COLEGIO DE POSTGRADUADOS

INSTITUCION DE ENSEÑANZA E INVESTIGACION
EN CIENCIAS AGRÍCOLAS

CAMPUS MONTECILLO

POSTGRADO DE SOCIOECONOMÍA, ESTADÍSTICA E INFORMATICA

COMPUTO APLICADO

MiScrt: UN PAQUETE PARA DEFORMAR Y RECUPERAR ARCHIVOS CON CIERTAS CARACTERÍSTICAS ADICIONALES

TITO MARCOS SÁNCHEZ GUTIÉRREZ

T E S I S

PRESENTADA COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS

MONTECILLO, TEXCOCO, EDO. DE MEXICO

2016

La presente tesis titulada: **MiScrt: Un paquete para deformar archivos con ciertas características adicionales**
realizada por el alumno: **Tito Marcos Sánchez Gutiérrez**

bajo la dirección del Consejo Particular indicado, ha sido aprobada por el mismo y aceptada como requisito parcial para obtener el grado de:

MAESTRO EN CIENCIAS
SOCIOECONOMÍA ESTADÍSTICA E INFORMÁTICA

COMPUTO APLICADO

CONSEJO PARTICULAR

CONSEJERO
DR. JUAN RICARDO BAUER MENGELBERG

ASESOR
DR. DAVID H. DEL VALLE PANIAGUA

ASESOR
M.C. EDGAR RAMIREZ GALEANO

Montecillo, Texcoco, Estado de México, Julio de 2016

MiScrt: UN PAQUETE PARA DEFORMAR Y RECUPERAR ARCHIVOS CON CIERTAS CARACTERÍSTICAS ADICIONALES

Tito Marcos Sánchez Gutiérrez, MC.
Colegio de Postgraduados, 2016.

RESUMEN

El diseño y la implementación de un producto de software llamado MiScrt resultó de agregar nuevas funcionalidades a las ofrecidas por muchos productos que encriptan archivos en diversas plataformas y dispositivos. Aunque su función principal es deformar y recuperar un archivo, se agregaron otras, cada una con un objetivo específico. Se pueden encriptar varios archivos en un solo archivo encriptado – y se recuperan en forma individual; evitar tener que comunicar una palabra clave al que usará el archivo; los archivos encriptados se pueden almacenar en una base de datos de donde se recuperan. Se ofrece la deformación a lo que se denominó Nivel-2: se deforman dos archivos juntos; además de las dos palabras clave correctas (la del creador del archivo y la del usuario potencial de éste) se agregan nueve palabras espurias. Proporcionar una de estas últimas ofrece el segundo archivo, mientras que las dos correctas resultan en el primero. Se agregaron opciones para eliminar archivos y fechas para limitar su uso. Además de los usos normales y los resultantes de las funciones agregadas, el software es útil para respaldar el trabajo del día en la computadora, ya sea reemplazando o como respaldo de los archivos involucrados.

Palabras clave: Encriptación de datos, deformación de archivos, comunicación confidencial, comunicación secreta, directorio de palabras clave de contactos, repositorio de mensajes encriptados

Miscrit: A PACKAGE deformability and retrieve files CERTAIN ADDITIONAL FEATURES

Tito Marcos Sánchez Gutiérrez, M.Sc.
Colegio de Postgraduados, 2016.

ABSTRACT

A series of features that are not offered by many products that encrypt files in different platforms and for all types of devices motivated the design and implementation for computers and mobile devices of a software product called MiScrt. Though its main function is to encrypt and recover a file, it offers several complementary features, each one added for a specific purpose, explained in the paper; they include: several files may be encrypted into a single file; avoid the need to communicate passwords between sender and receiver; encrypted files can be stored as records of a database, where they can be recovered by its recipients. What was called Level-2 encryption is available: two files are encrypted, and besides the correct passwords nine additional passwords are included: the use of one of these will result in the second file, whereas one of the first two (the user's and the recipient's) will result in the first file. A set of options allow specifications for deletion of files as well as dates that limit their use. Besides the usual ones and those added as new features, one may create a deformed version of one's work, replacing the files to be protected or simply as a (deformed) back-up.

Keywords*:* Data encryption, file scrambling, confidential communication, secret communication, contacts' password directory, repository for encrypted message

DEDICATORIAS

Dedico esta tesis

A mis esposa Sandra Grisel que siempre me brindado su tiempo, amor, apoyo y sobre todo paciencia para cumplir mis objetivos.

A mi hija Maryam a quien amo y es un orgullo para mí.

A mis padres Tito Sánchez y Guadalupe Gutiérrez quienes me dieron la vida y me alentaron siempre a estudiar.

A mis hermanos y hermanas, en especial a Lilia que se nos ha adelantado en el camino.

A mis suegros y familiares de mi esposa, ahora también míos que me han tratado con gran cariño y respeto.

AGRADECIMIENTOS

Dese expresar mi más sincero agradecimiento al Consejo Nacional de Ciencia y Tecnología y al Colegio de Postgraduados por haberme permitido avanzar en mi formación académica y profesional.

Al Dr. John Bauer Mengelberg un agradecimiento especial por dirigir esta tesis y por el apoyo que me ha brindado para graduarme, sin su apoyo no hubiera podido culminar esta meta. Sus conocimientos y experiencia en las tecnologías de información han transformado mi visión acerca del desarrollo de sistemas.

Al Dr. David del Valle y al M.C. Edgar Ramírez por complementar y mostrarme áreas de conocimiento de tecnologías de información que no había contemplado en mi formación profesional.

A las compañeros y compañeras con quien compartí en clase y a los profesores y demás personas que estuvieron en las aulas compartiendo lo mejor de ellas.

CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE CUADROS

INTRODUCCIÓN

Hay muchos productos de software para encriptar y proteger de otro modo archivos de cualquier tipo. Algunos de ellos son públicos, otros aún abiertos. Este conjunto de paquetes, que se describe en el Capítulo I de esta tesis, parecería indicar que es innecesario ofrecer otro. El origen de MiScrt (el paquete que resultó de la investigación que se describe en esta tesis) fue la formulación de ciertos usos de un paquete de deformación o protección de archivos, sobre la naturaleza de los mecanismos de encriptación y otros empleados para aumentar la confidencialidad.

De hecho se partió de un producto desarrollado hace algunos años por el Dr. Bauer, pero que nunca se publicó, especialmente porque se planteó la necesidad de modificarlo en diversos modos. Por lo tanto, se procedió a formular un conjunto de tareas que debería ofrecer un tal paquete, lo que resultó en la funcionalidad del mismo (expresadas en distintos atributos, características y funciones).

En este trabajo se usan indistintamente los términos *deformado* y *, encriptado,* a pesar de que este último tiene un significado muy preciso en la literatura matemática y de computación. Se ha dedicado un capítulo de esta tesis precisamente a elementos de la deformación de archivos que utiliza el MiScrt, con comentarios y comparaciones con técnicas comúnmente empleadas por paquetes de este tipo.

El otro término que se usa en forma especial es el de “enviar un mensaje”. El intercambio de información entre dos partes es comunicación, y en ésta, la unidad es el mensaje. Si se le envía un material a alguien, eso constituye un mensaje. Si uno se auto envía algo, también es un mensaje. De ese modo el vocablo *mensaje* en esta tesis significará *algo que se proporciona a alguien.* Los mensajes se *envían;* esto incluye la entrega (física) y también el envío por cualquier canal de comunicación (teléfono, internet, correo postal, mensajero,

etc.) De este modo, se hablará del que envía el mensaje y el destinatario (o receptor) del mismo.

Para organizar la descripción general del paquete, que se proporciona aquí para establecer el tema principal y permitir al lector comprender los diversos conceptos:

El MiScrt es un paquete de software (programa) que permite deformar un archivo para producir otro archivo, que tiene un usuario que lo encriptó, y uno o varios destinatarios, que podrán recuperar el archivo original si proporcionan la palabra clave correcta.

El término deformar se usa aquí con este significado: hacer que el archivo (original) no se pueda interpretar sin esfuerzos adicionales a usar un programa similar al que lo creó. Por ejemplo, si se creó un archivo con un editor, en general se puede leer con ese mismo editor (u otros, en algunos casos). Si se deforma este archivo, el programa no podrá recuperarlo. Por ejemplo, si a un archivo creado con el programa MS-WORD se le modifican los primero bytes, será ilegible para dicho paquete.

Se mencionó el concepto de esfuerzos adicionales. Esto conduce a uno de los aspectos más importantes de la encriptación (o protección) de archivos: el nivel de seguridad. En esta tesis se usará esta definición de este concepto: es el costo que se tendría que invertir para violar los procedimientos de seguridad incluidos en la deformación de un archivo. Esto conducirá a la determinación del nivel de seguridad necesario para un paquete, en este caso el MiScrt. Se trata de valorar el beneficio que alguien podría obtener si consiguiera obtener el material que se protegió, lo que en la literatura de denomina “costo del contenido”. Un nivel de seguridad adecuado debe garantizar que este beneficio sea considerablemente menor al del costo de violación de la protección. Una vez más, queda otro término por definir: la cuantificación de “considerablemente” . En MiScrt se decidió que si este término se interpreta como “100 veces más”, se consideraría adecuada la protección.

Una estrategia no evidente pero aplicada aquí, es que para lograr esto se puede proceder de dos modos: incluir dispositivos de deformación suficiente

robustos, es decir, aumentar el costo de la violación, o disminuir el costo de los contenidos. Esto último se hace limitando el tipo de material que se deformará con MiScrt. En su diseño se incluyeron ambas estrategias: el paquete no ofrece seguridad para mensajes secretos de gran valor, cualquiera que sea la aplicación de la cuantificación de dicho valor.

Continuando con la descripción del paquete, se muestran las funcionalidades adicionales que se impusieron como condición al diseño del mismo; aquí adicionales se refiere a las que tendrá además de la función principal de un paquete de protección de archivos: deformar un archivo y recuperar el original a partir del mismo, con ciertos atributos como exigir algún dato al que desee recuperarlo (palabra clave, autenticación por otro medio – incluyendo uno implícito, o proporcionar una clave numérica que necesite el algoritmo de encriptación.

- Permitirá el envío de archivos que normalmente son rechazados por los servidores de correo electrónico, en especial los ".exe" puesto que pudieran contener virus. El paquete los encriptaría y cambiaría su extensión para dificultar (o aun imposibilitar) determinar su naturaleza de programa ejecutable.
- Nunca se transmitirá una palabra clave: el que envía el archivo usará una palabra clave previamente acordada con el receptor. Además, no deberá ser necesario recordar palabras clave; serán almacenados en forma conveniente para su uso.
- Se podrá enviar un archive de modo que no deja huella alguna. Esto incrementará la confidencialidad de los mensajes.
- Deberá ser posible enviar un mensaje deformado a varios destinatarios, cada uno con su propia palabra clave (para evitar que personas conozcan las de otros);
- Se podrá incluir un mensaje espurio para personas no autorizadas a ver el mensaje.
- Deberá ser posible indicar que se eliminan (o aún destruyan) archivos: el original al encriptar y/o el encriptado cuando se recupere el original. Destrucción significa que se reemplaza el contenido por caracteres sin significado y a continuación se elimina (suprime).
- Por último, se proveerá un nivel de seguridad adecuado. El significado de este término se comentó previamente.

Como se verá en el resto del trabajo, hay varios aspectos del paquete que lo distingue de productos semejantes. La posibilidad de usar la deformación de nivel 2 se una de ellas: se trata de agregarle al archivo original otro – cualquiera. Estos dos archivos se encriptan juntos. Se generan 9 palabras clave adicionales a “la buena”, y si alguien indica una de estas 9 palabras, obtendrá el segundo archivo, mientras si proporciona la palabra correcta obtiene el primero.

Quizá la característica saliente del MiScrt consiste en su directorio de contactos. Un usuario tiene un archivo propio, mismo que usa el paquete en sus sesiones de trabajo. Como parte de este archivo está el directorio: incluye tantos contactos como desea, y para cada contiene una palabra clave “de” (pwd-de) y otra “a” (pwd-1). Sea X el dueño de este archivo. Cuando encripta un archivo para proporcionarlo a uno de sus contactos – llamémoslo D - le incluye como palabra clave la que almacenó como “a”. Su contraparte también tendrá un directorio, con la entrada “X”. Las palabras clave serán las duales de las que contiene el archivo de X (intercambiadas las palabras “de” y “a”. De ese modo, cuando le llegar un archivo que le envió X, usa la palabra “de” – que es la incluyó X en el archivo.

La otra particularidad del paquete es que permite subir estos archivos encriptados a una base de datos a la que tienen acceso el que envía el mensaje y el que lo recibe. Se incluye el archivo como registro - o registros) - de dicha base de datos. El destinatario, tras indicar la palabra clave apropiada, podrá bajar el archivo a su computadora, donde lo puede recuperar con el programa. Oberve que el modo más frecuente de enviar un archivo es con una entrega personal o con algún método de envió por una red. El usuario receptor debe proporcionar la palabra: si usa el MiScrt para procesar el archivo, sólo tendrá que indicar quién se lo proporcionó (y no tendrá que recordar la palabra clave acordada). El objetivo principal de esta funcionalidad es que nadie envía palabras clave: el receptor ya la conoce.

Un ejemplo del uso del directorio es una casa de bolsa. Acuerda con cada uno de sus clientes una palabra que incluirá en las comunicaciones con ese cliente, y otra que pondrá el cliente en sus mensajes a su bróker. Esto evita tener que

recordar (o anotarlo de algún modo) las palabras clave pactadas con cada cliente; lo mismo sucede con éstos, no deberán recordar las palabras clave que usan con esa casa de bolsa (puede tener varias de éstas o organismos similares).

Todos los elementos y datos están protegidos por una combinación de dos métodos: encriptación y cambio de orden de elementos del archivo. La otra actividad que se puede incluir es la inserción de datos sin significado (“basura”) pero se estimó que el paquete no la necesitaba – sólo incrementaría el tamaño del archivo encriptado.

Por último se señala una característica que se impuso como restricción al paquete de software: debería funcionar como una App o Aplicación móvil. Esto significa que a partir de ésta se realizan todas las actividades: no hay código en ningún otro lado (especialmente no en una página de internet). Se agregó una restricción más técnica: el programa debería ser instalable simplemente incluyendo el (único) archivo que lo contiene; no debe haber proceso de instalación.

El resto de esta tesis se ha organizado del siguiente modo. En el primer capítulo se describen algunos productos que tienen funcionalidades similares a las buscada, y se comparan éstas con las del MiScrt. A continuación se describe la metodología empleada en la investigación, y los materiales empleados. A partir del capítulo 3 se describe el producto a detalle: los archivos que utiliza, el diseño y uso de la base de datos para almacenar y proporcionar archivos, las técnicas de encriptación que se seleccionaron o crearon, las operaciones que se realizan descritas a detalle (incluyendo la descripción de las rutinas del programa que las ejecuta) y la descripción de una sesión de trabajo con el paquete MiScrt. Algunas observaciones sobre el uso y las conclusiones permiten valorar el producto en cuanto a su utilidad.

1 Otros productos comparados con MiScrt

1.1 Comparación de productos

La seguridad de datos es un problema en todo el mundo, y hay un amplio mundo de soluciones de encriptación disponibles para ayudar a resolver este problema. La mayoría de estos productos son desarrollados y vendidos por entidades con fines de lucro, aunque algunos son creados como proyectos de código abierto gratuitos. Están disponibles, ya sea para la venta o descarga gratuita, en todo el mundo (Schneier 2016 et al).

1.1.1 Diferentes productos para deformación de archivos

En el estudio de Schneier se pudo observar que hay diferente muchos tipos diferentes de productos de cifrado, incluido el correo electrónico cifrado, cifrado de mensajes, cifrado de archivos, cifrado de divisas, y así sucesivamente. El cuadro que se muestra a continuación

Cuadro 1 Diferentes tipos de productos para encriptación

Tipo de producto	Cantidad	Tipo de producto	Cantidad
Adblocker	2	**Mesh**	5
AnonProxy	10	**MessageEncryption**	198
AnonRemailer	5	**Microprocessor**	2
Browser	6	**Multi**	43
Camera	2	**Network**	41
Chat	1	**OperatingSystem**	6
Circumvention	9	**P2PFileSharing**	3
CloudEncryption	31	**PasswordMgr**	28
Currency	4	**PasswordProtection**	1

DevTools	65	**Radio**	2
DiskEncryption	50	**Router**	5
FileEncryption	81	**ScriptBlocker**	1
FileSync	4	**Telephone**	40
Financial	2	**USB**	21
HardDrive	6	**VideoCall**	7
Identification	4	**VoIP**	9
Keyboard	1	**VPN**	80
MailEncryption	89	**WebHosting**	1

Fuente: Schneider et al 2016

1.2 Algunos esquemas de protección

1.2.1 Dentro de los programas

Algunos programas proporcionan una protección básica para que la información que contienen no pueda ser leída otras personas. Por ejemplo, Microsoft Word condiciona la apertura de un archivo con una contraseña; Excel permite proteger el contenido de celdas o de hojas; Adobe Acrobat permite guardar documentos PDF protegidos por contraseña.

