones de la ACM sobre el futuro del software malicioso:

<http://www.counterpane.com/insiderisks2.html>

4. ROTO EL CIFRADO DE DVD

Por Bruce Schneier

Traducción: Daniel Cabezas

El sistema de protección de los DVDs ha sido roto. Ahora hay programas freeware en la red que eliminan la protección de copia en los DVDs, permitiendo que sean reproducidos, editados y copiados sin restricción alguna.

Esto no debería ser una sorpresa para nadie, y menos aún para la industria del ocio.

El esquema de protección es gravemente defectuoso en varios aspectos. Cada DVD está cifrado con algo llamado "Content Scrambling System (CCS)" - Sistema de embrollo de contenido. Tiene una clave de 40 bits. (No tengo idea de por qué. La NSA y el FBI no deberían preocuparse del cifrado DVD. No hay películas terroristas cifradas que necesiten observar). Ni tan siquiera es un algoritmo muy bueno. Pero incluso aunque el cifrado fuese triple-DES, este sistema sería defectuoso.

Cada reproductor de DVD, incluyendo las consolas hardware que se

enchufan al televisor, y los reproductores de software que se pueden descargar al ordenador, tienen su propia y única clave de acceso. (En realidad, cada uno tiene varias, no se por qué). Esta clave es usada para dar acceso a la clave de cifrado en cada DVD. Un DVD tiene 400 copias de la misma clave única de descifrado, cada una cifrada con cada código de acceso. Nótese el secreto a voces: si se las arregla para conseguir una clave de acceso para un reproductor, puede descifrar todos y cada uno de los DVD.

Pero incluso si esto fuese completamente perfecto, el sistema nunca podría funcionar.

El defecto se encuentra en el modelo de seguridad. El reproductor de software, finalmente, consigue la clave de descifrado, descifra el DVD y lo muestra por pantalla. Esa información descifrada del DVD está en el ordenador. Tiene que estar, no hay otra manera de mostrarla por pantalla. No importa lo bueno que sea el sistema de cifrado, la información del DVD está disponible en texto en claro (tal cual), para cualquiera capaz de escribir un programa de ordenador para obtenerla.

Otro tanto ocurre con la clave de descifrado. El ordenador debe descifrar el DVD. La clave de descifrado debe estar en el ordenador. Así que la clave de descifrado está disponible, de forma transparente, para cualquiera que sepa dónde buscar. Está protegida por una clave de acceso, pero el lector tiene que darnos acceso a ella.

Se suponía que los fabricantes de software para DVD encubrirían el funcionamiento del programa de descifrado, y posiblemente el programa reproductor, empleando algún tipo de técnicas de ocultamiento del software. Estas técnicas nunca han demostrado funcionar mucho tiempo; solo parecen obligar a los hackers a gastar un par de semanas extra haciéndose una idea de como funciona el software. Ya he escrito sobre esto anteriormente en relación a la protección de copia de software: no se puede ofuscar el software.

Puede que sea un mal trago que aceptar para la industria del ocio, pero la protección de contenidos de software no funciona. No puede funcionar. Se pueden distribuir contenidos cifrados, pero para poder permitir que sean leídos, vistos o escuchados, deberán ser pasados a texto en claro. Un hacker lo suficientemente inteligente, con herramientas de depuración de programas lo bastante buenas, siempre será capaz de invertir el funcionamiento del algoritmo, obtener la clave, o simplemente capturar el texto en claro tras el descifrado. Y puede escribir un programa de software que permita a otros realizar estas tareas automáticamente. Y esto no puede ser impedido.

Si en cambio asumimos hardware seguro, el sistema funciona. (De hecho,

la industria quiere extender el sistema por todo el camino hasta llegar al monitor, y finalmente realizar ahí el descifrado). El ataque funciona porque el hacker puede ejecutar un depurador y otras herramientas de programación. Si el dispositivo de descifrado y el de visionado (deben ser ambos) están dentro de una pieza de hardware a prueba de intromisiones, el hacker se queda atascado en su intento. No puede aplicar ingeniería inversa a nada. Pero el hardware a prueba de intromisiones es en gran manera un mito, así que en la realidad este caso tan sólo sería otra barrera que alguien finalmente superaría. La protección de contenidos digitales simplemente no funciona; pregunte a cualquiera que haya intentado proteger software contra copias.

Una lección más y una observación:

La lección: este es un ejemplo más de una empresa, reunida en secreto para diseñar un algoritmo de cifrado propietario, que termina siendo desconcertantemente débil. Nunca he entendido por qué la gente no emplea algoritmos y protocolos ya publicados, en los que se pueda confiar. Siempre son mejores.

La observación: la “solución” por la que la industria del ocio ha estado pugnando es ilegalizar la ingeniería inversa. Lo han conseguido en los Estados Unidos: el acta de Copyright Milenio Digital incluye disposiciones al efecto, a pesar de las protestas de comunidades científicas y de derechos civiles. (Sí, se podría ir a la cárcel por tener un depurador de código). Han conseguido hacer pasar una ley semejante en el Reino Unido, y están trabajando para lograrlo en la Unión Europea. Esta “solución” no funciona y no tiene ningún sentido. Primero, a menos que la ingeniería inversa sea ilegal en todo el planeta, siempre habrá alguien capaz de aplicarla en algún lugar. Y una persona es todo lo que se necesita, porque puede escribir software que usen todos los demás. Segundo, la ingeniería inversa puede, como en este caso, funcionar anónimamente. Las leyes no habrían ayudado en este caso. Y tercero, las leyes no pueden “meter de nuevo al gato en la bolsa”. Incluso aunque puedas atrapar y encausar a los hackers que lo hicieron, no afectaría a las herramientas de los hackers que ya han sido -y continúan siendo- escritas.

Lo que la industria del ocio sí puede hacer, y han hecho en este caso, es emplear amenazas legales para enlentecer la difusión de estas herramientas. Hasta ahora, la industria ha amenazado con acciones legales contra la gente que ha puesto estas herramientas de software en sus sitios web. El resultado es que estas herramientas existirán en las páginas web hackers, pero nunca estarán en software de dominio público (Linux, por ejemplo).

El tremendo fallo de todo esto es que la industria del ocio es perezosa,

y está intentando encontrar un solución tecnológica a lo que es un problema legal. Es ilegal robar copyrights o marcas comerciales, tanto si es una película DVD, una camisa de Ralph Lauren o un bolso Louis Vitton. Esta protección legal todavía existe, y todavía es fuerte. Por alguna razón la industria del ocio ha decidido que tiene un derecho legal a la protección de su tecnología, y eso no tiene sentido alguno. Por otra parte, están presionando a las parlamentos para aprobar leyes que afiancen esta protección tecnológica defectuosa. En los Estados Unidos y Reino Unido (y posiblemente, pronto en la Unión Europea), es ilegal saltarse su tecnología, incluso cuando nunca se haga para violar un copyright. Es ilegal iniciar investigaciones científicas sobre el cifrado utilizado en esos sistemas. Es ilegal intentar mirar dentro de algo que se adquirido legalmente. Así que no sólo el sistema no funciona, sino que además crea un mercado negro donde no lo había anteriormente, y sin hacer ningún bien a la sociedad durante el proceso. La rotura de la protección del DVD es algo bueno. No servía al interés de nadie que la industria del ocio depositara su confianza en un mal sistema de seguridad. Es una buena investigación la que lleva a mostrar lo malo que es el algoritmo de cifrado y lo mal concebido que está el modelo de seguridad en si. Lo aprendido en esta ocasión puede ser aplicado a hacer los sistemas futuros más resistentes.

