UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

INSTITUTO DE INVESTIGACIÓN EN COMUNICACIÓN
ÓPTICA DE LA UASLP

ENCRIPTACIÓN DE IMÁGENES Y SU USO EN ALGUNOS SISTEMAS DE COMUNICACIÓN

TESIS

QUE PARA OBTENER EL GRADO DE MAESTRO EN CIENCIAS
ESPECIALIDAD: INSTRUMENTACIÓN ELECTR-ÓPTICA

PRESENTA

ING. FERMÍN MORALES ROBLES

ASESORES

DRA. MARCELA MEJÍA CARLOS
DR. RAUL E. BALDERAS NAVARRO

San Luis Potosí, S.L.P. Marzo 2007

UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

INSTITUTO DE INVESTIGACIÓN EN COMUNICACIÓN
ÓPTICA DE LA UASLP

ENCRIPTACIÓN DE IMÁGENES Y SU USO EN ALGUNOS SISTEMAS DE COMUNICACIÓN

TESIS

QUE PARA OBTENER EL GRADO DE MAESTRO EN CIENCIAS
ESPECIALIDAD: INSTRUMENTACIÓN ELECTR-ÓPTICA

PRESENTA

ING. FERMÍN MORALES ROBLES

ASESORES

DRA. MARCELA MEJÍA CARLOS
DR. RAUL E. BALDERAS NAVARRO

San Luis Potosí, S.L.P. Marzo 2007

A Dios todo poderoso porque me ilumino siempre
En mis momentos de adversidad y estuvo presente
En mis momentos de gloria.

A mis padres por su cariño, confianza y apoyo
Incondicional en todas las etapas de mi vida.

A mis hermanos.

Reconocimientos

A mi asesora la Dra. Marcela Mejía Carlos por compartir sus conocimientos, por su gran ayuda y apoyo para la realización de esta tesis.

Al Dr. Raúl E. Balderas Navarro por su ayuda, consejos y sugerencias.

A mis compañeros y amigos que siempre estuvieron dispuestos a dedicar un poco de su tiempo para ayudarme, en especial a Carmen.

A todo el personal del Instituto de Investigación en Comunicación Óptica.

A CONACYT por la beca de posgrado No.

Este trabajo fue realizado en el Laboratorio de Comunicaciones del Instituto de Investigación en Comunicación Óptica.

Encriptación de imágenes y su uso en algunos sistemas de comunicación

Ing. Fermín Morales Robles
Instituto de Investigación en Comunicación Óptica

30 de marzo de 2007

Resumen

El objetivo de la tesis es diseñar e implementar un sistema que sea capaz de encriptar datos en tiempo real y transmitir la información encriptada a través de un medio de comunicación, desencriptar y recuperar los datos originales.

Índice general

Índice de figuras

CAPÍTULO 1

Introducción

Actualmente el envío de información utilizando cualquier sistema de comunicación esta expuesto al ataque que permite el acceso a todo los datos. Afortunadamente existen medidas de seguridad para evitar todo acceso no autorizado, una buena alternativa es la encriptación.

La encriptación es el proceso mediante el cual la información ó *texto plano* es cifrado de forma que el resultado sea ilegible a menos que se conozcan los datos necesarios para su interpretación. Es una medida de seguridad utilizada para que al momento de almacenar o transmitir información no pueda ser obtenida con facilidad por terceros. Al proceso inverso se le conoce como desencriptación.

Hay dos clases de algorítmos de encriptación y desencriptación: los algorítmos simétricos (o de llave secreta) y los asimétricos (o de llave pública). La diferencia es que los simétricos usan la misma llave para la encriptación y desencriptación o la llave de desencriptación es fácilmente derivada de la llave de encriptación. Mientras que los algorítmos asimétricos usan una llave diferente para encriptación y desencriptación y la llave de desencriptación no puede ser derivada de la llave de encriptación.