1.2.2 En los sistemas operativos

La mayoría de los sistemas proporcionan alguna funcionalidad para proteger la información:

1.2.2.1 Windows - BitLocker

Proporciona cifrado completo de disco, está incluido con ciertas ediciones de Windows Vista y versiones posteriores. Está diseñado para proteger los datos al proporcionar cifrado para volúmenes enteros.

BitLocker no se limita a cifrar los archivos del usuario, que cifra toda la partición del sistema operativo, incluyendo los archivos de Windows, todas las

aplicaciones de software, así como todos los datos del usuario almacenados en la unidad. Se crea una llave USB con la contraseña en él y conectarlo al PC con el fin de arrancarlo (como una llave para su coche.) BitLocker utiliza una pequeña partición de arranque para comprobar si la contraseña correcta, y sólo arranca si está presente, si no, el disco duro es completamente inaccesible.

1.2.2.2 OSX – FileVault

Encripta el directorio inicial del usuario y todos los archivos contenidos en ella. A medida que permite leer y escribir archivos en el directorio de inicio, en el fondo, FileVault cifra y descifra los archivos sobre la marcha. Sólo la contraseña de inicio de sesión del usuario puede descifrar y montar esta imagen de unidad, por lo que al iniciar la sesión en el directorio hogar disponible como de costumbre - su icono, sin embargo, se parece a un bloqueo. Para otros usuarios en el mismo Mac, sin embargo, todos los archivos en el directorio inicial del usuario no se puede acceder.

1.3 Comparativa de funciones de algunos productos de encriptación

La siguiente Cuadro muestra una comparación de las principales funciones que tienen cuatro programas para deformar y proteger archivos.

Cuadro 2 Principales funciones de los programas de encripción

Funcionalidad	AEP (1)	CryptoCrat (2)	AEPE (3)	CryptoExpert (4)
Permite trabajar con varios archivos al mismo tiempo	Si	Si	Si	Si
Almacena varios archivos en un solo archivo encriptado	No	Si	No	Si *funciona como un drive virtual
Permite generar archivos autoextraibles	Si	Si	Si	No

Permite borrar los archivos de forma segura	Si	Si	Si	Si
Se integra con el explorador de Windows	Si	Si	Si	No
Comprime los archivos durante la encripción	Si	Si	Si	No
Automatic ability to wipe original files after encryption	Si	No	Si	-
Encripta texto	Si	No	No	No

- (1) AEP Advanced Encryption Package (AEP) es un software de cifrado de archivos profesional que cifra los archivos usando varios algoritmos de cifrado fuerte. También cuenta con una gran cantidad de funciones de servicios especiales.
- (2) CryptoCrat También es un software de cifrado, pero aparte de AEP, se comprime múltiples archivos en un solo archivo cifrado.
- (3) AEPE Advancedn Encription Plugin Explorer es un plug-in para Windows Explorer que ofrece un menú contextual que permite cifrar / descifrar y envíar por email archivos
- (4) CryptoExpert en un programa de cifrado que permite mantener los archivos importantes en volumen virtual, lo consigue rápido, y lo protege del acceso no autorizado, mediante la autenticación de contraseña.

1.3.1 Resultado de la búsqueda

Tomando en cuenta los objetivos adicionales enumerados en la introducción, se determinó que **no hay un producto disponible que realice las tareas que permitirían cumplir con estos objetivos** (o mejor expresado, no se encontróun tal producto). De este modo se necesitan varios productos para los tipos de protección de datos necesarios lo que complica la administración y pudiera implicar costos adicionales (del software usado).

Conclusión: hay mercado para un producto como el MiScrt, donde la palabra mercado se refiere a que pudiera haber personas a quien les serviría y no implica alguna consideración de ventas o distribución.

2 Metodología del diseño y materiales

2.1 Materiales

En el presente capítulo se describen los aspectos técnicos de la implementación de MIScrt.

2.2 Lenguaje de programación seleccionado

Hay en el mercado diferentes herramientas para desarrollar aplicaciones. Tomando en cuenta que el objetivo planteado era mantener una sola base de código fuente para que el programa se ejecutara en una computadora y en un dispositivo móvil se tomó la decisión de utilizar el lenguaje Delphi XE8.

Cuadro 3 Comparativa de herramienta seleccionada

Característica	Programa	
	Delphi XE 8	**Android Studio**
Lenguaje	Object Pascal	Java
Sistema Operativo Soportado		
Android	SI	SI
IOS	SI	Instalando el plugin "Multi-OS-Engine" de Intel
Windows Mobile	SI	No
Windows 32	SI	No
Windows 64	SI	No

2.3 Base de datos seleccionada

Para la base de datos relacional se utilizó FirebirdSQL, con la observación de que el paquete contempla el uso de cualquier manejador.

FireBird es una base de datos relacional open source que ofrece muchas características de SQL ANSI estándar y que funciona en Linux, Windows, MacOSX y una variedad de plataformas UNIX.

FireBird es el único gestor de base de datos con licencia realmente Libre para uso comercial. Otros que dicen tener licencia libre sólo lo son mientras su uso no sea comercial. Por tanto para usarlo en producción, las empresas deben pagar por el uso de las licencias del manejador de bases de datos a su fabricante, además del pago por el uso de las licencias del software.

Otros como Microsoft permiten el uso del sistema en las empresas sin costo, pero totalmente limitadas. Por ejemplo, en el servidor no se direcciona más de 1 GB de memoria. Por tanto se desperdician las características del servidor.

Cuadro 4 Algunas características de la base de datos FirebirdSQL

Característica	Firebird 2.X
Tamaño maximo de la base de datos	Practicamente ilimitada usando partiendo la bd en varios ficheros.
Tamaño máximo bd en un solo fichero	Vários Terabytes en la gran mayoria de plataformas; El limite viene definido por el sistema de archivos (4Gb o 2 Gb en algunas plataformas, Ej. Windows 98)
Máximo numero de ficheros de base de datos	64.535
Máximo numero de tablas	64.535
Máximo tamaño de una tabla	32 Tb

Máximo tamaño de un fichero externo de tabla	2 Gb
Máximo número de filas por tabla	> 16 Billones (
Máximo tamaño de fila	64.535 bytes (64 KB)

2.4 Versiones del programa

2.4.1 Versión para Windows

Para la versión computadora se decisión hacer una versión para el sistema operativo Windows. De acuerdo a la información obtenida del portal de internet **statcounter.com** Windows en sus diferentes versiones tiene una participación de más del 85% del mercando.

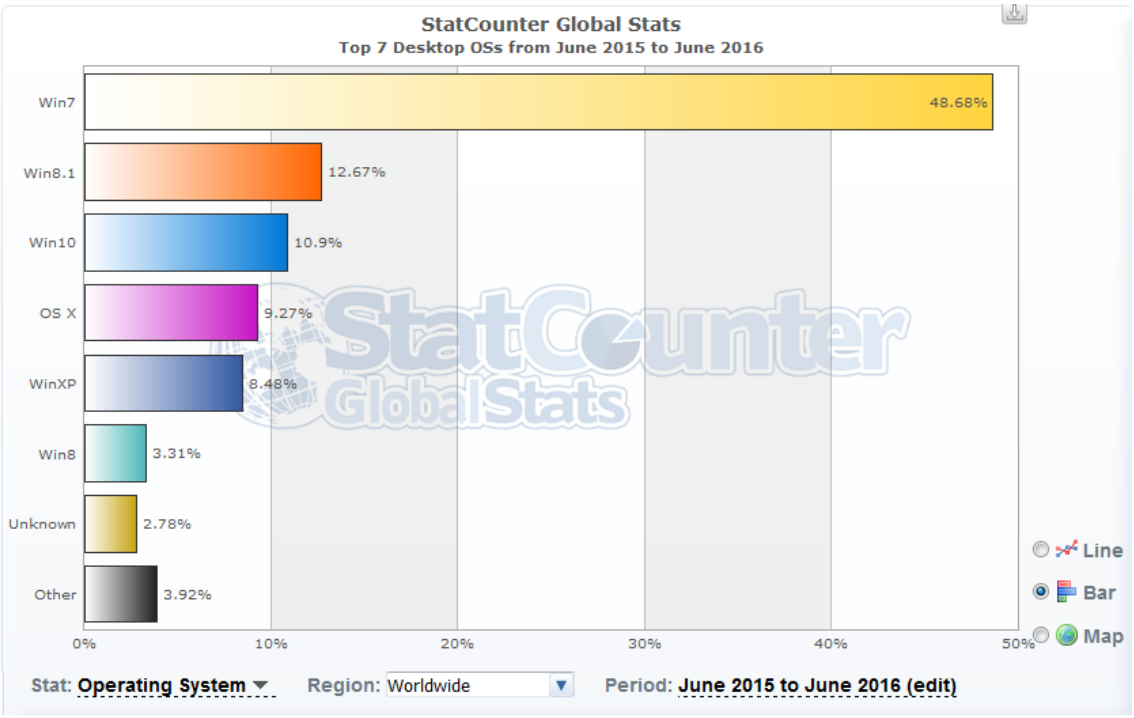
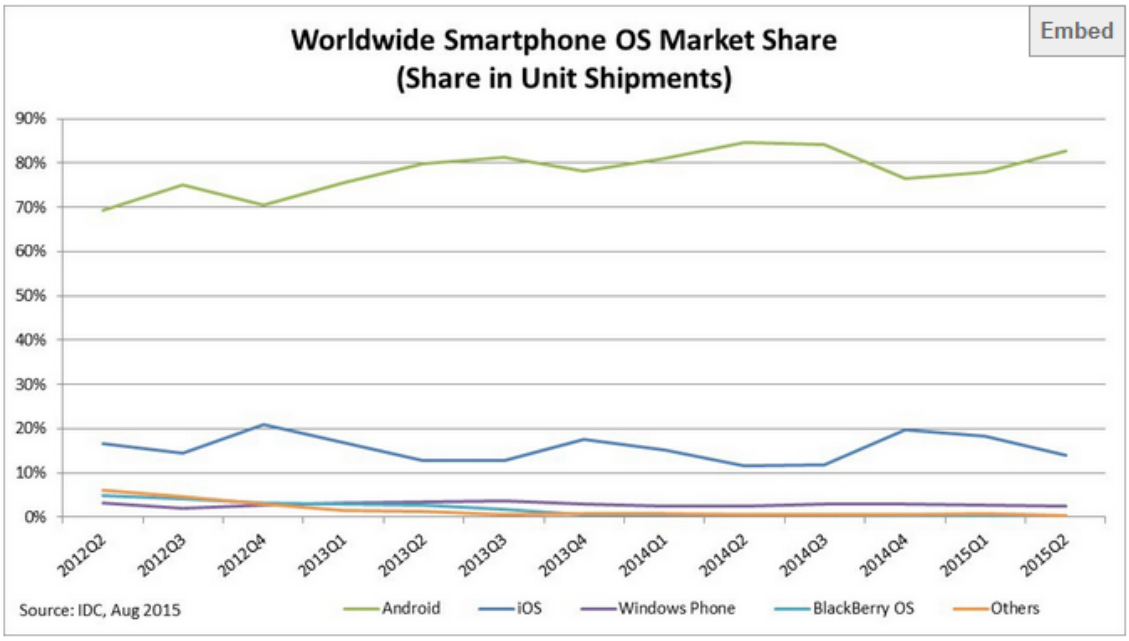


Figura 1 Uso de sistemas operativos en computadoras personales

Fuente: http://gs.statcounter.com/#desktop-os-ww-monthly-201506-201606-bar

2.4.2 Versión para Android

Tomando en cuenta une estudio de International Data Corportation (IDC) se decidió hacer una versión para Android tomando en cuenta que hasta agosto de 2015 tenía una participación del mercado de más del 80%.



Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

Source: IDC, Aug 2015

Figura 2 Participación de principales sistemas operativos móviles

Fuente: http://www.idc.com/prodserv/smartphone-os-market-share.jsp

2.5 Los objetivos que produjeron la funcionalidad incluida

Se señalaron en la Introducción los objetivos del MiScrt, además del principal que es encriptar o proteger de modo semejante un archivo. Aquí se presentan numerados para usar estos números en las explicaciones de algunos de ellos.

1. Permitirá el envío de archivos que normalmente son rechazados por los servidores de correo electrónico, en especial los ".exe" puesto que pudieran contener virus. El paquete los encriptaría y cambiaría su extensión para dificultar (o aun imposibilitar) determinar su naturaleza de programa ejecutable.
2. Nunca se transmitirá una palabra clave: el que envía el archivo usará una palabra clave previamente acordada con el receptor. Además, no deberá ser necesario recordar palabras clave; serán almacenados en forma conveniente para su uso.
3. Se podrá enviar un archive de modo que no deja huella alguna. Esto incrementará la confidencialidad de los mensajes.
4. Deberá ser posible enviar un mensaje deformado a varios destinatarios, cada uno con su propia palabra clave (para evitar que personas conozcan las de otros);
5. Se podrá incluir un mensaje espurio para personas no autorizadas a ver el mensaje.
6. Deberá ser posible indicar que se eliminan (o aún destruyan) archivos: el original al encriptar y/o el encriptado cuando se recupere el original. Destrucción significa que se reemplaza el contenido por caracteres sin significado y a continuación se elimina (suprime).
7. Por último, se proveerá un nivel de seguridad adecuado. El significado de este término se comentó previamente.
8. Se agregó a esta lista el uso de cifras de auditoría (checksums) para protección adicional del material.

2.6 Funciones y características resultantes de estos objetivos

Sólo se presentan los objetivos para los cuales no es evidente lo que implican.

El objetivo #1, “Nunca se transmitirá una palabra clave: el que envía el archivo usará una palabra clave previamente acordada con el receptor. Además, no deberá ser necesario recordar palabras clave; serán almacenados en forma conveniente para su uso.”

Este objetivo es quizá el que motivó el diseño de MiScrt. Para ofrecer la característica, se diseñó el **Directorio de Contactos:** para cada uno de los contactos de un usuario, podrá almacenar dos palabras clave (se verá más adelante que podrá almacenar una tercera).

Para simplificar la lectura, se denota con USU el usuario propietario del directorio) y con REC al contacto al cual ve dirigido el mensaje

Palabra_clave_a (Pwd-to) es la clave que tendrá que proporcionar REC contacto cuando reciba un archivo que le envió USU-

Palabra_clave_de (Pwd-from) es la clave que necesitará USU para recibir un mensaje enviado por REC.

En la mayoría de los casos, REC tendrá su directorio, en el cual habrá un contacto USU con las palabras intercambiadas. Su Pwd-to será el que USU usaba como Pwd-from, y su Pwd-from será la Pwd-to de USU.

El objetivo #2: Se podrá enviar un archive de modo que no deja huella alguna. Esto incrementará la confidencialidad de los mensajes.

El envío de “algo” por cualquier canal de comunicación siempre deja una huella. De ese modo, si mandamos el archivo encriptado por internet o teléfono, alguien puede saber que se efectuó tal envió, y en general tener acceso al material que se envió. Precisamente ése es uno de los motivos por los cuales se encriptan las cosas (no es único).

Se dedica mucho esfuerzo a la deformación, para que – en el caso de que alguien tenga el material y desee ver su contenido – sea lo más difícil posible (lo que se llama el nivel de seguridad discutido más abajo).

Si se pudiera hacer que nadie sabe que se ha enviado el archivo (y mucho menos contar con éste) se incrementa automáticamente el nivel de seguridad en forma muy significativa.

Para eso en MiScrt se ofrece depositar el archivo (su contenido) en una base de datos disponible para tal efecto. Se guarda el archivo usando el programa (en el dispositivo del usuario) para encriptar el material, crear el contenido y depositarlo en registros de una base de datos. El receptor usará su programa para recuperar el archivo (deformado) desde su dispositivo. En la base de datos se asigna un número de mensaje; el receptor (o los si hubiera más de uno) solicita ese mensaje, y tras proporcionar una palabra clave válida, recibe el contenido (deformado). No sólo el contenido nunca “viaja” en su forma natural (sin deformar) sino que nadie sabe que alguien actualizó la base de datos ni que la usó para obtener información.