<http://www.wired.com/news/technology/0,1282,32263,00.html>

<http://www.ntk.net/index.cgi?back=archive99/now1029.txt>

Resumen del modelo de encriptación DVD:

<http://crypto.gq.nu>

Material para expertos:

<http://livid.on.openprojects.net/pipermail/livid-dev/1999-October/00058.html>

<http://livid.on.openprojects.net/pipermail/livid-dev/1999-October/00059.html>

<http://livid.on.openprojects.net/pipermail/livid-dev/1999-October/00069.html>

<http://livid.on.openprojects.net/pipermail/livid-dev/1999-October/00061.html>

Mi ensayo sobre protección de copia de software:

<http://www.counterpane.com/crypto-gram-9811.html#copy>

Mis comentarios sobre el Acta de Copyrights Digital Milenio:

<http://www.zdnet.com/pcweek/news/0622/22wipo.html>

Nuevas técnicas de ofuscación de software de Intel que, pronostico, serán pronto rotas:

<http://www.intel.com/pressroom/archive/releases/in110999.htm/>

5. EN LA RATONERA: MICROSOFT WINDOWS CE

Por Bruce Schneier

Traducción: Miguel Camacho

Microsoft cifra su contraseña Windows NT cuando se guarda en una unidad Windows CE. Pero si uno mira cuidadosamente el algoritmo de cifrado, simplemente realizan una operación XOR con "susageP" (Pegasus escrito al revés). Pegasus es el nombre clave de Windows CE. Es tan patético que resulta asombroso.

<http://www.cegadgets.com/artsusageP.htm>

6. PUESTA AL DÍA DE LA LEGISLACIÓN EN ESTADOS UNIDOS

Por Bruce Schneier

Traducción: Miguel Camacho

A veces parece que nunca cambia nada. Los proyectos para relajar los controles de exportación en materia criptográfica están avanzando a rastras tanto a través del Congreso como del Senado. Pero nunca se llega a votar nada.

En el parlamento, más de 250 miembros del Congreso han copatrocinado H.R 850, el acta sobre seguridad y libertad mediante cifrado (SAFE) presentado por los Republicanos Bob Goodlatte (R-VA) y Zoe Lofgren (D-CA). El proyecto de ley permite para la libre exportación de dispositivos criptograficos (tanto en hardware como en software) si un producto equiparable está disponible desde una compañía extranjera, y suprime la obligatoriedad de la custodia de claves. Por otra parte, también incluye una controvertida cláusula que crea un nuevo delito federal al usar criptografía para "favorecer actuaciones criminales".

Este proyecto ha sido aprobado por cinco comisiones, lo que llevaría a pensar que ya existiría suficiente como para emitir una decisión. Sin embargo, en dos de los comités (Inteligencia y Servicios Armados) los proyectos fueron enmendados para mantener en la práctica los controles de exportación. El Comité de reglamentacion parlamentaria tendra que decidir sobre cuál de las versiones será sometida a votación.

Por parte del Senado, el presidente de la Comisión de Comercio y candidato presidencial John McCain (R-AZ), dio marcha atrás en sus posiciones iniciales oponiéndose a la criptografía e introdujo S. 798,

el acta de 1999 que promueve las transacciones en línea seguras para estimular el comercio y negocios (PROTECT). El proyecto de ley permite la libre exportación de productos de 64 bits o inferiores. El cifrado más potente puede ser exportado a comerciantes en línea, grandes empresas de administración pública, industrias controladas por el gobierno, filiales o subsidiarias de empresas americanas y gobiernos pertenecientes a la NATO, OECD y ASEAN. (Es de destacar esto último; ¿Quieren vender cifrado fuerte a Vietnam y Birmania, pero no a Brasil o Argentina?)

Un departamento de consejo de exportación criptográfica puede aconsejar relajar las restricciones. Finalmente, sobre Enero del 2002 los productos que adopten el AES o su equivalente serán libremente exportables. El proyecto de ley fue aprobado por la Comisión de Comercio en junio y espera actualmente la votación del Senado.

Mas información:

http://www.epic.org/privacy/bill_track.html

7. CRIPTOGRAFÍA DE CLAVE PÚBLICA DE CURVAS ELÍPTICAS

Por Bruce Schneier

Traducción: Angel Galindo Sánchez

En septiembre de este año, alrededor de 700 personas, utilizando 740 ordenadores, fueron capaces de romper un mensaje cifrado con criptografía de curva elíptica de 97 bits. El proceso necesitó 16.000 MIPS-años de cálculos, aproximadamente el doble de lo que necesitó otro equipo que recientemente rompió una llave de cifrado RSA de 512 bits.

Certicom, la empresa que patrocinó este concurso, ha ofrecido estos resultados como evidencia de que la criptografía de curva elíptica es más fuerte que RSA. Analicemos esta afirmación un poco más despacio.

Todos los algoritmos de llave pública, ya sean para intercambio de claves, cifrado o firma digital, están basados en uno de estos dos problemas: el problema de factorización o el problema del logaritmo discreto. (Hay otros algoritmos en círculos académicos, pero son demasiado difíciles de manejar como para ser utilizados en el mundo real). La seguridad de RSA proviene de la dificultad de factorizar números grandes. Los sistemas fuertes basados en RSA utilizan números de 1024 bits, e incluso mayores.

La seguridad de la mayoría del resto de los algoritmos de llave pública

-ElGamal, DSA, etcétera- se basa en el problema del logaritmo discreto. Los dos problemas son muy similares, y todos los algoritmos de factorización modernos pueden utilizarse para calcular logaritmos discretos en un grupo multiplicativo de dimensión finita. De forma aproximada, factorizar un número de un cierto tamaño y calcular el logaritmo discreto de otro número del mismo tamaño, supone la misma cantidad de trabajo. Esto significa que, para un tamaño de clave dado, RSA, ElGamal, DSA, etcétera, son aproximadamente igual de seguros. (Esto no es estrictamente cierto, pero es una aproximación suficientemente buena para este artículo).

Todos estos algoritmos requieren el uso de algo llamado "grupo algebraico". Cuando se inventó la criptografía de llave pública, todos los algoritmos se implementaron en el grupo algebraico más sencillo: los números de módulo n . Por ejemplo, la cifra RSA es $m^e \bmod n$, y una llave pública Diffie-Hellman es $g^y \bmod n$. De hecho, cualquier grupo algebraico valdría. Las curvas elípticas son, simplemente, otro grupo algebraico.

En criptografía de curva elíptica, las llaves públicas y privadas se definen como puntos situados sobre un objeto matemático llamado curva elíptica. (No se preocupe: en realidad no importa qué es lo que significa esto). La suma es una operación que combina dos puntos y produce un tercero. El algoritmo parece lo mismo, pero las operaciones concretas son muy distintas.

Pero si sirve cualquier grupo algebraico, ¿por qué hay gente que se preocupa por las curvas elípticas?. Parece que para algoritmos de curva elíptica de logaritmos discretos, tal vez podamos utilizar llaves más pequeñas. (Esto no es cierto para RSA, y por ello nunca verá una variante RSA de curva elíptica).