Algunos de los usos más comunes de la encriptación son el almacenamiento y transmisión de información sensible como contraseñas, números de identificación legal, números de tarjetas de crédito y conversaciones privadas. Para el desarrollo de la tesis se utiliza un algoritmo de encriptación basado en autómatas celulares, esto como base para el envío de información encriptada

a través de algunos sistemas de comunicación.

En el capítulo 2 se define y se muestra un sistema de Encriptación por Sincronización de Autómatas Celulares (ESAC). El cual es un cifrador simétrico que encripta bloques de $2k-1$ bits, utilizando para cada bloque una llave generada por una semilla inicial. El ESAC esta formado por un generador pseudoaleatorio y las funciones ϕ y ψ, las cuales realizan la encriptación y la desencriptación respectivamente.

Con el fin de probar la versatilidad del ESAC en el capítulo 3 se presentan las siguientes aplicaciones.

Envío de datos en la red GSM/GPRS

Debido al incremento en el uso de los sistemas celulares y a que existen diversas tecnologías que permiten el envío de una gran cantidad de datos a través de la red celular tales como: texto, voz, imágenes, música, etc. Es importante contar con sistemas de seguridad que permitan el envío de tales datos de forma segura y confiable. Como se menciono anteriormente la encriptación es una buena alternativa.

Se presenta un sistema capaz de enviar información encriptada a través de la red GSM. Se introducen las características básicas de los sistemas celulares y el envío de información utilizando el servicio de mensajes cortos (SMS).

El algoritmo de encriptación y desencriptación es implementado en el módulo GT48 de Sony - Ericsson. El módulo GT48 es una terminal de control GSM/GPRS inteligente que incluye todo lo necesario para su uso en telemetría. Cuenta con su propio lector de tarjeta SIM que permite acceder a todos los servicios ofrecidos por el operador.

El GT4× puede ser manejado mediante el uso de comandos AT ó mediante tareas programadas por el usuario, para esto cuenta con su propio software que incluye todas las librerias necesarias para su aplicación. La programación esta basada en el lenguaje C.

Autentificación de huellas digitales y su validación

Entre todas las técnicas biométricas, la autentificación basada en el uso de huellas digitales es la mas utilizada. Algunas de las razones principales son que debido al uso de nuevas tecnologías es posible contar con un gran número de sensores baratos y confiables para la captura de huellas digitales y además la identificación basada en huellas digitales tienes errores reportados con menos del 1 %.

Se hace una decripción de las etapas ó módulos en los que se divide

el sistema de autentificación por huellas digitales. El sistema es capaz de adquirir la huella digital de un usuario, previamente registrado en una base datos, encriptar la imagen y enviarla a través de un medio de comunicación inalámbrico hacia una estación base donde se desencripta la imagen y se realiza la verificación y autentificación de la huella.

En el capítulo 4 Se realizan diversas pruebas con el fin de determinar la confiabilidad en el envío y recepción de las huellas encriptadas en el sistema de autentificación biométrica.

Con el fin de simular el ruido que afecta a la información cuando se utiliza cualquier medio de comunicación, se añade ruido a los datos encriptados. Se utilizan dos tipos de ruido: aleatorio y gaussiano y finalmente se presentan los resultados obtenidos.

CAPÍTULO 2

Sistemas de Encriptación

Actualmente las aplicaciones de transmisión de información en tiempo real llegan a ser vulnerables a ataques que permiten el acceso a todos los datos. Una alternativa a este problema es la encriptación [5].

La encriptación es el proceso de disfrazar un mensaje de tal manera que esconda su contenido. Al mensaje disfrazado se le conoce como texto cifrado (ciphertext) o texto encriptado. Al proceso inverso se le conoce como desencriptación [11].