El objetivo #3: Deberá ser posible enviar un mensaje deformado a varios destinatarios, cada uno con su propia palabra clave (para evitar que personas conozcan las de otros).

Al encriptar un archivo, se incluyen palabras claves que harán que quien las proporcione pueda recuperar el contenido. En general estas palabras son

- La del usuario (la que tiene su archivo de usuario) para que siempre pueda recuperar el material
- La del contacto o persona para el que se prepara el archivo. Esto se hace indicando el contacto o introduciendo una palabra clave (tecleada).
- El MiScrt permitirá agregar otros contactos (en total no más de 10 de ellos): incluirá otras palabra clave, una vez más tomadas del directorio de contactos o tecleadas.

El objetivo #4: Se podrá incluir un mensaje espurio para personas no autorizadas a ver el mensaje. Este es uno de los objetivos extraños que tiene el MiScrt.

En lugar de enviar un texto (o un archivo) encriptado protegido por una palabra clave – la del receptor del mensaje – se incluye un segundo texto, en general con contenido diferente. Si alguien intenta "leer" el mensaje, probará una y otra palabra clave. Para disminuir esta vulnerabilidad se agregan 9 palabras clave: si proporcionan alguna de éstas, se mostrará el segundo texto (el "malo" o "falso) mientas que la palabra del contacto le mostrará el primero (el "bueno" o "verdadero"). De este modo, es 4.5 veces más probable que sus intentos resulten en el segundo archivo (9 "malos" y 2 "buenos"). El número 9 no tiene significado u origen: se adoptó ese número para este uso: las probabilidad 81% de obtener uno malo fue considerada suficiente (el autor no cree en el 5% de Fisher para este tipo de cuestiones, de lo contrario necesitado 35 palabras clave espurias).

El programa genera estas palabras espurias con una serie de algoritmos; se intentó que satisficieran una de dos condiciones:

- Que se parecieran lo más posible a la verdadera
- Que fuera probable que un violador de la seguridad la probara.

En el capítulo sobre el uso de palabras clave hay más información sobre este tema.

El objetivo #5: Por último, se proveerá un nivel de seguridad adecuado. El significado de este término se comentó previamente. La determinación e implementación de este nivel constituye uno de los elementos más importantes de esta tesis, y se describe a detalle en varias secciones de la misma.

El objetivo #6: Se agregó a esta lista el uso de cifras de auditoría (en la literatura se las llama checksums o aún CRC) para protección adicional del material. Se trata de calcular un número como función de una serie de caracteres. Cuando alguien desea hacer uso de dichos caracteres, se valida primero que no hubo errores (o cambios no autorizados) para lo cual se aplica

el mismo algoritmo a los que hay ahora y se comparan los números obtenidos. Esto se hace para prevenir que alguien con los conocimientos técnicos necesarios cambie algún elemento de un archivo de modo que le permita usarlo aún sin tener una palabra válida. Se proporcionan más datos en una sección del capítulo dedicado al archivo de usuario.

3 Organización del archivo de usuario

3.1 Propósito y contenido del archivo de usuario

Un usuario de MiScrt es alguien que utiliza el programa en cualquier circunstancia y dispositivo. Usará un archivo “suyo) en el cual almacena ciertos datos que le ayudan a usar el programa, y por consiguiente, lograr el objetivo que lo motivó a iniciar una sesión de trabajo.

Una misma persona puede tener tantos archivos de este tipo como desea, pero deberán residir en dispositivos o carpetas diferentes. Estos dispositivos incluyen memorias de cualquier tipo (memorias USB, SSD, etc, CD actualizable, etc.)

El archivo de usuario es un archivo PLANO que se usa siempre en modo BINARIO.

3.2 Datos que contiene el archivo de usuario

A continuación de describen los elementos de datos que contienen estos archivos, agrupados en partes que utilizan las rutinas de lectura y grabación.

Cuadro 5 Contenidos del archivo de usuario

Parte	Contenido
1	Datos del usuario y opciones activas
2	Opciones activas (inicio sesión, cambiables)
3	Las bases de datos (máximo 9 bases)
4	Números de contacto disponibles (max 1000)
5	Los contactos registros de 100 contactos

Los datos del usuario incluyen: palabras clave, un apodo (que pondrá en sus mensajes enviados a la base de datos) y algunas opciones que le faciliten su uso.

- Las opciones activas son decisiones que afectan la oferta de funciones. Algunas pueden ser impuestas por la versión del programa; otras las puede actualizar el usuario en cualquier momento.
- Un usuario puede tener acceso a más de una base de datos de mensajes (como se verá, puede crear una propia). Se almacena en su archivo una lista de hasta 9 bases, y cuando desee usar alguna, seleccionará la apropiada de esta lista.
- La última parte es el directorio de contactos. Sin embargo, se decidió incluir un modo de reutilizar números de contacto (cuando se elimina un contacto, puede quedar ese número vacante).
- El directorio mismo consiste de registros (lógicos, son segmentos del archivo) que contienen 100 contactos cada uno.

Observaciones generales:

- Todas las palabras clave tienen una longitud máxima de 12 caracteres y se almacenan deformadas en 24 bytes
- Los números de usuario son enteros de 2 bytes (máximo número es 32.000)
- El apodo de un contacto no puede tener más de 16 caracteres.
- Toda la información importane se protege con cifras de auditoría. Los algoritmos se incluyeron al final de este capítulo. Las cifras de auditoría siempre son decimales (de 8 bytes).
- A continuación se muestran los campos a detalle. Observe que se muestran las estructuras en la terminología de Lazarus (structures). Para facilitar la lectura, se incluyeron comentarios sobre los campos que necesitan una explicación.

3.3 Estructuras utilizadas para el manejo de datos en las diversas partes

```
type
  T_USER_FILE_PARAMS = record // por ahora tiene 180 bytes
    descripcionDelArchivo: packed array [1 .. 40] of char;
    // aqui pone lo que quiera el user
    pwd: packed array [1 .. 24] of char; // para usar ese archivo
```

```
    pwdFuerte: packed array [1 .. 24] of char; // para opciones
confidenciales
    uwaPwdFuerte: Byte; // 0/1
    cuandoPidePwd: Byte;
    // cuando desea que le pidan su password de usuario
    // 0 = nunca   1= abc a contactos  (pide pwd normal)
    // 2  siempre (inicio sesión)
    // 3   invoca otras funciones
    // nunca le pide el pwd 2 veces en la misma sesión
    // si hay pwd fuerte se lo pide sólo para recuperar pwds
    // esto es default pero no se puede cambiar

    // opciones
    usaDirectorio: Byte; // 0/1
    idioma_default: Byte; // 1=ENGLISH  2= ESPAÑOL
    // este campo no se usa en la versión 2016 pero lo dejamos
    ufu_byte: Byte;
    color_background: Integer; // en la forma para actualizar opciones
    // le dejamos elegir el background
    puedeUsarBDParaPublicar: Byte;
    // 0 = no PUEDE    1= puede usar 1   2=PUEDE USAR VARIAS
    usaraBDParaLeer: Byte; // 0/1
    puedeCrearUnaBD: Byte; // 0/1=SI
    usaNivel2: Byte; // 0/1
    quierePoderVariosArchivos: Byte; // 0/1
    // ALGUNAS FECHAS
    fechaExpiraSuCopiaMiscrt: TDateTime;
    Otras_fechas: packed array [0 .. 2] of TDateTime;
    // 0 = creación archivo  1=expira el archivo  2=la que quiera
    // PARAMETROS DIVERSOS  USO SECRETO
    PARAM_1: Integer;
    PARAM_2: Integer;
    PARAM_3: Integer;
    Audit1: Integer;
    Audit2: Integer;
    UFU: packed array [1 .. 24] of char;
    Mayor_num_contacto_usado: Smallint;
  end;
```

3.3.1 Parte 2: opciones activas al inicio de la sesión

El usuario puede cambiar estas opciones (excepto las que vienen como fijas en la copia que usa). Se cambian durante una sesión, pero cuando se confirman los cambios termina la sesión (inicia otra con las opciones cambiadas).

```
T_OPCIONES_ACTIVAS_INICIO_SESION = record
    // _____________  OPCIONES ACTIVAS
    // se usan para esconder elementos de la forma
    // (LOS PUEDE CAMBIAR O DESACTIVAR CUANDO QUIERA)
    // todas son 0=false 1=tru excepto la INTEGER, claro

    SOLO_1_ARCHIVO: Byte;
```

```
    VARIOS_ARCHIVOS_PERO_SIN_NIVEL2: Byte;
    TEXTO_TECLEADO_SIN_NIVEL2: Byte;
    TEXTO_TECLEADO_CON_POSIBILIDAD_NIVEL2: Byte;
    No_ofrecer_bases_de_datos: Byte;
    //   cuidado    true es no ofrecer
    Poner_Archivos_encriptados_solo_en_mismo_directorio: Byte;
    SOLO_poner_archivos_deformados_en_mismo_directorio: Byte;
    Borrar_archivo_original_al_encriptar: Byte;
    Borrar_archivo_encriptado_al_recuperar: Byte;
    No_permite_cambiar_nombre_archivo_encriptado: Byte;
    AL_encriptar_Siempre_teclear_pwd: Byte;
    AL_encriptar_Siempre_usar_directorio_para_pwd: Byte;
    AL_recuperar_Siempre_teclear_pwd: Byte;
    AL_recuperar_Siempre_usar_directorio_para_pwd: Byte;
    AL_encriptar_indicar_contacto_solo_por_apodo: Byte;
    AL_recuperar_indicar_contacto_solo_por_apodo: Byte;
    Mostrar_nivel2_despues_del_intento_numero: Integer;
    // Estas cifras se ponen en la parte 1
    Audit1: double;
    Audit2: double;

  end;
```

3.3.2 Parte 3: las bases de datos

```
T_ONE_DATABASE = record
    servidor: array [0 .. 90] of char;
    nombreBase: array [0 .. 24] of char;
    extension: string[90];
    pwd_this_database: array [0 .. 31] of char;
    provider: array [0 .. 79] of char;
  end;

  T_USERFILE_DATABASES = record // maximo 8 bases de datos
    // cuidado  se usa la posición 0 de la lista
    // longitud 2 + 8  *208 = 2 + 1664 = 1666 bytes
    cuantasEnUso: Smallint;
    one_database: array [0 .. 7] of T_ONE_DATABASE;
  end;
```

3.3.3 Parte 4: los números de contacto disponibles

```
    // los números de contacto disponibles
  T_NUMEROS_DISPONIBLES = record
    Cuantos_disponibles_hay: Smallint;
    Disponible: array [0 .. 99] of Smallint; // usamos posición 0
    // solo caben 100 numeros disponibles siempre se usa el ultimo de
la lista
    // la lista NO está ordenada
    // opción (futura)  cuando llega un num (de 1 baja) se lo pone
    // descending  en orden
  end;
```

Parte 5 el directorio de contactos

```
// parte 5   el directorio de contactos
T_UN_CONTACTO = record // 172 bytes
  Apodo: array [0 .. 15] of char;
  Pwd_a: array [0 .. 23] of char;
  Pwd_de: array [0 .. 23] of char;
  Pwd_mal: array [0 .. 23] of char;
  Email: array [0 .. 23] of char; //
  Nombre: array [0 .. 31] of char;
  Phone: array [0 .. 16] of char;
  Param: Integer; // uso futuro
  Audit1: Integer;
  Audit2: Integer;
end;
```

3.3.3.1.1.1 Type T_otros_100_contactos

```
T_CONTACTOS = record
  // cada "registro" de  100 de esos mide   17200 + 4 + 2 = 17206
  Centena: Integer; // a partir de 101 es la 2  la 1 va de 1 a 100
  Ultima_posicion_ocupada: Smallint;
  Un_Contacto: array [0 .. 99] of T_UN_CONTACTO;
end;
```

3.4 LAS CIFRAS DE AUDITORÍA

3.4.1 Las cifras de auditoría para detectar errores en datos

3.4.1.1 Las cifras de la parte 1 (datos del usuario)

Con los datos del usuario se calculan 2 cifras usando la misma cadena (pero en distinto orden).

Cuidado: los pwds se agregan en su versión original (tecleada) es decir se recuperan cuando se trata de un cálculo (ya no está disponible la palabra tecleada).Las fechas se agregan como INT(DATE). Los enteros “longs” se agregan tal cual (no hace falta convertirlos a “string”).

With user_file_params

```
text = .pwd & .pwd_fuerte & .usa_pwd_fuerte & .CUANDO_PIDE_Pwd & Int(.Fecha_expira_su_copia_Miscrt) & .PUEDE_USAR_bases_de_datos_para_publicar & .puede_crear_UNA_BASE_DE_DATOS _
   & Int(.Otras_fechas(0)) & Int(.Otras_fechas(1)) & Int(.Otras_fechas(2))
.Audit1 = RUTINAS_DEFORMACION.calcula_cifra_audit(text, 1)
text = .PUEDE_USAR_bases_de_datos_para_publicar & _
.puede_crear_UNA_BASE_DE_DATOS & .pwd_fuerte & Int(.Otras_fechas(0)) & _
Int(.Fecha_expira_su_copia_Miscrt) & .pwd & Int(.Otras_fechas(2)) & _
.usa_pwd_fuerte & Int(.Otras_fechas(1)) & .CUANDO_PIDE_Pwd
.Audit2 = RUTINAS_DEFORMACION.calcula_cifra_audit(text, 2)
End With
```

3.4.1.2 Las cifras de auditoría de cada contacto

En los contactos (en cada uno) se calculan 2 cifras

Para Audit1 usa Apodo Pwd_mal Param Pwd_de Pwd_a

Para Audit2 usa Pwd_a Pwd_de Pwd_mal ApodoParam

3.5 El cálculo de una cifra de control

Es una función que usa 2 dos parámetros:

- el texto para el cual se calcula y
- el número de cifra (1 o 2). Esto porque cambian los algoritmos de la primera a la segunda cifra.

Devuelve el número calculado.

El cálculo de la cifra misma no se describe aquí. Se usan operaciones aritméticas diversas sobre números formados a partir del texto que llega.

4 Organización de un archivo deformado

4.1 Introducción

Cuando se deforma el contenido de un archivo se crea otro. De hecho, se puede depositar como un mensaje en una base de datos, y ésta se describe en el siguiente capítulo.

La lectura de este material asume que el lector ha asimilado algunos conceptos de los capítulos anteriores.

El cuadro 6 muestra las partes en las que se divide el archivo creado (encriptado).

Cuadro 6 Partes en las que se dividió un archivo encriptado

Parte	Contenido
Registro base	Descripciones, palabras clave, naturaleza
Las 9 palabras adicionales	9 palabras deformadas
Cada archivo incluido	Uno por cada archivo

El esquema de un tal archivo se ilustra en la Figura 3

<table>
<tr><td rowspan="2">Registro base</td><td rowspan="2">Palabras clave adicionales</td><td colspan="2">Archivo 1</td><td colspan="2">Archivo 2</td><td rowspan="2">Etc.</td></tr>
<tr><td>Nombre Long (bytes)</td><td>El archivo encriptado</td><td>Nombre Long (bytes)</td><td>El archivo encriptado</td></tr>
</table>

Figura 3 La estructura del archivo encriptado.

Las palabras clave adicionales sólo se agregan si se usarán:

- Si se trata de nivel-2
- Si hay varios contactos a los cuales se les envía este mensaje.

Como se verá en las secciones correspondientes a la deformación de archivos y textos, difieren uno del otro. Para textos, se completa la longitud de la cadena para que sea múltiplo de 4 (esto se hace en memoria y no tiene impacto en otro lado),

En el caso de un archivo, agregar bytes implicaría grabar información falsa en el archivo original, de modo que se hará de otro modo. Los bytes que sobran (Longitud del archivo mod 4, es decir, módulo 4) se procesan por separado.

Figura 4 Cómo se graba cada archivo encriptado (se muestran 2 pero puede haber muchos más

Para cada archivo, hay 2 segmentos. En el primero se anota el nombre del archivo (para que se pueda recuperar con el mismo nombre que tenía) y la longitud en bytes del archivo a encriptar. Luego se graba el arreglo COMP (que como verá contiene parámetros de la encriptación, seguido del arreglo de enteros (formado a partir del contenido) y encriptado. Finalmente se agrega un entero que contiene una versión deformada de los bytes sobrantes.