Todos los algoritmos más rápidos para calcular logaritmos discretos - el filtro del campo numérico y el filtro cuadrático - utilizan algo llamado índice de cálculo y una propiedad de los números de módulo n llamada uniformidad [smoothness], y, por tanto, para romper algoritmos de curva elíptica hay que utilizar métodos antiguos: la rho de Pollard, por ejemplo. Entonces, sólo tenemos que utilizar claves lo suficientemente largas como para que sean seguras frente a estos viejos y lentos métodos. De ahí que las claves puedan ser más cortas.

Y pueden ser significativamente más cortas. Debido a la rotura de la que hablábamos, Certicom recomienda llaves de 163 bits. Compare esto con la longitud de clave recomendada para algoritmos convencionales de logaritmo discreto, que es como mínimo de 1024 bits.

El hecho de que esta recomendación tenga sentido depende de lo rápido

que puede hacerse que los algoritmos trabajen con las curvas elípticas. La cuestión a preguntarse es: "¿Esta falta de uniformidad es una propiedad fundamental de las curvas elípticas, o simplemente es un vacío en nuestros conocimientos sobre ellas?". O, más en general: "¿Es inherentemente más difícil calcular logaritmos discretos en las curvas elípticas, o podremos tal vez encontrar algún método para hacerlo de forma tan eficiente como con los números de módulo n ?".

Si creemos en lo primero, las curvas elípticas serán siempre más seguras -para la misma longitud de clave- que los números de módulo n . Si creemos lo segundo, sólo es cuestión de tiempo que se consiga romperlas. Certicom desea fervientemente creer en lo primero. Dicen cosas como: "Las curvas elípticas como entidades algebraicas/geométricas han sido estudiadas de forma extensiva durante los últimos 150 años, y de estos estudios ha surgido una rica y profunda teoría". Concluyen que, debido a ésto, podemos confiar que los nuevos avances algorítmicos no sean demasiado devastadores.

Para mí, esto es una montón de buenos deseos. Sería maravilloso si dispusiéramos de 150 años de investigaciones en las propiedades criptográficas de las curvas elípticas. Pero no es así; por el contrario, tenemos 150 años de trabajos en las propiedades de las curvas elípticas que interesan a los matemáticos, y la mayoría de ellas sólo se relacionan accidentalmente con las que interesan a los criptógrafos. La criptografía de curva elíptica fue inventada apenas en 1985, y sólo se ha estudiado seriamente durante unos pocos años.

Incluso hoy, la mayoría del trabajo sobre curvas elípticas en un típico departamento universitario de matemáticas es bastante irrelevante para nosotros, los criptógrafos. Seguramente, algunos de sus resultados podrían ocasionalmente ayudarnos a comprender la fortaleza de los algoritmos de curva elíptica; pero éste no ha sido casi nunca el objetivo de las investigaciones en matemáticas. Esta situación está cambiando ahora, pero muy lentamente.

Más aún, la investigación en algoritmos eficientes para curvas elípticas es muy nuevo. La propia noción de algoritmo eficiente, incluso, no apareció hasta los 60 o 70, y la teoría algorítmica de números sólo se ha popularizado en las pasadas dos décadas. Antes de los ordenadores, ni siquiera era relevante.

La respuesta real a la pregunta es "No lo sabemos". No sabemos si hay formas eficientes de calcular logaritmos discretos en grupos de curvas elípticas. No sabemos si hay una definición de uniformidad que nos permita aplicar el filtro del campo numérico a las curvas elípticas. No sabemos si, a largo plazo, pueden utilizarse claves más cortas con algoritmos de curva elíptica.

A corto plazo, las recomendaciones de Certicom son razonables. Hoy, no podemos calcular logaritmos discretos en curvas elípticas de forma tan eficiente como con los números de módulo n . Los sistemas pueden utilizar claves más cortas con curvas elípticas. Pero, a largo plazo, no lo sabemos.

Hay otras diferencias que también es necesario considerar. Comprobar firmas de curva elíptica es aún complicado, comparado con la comprobación de las firmas RSA. Y todos los usuarios de un sistema de curva elíptica tienen que utilizar la misma curva. (Si no se hace esto, se pierden la mayor parte de los beneficios de tamaño en las claves de curva elíptica). Esto tiene implicaciones en seguridad: es más fácil romper la clave de un usuario al azar de un sistema, que romper la clave de un usuario concreto. Me gustaría ver más análisis sobre este aspecto para los sistemas de curva elíptica.

Mi recomendación es que si está usted trabajando en un sistema con limitaciones en el tamaño de clave -tarjetas inteligentes, algunos teléfonos móviles o receptores de mensajes, etcétera-, considere la utilización de curvas elípticas. Si no tiene restricciones de tamaño, utilice RSA. Si necesita seguridad durante varias décadas (casi ningún sistema lo necesita), utilice RSA.

Tenga en cuenta que algún día -el próximo año, en diez años, en un siglo-, alguien puede encontrar una forma de definir la uniformidad, o algo incluso más útil, en curvas elípticas. Si ocurre esto, tendrá que utilizar la misma longitud de clave que se emplearía en algoritmos de logaritmo discreto, y entonces ya no habrá ninguna razón para utilizar algoritmos de curva elíptica.

Postdata: Este análisis puede aplicarse también a la factorización. A la gente de RSA Security, Inc. les gusta hablar de la larga historia del problema de factorización, y de cómo nos da confianza sobre la seguridad de RSA. Sí, se ha estudiado durante siglos, pero sólo recientemente esos estudios han estado remotamente relacionados con la criptografía. Más aún, el trabajo en factorización no ha sido considerada como un área respetable de estudio hasta recientemente; antes de eso, se consideraba como un hobby excéntrico. Y los algoritmos eficientes para la factorización sólo se han estudiado en el último par de décadas. Realmente no tenemos ni idea de la fortaleza de la factorización fuerte.

La verdad es que las empresas tienen tendencia a exagerar sus productos. Antes de tomar una decisión sobre algoritmos criptográficos, los clientes deberían recabar una amplia variedad de opiniones independientes (de personas no involucradas económicamente en los resultados de la decisión que se tome) sobre qué están comprando.

Noticias sobre los recientes esfuerzos de craqueo de las curvas elípticas:

<http://www.computerworld.com/home/news.nsf/all/9909282ellip>

<http://www.certicom.com/press/99/sept2899.htm>

Una excelente introducción matemática a las curvas elípticas:

<http://www.certicom.com/ecc/enter/index.htm>

Una excelente comparación sobre las longitudes de claves, incluyendo RSA y curvas elípticas:

<http://www.cryptosavvy.com>

Del Boletín #147:

CRIPTOSISTEMAS INEXPUGNABLES (I)

Por Manuel Lucena López

<lucena@kriptopolis.com>

"Wolff se dirigió al armario donde había metido la radio. Sacó la novela inglesa y la hoja de papel que tenía la clave del código. Se puso a estudiar la clave. Aquel día era 28 de mayo. Tenía que añadir 42 -el año- a 28 para llegar a la página de la novela que debería utilizar para cifrar el mensaje. Mayo era el quinto mes del año, así que se descontaría cada quinta letra de la página.