Figura 5 Cómo se graban los textos deformados

A continuación se presentan las estructuras de datos (*Type, structure* o como las llamen los lenguajes diversos) con los cuales se graba y lee la información de estos archivos.

Para el archivo encriptado se usa un archivo plano que se procesa en modo binario (byte por byte).

```
type
  T_REGISTRO_BASE = record // POR AHORA TIENE  182 BYTES
    // el orden se modificará luego
    // cuantos archivos contiene as byte
    // longitud_primer_archivo as long
    // en los subsecuentes se pone el siguiente
    cuantos_archivos_tiene: Byte;
    es_nivel_2: Byte; // 0/1
    // CUANTOS DESTINATARIOS
    CUANTOS_DESTINATARIOS : Byte;
    // las palabras clave tienen un máximo de 16 caracateres
    // se guardan como 24 bytes (deformados)
    pwd_creador: array[0..23] of char;
    pwd_destinatario: array[0..23] of char; // max pwd 16 caracteres
    num_contacto_destinatario: Integer;
    APODO_DEL_CONTACTO_QUE_LO_CREO: array[0..16] of char;
    APODO_DE_A_QUIEN_VA_DIRIGIDO: array[0..16] of char;
    param1: Integer; // uso pendiente
    param2: Integer; // uso pendiente
    // fechas
    fecha_creacion: TDateTime;
    fecha_inicial: TDateTime; // no se puede desencriptar antes de …
    fecha_final: TDateTime; // no se puede desencriptar después de …
    num_max_lecturas: Smallint; // cuando llega a este número se
destruye
    numero_de_lecturas: Smallint;
    directorio: array[0..40] of char; // se llega el directorio es
porque se intenta
    // recuperarlo en el mismo
    Cuantos_intentos_pwd_mal: Smallint;
    Eliminar_encriptado_despues_de : Smallint;
    Como_eliminar: Smallint; // 0=NO 1=DELETE, 2=borrado total
    ENVIO_EMAIL: TDateTime; // LA FECHA EN LA CUAL SE ENVIÓ
    MARCADO_PARA_ELIMINAR: Boolean;
    Audit1: Integer;
    Audit2: Integer;
  end;

  T_PALS_ADICIONALES = record // mide 220 bytes
    Cuantos_pwds_se_Generaron: Smallint; // por ahora siempre vale 9
    Otro_generado: array [0 .. 8] of String[24];
    audit: Double;
  end;

  T_PRIMERA_PARTE_DE_CADA_ARCHIVO = record // mide 108 bytes
    longitudEsteArchivoADeformar: Integer; // es en bytes
    //nombre: packed array[0..99] of char; // QUE TENIA, con extensión
    nombre: String[100]; // QUE TENIA, con extensión
  end;
```

5 Actualización del archivo de usuarios

Un usuario puede modificar su archivo en estos aspectos

- Sus palabra claves
- Sus datos complementarios
- Las bases de datos que puede utilizar
- Sus contactos.

5.1 Actualización de sus datos

La Figuras 6 muestra el formulario con el que el usuario puede cambiar sus propios datos. La palabra clave “fuerte” se incluyó para que no cualquiera que use el archivo (quizá conociendo la palabra clave del usuario) puede recuperar la palabra clave a partir de su versión deformada, que es la que esté en el archivo.

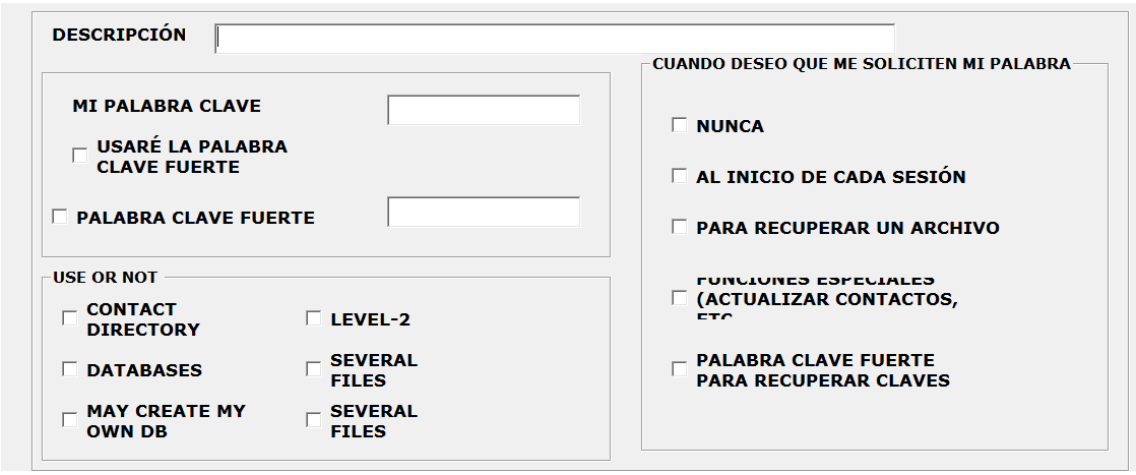


Figura 6 La interface con la que se actualizan los datos del usuario

En el programa hay otros campos para actualizar las diversas opciones.

Se puede indicar cuándo se desea que el sistema le exija su palabra. Por ejemplo si una persona tiene la seguridad de que otros no pueden usar su archivo, puede indicar que no le exijan su palabra clave al inicio de cada sesión. Las otras opciones se explican por sí mismas.

Observe que para actualizar los datos debe estar en una sesión; si no le habían pedido su palabra clave, deberá proporcionarla cuando la invoque. Si olvida su palabra clave, ya no podrá usar el archivo. (Si recuerda solamente la palabra fuerte podrá entrar con ésta).

5.2 Actualización de la lista de bases de datos

Un usuario puede o no tener acceso a las diversas bases de datos creadas por él mismo o terceros. Para poder usar una tal base de datos deberá agregarla a la lista de éstas en su archivo de usuario. La Figura 7 muestra la interfaz para hacer esta operación. La base de datos deberá llamarse MiScrtbase, y la extensión dependerá del proveedor de la base de datos relacional correspondiente.

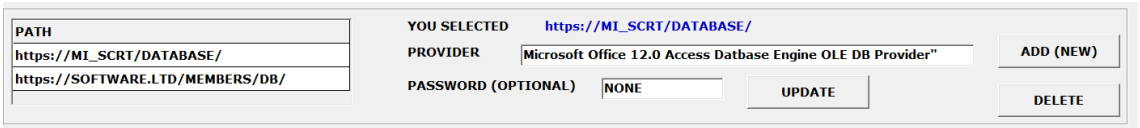


Figura 7 Interfaz para actualizar la lista de bases de datos disponibles

5.3 Actualización de contactos de su directorio

Podrá agregar y quitar contactos en cualquier momento. También podrá cambiar los datos de un contacto. Para ubicar el contacto a modificar, usa una forma como la ilustrada en la Figura 8. Indica o no un criterio de búsqueda (filtro): si lo hace, sólo se le mostrarán en la lista de contactos aquéllos que cuyo apodo satisfaga dicho criterio. Observe que sólo usará uno de los 3 campos: si indica “comienza con”, el sistema proporciona el filtro mostrado; lo mismo sucede con “contiene”; también puede indicar directamente el filtro o criterio si sabe cómo se hace (agregar asteriscos al final, y excepto en el caso “comienza con”, otro al inicio.

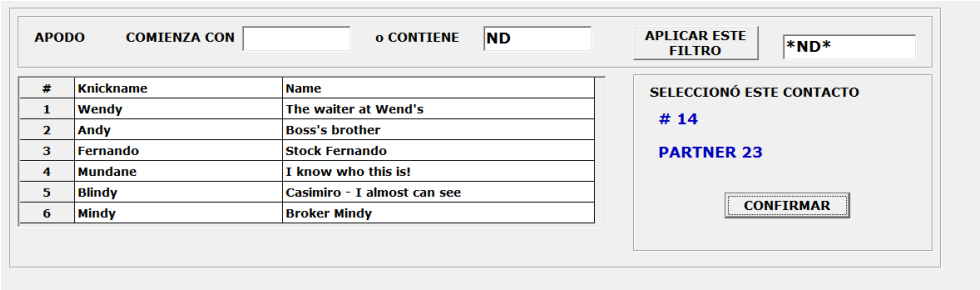


Figura 8 Formularios usado para encontrar el contacto deseado

Un contacto tiene un APODO (el nombre es para conveniencia del usuario, para recordar de quién se trata). Sus palabras claves son “a” para usar al enviar algo a ese contacto;”de” para cuando reciba material de éste, y finalmente “mala” que se explica por separado. Los datos complementarios son optativos: por ahora MiScrt no usa estos campos.

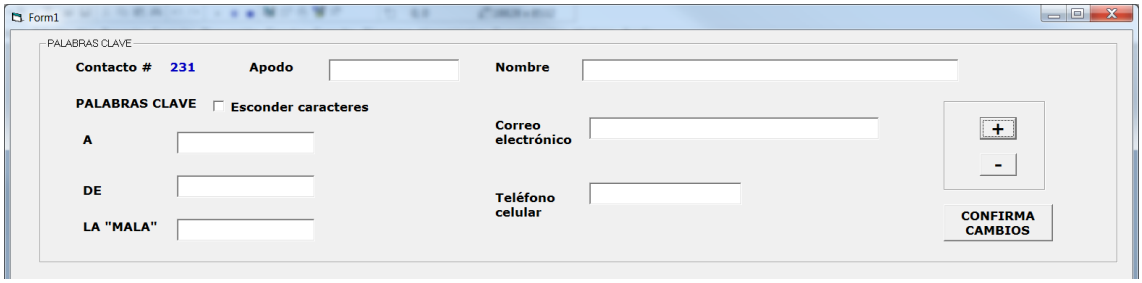


Figura 9 Formulario que permite actualizar los datos de un contacto

La palabra “mala” se usa sólo cuando se envía un material usando el nivel-2. En lugar de generar 9 palabras espurias, se incluye esta palabra como la primera de ellas. De ese modo el receptor (que conoce esta palabra porque la proporcionó) puede invocar el archivo “malo” si lo desea (por ejemplo si le quiere mostrar algo equivocado a alguien).

6 La base de datos para depositar mensajes

6.1 Introducción

La base de datos se usa para almacenar archivos (o textos) deformados. Alguien “crea” el mensaje (deposita algo deformado) y otro lo recupera.

Sólo se puede incluir en un mismo mensaje 1 archivo o texto, excepto si se trata de Nivel-2 (en ese caso, serán 2).

- Si es un texto, éste (deformado) se guarda en un registro único de la tabla de Textos.
- En el caso de un archivo, puede ocupar varios si es grande.
- Para archivos, el nombre del archivo es el del “bueno”. El nombre “sugerido” para el archivo “malo” lo proporciona el usuario que crea el mensaje (si no lo hace, le pondrá el mismo nombre que al bueno).

6.2 Las tablas de la base de datos

TABLA MENSAJES

Campo	**Tipo**	**Longitud**	**Descripción**
NUM_MENSAJE	Integer		Número de mensaje
QUIEN_GENERO	Char	24	Nombre o apodo de quien creo el mensaje
FECHA_INICIAL	DATE		Fecha a partir de la cual se puede leer el mensaje
FECHA_FINAL	DATE		Fecha hasta la que se puede leer el mensaje
NUM_LECTURAS	SMALLINT		Número de veces que se ha leido el mensaje
NUM_MAX_LECTURAS	SMALLINT		número de veces como máximo que se puede leer el mensaje
CUANTOS_ARCHIVOS	BYTE		Si dice 2, es nivel 2 de deformación
APODO_DESTINATARIO	CHAR	16	Apodo del destinatario
A_CUANTOS_DESTINATARIOS	BYTE		Número de destinatarios a los que se envía el mensaje (solo si no es

			nivel2)
Pwd_1	CHAR	24	Contraseña del creador
Pwd_2	CHAR	24	Contraseña del destinatario
Pwad_1	Char	24	Contraseña 1
Pwad_2	Char	24	Contraseña 2
Pwad_3	Char	24	Contraseña 3
Pwad_4	Char	24	Contraseña 4
Pwad_5	Char	24	Contraseña 5
Pwad_6	Char	24	Contraseña 6
Pwad_7	Char	24	Contraseña 7
Pwad_8	Char	24	Contraseña 8
Pwad_9	Char	24	Contraseña 9
CUANTOS_INTENTOS_PWD_MAL	SMALLINT		Número de intentos fallidos
CUANTOS_REGISTROS_TEXTOS_ARCh1	SMALLINT		Número de registros para el texto del archivo 2
CUANTOS_REGISTROS_TEXTOS_ARCh2	SMALLINT		Número de registros para el texto del archivo2
LONG_ARCHIVO_1	INTEGER		Longitud del archivo1
LONG_ARCHIVO_2	INTEGER		Longitud del archivo 2
LONG_TEXTO_DEFORMADO_1	INTEGER		Longitud del texto deformado 1
LONG_TEXTO_DEFORMADO_2	INTEGER		Longitud del texto deformado 2
NOMBRE_ARCHIVO_1	CHAR	SIN ESPECIFICACIÓN	Nombre del archivo 1
NOMBRE_ARCHIVO_2	CHAR	SIN ESPECIFICACIÓN	Nombre del archivo 2
MARCADO_PARA_BAJA	DATE		Si está marcado para baja
BORRAR_DESPUES_DE_N_LECTURAS	SMALLINT		Para indicar después de cuantas lecturas se debe borrar el mensaje
TIPO_DE_BORRADO	SMALLINT		0 = no 1=dar delete mensaje y textos 2=regrabar con ceros y luego delete
PARM	INTEGER		Para uso futuro
AUDIT1	decimal		Cifra de auditoría 1
AUDIT2	DECIMAL		Cifra de auditoría 2

TABLA LOS_TEXTOS

Campo	Tipo	Longitud	Descripción
NUM_MENSAJE	Integer		Número de mensaje
CONSECUTIVO	SMALLINT		Consecutivo de mensajes
EL_TEXTO	BLOB		Contenido del mensaje
marcado_para_baja	BOOLEAN		Si el mensaje está marcado para eliminar

COMENTARIOS SOBRE ESTAS TABLAS

Un “mensaje” se guarda con un número (único). Los datos del mensaje (especialmente las palabras clave que permiten leerlo) están en la tabla mensajes.

Los textos (arreglos deformados) se guardan en registros de la tabla TEXTOS. En textos, se graba un registro por texto (no debe exceder la longitud máxima de un campo MEMO pero de hecho, no puede ser que suceda esto).

En archivos, si el archivo es grande, se graban “cachos”. El tamaño sugerido es de 1 megabyte por registro. Se incluye un consecutivo para cada “cacho” que se graba. El primer (o único) cacho se procesa en un modo ligeramente diferente a los demás (pero esto es transparente al uso de los campos).

Si hay dos archivos, el BUENO se graba antes del MALO.

Se protegen los registros de la tabla MENSAJES para evitar que alguien sustituya algún dato por otro, y evitar que alteren los contadores de número de lecturas o intentos de password.

Se caculan dos cifras de auditoría: Se concatenan los campos

Fecha_final¸ Num_lecturas,Num_max_lecturas, Apodo_destinatario,el 1er pwd y se usa la rutina de cálculo de una cifra de auditoría basada en una cadena de caracteres.

Se concatena el segundo password, num_intentos_pwd_mal, param y audit1 (cuidado con éste)

Y se calcula la cifra. Observe que cuando cambia algo se tienen que calcular ambas cifras audit (lo mismo pasa cuando se trata de validar un registro antes de ofrecerlo).

En la tabla textos se introduce una cifra para evitar que reemplacen el texto por otro, aunque no se sabe para qué harían esto. Se concatena el numero de mensaje, el consecutivo y los primeros 16 byes del texto y se calcula la cifra de auditoría.

6.3 Uso de la base de datos

Para “enviar” un archivo deformado, se ofrece la alternativa de almacenarlo en una base de datos de modo que el destinatario puede bajarlo a su computadora. De ese modo habrá rutinas para las siguientes funciones.

6.3.1 Grabar archivos o textos

Observación: están todos los valores necesarios ya en memoria.

Se crea un registro de MENSAJES. Se asigna un número de mensaje (ya sea el siguiente al último registro o el primero disponible – agujero.

Se puede usar la rutina de decisión (o indicar que siempre se usará alguna de las anteriores). IF recordcount de la tabla mensajes < ultimo numero de mensajes / 2 Y ultimo numero de mensaje > 50 decimos que hay desperdicio. En ese caso, se usan los agujeros; de lo contrario, se agrega al final de la tabla.