Decidió enviar el siguiente mensaje: 'He llegado. Confirmación'. Comenzando en la parte superior de la página 70 del libro, recorrió la primera línea buscando la letra h. Era el décimo carácter tipográfico, descontando cada quinta letra. En su código, por lo tanto, sería representado por la undécima letra del alfabeto, la "j". Ahora necesitaba una "e".(...) Para descifrar el mensaje, un oyente había de poseer tanto el libro como la clave, siendo así completamente impenetrable tanto en teoría como en la práctica"

(Ken Follet, "La clave está en Rebeca")

Todo criptólogo ha soñado alguna vez con encontrar un sistema de

codificación seguro al cien por cien aún en las condiciones más adversas, invulnerable incluso frente enemigos con cantidades ingentes de recursos computacionales. Lo cierto es que dichos sistemas existen, como cualquier buen aficionado a la Criptografía sabe, y ya fueron caracterizados por Claude Shannon hace medio siglo. Pero si existen, ¿por qué -al parecer- nadie los emplea?

Veamos en primer lugar la caracterización que hace Shannon de un sistema criptográfico seguro (no se asusten, no vamos a emplear fórmulas):

"Un criptosistema es seguro si la cantidad de información que nos aporta el hecho de conocer el mensaje cifrado sobre la entropía del texto en claro vale cero".

Estos criptosistemas, como veremos a continuación, son imposibles de romper incluso para un adversario con una capacidad de computación infinita.

Algunos de ustedes se preguntarán: ¿Qué es eso de la "entropía"? Digamos que en nuestro caso es una medida de incertidumbre. Cuanta más entropía posea un suceso, mayor incertidumbre tendremos a la hora de intentar predecirlo. Un suceso con mucha entropía es, por ejemplo, el Sorteo de Lotería de Navidad, y otro con poca entropía podría ser enchufar la tele: tenemos una gran certeza acerca de lo que nos vamos a encontrar...

Mi profesor de Algebra en la Universidad solía decir: "vamos a ver un ejemplo, que con ejemplos es como se ven las cosas". Supongamos que Pepe envía un mensaje a Luisa. Nosotros sabemos de antemano que Pepe sólo puede enviar uno de estos tres mensajes: "Jorgito" (el 50% de los casos), "Juanito" (el 30%) y "Jaimito" (el 20%). Estas probabilidades son conocidas antes de interceptar el texto cifrado, y darán lugar a una entropía. También sabemos que el mecanismo que emplea Pepe para cifrar su mensaje es sumar el mismo número a todas las letras del texto en claro, por lo que tenemos 26 posibles claves de cifrado.

Pues bien, resulta que por el canal circula el mensaje "MXDQLWR". Como conocemos el método de cifrado, vamos a poner en una tabla todos los posibles textos en claro que pueden dar lugar a este mensaje (vamos a realizar un ataque por la fuerza bruta):

MNOPQRSTUVWXYZABCDEFGHIJKL
 XYZABCDEFGHIJKLMNOPQRSTUVW
 DEFABCDEFGHIJKLMNOPQRSTUVWXYZABC
 QRSTUVWXYZABCDEFGHIJKLMNOP

LMNOPQRSTUVWXYZ ABCDEFGHIJK
 WXYZABCDEFGHIJKLMNOPQRSTU
 RSTUVWXYZABCDEFGHIJKLMNO P
 ^ *

Texto cifrado

Obsérvese que en la columna marcada con * aparece la palabra "Juanito" (utilice una fuente tipo Courier o similar para ver correctamente la tabla). Podemos decir que Pepe ha sumado 3 a cada letra del mensaje original y que éste era "Juanito", con probabilidad 100%. Ninguno de los otros dos mensajes podía dar lugar al criptograma "MXDQLWR".

Como podemos ver, nuestra incertidumbre ha descendido -en este caso ha desaparecido por completo- al conocer el texto cifrado, por lo que este sistema no es seguro según el criterio de Shannon.

Pero, ¿qué pasaría si la clave que emplea Pepe consistiera en sumar un número diferente a cada letra del mensaje? Pues que el criptograma "MXDQLWR" se obtiene a partir de "JUANITO" con la clave (3,3,3,3,3,3,3), a partir de "JORGITO" con (3,9,12,10,3,3,3), y a partir de "JAIMITO" con (3,23,21,4,3,3,3). A priori, ninguna de las tres claves indicadas es más probable que otra, por lo que, desgraciadamente, el hecho de conocer el criptograma en este caso, no nos dice nada sobre el texto original que ya no sepamos, es decir, la entropía no disminuye. Este será un criptosistema seguro de Shannon, invulnerable incluso frente a ataques por la fuerza bruta, siempre que Pepe no repita la misma clave para otro mensaje.

Y ahora, le reto a adivinar cuáles son los mensajes originales si Pepe repite clave y usted escucha por el canal el criptograma "MDMXLWR".

Para que un sistema cumpla la condición de Shannon necesitamos que, sea cual sea el criptograma, exista una clave que lo decodifique en cada uno de los textos en claro posibles; de esta forma, un adversario no sabrá con qué clave quedarse. Necesitaremos pues que haya al menos tantas claves como posibles textos en claro. Y este es el principal problema de los sistemas de Shannon: la clave debe ser igual o más larga que el mensaje original. Como se puede observar, esto es exactamente lo que ocurre con el espía de la cita que he incluido al principio: la clave era todo un libro, y además no se repetía nunca, al usar cada día una página diferente para cifrar.

En resumen: si utilizamos una clave de un único uso, tenemos garantizado que nuestro sistema es virtualmente inexpugnable, siempre y cuando podamos mantener la clave a salvo... ¿qué nos impediría,

pues, emplear un CD-ROM lleno de información como clave?

[Continuará...]

Manuel Lucena

Del Boletín #148:

CRIPTOSISTEMAS INEXPUGNABLES (y II)

Por Manuel Lucena López
<lucena@kriptopolis.com>

[NOTA: La primera parte del artículo puede consultarse en:
<http://www.kriptopolis.com/boletin/0147.txt>]

Planteábamos en la anterior entrega la posibilidad de emplear un CD como clave de uso único para poder construir un criptosistema de carácter invencible. Pero, ¿hasta qué punto podemos considerar útil y práctico un sistema de estas características? ¿Merece realmente la pena o es mejor "conformarse" con sistemas razonablemente seguros? ¿Sería en la práctica un sistema de estas características tan inexpugnable como teóricamente parece?

Pues bien, manos a la obra. Vamos a intentar diseñar nuestro sistema. Para ello supondremos que nuestros mensajes son cadenas de bytes, y que empleamos el contenido de un CD-ROM como clave, haciendo un Or-Exclusivo (XOR) entre los bytes del mensaje y la clave. Se supone que existe un mecanismo para que los dos interlocutores sepan qué trozo del CD deben emplear para codificar/decodificar cada mensaje, y que el mismo trozo no se emplea dos veces. ¿Preparados...?

*** PROTOTIPO 1:**

Grabar dos CD-ROM's idénticos con ruido blanco (números aleatorios puros) y emplearlos como clave de un sólo uso.

Problemas: a) Hacer llegar una copia de forma segura a cada interlocutor. b) El CD cantaría (y no precisamente ópera) si alguien lo intercepta.

Este prototipo puede ser válido si queremos almacenar información *muy* sensible, de forma que metamos los datos codificados en una caja fuerte y la clave en otra (ya que si perdemos alguna de las dos cosas estaremos perdidos, valga la redundancia). Pero eso parece más bien el argumento de una película de 007 y, como no queremos pisarle el terreno al agente de Su Majestad, pasaremos al:

*** PROTOTIPO 2:**

Emplear como clave un CD comercial cualquiera, acordado por los dos interlocutores.

Este algoritmo presenta la ventaja de que no es necesario hacer llegar desde un interlocutor a otro el CD, ya que ambos lo pueden comprar, y además su pérdida ya no constituirá un hecho irreparable. Eventualmente, también evita que, en caso de robo, el ladrón identifique cuál de nuestros CD's contiene las claves.

En este caso, conviene tener en cuenta que el tipo de contenido del CD no es aleatorio, por lo que un atacante podría modelizar la distribución de probabilidad de la clave. Esto daría lugar a un posible (y fatal) ataque. Ejemplo al canto: Supongamos que el enemigo sabe que nuestro CD contiene fundamentalmente código ejecutable, y que nosotros transmitimos texto ASCII en castellano. Bastaría con coger cada mensaje y analizarlo por la fuerza bruta, pero empleando sólo como claves de prueba secuencias que tengan sentido como código ejecutable, hasta que nos salga como resultado algo en castellano. (La verdad es que habría que probar cantidades astronómicas de combinaciones, y el sistema sería relativamente seguro, pero ya no sería totalmente inexpugnable, y además dependerá de la "calidad" de la información almacenada en el CD). Intento fallido.

*** PROTOTIPO 3:**

Emplear un CD comercial cualquiera, pero "aleatorizando" su contenido. Esto se puede conseguir aplicando un algoritmo que "aplane" la distribución de probabilidad de los datos en el CD (o sea, que elimine la redundancia que éstos poseen). Lógicamente, para conseguir un trozo de clave "buena", hará falta un fragmento más grande de información del CD. (Como ahora no es el momento, les emplazo para un próximo artículo en el que profundizaremos sobre esto de la redundancia y la "aleatorización")

NUESTRO GOZO EN UN POZO...

Desgraciadamente, todos los métodos que acabamos de proponer adolecen del mismo defecto: si el enemigo accede al CD empleado como clave, conoce el algoritmo de codificación, y en algún momento descubre un par texto en claro-texto cifrado, puede descubrir una subcadena del contenido del CD. Bastará entonces con buscar todas las ocurrencias de dicha subcadena en el CD (o en su versión "aleatorizada"), y tratar de decodificar el resto de los mensajes a partir de las posiciones obtenidas.

LA PERSISTENCIA DE LA MEMORIA...

Muchos criptosistemas empleados en la actualidad (RC4, SEAL, ...) basan su funcionamiento en la generación de secuencias pseudo-aleatorias empleando una semilla (que hace las veces de clave). Las secuencias generadas son tan largas que tenemos una gran certeza de que no se van a producir ciclos. Si el sistema está bien diseñado, y no codificamos dos mensajes con la misma clave, habrá que buscar la semilla por la fuerza bruta para poder romperlos, y siempre tendremos una ventaja: la clave puede almacenarse en nuestro cerebro (en forma de contraseña), y mientras siga siendo imposible leer la memoria de un ser humano (vale, vale, sin contar el "suero de la verdad"), el nivel de seguridad seguirá siendo mucho mayor que el de los tres prototipos que acabamos de diseñar. Una vez más, el Hombre vuelve a vencer a la Máquina.

MORALEJA:

Las cosas están bien como están, lo que parecía "perfecto" acabó presentando más inconvenientes que ventajas. Sigán usando su "imperfecto" PGP de toda la vida. En cualquier caso... ¿a que todos hemos aprendido algo?

AGRADECIMIENTOS:

- * A Javier Candeira: Por proponer la idea.
- * A Pepe Cervera: Por hacer las aclaraciones oportunas.
- * A Carlos Sánchez Almeida: Por animar el cotarro y por su energía positiva.
- * A Javier Maestre: El quinto hombre...

* A la Fabada Asturiana: Por inspirarnos estas "veleidades".

SOLUCIÓN AL RETO DE LA SEMANA PASADA:

Como todos ustedes recordarán, el reto que planteamos en la anterior entrega consistía en averiguar cuál era la clave y los mensajes si Pepe repetía clave y se capturaba el criptograma "MDMXLWR".

Sabemos que Pepe sólo emite cada vez uno de estos tres mensajes: "Jorgito", "Juanito" y "Jaimito". Si en la primera ocasión habíamos interceptado "MXDQLWR", podemos deducir que Pepe ha empleado una de estas tres posibles claves:

(3, 3, 3, 3,3,3,3) si ha enviado "JUANITO"
(3, 9,12,10,3,3,3) si ha enviado "JORGITO"
(3,23,21, 4,3,3,3) si ha enviado "JAIMITO"

Puesto que ahora hemos recibido "MDMXLWR", emplearemos las tres posibles claves para ver qué obtenemos:

Descifrando con (3, 3, 3, 3,3,3,3) obtenemos "JAJUITO"
Descifrando con (3, 9,12,10,3,3,3) obtenemos "JUANITO"
Descifrando con (3,23,21, 4,3,3,3) obtenemos "JGRTITO"

El único texto en claro que corresponde a un mensaje posible de Pepe es "JUANITO", por lo que la solución es que Pepe envía los mensajes "JORGITO" y "JUANITO", y la clave en ambos es (3,9,12,10,3,3,3)

Obsérvese que, como ya se dijo, si Pepe no hubiera empleado la misma clave para dos mensajes, el criptosistema hubiera permanecido invencible.

Del Boletin # 151:

GriYo: CREADOR DE VIDA ARTIFICIAL

NICKNAME: GriYo

Bios release date: XX.XX.1972

Manufactured: Spain

Group: 29A

Contact adress: griyo@bi0.net

End of file.

Quizás la vida es más sencilla de lo que nos quieren hacer creer. La prueba son los Virus: no estoy seguro de que la Madre Naturaleza sepa a estas alturas quién es peor, si ellos o nosotros. Para aclararme un poco, riego mi depresión en compañía de GriYo y Maia, una pareja encantadora. Nos acompañan FrAKaSo, NetSavage, y algún amigo más. Kabul, Plaza Real, Barcelona, diciembre de 1999.

P: ¿Qué se siente al crear vida artificial?

"Los virus informáticos, como los biológicos, resultan en ocasiones muy simples y muy complejos al mismo tiempo. Como programador siento cierto cosquilleo tan sólo por atreverme a emular semejante mecanismo, aunque se trate del modelo más básico de vida.

Tengo que confesar que cuando encuentro alguno de mis virus en la *wild-list* me siento bien. No me dedico a distribuir los virus que escribo (suficientemente ocupado estoy ya escribiéndolos). Pero tener noticias de que el virus está propagándose, funcionando tal y como lo haría un virus real en el mundo real, es tener buenas noticias: por un lado significa que la estrategia utilizada en su diseño ha funcionado (nada como un buen plan, eh!?). Por otro lado, significa que la implementación ha sido precisa, perfecta; de lo contrario hubieran aparecido problemas, incompatibilidades, que seguramente hubieran impedido la propagación del virus.

Ya sé que detrás de esos números hay usuarios, empresas... pero también hay usuarios, y sobre todo empresas, tras los números en la cuenta de ganancias de las empresas que producen software antivirus.

Me siento bien con lo que hago, por eso lo hago. Me hace sentir *realizado* (haha!)...

Me permite retar a otros programadores, retar a los programas

antivirus, retar a los usuarios del mundo..., todo ello sin moverme de casa y haciendo lo que más me gusta: programar. Algunos dicen: cobardía, retar a nadie sabiéndose protegido detrás del monitor. Yo creo que los tiros no van por ahí. No son necesariamente cobardes los jugadores de ajedrez por retar a otros jugadores en un terreno imaginario en el que lo importante no es la fuerza, no es el dinero, ni el poder... Como en diseñar un virus: La estrategia prima, y la metodología. Hacerlo bien resulta reconfortante por eso, no por sentirse un Dios creador de vida o algo así ;-) puesto que realmente se trata de torpes imitaciones."