Se agregan en TEXTOS los textos o archivos deformados. Se calcula la cifra de auditoría y se le da update.

Se actualizan (a medida que se agregan los textos) los campos involucrados en MENSAJES

Se calculan las cifras de auditoría de MENSAJES y se da UPDATE

6.3.2 Recuperar archivos o textos

El usuario indica un número de mensaje que le interesa. (el programa ofrecerá modos de encontrar un mensaje por ciertos datos del mismo).

Se lee el registro del mensaje en la tabla: si no está se informa al usuario.

Se solicita la palabra clave necesaria para leer (procesar) el mensaje.

Se usa lo de NIVEL 2 (explicado en otro lado).

Se leen: el texto o los registros de archivos de acuerdo a lo que indica el MENSAJE.

Si es el “malo” se inicia la lectura en el registro de TEXTOS que contiene el primer “cacho” del archivo deformado.

Se desencripta el texto de ese registro, y se agrega a lo recuperado

Ya sea el archivo que se crea o el texto que se devuelve).

Se incrementa en 1 el número de lecturas del mensaje.

Cuando se llega al máximo, se marca para baja el mensaje.

6.4 Cambios a un MENSAJE

Se indica el cambio, y el PWD deber ser el primero (del creador).

Se permite cambiar SóLO EL TEXTO (para archivos será necesarios eliminar el mensaje y crear otro).

Se puede cambiar cualquiera de los 2 textos (en caso de Nivel2).

No se puede cambiar un mensaje de Nivel-2 a “NO” ni vice-versa.

En caso de proceder los cambios, el usuario proporciona “cuál texto cambiaré”

(cuando es nivel2) y lo proporciona nuevamente.

Puede cambiar ambos textos (invoca cambios con el primer texto, y hace lo propio con el segundo).

Puede cambiar la fecha inicial y/o la fecha final.

Puede inicializar el número de lecturas o cambiar el máximo número de éstas.

Puede cambiar el destinatario: debe cambiar el APODO, el PWD2 y si hay.,el PWD3.

Esta operación sólo se puede hacer con PWD de directorio.

Puede cambiar la PWD 2 “tecleando una nueva”. Es su responsabilidad hacer esto sin que produzca situaciones inconvenientes (por ejemplo, que el destinatario “original” no pueda leer ese mensaje.

6.5 Eliminación de mensajes de la base de datos

Se invoca esta función. Exigirá la palabra clave del usuario de la SESIÖN.

Selecciona opciones: cuál base de datos, criterio de eliminación:

Los marcados para baja

Los “vencidos”(fecha final > hoy)

Los que ya han alcanzado el número máximo de leclturas

Uno por uno varios mensajes

En todos los casos, el sistema VERIFICA que el pwd proporcionado es el pwd-1 del mensaje (es decir, este usuario generó ese mensaje).

7 Una sesión de MiScrt

7.1 Descripción general de una sesión

Las siguientes características se impusieron al diseño de todas las interfaces:

- Disposición de los objetos deberá facilitar el uso de las funciones;
- No solicitarán nunca información superflua;
- No se pedirán opciones o valores que no son aplicables;

- Las pantallas siempre reflejarán las selecciones y opciones de pasos anteriores; es decir, en cada paso se muestra o se pide sólo la información pertinente.

7.2 Inicio de la sesión

Una sesión se inicia cuando se invoca el programa. En ese momento, el sistema abrirá el (archivo) expediente del usuario, que reside en el mismo directorio que el programa ejecutable. Si no se encuentra un archivo de este tipo, el usuario puede optar por

- crear uno (y proporcionar los elementos de datos esenciales, especialmente su contraseña). El programa contiene rutinas para crear este tipo de archivos.
- utilizar un archivo existente desde una carpeta diferente.
- No iniciar la sesión. Puede ser que intente con otra copia del MiScrt que reside en otra carpeta o aun en otro dispositivo.

De este modo, a través de algunas de las opciones almacenadas en su archivo de usuario, el sistema está consciente de lo que puede o decidió utilizar de las características de MiScrt..

7.3 Selección fundamental: usará o no bases de datos

Primero se determina si hay o no una conexión a Internet activa. De no haber tal conexión, no se ejecuta este paso y se indica que NO usará bases de datos.

Aparece la forma ilustrada en la Figura 10. Indica si desea o no usar bases: si indica que no, termina este paso.

Si selecciona SI, si tiene una sola base de datos en su lista de éstas (en su archivo) se le muestra y la confirma. Si no la confirma, puede invocar el menú de funciones privadas o adicionales, donde podrá crear una nueva base (o incluir una que ya está creada pero no figuraba en su lista).

Figura 10 El usuario indica si va a trabajar con mensajes (almacenados en una base de datos) o no. También indica qué base de datos que va a usar.

Mediante la forma ilustrad en la Figura 11, se podrá seleccionar una base de datos de la lista de éstas almacenada en el archivo del usuario.

Figura 11 Selección de la base de datos deseada

Una vez seleccionada la base (o la indicación de que no usaría base de datos)

Aparece una de las formas ilustradas en la Figuras 4 y la Figura 5, dependiendo de la selección anterior: si indicó que usaría la base de datos, le aparece la correspondiente a esa situación, y viceversa.

7.4 Sesión con bases de datos

En un formulario como el que ilustra la Figura 12, selecciona si desea crear un mensaje con material encriptado o quiere recuperar lo que se había guardado como un mensaje.

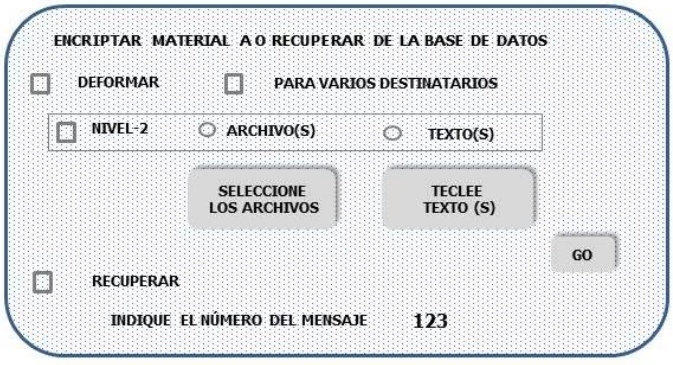


Figura 12 Selección de la función deseada con uso de la base de datos

Tras seleccionar el o los 2 archivos o proporcionar el o los 2 textos, invoca la función deseada con la tecla GO. Cabe señalar que no se pueden deformar

más de 2 archivos y guardar el material encriptado en una base de datos. Adicionalmente, sólo serán 2 archivos o textos si se trata de nivel-2.

El número de mensaje para la base de datos lo genera el sistema: al finalizar el proceso de encriptación informa el resultado (el número de mensaje).

7.5 Sesión con archivos (no bases de datos)

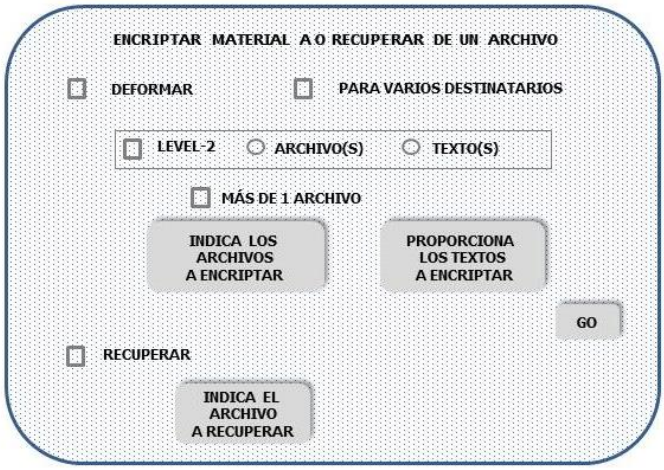


Figura 13 Selección de la función deseada cuando no se usa la base de datos

El programa reacciona a la primera selección (Deformar o recuperar): aparece la forma sólo con esos 2 campos. A medida que el programa "sabe" lo que quiere solicita las otras opciones una tras otra. Cuando procede cambia las leyendas (por ejemplo no dirá "texto(s)" sino que cuando según aplique la etiqueta será "texto" o "textos".

Los textos se teclean en una forma que ofrece uno o dos espacios (según lo que se desprende de las opciones) para hacerlo. No hay ningún límite de la longitud de los textos.

7.6 Selección de archivos

Para indicar los archivos a utilizar (tanto para recuperar el contenido como para saber cuáles se quieren encriptar juntos) se usa una interfaz como la ilustrada en la Figura 14. Esta forma se invoca para diversos propósitos y por lo tanto se adapta a cada uso (desaparecen campos, etc.

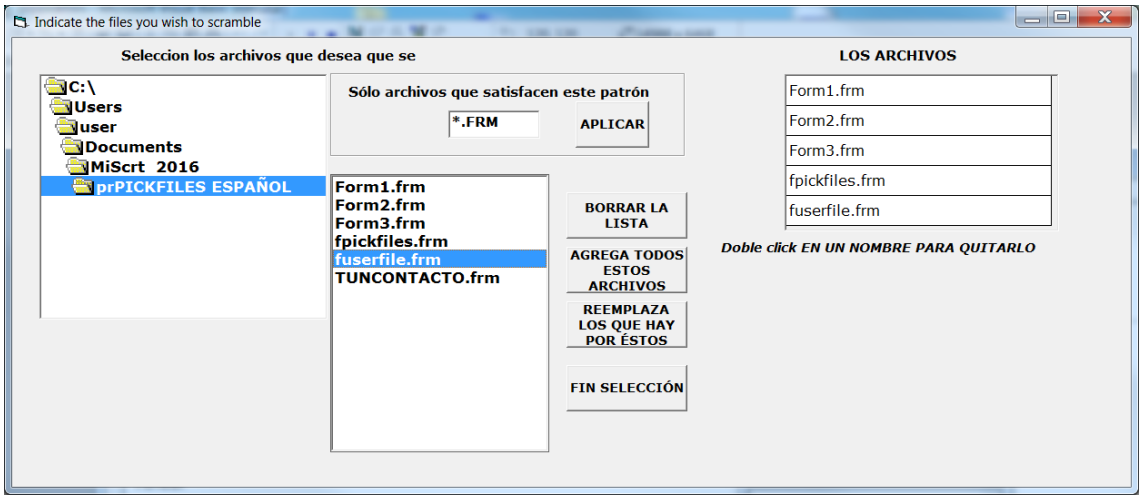


Figura 14 La pantalla que se utiliza para indicar los archivos que se van a cifrar

Cuando sólo se seleccionará un archivo, desaparecen los botones para agregar varios o quitarlos, y en la lista de los archivos aparecerá el archivo indicado, con la leyenda EL ARCHIVO.

SI se trata de nivel-2, la lista desaparece y en cambio habrá dos campos que indican los 2 archivos indicados.

Si se trata de recuperar un archivo, cambian las leyendas y aparece la lista donde se pondrá el único archivo. En la “máscara” tendrá el valor “*.jbr” que es la extensión de los archivos deformados en MiScrt.

Cuando termina de indicar los archivos (o los textos) el botón GO de la forma invoca la función elegida: encriptar o recuperar. Se invocan las rutinas que efectúan estas operaciones – descritas en los siguientes capítulos – donde termina el proceso.

8 Deformación de archivos y textos

Siempre llega (en memoria) la información para la cual se solicita la deformación (la actualizan las opciones seleccionadas en la sesión de trabajo – de la cual forma parte la encriptación que se describe aquí. Estos datos son:

- Si se trata de textos no archivos a encriptar
- La lista de archivos o el texto (o los textos, en nivel.2)
- Si aplica nivel-2 o no

- La lista de archivos a encriptar (en memoria)

Además, están los datos del usuario en memoria.

8.1 Descripción general de lo que realizan estas funciones del programa

A pesar de las variantes de los 4 casos que se describirán, se puede decir que todos hacen lo siguiente.

- Preparan el archivo de salida o la base de datos
    - Generan un nombre o un número de mensaje (numero nuevo)
    - Abren (crean) el archivo con un nombre generado por sistema
    - Preparan y graban el registro base del archivo encriptado

      En el caso de base de datos

      Preparan una tabla (en memoria) para cada una de las de la base de datos. Típicamente se hace con las instrucciones SQL

      Select * from Mensajes where num_mensaje = numero nuevo

      Select * from Textos where num_mensaje = numero nuevo

      Observe que estas tablas estarán vacias si no hubiera un error de asignación del número de mensaje.
    - O el registro de la tabla MENSAJES de la base de datos
    - Graban el registro base en el archivo o el registro de mensajes en la base de datos (luego se actualizarán éstos)
    - Procesa uno por uno los archivos o textos indicados
        - Los prepara para la encriptación, los invoca y graba los arreglos encriptados resultanets
    - Cuando termina, solicita información sobre el “mensaje”
        - En especial, la o las palabras clave
        - Algunas opciones que ofrece
    - Regraba el registro base en el archivo o el registro del mensaje en la tabla de Mensajes.

8.2 Encriptar uno o varios archivos a un archivo (encriptado)

El programa usa las constantes

Donde_empiezan_palabras_adicionales

Donde_empieza_ primer_archivo

Cuanto_mide_segmento_nombre_y_longitud

Para ubicar el cursor en el lugar adecuado del archivo.

Genera un nombre de archivo (basado en la hora del sistema)

Crea un archivo (lo abre) con ese nombre

Prepara el registro base del archivo encriptado, en especial

- Si es nivel-2
- que se trata de archivos (en este caso es así)
- si va dirigido a varios contactos. Lo graba en el archivo.

Si es nivel -2 o está dirigido a varios contactos, graba el segmento correspondiente (con las palabras con valor “espacios”, las agregará al final del proceso para no tener que proporcionarlas antes de que termine con éxito la encriptación.

Ahora procesa, uno tras otro, cada uno de los archivos de la lista de archivos a encriptar.

Abre el archivo (si no está, avisa y procede al siguiente archivo)

Proporciona el nombre (sin directorio) y la longitud en bytes del archivo (LEB) en el segmento inicial de cada archivo y lo graba (en el archivo de salida)

Calcula la dimensión del arreglo ARR (dimarr)

Dimarr = LEB \ 4 - 1 donde la diagonal invertida indica división entera

Redimensiona el arreglo ARR (dimarr) usará la posición 0

Determina cuántos bytes sobraron : Sobraron = LEB mod 4 (módulo)

Lee el ARR del archivo

Prepara un entero IntSobraron

Lee uno por uno los bytes hasta que se acaba el archivo

Arma Inssobraron agregando los bytes a medida que los lee

Para k = 1 to SOBRARON

Intsobraron = intsobraron + byteleido * 256 $^{k-1}$

Observe que se agregan de derecha a izquierda: el primer byte será el menos significativo.

Ahora está en condiciones de invocar al rutina que encripta los arreglos.

Graba COMP, ARR en el archivo de salida (ambos están en memoria)

A partir de Intsobraron agrega los últimos bytes a ese segmento

Dimensióna Temp as integer

Temp = Intsobraron

For k = 1 to Sobraron

Bytecito = temp mod 256

Agrega bytecito al archivo

Temp = temp \ 256

Siguiente k

Con esto finaliza el proceso.

Se agrega la palabra clave del usuario al campo correspondiente del registro base del archivo encriptado.

Ahora se invoca la FINALIZACIÓN de la preparación del archivo que se describió en una sesión de trabajo: con estos datos se actualiza el registro base y (si procede) las palabras adicionales y se regraban ambos segmentos del archivo.

Se describe brevemente el caso de un archivo muy grande (en la versión actual esto significa que tiene más de 2 megabytes).

En este caso, se procesa el archivo en segmentos de 2 megabytes

(2^{21} = 2097152 bytes), es decir, 2^{19} = 524288 enteros de 4 bytes. Ambos números están almacenados como constantes en memoria.

Se leen, uno tras otro, segmentos de 524288 enteros, hasta que ya no haya tal. (El último cacho se procesará de otro modo).

Dimensiono el ARR, indico que es el primer cacho (sólo en el primero)

Indico que no es el último e invoco la encriptación. Cuando regresa

Si es el primer cacho, grabo COMP

Grabo ARR en el archivo.

Para el último cacho, calculo la dimensión del ARR; leo ARR

Armo IntSobrantes como se describió anteriormente

Invoco encriptación (le digo que es el último)

Cuando regresa, grabo ARR y el IntSobrantes en el archivo de salida.