P: La evolución de la industria informática ha creado un monocultivo de Windows ¿eso hace más fácil la proliferación de virus?

"Está claro. Siempre me sorprenden estas cosas, pese a que siempre son así: lo que es cierto para el mundo real lo es para el mundo virtual.

Me refiero a la multiplicidad. Si todos los seres humanos fuésemos iguales podríamos morir todos en una sola epidemia. Ser diferentes nos protege... El *monocultivo Windows* como tú le llamas (hahaha ;) es sin duda algo muy favorable de cara a la proliferación de virus.

Si cada individuo es *de su padre y de su madre* (como sucede con el Linux) esto no sucede, o es más complicado."

P: Nadie se atreve con los fabricantes de armas, pero todos quieren encerrar a los creadores de virus ¿qué piensas al respecto?

"Me cuesta creer que las cosas sean como son. ¿Por qué un gobierno puede hacer caer una bomba atómica en el planeta en que todos nosotros vivimos sólo para ver si funciona, y luego se permite encarcelar a un chaval por escribir un virus? No sé la respuesta, pero me parece una gran sátira."

P: ¿Consideras los virus como algo positivo para la evolución, en su doble sentido, natural e informático?

"Sin duda la naturaleza, la evolución, los ecosistemas, todo, pertenece a un orden global que se rige por unas reglas. Determinados virus biológicos parecen *sistemas de protección* que la naturaleza emplea frente al invasor humano. Me cuesta creer que el ser humano pueda ser superior en inteligencia o capacidad a aquello que lo ha creado. Tiene que existir, por necesidad, un mecanismo capaz de proteger al sistema de la amenaza que representa una especie mucho más evolucionada que el resto.

La parte *cibernética* de esto me parece menos clara: ¿quizás un virus que acabe para siempre con el *monocultivo Windows* que mencionabas antes? ;-) Como mínimo estamos hablando de la evolución de una técnica, lo cual implica muchas cosas en términos de eficacia, eficiencia... Y, por añadido, supondrá una evolución en los sistemas de protección."

P: ¿Y más allá de la evolución, se vendría un virus a la revolución?

"Esto sí que esta clarísimo. Los virus informáticos pueden ser utilizados como armas, con fines militares o revolucionarios como tú dices. Se trata de acceder a sistemas informáticos en los que no estamos autorizados, algo evidentemente útil en este tipo de temas."

P: Entonces, ¿te parece que el ejército puede estar interesado en temas como los virus informáticos?

"Exactamente. Para los ejércitos, y en consecuencia para los gobiernos, pueden tener importancia. No olvidemos que en una guerra juegan un papel muy importante temas que no están directamente relacionados con pegarse tiros, temas como la encriptación, el espionaje, la publicidad..."

Podría pasarme horas escribiendo ideas delirantes sobre las posibles aplicaciones militares de los virus informáticos. Rápidamente diría algunas: neutralizar sistemas, robo de contraseñas de cifrado, adulterar información..."

Tampoco olvidemos *Independence Day*, donde un virus informático acaba con los marcianos que nos amenazan... ;-)"

P: No eres del todo apolítico, creaste un virus anti-ETA. Seguro que tenías razones para ello. ¿Nos las quieres contar?

"Sí; no hay problema. Cuando comencé a escribir este virus no pensé en ningún nombre. Tan sólo estaba volcado en acabar un virus que explotaba varios aspectos nunca antes utilizados. Estaba tan emocionado con el tema que tardé sólo dos semanas en acabar el virus."

Por aquel entonces sucedió lo de Miguel Angel Blanco, ya sabes... El nuevo número de 29A estaba a punto de salir y en el grupo pensamos que nos gustaría contar algo sobre el tema de ETA.

Dado que el virus no tenía nombre, podríamos contar lo que había sucedido como introducción al artículo que se publicó sobre él. Anti-ETA era entonces el nombre que mejor nos pareció para el virus. No hubiese tenido mucho sentido hablar de ETA en el artículo y luego

mostrar el código fuente de un virus que nada tiene que ver con el tema.

Para relacionar aun más el código del virus con el tema que tratábamos en el artículo, le añadí un efecto gráfico en la activación, que mostraba una mano blanca sobre la que se leía un rótulo: Anti-ETA.

Con ello mostrábamos nuestra indignación por lo que sucedió, nada político realmente, sólo en referencia a un asesinato tan sucio."

P: 31 de diciembre, 24 horas. Tus mejores deseos para el año 2000.

"Me tiene un poco triste esto del 2000... De pequeño siempre había leído e incluso visto en la tele, que en el 2000 estaríamos viviendo en la Luna, o en Marte, cosas así... Como no se den prisa no me dará tiempo a viajar por el espacio!!!! (heh)

Pero bueno, os deseo un feliz año a todos, espero que no tengáis problemas con los virus en vuestros ordenadores...

Felicidad y bueno, ya sabes, lo de siempre: love, peace and fuck the police ;-)) Un abrazo a todos."

- - - - -

Me pierdo en las Ramblas, en dirección al centro de la ciudad. Miro la luna, y deseo que algún día sea la patria de los hijos de Maia y GriYo. Y pienso en los ojos oscuros de FrAKaSo: los de EXiT0 son azules, pero lleva lentillas. La vida es complicada, no sé por qué quieren imitarla las máquinas. Ser humano significa ser vulnerable.

Feliz Navidad y allá Vds. con el 2000.

Carlos Sánchez Almeida
almeida@kriptopolis.com
<http://www.bufetalmeida.com>

Del Criptograma #20:

1. LA SEGURIDAD NO ES UN PRODUCTO; ES UN PROCESO

Por Bruce Schneier

Traducción: José Manuel Gómez

En Abril de 1999 alguien descubrió una vulnerabilidad en MDAC (Microsoft Data Access Components) que pudiera permitir a un atacante tomar el control de un sistema remoto con Windows NT. Esta vulnerabilidad fue publicada inicialmente en una lista de correo pública. Aunque el moderador de la lista mantuvo los detalles en secreto durante más de una semana, algún despierto hacker fue capaz de utilizar los detalles disponibles para divisar una forma de explotar el defecto.

A continuación, un 'script' escrito en PERL fue difundido públicamente en Internet. Casi al mismo tiempo, Microsoft creaba un parche y un método para evitar que posibles atacantes pudieran aprovechar esta vulnerabilidad en los sistemas de los usuarios. Microsoft publicó también un boletín de seguridad sobre el tema, tal y como hicieron varios otros sitios especializados en noticias de seguridad. Pero los parches no resuelven mágicamente los defectos de seguridad. El fin de semana en torno a la festividad de Halloween, unos intrusos atacaron e hicieron claudicar más de 25 sitios web basados en NT.

Este tipo de cosas suceden constantemente. Otro ejemplo: Microsoft publicó un boletín y un parche para una vulnerabilidad en el acceso a datos de IIS (Internet Information Server) el año pasado. Recientemente, varios expertos han demostrado que Compaq, Dell, Compuserve, PSINet y NASDAQ-AMEX nunca se molestaron en instalar el parche y seguían siendo vulnerables.

Se informa de una vulnerabilidad y se publica un parche. Si se confía en las noticias, ahí se acaba la historia. Pero en la mayoría de los casos los parches nunca se instalan. Por esta razón la mayoría de los sistemas en Internet son vulnerables a ataques ya conocidos para los que existen parches.

La seguridad no es un producto; es un proceso. Es el proceso de permanecer atento a las actualizaciones que el fabricante realiza en sus productos. No sólo en programas para redes o productos de seguridad para redes (navegadores, cortafuegos, sistemas operativos de red, programas de servidores web...), sino en todo programa que se ejecute. Las vulnerabilidades de su procesador de textos pueden comprometer la seguridad de su red.

Es el proceso de vigilar sus sistemas cuidadosamente en busca de señales de ataque. Su cortafuegos elabora informes de actividad. Otro tanto

hacen sus servidores UNIX y NT. Y sus enrutadores y servidores de red. Aconstúmbrese a revisarlos todos los días. Aprenda cómo se refleja un ataque y cómo reconocerle.

Ningún producto de seguridad funciona como una varita mágica; todos necesitan tiempo y experiencia para que funcionen de forma apropiada. Tiene que estar pendiente de ellos cada día.

El bug de Microsoft mencionado arriba:

<http://www.microsoft.com/security/bulletins/ms99-025.asp>

<http://www.microsoft.com/security/bulletins/ms99-025faq.asp>

En las noticias:

<http://www.fcw.com/pubs/fcw/1999/1101/fcw-newsfedwire-11-01-99.html>

Por qué las vulnerabilidades no se reparan:

<http://www.computerworld.com/home/print.nsf/all/991122CD52>

2. El algoritmo de clave pública de Sarah Flannery

Por Bruce Schneier

Traducción: José Manuel Gómez

En Enero de 1999, una chica irlandesa de 16 años llamada Sarah Flannery apareció en los noticiarios internacionales anunciando un nuevo algoritmo de clave pública denominado Cayley-Purser, supuestamente mejor y más rápido que RSA y ElGamal.

El único problema era que nadie conocía el algoritmo.

Bueno; pues por fin es público.

El documento de Flannery que describe el algoritmo Cayley-Purser ha sido publicado en Internet por una fuente desconocida. Es un trabajo interesante, pero no es seguro. La propia Flannery publica una forma de romper el algoritmo en un apéndice.

Para mí, esto convierte a Flannery en mucho más impresionante como joven criptóloga. Como ya he dicho muchas veces, cualquiera puede inventar un nuevo criptosistema. Muy poca gente es lo bastante inteligente y capaz como para romperlo. Al romper su propio sistema, Flannery resulta aún más prometedora como criptóloga. Espero con interés sus próximos trabajos.

El documento de Flannery:

<http://cryptome.org/flannery-cp.htm>

Las noticias de Enero:

<http://www.zdnet.com/zdnn/stories/news/0,4586,2189301,00.html?chkpt=zdnmsmsa>

<http://www.wired.com/news/technology/0,1282,17330,00.html>

Del Boletín #155:

LA LOTERÍA CHINA

Por Manuel Lucena López

[<mlucena@kriptopolis.com](mailto:mlucena@kriptopolis.com)

En 1991, J.J Quisquater y J.P Delescaille propusieron la posibilidad de construir un sistema masivamente paralelo para criptoanalizar mensajes por la fuerza bruta. La idea era ingeniosa: incluir de forma secreta un "chip" capaz de probar un millón de claves por segundo en cada receptor de radio y televisión. En la señal que emitieran los repetidores se camuflaría un texto cifrado reconocible por el chip en cuestión, que comenzaría a analizarlo de forma automática. Cada vez que un gobierno (los autores proponen el de China, por ser el país más poblado de la Tierra) quisiera romper un mensaje, sólo tendría que transmitirlo durante algún programa de televisión de máxima audiencia. La forma de recuperar la respuesta sería tan simple como ofrecer un premio en metálico al afortunado (e inadvertido) descubridor de la clave.

El sistema es tan efectivo que un mensaje de 64 bits podría ser roto en 7 horas (¡y uno de 56 en menos de dos minutos!) si se incluyera un chip de estas características en cada receptor de radio o televisión de EE.UU. (suponiendo un total de aproximadamente 700.000 receptores). Aparte de las evidentes consideraciones prácticas (que todos los habitantes del país sintonicen el canal apropiado en el momento preciso, que los televisores permanezcan encendidos durante el número suficiente de horas, etc.) convendrán conmigo en que este ataque podría parecer a primera vista *muy* interesante.

En la actualidad, de todos es sabido que en el cuarto de juegos de un niño hay mayor capacidad de proceso que en la NASA hace veinte años. En efecto, uno sólo de los chips que incorpora la famosa máquina DES-Cracker, capaz de descifrar un mensaje DES de 56 bits en pocos días, puede probar la friolera de 60 millones de claves por segundo. Teniendo en cuenta que se trata de un chip genérico

programado expresamente y que corre a 40 MHz, no cabe duda de que un circuito integrado diseñado expresamente y corriendo a una muy superior velocidad de reloj podría probar una cantidad de claves por segundo considerablemente mayor. Evidentemente, el diseño de tal chip sería muy caro, pero una vez finalizado su producción en masa sería ridículamente barata.

Otro de los temas que más desarrollo ha conocido en este final de siglo es la televisión digital, que está abriendo la brecha a la conexión masiva de los hogares a canales de transferencia de información de considerable capacidad, y además de manera bidireccional. La consecuencia es que estamos asistiendo al nacimiento de la denominada "Era Digital", llena de nuevas posibilidades, algunas de ellas ciertamente inquietantes.

Con todos estos ingredientes en nuestra coctelera sólo nos queda agitar un poco y el resultado puede ser asombroso. Supongan un chip decodificador que pruebe cien millones de claves por segundo (nada del otro mundo), y cien mil aparatos decodificadores de televisión digital funcionando a la vez: se podrían probar diez billones de claves por segundo. Eso supondría romper un mensaje de 40 bits (los que usan los navegadores de internet de fuera de EE.UU) en una décima de segundo, uno de 56 bits (DES) en dos horas, y uno de 64 en aproximadamente veinte días. Nótese que las cifras que propongo (cien millones de claves por chip y segundo y cien mil chips) no son en absoluto exageradas, más bien se diría que todo lo contrario.

Que nadie se alarme, no estoy insinuando que un sistema de estas características ya exista. Lo único que pretendo es demostrar que es técnicamente factible, y representa una clara prueba de que las claves que las leyes norteamericanas nos dejan usar son manifiestamente incapaces de ofrecernos un sistema de protección realmente fiable.

Personalmente, no creo que nada en mi información personal valga tanto dinero como para que merezca la pena someterlo a un ataque de estas características (aunque, una vez montada la red, el coste para descifrar un mensaje sería casi nulo), pero ese no es el problema. El problema es que la información privada no tiene precio, por lo que debe ser protegida por métodos que garanticen su inviolabilidad. Mi consejo es que usen 128 bits siempre que puedan, que les van a durar muchos años.

Vulnerabilidades de
P.G.P.
(Pretty Good Privacy)

Vulnerabilidades de PGP:

Según todo lo dicho hasta ahora, parece claro que PGP proporciona un nivel de seguridad que nada tiene que envidiar a cualquier otro sistema criptográfico jamás desarrollado. ¿Qué sentido tiene, pues, hablar de sus vulnerabilidades, si estas parecen no existir?