8.3 Encriptar uno o varios archivos y depositarlos como un mensaje en la base de datos

Genera un número de mensaje para el (nuevo) que se agrega a la base

Prepara unA datatable, table o recordset para cada tabla

MENSAJES Y TEXTOS

Crea un registro de la tabla de MENSAJES

Le indica

- Si es nivel-2
- que se trata de archivos (en este caso es así)
- si va dirigido a varios contactos.

Lo actualiza en la base de datos (esto es optativo, pero se hace así).

Ahora procesa, uno tras otro, cada uno de los archivos de la lista de archivos a encriptar.

Abre el archivo (si no está, avisa y procede al siguiente archivo)

Proporciona el nombre (sin directorio) y la longitud en bytes del archivo (LEB) en el segmento inicial de cada archivo y lo graba (en el archivo de salida)

Calcula la dimensión del arreglo ARR (dimarr)

Dimarr = LEB \ 4 - 1 donde la diagonal invertida indica división entera

Redimensiona el arreglo ARR (dimarr) usará la posición 0

Determina cuántos bytes sobraron : Sobraron = LEB mod 4 (módulo)

Lee el ARR del archivo

Prepara un entero IntSobraron

Lee uno por uno los bytes hasta que se acaba el archivo

Arma Inssobraron agregando los bytes a medida que los lee

Para k = 1 to SOBRARON

Intsobraron = intsobraron + byteleido * 256^{k-1}

Observe que se agregan de derecha a izquierda: el primer byte será el menos significativo.

Ahora está en condiciones de invocar al rutina que encripta los arreglos.

Arma una variable de texto (string) TXTaGRABAR

Le agrega COMP : esto se hace con un copymemory (o como se llama) o byte por byte, Se agrega a TXTaGRABAR

Hace lo mismo con ARR – lo agrega a TXTaGRABAR

Finalmente, graba el Intsobraron (como texto de 4 bytes)

Graba TXTaGRABAR en la tabla TEXTOS (crea un registro, le pone el consecutivo (índice del archivo que se procesó – 1, puesto que inicia con 0)

Con esto finaliza el proceso de encriptación

Ahora se invoca la FINALIZACIÓN de la preparación del archivo que se describió en una sesión de trabajo: con estos datos se actualiza el registro de la tabla MENSAJES y (si procede) las palabras adicionales y se regraban el registro de MENSAJES.

Se describe brevemente el caso de un archivo muy grande (en la versión actual esto significa que tiene más de 2 megabytes).

En este caso, se procesa el archivo en segmentos de 2 megabytes (2^{21} bytes, 2097152 bytes), es decir, 2^{19} (524288) enteros de 4 bytes. Ambos números están almacenados como constantes en memoria.

Se leen, uno tras otro, segmentos de 524288 enteros, hasta que ya no haya tal. (El último cacho se procesará de otro modo.

Dimensiono el ARR, indico que se el primer cacho (sólo en el primero)

Indico que no es el último e invoco la encriptación. Cuando regresa

Uso una variable TXTaGRABAR de cadena de caracteres

Si es el primer cacho, le agrego COMP (como texto)

Le agrego ARR a TXTaGRABAR

Grabo esta variable en el siguiente registro de TEXTOS (de este mensaje)

Para el último cacho, calculo la dimensión del ARR; leo ARR

Armo IntSobrantes como se describió anteriormente

Invoco encriptación (le digo que es el último)

Cuando regresa, agrego ARR y el IntSobrantes a TXTaGRABAR

Y lo grabo en el siguiente registro de TEXTOS

8.4 Encriptar uno o dos textos a un archivo (encriptado)

Observación: nunca se graban más de 2 textos (y sólo son 2 si se trata de nivel-2).

La diferencia fundamental es el manejo de los bytes que sobran. En el caso de textos, para cada uno de ellos:

se registra (en memoria) la longitud del texto que llega (LTXT). A continuación se agregan al texto LTXT mod 4 bytes (con cualquier valor).

Se dimensiona el arreglo nuevo valor de LTXT \ 4.

Todo es igual que cuando se trata de archivos.

Se invoca la finalización del mensaje.

Se agrega COMP y ARR al archivo (aquí no procede el primer segmento que contiene el nombre y longitud del archivo original ni el entero de los sobrantes).

8.5 Encriptar uno o dos textos y depositarlos como mensaje en la base de datos

Se procede del mismo modo que cuando se graban en un archivo, excepto que aquí se crea el mensaje en la tabla, se encripta y se arma un texto TXTaGRABAR con COMP y ARR. Se lo agrega a la tabla TEXTOS.

Aquí no sucede que el texto sea demasiado grande (no llegarán textos de ese tamaño, puesto que alguien los tecleó).

8.6 Finalizar el archivo o mensaje en la base de datos

Al archivo encriptado se le agregan algunas opciones y especialmente, las palabras clave para abrirlo.

Cómo se agregan las palabras clave está descrito a detalle en el capítulo 11.

Se invocan las rutinas correspondientes que devuelven en el arreglo PAL() la o las palabras. Ahora se incorporan al registro base del archivo o a los campos correspondientes de la tabla MENSAJES de la base de datos.

Las restantes opciones se actualizan (en el registro base) usando una interfaz similar a la que muestra la Figura 15.

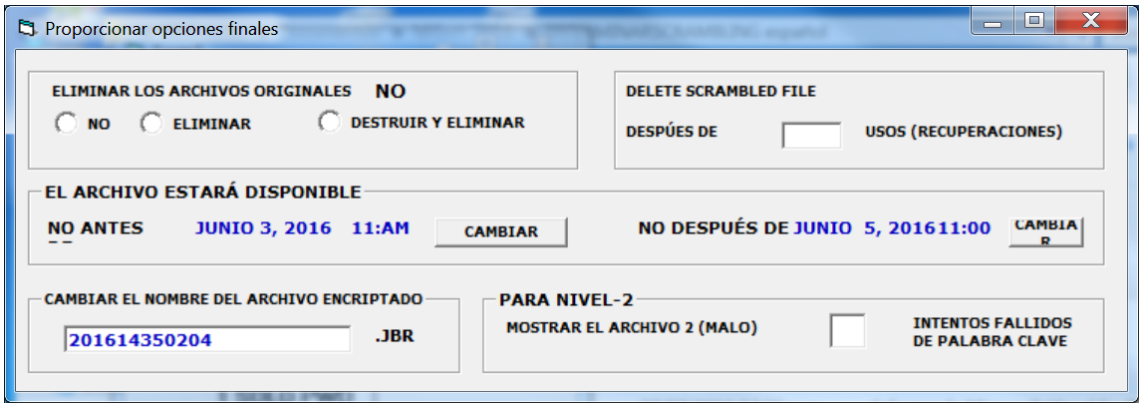


Figura 15 Las opciones activas para el archivo encriptado

Los campos referentes al cambio del nombre del archivo encriptado desaparecen si se ha enviado el mensaje a la base de datos .Análogamente, si

no se aplicó el nivel-2, no aparecerán los campos referentes exclusivamente a dicho nivel.

9 Recuperación de archivos y textos

Datos que utiliza: Le llega el archivo a recuperar o el número de mensaje, y están los datos del usuario en memoria.

Datos que produce: el (o los) archivos originales o el texto proporcionado originalmente.

Observación: en todos los casos, la primera actividad que se hace es determinar si el que está usando la función está autorizado a hacerlo. En todos los casos (de archivo o de la base de datos) se arma un arreglo de PAL(11) donde se almacenan las palabras clave recuperadas del archivo o del registro de la Tabla Mensajes.

9.1 Se presentan los diversos casos en este orden

- Se recupera el contenido de un archivo
    - El contenido original se encriptó con nivel-2
    - El contenido original no se encriptó con nivel-2
- Se recupera el contenido de un mensaje que está en la base de datos
    - El contenido original se encriptó con nivel-2
    - El contenido original no se encriptó con nivel-2

En todos los casos, hay dos situaciones: se trata de un texto o un archivo deformado.

9.2 En todas las situaciones hay que desencriptar lo que llega. Hay una rutina única que hace esto: se la invoca actdualizando antes los siguientes campos: el ARR, COMP leído del archivo o base de datos

El intSobrantes (sólo llegará algo cuando son archivos)

En el caso de textos, llegarán también la longitud del texto y los parámnetros desde y de-a. Siempre está en memoria el dato son_textos_o_no..

El programa usa COMP(2) para: interpreta y extrae los parámetros desde y de-a (sólo lo hace si se trata de archivos encriptados)-

Recupera CLAVE (2) a partir de COMP(0) (invoca la deformación)

Calcula K a partir de esta clave (2) ; intercambia comp(1) y ARR(kl)

Recupera CLAVE (1) de COMP(1) - deformando el valor que trae.

Calcula la permutación de (1,2,3) a partir de CLAVES (2)

Aplica el reordenamiento

Aplica las claves. Siempre las aplica también al INTSOBRANTES (aunque en textos no contiene información)

Ahora ARR e INTSObrantes contienen la versión recuperada-

9.3 Recuperar contenidos a partir de un archivo

9.3.1 Preparación del proceso a partir del registro base del archivo

- El registro base . De ahí se determinan
    - Si se encriptó un texto o no (es decir, archivos)
    - Si fue encriptado con nivel-2
    - Si se incluyeron varios destinatarios
    - Las palabras clave del usuario y el destinatario
    - (se almacenan como PAL(1) y PAL(2) de un arreglo dimensionado como PAL(11)
    - El número de archivos o textos que es encriptaron
- Si procede (nivel-2 o varios destinatarios)
    - Se leen las 9 palabras adicionales y se almacenan en PAL(3) a PAL(11)

9.3.2 Recuperación de los archivos deformados

Se recuperan los archivos uno por uno. Para cada uno (el caso de nivel-2 se describe más abajo) el archivo está preparado para leer “el siguiente byte”. Sin embargo, en los programas siempre se calcula esta posición cuando termina un archivo y comienza el siguiente.

A) Se lee el segmento que indica (nombre y longitud)
B) Se crea un archivo con ese nombre

C) Se calcula la dimensión del arreglo (longitud / 4 – 3), la longitud siempre es múltiplo de 4). Se lee el último entero por separado. Se le restan 3 puesto que se leen los 3 primero enteros como COMP
D) Se lee el COMP del archivo
E) Se lee el arr del archivo
F) Se lee el último entero (de ere archivo): Intsobrantes)
G) Se invoca la recuperación de arreglos deformados
H) Se graba el arreglo ARR en el nuevo archivo
I) Se procesa el InsSobrantes (ya llega recuperado)
    a. Se lo divide en 4 bytes y se agregan al archivo los que se originaron en el archivo original. El número de éstos se calcula con LOF mod 4 donde una vez más LOF es la longitud del archivo original (el que se deformó).

Se cierra el archivo creado y con ello termina el proceso de ess archivo.

Ahora el programa regresa a procesar el siguiente archivo hasta que ya no haya tales; en ese momento termina la recuperación.

9.3.3 Recuperación de los textos deformados

Las diferencias son

No pueden llegar más de 2 textos;: y sólo habrá 2 si se trata de nivel-2.

Se se trata de entregar el primer archivo, se lee “desde el principio”; en cambio se es el segundo, hay que saltar sobre el primero.

Se leen COMP- los 3 primeros enteros del archivo .

El elemento COMP(2) contiene el entero especial. Se lo interpreta lo que arroja

Los parámetros desde y de-a usados para el reordenamiento y la longitud del texto original.

Se calcula la dimensión del arreglo (primer múltipo de 4 mayor que esta longitud) es decir dimensión del arreglo = (longitud texto + 3) \ 4 donde la diagonal invertida indica la división entera.

Se lee el arr del archivo

Se recuperan los contenidos originales (se desencripta)

Se transforma el arreglo a texto. Se quitan los últimos HH bytes, donde HH = 4 – Long-texto mod 4

Se entrega ese texto al programa que solicitó la recuperación.

En el caso de nivel- para entregar el Bueno se acaba aquí el proceso.

En el caso de nivel 2, después de calcular la longitud del arreglo, se puede leer éste o calcular la posición sumando los bytes que ocupa el arreglo a la posición y sólo se procesa el segundo archivo.

9.4 Recuperar contenidos a partir de un mensaje de la base de datos

Cuando se recuperará “un mensaje” de la base de datos, los contenidos de los archivos o textos están en la tabla TEXTOS. Ya se ha leído en mensaje, de modo que ya se determinó

Si se trata de textos o de archivos

Si se encriptó el material con nivel-2: en este caso habrá dos extos o archivos; de lo contrario, sólo puede haber uno de cada uno.

También ya se ha validado al usuario, y de este modo se sabe cuál archivo se recuperará: será el archivo o texto número 1 excepto cuando es nivel 2 y la rutina de validación informó que se recuperará el segundo archivo (el malo).

9.4.1 Recuperar archivos del base de datos

La longitud y el nombre del archivo están grabados en la tabla mensajes.

También está dónde inicia el segundo archivo (el número de consecutivo de la tabla TEXTOS. Es importante señalar que puede haber más de uno de estos registros que contiene partes del archivo encriptado.

Se abre (crea) el archivo con el nombre del archivo original. Se leen

Comenzando por el primer consecutivo (0 si es el bueno, el que se determinó si es el malo) se carga a memoria el campo que contiene el texto deformado.

Si se trata del primer registro (de ese archivo)

se recupera el arreglo COMP (los 12 primeros bytes) del texto; a partir de la longitud del campo leído, se determina el tamaño del arreglo ARR; si es el último registro correspondiente a ese archivo, se leen aparte los últimos 4 bytes al entero IntSobrantes.

Procede hacer una observación técnica: para armar un arreglo de números enteros a partir de un texto en memoria (recordando que éste está almacenado con UNICODE, es decir, 2 bytes por carácter) se puede

- Invocar una función que haga precisamente eso (si el lenguaje la ofrece)
- Usar los bytes de la cadena uno por uno y armar los enteros del ARR
- Grabar la cadena en un archivo (temporal) y leer el arreglo ARR ahí mismo.

El resto de la recuperación es igual al de la recuperación de un archivo a partir del archivo encriptado y no se repite aquí la descripción.

9.4.2 Recuperar un texto de registros de la base del datos

Es útil señalar que

No puede haber más de un registro por texto: de ese modo, si hay que recuperar el bueno, se usa el consecutivo 0; para el malo, se procesa el consecutivo .1.

Se lee el registro en cuestión de la tabla TEXTOS:

Se recupera el arreglo COMP – 3 enteros y se arma el arreglo ARR

Del mismo modo que si se tratara de recuperar un archivo de la base descrito en la sección anterior.

Se procede igual que la recuperación de un texto a partir de un archivo: esto resulta en un Texto en memoria, que es el que se entrega a la función que invocó estas rutina (y que es el texto que se mostrará al usuario):

9.5 Validación del usuario como receptor del mensaje

‘ estas rutinas siempre actualizan el campo Cual_archivo_muestra

‘ vale 0 si no puede leer nada; 1 = el bueno (o todos); 2 = solo el malo´

Aquí siempre llega el arreglo PAL(11) donde están las palabras clave recuperadas y CUANTAS_PALS que indica precisamente eso: la última posición de este arreglo poblada con una palabra clave.

PAL (1) contendrá la palabra clave del usuario que creó el mensaje, y en PAL(2) la del destinatario indicado. En todos los elementos de PAL (j) se pone la palabra DEFORMADA. Las otras 9 son las “adicionales” – si no hubo, CUANTAS_PALS valdrá 2.

Primero se determina si la palabra clave correspondiente al usuario que generó el archivo coincide o no con la que tiene el archivo del usuario. Si coincide, no se invoca la función de validación. Es decir, esto resulta en

Cual_archivo_muestra = 0 seguir, o Cual_archivo_muestra = 1 y salir de la rutina de validación

Se invoca la función descrita en el capítulo siguiente

Subíndice_pal = determina_si_esta_autorizado ()

La función devuelve 0 = no proporcionó palabra válida

J cuando indició la palabra almacenada en PAL(j)

Cual_archivo_muestra = 0

If subíndice_pal < 3 --- Cual_archivo_muestra = 1

If subíndice_pal > 2 ---

Cual_archivo_muestra = 1

If nivel-2 then Cual_archivo_muestra = 2

10 El uso de palabras clave en MiScrt

10.1 Introducción

Hay dos situaciones en las cuales se usan las palabras clave

- Para incluirlas en un archivo (o en el mensaje)
- Para determinar si el que desea recuperar un material está autorizado a hacerlo.

10.2 Inclusión de palabras clave en el material encriptado

Le llega: Determina cuantas palabras se generaran:

Siempre se genera la segunda (la primera, la del usuario,) ya la puso el programa de encriptación).

Invoca la función

```
FUNCIÓN Proporciona_las_palabras
    Usa    varios_destinatarios as boolean
    ‘  devuelve el número de palabras clave que se agregaron
    ‘  siempre incluye las palabras en un arreglo PAL (11) donde no se utiliza
    ‘ la posición 0
If not varios_destinatarios then
```

10.3 Validación de la palabra clave proporcionada para procesar el archivo

El término palabra válida indica que el usuario pudo indicar una palabra clave que coincide con alguna de las que trae el archivo encriptado.

```
Función determina_si_esta_autorizado () as integer
‘  devuelve 0 si no pudo proporcionar una palabra válida
‘     el subíndice del arreglo  PAL() en el que encontró la palabra proporcionada
Dim le_atino as boolean
```

Le_atino = false

While not le_atino

With funapalabra ‘ figura 34

Limpiia todos los campos

La ofrece

Regresa

Si indico contacto, se consigue la palabra “de” y el sistema

Accesa los datos de ese contacto y de allí la toma

Si tecléo la palabra, se deforma

Se compara con las PAL

Si la encuentra, devuelve el subíndice y termina

Si no la encuentra

Suma 1 a número-de-intentos-fallidos

- determina si le deja pedir otra

wend

Determina si le deja pedir otra Hay 3 casos: Se indicó (en el mensaje deformado)

Mostrar_el_malo_despues-de-n-intentos = NINT (no 0)

Este campo nunca tendrá un valor 0 si no es nivel-2

Si numero de intentos = mostrar-el- Mostrar_el_malo_despues-de-n-intentos subíndice = 3 y fuera (mostrar el malo)

Aplicar numero-máximo-de-intentos-permitidos (por ahora no hay máximo

If excede del máximo, se niega el permiso de intentar otra.

En otro caso, se aplica una demora (el usuario espera el siguiente prompt)

Demora (campo en memoria) = $5^{num\_intentos}$

Invoca una función que causa una pausa de demora milisegundos. Esta demora es una progresión geométrica 1,5,25,125, etc. Ya 5 intentos fallidos causan una pausa (no puede hacer nada el usuario) de 3 segundos. A los 10 intentos, son aproximadamente 9,000 segundos (2.5 horas!). Es por eso que no hace falta indicar un número máximo de intentos.

Observación: Probablemente en el caso de nivel-2 se cambie este número 5 (la base de la demora) por 2, o aún por 1 (es decir, no habrá demora) porque de hecho se desea que el que intenta violar la seguridad piense que lo ha logrado). Otras consideraciones pueden hacer que cambiemos de 5 a 3 siempre.

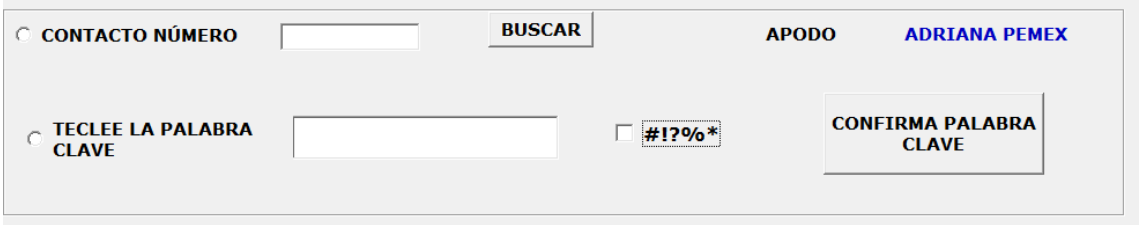


Figura 16 Interfaz utilizada para autenticarse como destinatario válido

En la Interfaz descrita en la Figura 16 el usuario puede

- Indicar un contacto (puede invocar la búsqueda del contacto deseado usando la función descrita en la siguiente sección).
    - En este caso, el sistema usa la palabra FROM (pal-de) y la almacena en Pal_a_comparar
- Teclear la palabra clave - se almacena en Pal_a_comparar

Observe que CONFIRMA indica al sistema que pruebe esa palabra clave.

10.4 Proporcionar palabras clave que se incluirán en el menaje

Siempre se solicita primero la palabra clave del destinatario PAL (2).

Se envía la forma ilustrada en la Figura 17.

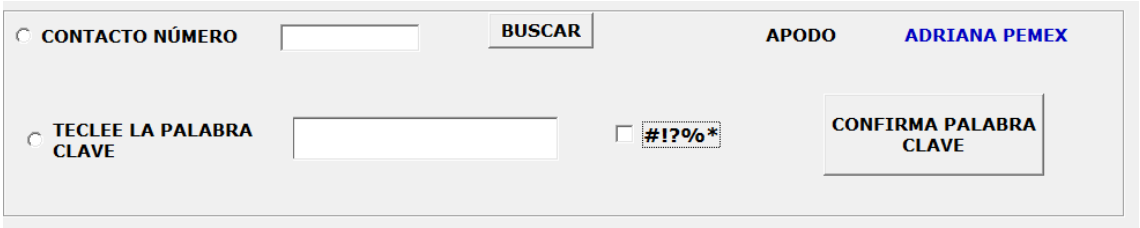


Figura 17 Interfaz para indicar la palabra clave del destinatario

Si no hay múltiples destinatarios, termina el proceso de agregar las palabras clave.

Si se dirige a varios contactos, la forma que se muestra en la figura 18 permite agregar (hasta un máximo de 9).

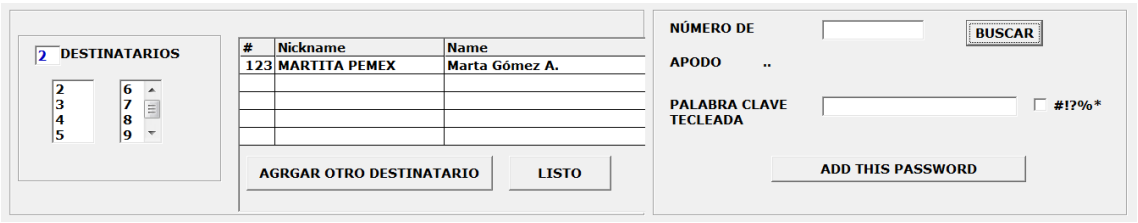


Figura 18 Forma que permite indicar varias palabras clave

Si el usuario teclea una palabra clave, se agregará a la lista un contacto con la leyenda “tecleado” (en memoria se guarda la palabra clave deformada).

Puede usar la función Buscar un contacto que ya se ha explicado y está descrita en la sección siguiente. A medida que se agregan contactos, se usan sus palabras clave Pal-TO en el arreglo PAL.

10.5 Ubicar un contacto en el directorio de contactos por su apodo

La Figura 18 ilustra le interfaz que permite encontrar un apodo por una parte del mismo. Se invoca como función y devuelve el número del contacto encontrado (0 si no encontró lo que buscaba) y el apodo. Estos datos los graba en memoria en los campos Num_contacto_encontrado y Apodo_contacto_encontrado

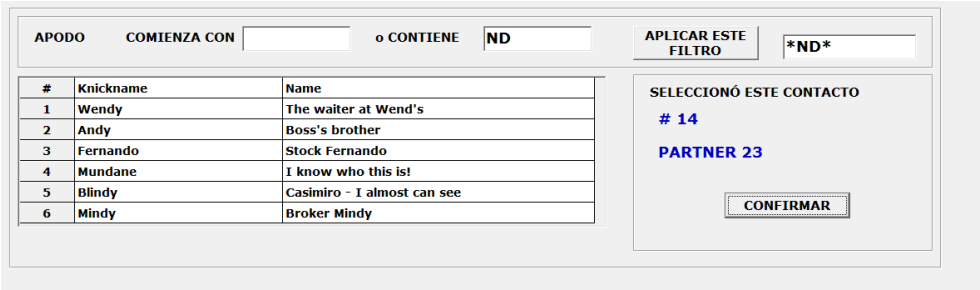


Figura 19 Encontrar un contacto

Se indica algún criterio de selección (o filtro). Se puede indicar el criterio mismo o solicitar que lo arme el programa indicando ya sea los caracteres iniciales de (o todo) el apodo o bien una cadena de caracteres que forma parte del apodo. El sistema antepondrá un asterisco en el este último caso, y siempre incluirá un asterisco al final de la cadena que se usará como criterio.

Aparece la lista de contactos y se selecciona uno, se cambia el criterio o se indica que no se encontró. Cuando se cambia el criterio, se borra alguna selección anterior, de modo que si se CONFIRMA sin haber seleccionado un contacto los campos num_contacto_encontrado y apodo_contacto_encontrado tendrán 0 y “” respectivamente.

10.6 Generación de N palabras espurias

Se ha mencionado numerosas veces que cuando se trata de una encriptación que usa el nivel-2 se generan palabras clave tales que si el usuario proporciona una de ellas, el sistema le muestra el archivo malo. Si este dato llega con 8, indica que ya se incluyó una de las 9 (la palabra-mal de un contacto).

La rutina que determina otra palabra (una tras otra a partir de la indicada como “la del destinatario”) no se documenta: es muy compleja y además, secreta.

Las palabras se generan a partir del reordenar ciertos elementos. Se determina la longitud LPALD de la palabra clave del destinatario. Se generarán palabras de LPALD -1 u LPALD – 2 caracteres, siempre que estos números sean mayores que 5. Se construye una palabra y se usan permutaciones (de a grupos de caracteres) y cambios de mayúscula por minúscula.

Estas palabras se agregan al arreglo PAL () pero ya deformadas.

10.7 Deformación y recuperación de palabras clave

La deformación es casi idéntica a la de un texto tecleado, excepto que se agregan los caracteres que completan los 12 caracteres (máxima longitud de una palabra clave). De ese modo se graban como deformado 24 caracteres.

Para la recuperación, se procede como en el caso de textos, excepto que ahora se eliminan del texto recuperado los caracteres que se agregaron para completar los 12.

11 Algoritmos de deformación de datos en MiScrt

11.1 Introducción

En este capítulo se describirán los algoritmos utilizados por MiScrt para deformar y almacenar textos. Un aspecto fundamental, como se ha mencionado, es que provea un nivel de seguridad adecuado, concepto que se definió anteriormente pero cuyo significado se reproduce aquí para conveniencia del lector.

El término deformar un archivo en este trabajo significa: hacer que el archivo (original) no se pueda interpretar sin *esfuerzos adicionales* a lo que implicaría usar un programa similar al que lo creó. Por ejemplo, si se creó un archivo con un editor, en general se puede leer con ese mismo editor (u otros, en algunos casos). Si se deforma este archivo, el programa no podrá recuperarlo. Por ejemplo, si a un archivo creado con el programa MS-WORD se le modifican los primeros bytes, será ilegible para dicho paquete. Lo mismo sucede con muchos archivos multimedia.

Se mencionó el concepto de esfuerzos adicionales. Esto conduce a uno de los aspectos más importantes de la encriptación (o protección) de archivos: **el nivel de seguridad**. En esta tesis se usará esta definición de este concepto: es el costo que se tendría que invertir para violar los procedimientos de seguridad incluidos en la deformación de un archivo. Esto conducirá a la determinación del nivel de seguridad necesario para un paquete, en este caso el MiScrt. Se trata de valorar el beneficio que alguien podría obtener si consiguiera obtener el material que se protegió, lo que en la literatura de denomina “costo del contenido”. Un nivel de seguridad adecuado debe garantizar que este beneficio sea considerablemente menor al del costo de violación de la protección.

Una vez más, queda otro término por definir: la cuantificación de “considerablemente”. En MiScrt se decidió que si este término se interpreta como “100 veces más”, se consideraría adecuada la protección.

Una estrategia no evidente pero aplicada aquí, es que para lograr esto se puede proceder de dos modos: incluir dispositivos de deformación suficiente robustos, es decir, aumentar el costo de la violación, o disminuir el costo de los contenidos. Esto último se hace limitando el tipo de material que se deformará con MiScrt. En su diseño se incluyeron ambas estrategias: el paquete no ofrece seguridad para mensajes secretos de gran valor, cualquiera que se la aplicación de la cuantificación de dicho valor.

11.2 Algunas técnicas de encriptación

Procede una advertencia sobre esta tesis: FURHT et. Al. (2005), refiriéndose a dicho libro, señala que 'it is important to point out "that this book is not about cryptography, but about applying cryptography to achieve the desired multimedia security and protection compare the cost of the multimedia information to be protected versus the cost of the protection itself.'

Como se ve, la definición adoptada del nivel de seguridad adecuada es la m." Aquí aplica el mismo comentario pero a una tesis, en lugar del libro. En el mismo libro Furht opina que "Deciding upon what level of security is needed is harder than it looks. To identify an optimal security level, we have to carefully compare the cost of the multimedia information to be protected versus the cost of the protection itself." Observe que es la misma que utilizamos nosotros,y la que se usa en muchos productos (y cursos de protección o encriptación de datos).

Por lo tanto no se expondrán diversas técnicas de encriptación: son temas matemáticos aplicados a la computación. A efectos de incluir protección de ese tipo en MiScrt no es necesario conocer todos ellos.

Sólo se incluyen aquí algunos comentarios sobre características de los procesos de encriptación. El primero es que sea "reversible", es decir, debe poder ser posible recuperar el archivo original. Para esto, sólo se usan funciones *inyectables*, lo que significa en términos matemáticos que tienen inversa o que son isomorfismos. En otras palabras, si $x \neq y$, $f(x) \neq f(y)$. En

algunos algoritmos (como el que se usó aquí) se exige una condición mucho más estricta: $f^{-1}(x) = f(x)$ o $f(f(x)) = x$.

Ante la disyuntiva de adoptar una encriptación de las muchas ofrecidas y de crear una propia, se seleccionó esta última. El motivo se basa precisamente en una vulnerabilidad que tiene el MiScrt, ausente en muchos otros productos.

Se trata de qué datos tiene que proporcionar el que desea recobrar el archivo encriptado. En la mayoría de los productos, se usan 2 claves de deformación (números enteros de diversa longitud) que se combinan de diversos modos para aplicarlos a la cadena de bytes que se desea deformar. Una de las claves está en el archivo, mientras que la otra se proporciona para usarlo.

Se puede usar la palabra clave – misma que servirá de base para calcular dicha clave, y por ende el usuario no tiene que conocer ni proporcionar la clave, más que en esta forma indirecta.

Esto no funciona en MiScrt porque los archivos no están protegidos por una palabra, sino varias: dos en el caso de un solo destinatario, porque siempre se puede usar la palabra clave del que creó el archivo, o más si el mensaje se dirigió a más de un destinatario (o contacto).

Esto obligó a diseñar un modo de “esconder” los parámetros utilizados para deformar el archivo que se describirá porque forma parte del algoritmo.

11.3 Los números enteros que usan los procesos de encriptación en MiScrt

EL proceso que se describe siempre se aplica a un arreglo de enteros (long). La longitud depende del sistema operativo. Cuando inicia el programa, se determina si el sistema operativo es de 64 bits: en ese caso los enteros lont serán integer64 (8 bytes). Si es de 32 bits, se usan enteros long de 4 bytes. Observe que esto es transparente para el usuario (ni siquiera sabe cuáles se utilizaron). El programa está parametrizado para usar los números resultantes en todos los diversos procesos en los que intervienen.

11.4 Elementos básicos utilizados (encriptación por 2 claves)

Se puede decir que la mayoría de los procesos que encriptan datos usan el concepto de: aplicar una clave de deformación (que es un número entero) a los bytes que llegan. Esto se hace agrupando ésta en números de la longitud de la clave. Sea ARR el arreglo de números long resultante.

La encriptación consiste en aplicar a cada elemento Arr(j() la clave (CL) mediante la operación XOR (el OR exlusivo, o disyunción exclusiva). La propiedad que se aprovecha es que ARR(j) xor CL xor CL = ARR (j), es decir la función de encriptación cumple la condición descrita anteriormente: la función inversa coincide con la función misma.

De hecho, en la mayoría de los algoritmos usados, incluyendo los que se usaron aquí, se usan 2 claves (en ocasiones aún más) para aumentar el nivel de protección, porque ahora alguien tendría que “adivinar” dos números en lugar de uno solo. Una forma de ataque a los mecanismos de protección (muy utilizado en la práctica de los hackers, especialmente en películas donde se despliega este modo de ataque) es generar consecutivamente “todos” los números y probarlos para ver si son los utilizados. Es como ir a una puerta con un enorme manojo de llaves (muchos millones en el mejor de los casos) y probar una por una hasta que se encuentre la que abre la puerta. Al haber dos, tendría que probar dos cerraduras (cada una con una llave diferente):

Sean CL1 y CL2 estas dos claves. Sea CL3 = CL1 xor CL2.