Como cualquier herramienta, PGP proporciona un gran rendimiento si se emplea correctamente, pero su uso inadecuado podría convertirlo en una protección totalmente inútil. Es por ello que parece interesante llevar a cabo una pequeña recapitulación acerca de las buenas costumbres que harán de PGP nuestro mejor aliado.

- Escoger contraseñas adecuadas que cumplan algunas condiciones para que sean consideradas seguras:
- Deben ser memorizadas. Una contraseña jamás debe ser escrita, por razones obvias.
- Suficientemente complejas. Una buena contraseña debe constar de al menos ocho letras. Pensemos que si empleamos únicamente seis caracteres alfanuméricos, tenemos únicamente unos dos mil millones de posibilidades. Teniendo en cuenta que hay programas para PC capaces de probar más de cuarenta mil claves en un segundo, una clave de estas características podría ser descubierta en menos de 15 horas.
- Fáciles de recordar. Puesto que una palabra clave ha de ser memorizada, no tiene sentido emplear contraseñas difíciles de recordar. En este sentido podemos seguir reglas como que la palabra se pueda pronunciar en voz alta, o que responda a algún acrónimo más o menos complejo. En este punto no debemos olvidar que hay que evitar a toda costa palabras que signifiquen algo.
- Deben ser modificadas con frecuencia. Hemos de partir de la premisa de que toda palabra clave caerá tarde o temprano, por lo que será muy recomendable que nuestras contraseñas sean cambiadas periódicamente. La frecuencia con la que se produzca el cambio dependerá de la complejidad de las claves y del nivel de seguridad que se desee alcanzar. Y lo más importante: ante cualquier sospecha, cambiar todas las claves.
- Proteger adecuadamente los archivos sensibles. Estos archivos serán, lógicamente, nuestros llaveros (anillos de claves) y el fichero que alberga la semilla aleatoria. Esta protección debe llevarse a cabo tanto frente al acceso de posibles curiosos, como frente a una posible pérdida de los datos (¡recuerde que si pierde el archivo con su clave privada no podrá descifrar jamás ningún mensaje!).

- Emitir revocaciones de nuestras claves al generarlas y guardarlas en lugar seguro. Serán el único mecanismo válido para revocar una clave en caso de pérdida del anillo. Afortunadamente, la versión 6 de PGP permite nombrar revocadores para nuestras claves, de forma que estos podrán invalidarla en cualquier momento sin necesidad de nuestra clave privada.
- Firmar solo las claves de cuya autenticidad estemos seguros. Es la única manera de que las redes de confianza puedan funcionar, ya que si todos firmáramos las claves alegremente, podríamos estar certificando claves falsas.
- Quizás el peor problema con el que nos podríamos encontrar al utilizar PGP es al de la autenticidad de las Claves Públicas. Esto es, por ejemplo, que alguien (llamémosle embaucador) podría hacerse pasar por otra persona (llamémosla X) y generar sus claves pública y privada. Así cuando nosotros le enviáramos algo a la persona X y el embaucador nos interceptara el mensaje, este último podría descifrarlo sin mayor problema. Para evitar este problema es que existen las Autoridades Certificantes y el hecho de poder firmar una clave pública de la cual uno tenga plena seguridad de su origen.
- FACTORIZADA CLAVE PÚBLICA DE 512 BIT

El pasado 22 de agosto, un equipo de investigadores de seis países logró hallar los factores primos de un número de 512 bits, es decir, una clave pública RSA, idéntica a las que aún protegen un gran número de transacciones electrónicas vía SSL.

Bibliografía

Bibliografía:

- NETWORK SECURITY: PRIVATE COMMUNICATION IN A PUBLIC WORLD – **Charlie Kaufman, Radia Perlman y Mike Speciner** – PTR Prentice Hall.
- NETWORK AND INTERNETWORK SECURITY – PRINCIPLES AND PRACTICE **William Stallings** – Prentice Hall – IEEE Press.
- TECNICAS CRIPTOGRAFICAS DE PROTECCION – **Fuster**.
- PUBLIC-KEY CRYPTOGRAPHY – **Salomaa**.
- APPLIED CRYPTOGRAPHY – **B. Schneier**.
- THE OFFICIAL PGP USER'S GUIDE – **Zimmermann Phillip**.
- SECURE ELECTRONIC COMMERCE: BUILDING THE INFRASTRUCTURE FOR DIGITAL SIGNATURE AND ENCRYPTION, **W. Ford, M. S. Baum**, Prentice-Hall, Englewood Cliffs, NJ 1997
- DIGITAL CASH – **Peter Wayner**.
- Distintas paginas Web que nos garantizan credulidad, como por ejemplo:
 - <http://www.jus.gov.ar/firma/> **SubComité de Criptografia y Firma Digital**
 - <http://www.pgpi.com/> **The International PGP Home Page**
 - <http://www.rsa.com/> **RSA Data Security, Inc.**
 - <http://www.cryptography.com/> **Cryptography.Com**
 - <http://www.verisign.com/> **Digital Certificates – Verisign**
 - <http://www.cacr.math.uwaterloo.ca/~dstinson/ssbib.html> **Secret Sharing**
 - <http://www.entrust.com/> **Digital Certificates – EnTrust**
 - <http://www.kriptopolis.com/> **Kriptopolis**
- HACKERS – LOS CLANES DE LA RED – **Claudio Hernandez** © 1999
- CRIPTOGRAFIA Y SEGURIDAD DE COMPUTADORES – **Manuel J. Lucerna López** © 1999

- DIGITAL CERTIFICATES APPLIED INTERNET SECURITY, **J. Fegghi, J. Fegghi, P. Williams**, Addison Wesley, 1999
- CRIPTOGRAFIA – **José de Jesús Angel Angel** © 1999
- USING SET FOR SECURE ELECTRONIC COMMERCE, **G. N. Drew**, Prentice Hall, NJ 1999
- A METHOD FOR OBTAINING DIGITAL SIGNATURE AND PUBLIC-KEY CRYPTOSYSTEMS, Communication of the ACM - **R.L. Rivest, A. Shamir, L. Adleman**.
- CRYPTOGRAPHY THEORY AND PRACTICE, **D.R. Stinson**, CRC Press Inc. 1995
- HOW TO SHARE A SECRET, Communications of the ACM **A. Shamir**.
- CURSO DE SEGURIDAD INFORMATICA, CRIPTOGRAFIA, FIRMA DIGITAL, COMERCIO EN INTERNET, **Daniel Sintelli**. Puerto Madero. Cap. Fed. 7 y 9 de Septiembre de 1998.

He preparado una serie de documentos Power Point a fin de poder guiarme en los temas a tratar en la exposición. Los temas que abarcaremos serán:

- ❖ Criptografía de Clave Simétrica
- ❖ Criptografía de Clave Pública (Asimétrica)
- ❖ Distintos Algoritmos de Encriptación
- ❖ Autenticación y Firma Digital (Certificación de Autor, Contenido y Fecha)
- ❖ Estándares Criptográficos Internacionales
- ❖ Validez Legal de un Documento Electrónico
- ❖ Hashing para garantizar la Inalterabilidad
- ❖ Usos Actuales de Criptografía
- ❖ El Comercio Electrónico en Internet
- ❖ Pretty Good Privacy (PGP)
- ❖ Formas de Crackear los distintos Sistemas de Seguridad
- ❖ Posibles soluciones a los ataques del Sistema