Entonces ARR(j) xor CL1 xor CL2 = ARR (j) xor CL3. En el tema de la puerta, hay dos cerraduras pero hay una tercera que es suficiente para abrir la puerta: se dividió por dos el trabajo del atacante.

De ese modo, en el algoritmo del MiScrt (y naturalmente en el de muchos otros) las claves no se aplican a los mismos números. La rueda de Feisel (SSSS) es quizá el modo más popular de determinar a cuáles elementos del arreglo se le aplican cada una de las llaves. En las sección siguiente se presenta el algoritmo que hace esto.

11.5 Cómo se aplican las claves

Para determinar cuál de las claves se aplica a cada elemento del arreglo, se creó un algoritmo. El modo evidente de hacer esto sería

Si j es par, ARR(j) xor CL1 ; si es impar, ARR(j) xor CL1

Vale la pena señalar aquí que uno de los medios de violar un procedimiento de seguridad (o código) es invocarlo repetidamente e intentar descubrir un patrón. Para evitar esto, en MiScrt se usaron varios modos de cambiar constantemente los algoritmos para dificultar en lo posible este ataque. Observe que se usa un ejemplo para aclarar algunas operaciones.

Se crea un arreglo CL (0) – no se usa el elemento con índice 0. Los valores de CL(1) y CL(2) son las dos claves de encriptación CL1 y CL2 generadas anteriormente. Se completa el arreglo con CL(3) = CL1 XOR CL2.

A continuación se genera, en base a ciertos dígitos de CL2, una permutación de los números (1,2,3) . Sea (en este ejemplo) (2,3,1).

Se procede a “reacomodar” los valores del arregl CL:

CL(1) = CL(2), CL(2) = CL(3) y CL(3) = CL(1) donde naturalmente se usan los campos intermedios necesarios.

Ahora se aplican las claves del siguiente modo:

ARR(j) = ARR(j) xor CL(j mod 3 – 1) donde “mod3” indica el módulo.

Se aplicarían las claves

ARR(0) : CL(1) = CL2

ARR(1) = CL(2) = CL1 xor CL2

ARR(2) = CL(3) = CL1

Y en ARR(3) comienza otra vez el ciclo (igual que ARR(0))

11.6 Qué se encripta en MiScrt

Es importante señalar un aspecto de programación. Esta encriptación se invoca desde varias rutinas que necesitan encriptar un arreglo de enteros. Por lo tanto se la programa como bloque de instrucciones (o “begin”) o en términos de VB, una subrutina. La rutina encripta arreglos provenientes de un archivo o de un texto. Se señalará la diferencia en su explicación, pero de hecho es que para archivos, además del ARR (el arreglo de enteros) hay que encriptar otro entero que llamaremos IntSobrantes.

El encriptado se efectúa sobre el mismo arreglo (es decir, no se crea otro).

La rutina usa el campo (global)

Es-Texto (Falso indica que se trata de un archivo)

Esta variable se establece cuando se prepara la encriptación solicitada (en la rutina que la invoca).

Además usa 3 parámetros:

- es_el_primer_cacho as boolean. Esto sirve para determinar si hay que efectuar ciertas operaciones que sólo se hacen con la primera parte del arreglo. Como es verá en el capítulo de encriptación de textos y archivos, si el archivo es muy grande, se encripta de a segmentos (en el programa se llaman cachos).
- Es-el-ultimo-cacho as boolean: si no es el último, no se encriptará el entero IntSobrantes que llega además del arreglo
- Longitud_del_texto_original (si se trata de archivos, tendrá el valor 0)

11.7 La encriptación del arreglo.

Hay (disponible) un arreglo de 3 números enteros que llamaremos COMP. De ese modo nos referiremos a COMP(0), COMP(1) y COMP(2).

Se presenta el algoritmo en una serie de pasos, pero antes hay que señalar que se usa un esquema de reordenamiento de elementos (se intercambian posiciones del arreglo encriptado). Para ellos se usan dos parámetros,

Es importante señalar que lo que se describe sólo aplica si

es_el_primer_cacho = Verdadero (el único también llega con verdadero)

En cada paso, se indica si se ejecuta o no si no se trata del primer cacho. Pero se puede explicar esto de modo “no programación”: si no es el primer cacho, sólo se ejecuta el PASO 5.

11.7.1 PASO 1: se generan dos claves enteras CL1 y CL2. Esto se hace, para cada uno

If NO ES EL PRIMER CACHO no se ejecuta este paso

- Se Generan 2 números aleatorios
- se multiplican estos números;
- se aplica una operación adicional que usa un parámetro diferente para las dos claves con lo que se obtiene la clave.

Se ha mencionado que el MiScrt usa números enteros “long”. La longitud es estos enteros puede ser Int32 (4 bytes) o Int64 (8 bytes). Cuando se ejecuta el programa, obtiene el número de bits que usa el sistema operativo, puesto que ése será el que proporciona el lenguaje de programación.

De ese modo, en la generación de claves, hay dos algoritmos (uno para 32 bits, o para 64) puesto que si se pueden usar números de 8 bytes, la protección será mucho mayor (de hecho, el doble de robusta). La diferencia está en el uso de números aleatorios.

Para 32 bits, recordando que si se usan con signo, no almacenan números mayores a $2^{31} - 1$ se usan enteros obtenidos de Rn1 * 10^5 y Rn2 * 10^4 donde Rn indica un número aleatorio. En cambio si se trata de enteros de 64 bits, el factor es Rn1 * 10^{10} y Rn2 * 10^9

Al usar

Rn1 * 100,000 el número más grande sería 99,999

Y Rn2 * 10,000 el número más grande sería 9,999

Si se multiplican 2 números cómo éste, el resultado máximo es

99,999 * 9,999 < 10^9 que siempre es menor que $2^{31} – 1$

Para enteros de 64 bits, el argumento es similar.

La otra operación depende del resultado de este producto: en ciertas condiciones se le suma otro número aleatorio, diferente para las dos claves.

11.7.2 Paso 2: se generan los parámetros para el reordenamiento

If NO ES EL PRIMER CACHO no se ejecuta este paso

Se generan 2 números enteros para las variables desde y de_a

Los números deben satisfacer $0 < x < 6$

11.7.3 Paso 3. Se genera un número entero especial

If NO ES EL PRIMER CACHO no se ejecuta este paso

Este número contiene

- La longitud del texto original (que le llegó como parámetro)
- Los valores de las variables desde y de-a generadas previamente.

La operación es (aproximadamente)

Lespecial = longitud del texto * 1000 (no habrá textos de megabytes, puesto que fueron tecleados)

Lespecial = desde * 100 + de-a * 10 + Digito (de 0 a 9)

11.7.4 Paso 4. Se arma el arreglo COMP

If NO ES EL PRIMER CACHO no se ejecuta este paso

COMP (0) = Deformacion (CL1)

COMP (1) = Deformacion (CL2)

COMP (2) = Lespecial

Deformación es una función que deforma un número entero de un modo secreto, pero que tiene la característica de

Deformacion (Deformacion (CL1))= CL1

Es decir, coincide con la función inversa.

11.7.5 Paso 5. Se aplican las claves al arreglo y al entero IntSobrantes

Esto se explicó en la sección correspondiente. Si se trata del último cacho (o del único) se le aplican ambas claves al entero IntSobrantes (que llega armado del programa invocador)

11.7.6 Paso 5. Se aplica el reordenamiento

If NO ES EL PRIMER CACHO no se ejecuta este paso

Comenzando por j = desde se intercambian

ARR(j) = ARR(k) ARR(K) = (J) donde k = desde + de_a

Y ahí sucesivamente j = j+1 k = k +1

Pero no se intercambia nunca un elemento dos veces.

Ejemplo Sean desde = 2 y de_a = 3

Se intercambian los elementos (señalados como pares de enteros)

(2,5) (3,6) (4,7) (6,9) etc.

No se intercambia nunca un elemento dos veces, de modo que se “brinca” el elemento que ya se ha intercambiado por otro (en este caso, ARR(5).

Este operación, una vez más, es su propia inversa

11.7.7 Paso 6. Se intercambian CL1 y un elemento del arreglo

If NO ES EL PRIMER CACHO no se ejecuta este paso

Se calcula un entero DONDE a partir de los primeros elementos del arreglo, y se obtiene un entero que será

Mayor que el último utilizado en la operación anterior

No mayor que la dimensión del arreglo.

Se procede a intercambiar COMP(0) y ARR (donde).

Con esto concluye la encriptación del arreglo.

Si se trata del último “cacho” o segmento del archivo, se aplican las claves al entero IntSobrantes

Intsobrantes = Intsobrantes xor CL1 xor Cl2

12 Algoritmos de recuperación de datos en MiScrt

12.1 Introducción

En este capítulo se describirán los algoritmos utilizados por MiScrt para recuperar el contenido original a partir de los archivos encriptados. Éstos pueden estar como registros de la base de datos o ser archivos planos en algún dispositivo de almacenamiento.

Siempre “llegan” en memoria a estas rutinas (se usan los términos especificados en el capítulo anterior):

- En ARR, el arreglo leído o armado a partir del material encriptado
- En COMP el arreglo especial
- En el caso de que el material encriptado fue un archivo (y no un texto) también llega IntSobrantes.
- Es-el-primer-cacho (verdadero o falso)
- Es-el-ultimo-cacho (verdadero o falso)

Las rutinas siempre devuelven el ARR “desencriptado” y cuando procede, el IntSobrantes también recuperado.

12.2 Recuperar el contenido original a partir de los arreglos

Se procede esencialmente del mismo modo que cuando se deforman los arreglos, pero en sentido contrario.

Paso1: recuperación de parámetros

Sólo se ejecuta si es el primer cacho.

Se recupera el Intespecial de Comp(2)

Se interpreta este entero: recupera los parámetros desde y de-a y la longitud del archivo original. Se graban estos datos en los campos respectivos de memoria.

Se recupera CL(2) la segunda clave de encriptación de COMP(0) (aplicando la deformación de claves a la inversa)

A partir de CL(2) se determinan

La posición K y se intercambian ARR(k) y COMP(1)

La permutación utilizada para encriptar de la terna (1,2,3)

A continuación se aplica el reordenamiento (que es igual a su inversa)

Paso 2: Se aplican las claves del mismo modo (con la misma rutina) que para encriptar el arreglo. Se apican también a IntSobrantes (de hecho sólo si se trata de archivos encriptados (los textos no tienen tal entero) y cuando es el último cacho (los demás no usan este campo).

Con esto finaliza el proceso de desencriptación de los arreglos.

13 Ciertos usos resultantes del diseño

Miscrt fue creado para satisfacer los usos potenciales que se agregaron a la de perimitir el intercambio de materiales protegidos entre personas o grupos. Se trata de que sólo el destinatario del mensaje pueda aprovechar el contenido, para lo cual lo recuperará a partir de la versión deformada que le llega. En otras palabras, se intenta prevenir en lo posible que el material sea visto por personas no autorizadas.

Con la funcionalidad incluida en MiScrt, además de los usos normales de software como éste (encriptar material y envíalo a alguien), un usuario pude aprovecharlo para ciertas tareas que incluyen las que se describen a continuación.

- Almacenar en forma confidencial todas sus palabras clave (de todos sus sitios, cuentas bancarias, adscripciones, etc.) Para cada una, indicará un apodo y un nombre que le permita saber a qué se aplica esa palabra clave, preferiblemente de modo de dificultar esto para otros (por si el archivo cayera en manos de otros). Observe que el MiScrt ofrece la recuperación de las palabras clave encriptadas, pero no sin solicitar la palabra clave fuerte del usuario.
- Respaldar y protegersu trabajo diario o del período que desee. Los archivos se deforman resultando en un archivo único (similar a un zip). Se puede indicar que los archivos originales sean eliminados del disco. Podrá hacer esto depositando el archivo encriptado en cualquier dispositivo.
- El directorio de contactos se diseñó para que uno pueda olvidar las palabras clave acordadas con terceros (sus contactos). Esto es especialmente útil cuando se deben enviar distintos mensajes (o aún el mismo) a un conjunto numeroso de contactos, por ejemplo como actividad de una organización, banco, casa de bolsa o similar. Con el directorio ni el organismo ni el cliente tendrá que recordar la palabra clave acordada para sus informes. El cliente tendrá MiScrt (como app o

app móvil) y su propio directorio de contactos (esto último es optativo, pero conveniente).

- Un uso un tanto especial para ciertos usuarios es que mucos servicios de correo electrónico rechazan archivos adjuntos del tipo “.exe” (puesto que pueden contener software maligno). Al cambiar la extensión y hacer que el contenido no sea interpretable, esto ya no sucede con los archivos encriptados.

14 Conclusiones

El objetivo del Proyecto de investigación era desarrollar un product de software que ofreciera ciertas funcionalidades pero con algunas características específicas, en especial la facilidad de uso que permitiera a usuarios aprovechar las funciones sin conocimiento previo y no tener que recordar aspectos del mismo para iniciar una sesión.

Se partió de ciertos usos que se consideraron deseables en una programa como éste, los que se transformaron en objetivos y a partir de estos, funciones que se podrían hacer con el programa. Asimismo, el programa se podría ejecutar en diversos sistemas operativos, y en distintos dispositivos, tales como teléfonos móviles o tabletas, además de en una computadora. A pesar de que fue desarrollado en idioma español, habrá una versión en inglés. El proyecto no se hizo para obtener un producto que se vendería o regalaría, sino para ilustrar un software que satisficiera todas las necesidades que se le impusieron.

15 Referencias

Carey G S., Widel M., Truong. The use of checksums to ensure data integrity in the healthcare industry. Pharmaceutical Programming. Volume 5, Issue 1-2, pp. 3841, 2012

Durán, R; Hernández L, Muñoz J. (2005). El criptosistema RSA. Ra-MA

Furht, B., Muharemagic E., Socek, D. Multimedia Encryption and Watermarking. Springer, 2005. pp. 24-27

Gallo. G; Coello I.; Parrondo, F.; Sánchez , H. (2003). La protección de datos personales: Soluciones en entornos Microsoft. Microsoft Ibérica.

Granados, G. (2006). Introducción a la Criptografía. Revista Digital Universitaria, vol. 7, Número 7. ISSN: 1067-6079

Lu, C. S. (2005). Multimedia security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property, Idea Group Publishing.

Kaur M., Singh H. A review of cloud computing security issues. International Journal of Grid Distribution Computing, Vol. 8. No. 5, pp. 215-222. ISSN. 2005-4262, 2015

Kumar D, Suneetha C., Chandrasekhar. A block cipher using rotation and logical XOR operations. International Journal of Computer Sciences Issues, Vol. 8, Issue 6, No. 1 November,2011.. ISSN1694-0814

Piper. F; Murphy, S. (2002). Cryptography: A very short Introduction. Oxford University Press.

¿Qué es Firebird?. Firebird Manual . http://www.firebirdmanual.com/firebird/es/firebird-manual/2/-que-es-firebird-/8 consultada al 30 de abril de 2016

Pino. C. (2002) Introducción a la Criptografía, 2da. Edición, Ra-Ma.

Ramio, J. (2006). Libro Electrónico de Seguridad Informática y Criptografía, Versión 4.1 de marzo de 2006. http://www.criptored.upm.es/guiateoria/gt_m001a.htm

Rout H., Kishore B. Pros and cons of cryptography, steganography and perturbation techniques. IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), e-ISSN: 2278-2834, PP 76-81, 2014

Smartphone OS Market Share, 2015 Q2. International Data Corportation. http://www.idc.com/prodserv/smartphone-os-market-share.jsp consultada al 5 de junio de 2016

Schneier B., Seidel K., Vijayakumar S. A Worldwide Survey of Encryption Products. Berkman Center Research Publication No. 2016-2. Available at SSRN: http://ssrn.com/abstract=2731160 or http://dx.doi.org/10.2139/ssrn.2731160 February 11, 2016

Silva, J.; Morales, G. Desarrollo de una plataforma de seguridad en dispositivos móviles de comunicación, ¿Necesidad o Paranoia?. Revista Digital Universitaria. 1 de agosto 2012, volumen 13, número 8. ISSN: 1067-6079.

Top 7 Desktop OSs from June 2015 to June 2016.

http://gs.statcounter.com/#desktop-os-ww-monthly-201506-201606-bar consulta al 29 de Junio 2016

16 Anexos

16.1 CD

17 Contenido del CD

- Código fuente
- Instalador de la base de datos FirebirdSQL