Hay dos clases de algoritmos de encriptación y desencriptación: algoritmo simétrico o de llave secreta y algoritmo asimétrico o de llave pública. La diferencia es que los algoritmos simétricos usan la misma llave para encriptación y desencriptación o la llave de desencriptación es fácilmente derivable de la llave de encriptación. Mientras que los algoritmos asimétricos usan una llave diferente para encriptación y desencriptación y la llave de desencriptación no puede ser derivada de la llave de encriptación [11]. El propósito básico de la encriptación es proteger la información de accesos no autorizados.

2.1 ESAC

En el desarrollo de la tesis se utiliza un sistema de Encriptación por Sincronización de Autómatas Celulares (ESAC) [2]. El ESAC es un cifrador simétrico que encripta bloques de $2k-1$ bits, utilizando para cada bloque una subllave generada a partir de una llave inicial. El fenómeno de sincro-

nización en autómatas celulares acoplados, es usado para implementar dos familias de permutaciones de palabras binarias y una transformación de palabras binarias, que son los elementos básicos del sistema ESAC. Las familias de permutaciones son utilizadas para la encriptación y desencriptación y la transformación de palabras binarias es utilizada en el generador de llaves.

El sistema ESAC (Encriptación por Sincronización de Autómatas celulares) encripta y desencripta un mensaje dividido en una secuencia de bloques de 15 bits, utilizando una llave diferente para cada bloque. La misma llave debe utilizarse en la encriptación y en la recuperación del mensaje original.

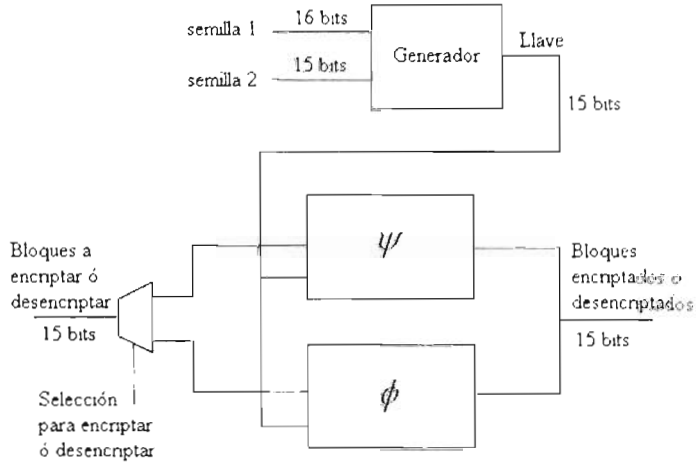

Figura 2.1: ESAC

El ESAC esta formado por un generador pseudoaleatorio y las funciones ϕ y ψ, las cuales realizan la encriptación y la desencriptación respectivamente. En la figura 2.1 se muestra un diagrama a bloques del ESAC. El generador pseudoaleatorio contiene las semillas iniciales con las cuales se genera una llave para cada bloque de 15 bits que se quiere encriptar ó desencriptar. El demultiplexor permite seleccionar la opción de encriptar ó desencriptar.

2.1.1 Generador de Llaves del ESAC

La parte principal de un sistema de encriptación es el generador de llaves o generador de secuencias pseudoaleatorias. Las secuencias en criptografía

deben seguir los siguientes criterios: (1) buenas propiedades estadísticas, (2) computacionalmente impredecibles y (3) fáciles de realizar mediante sistemas electrónicos de alta velocidad [2].

Un generador aleatorio de bits es un dispositivo o algoritmo el cual produce una secuencia de dígitos binarios con igual probabilidad de aparición, estadísticamente independientes y que requiere de una fuente natural de aleatoriedad. En el caso óptimo, los generadores aleatorios están basados en verdaderas fuentes físicas de aleatoriedad que no son predecibles. Tales fuentes pueden incluir el ruido de un semiconductor, los últimos bits significativos de una entrada de audio, etc. Sin embargo estos métodos de generación de bits aleatorios es un procedimiento ineficiente en muchas aplicaciones practicas. El problema puede ser aminorado si se sustituye un generador de bits aleatorio por un generador de bits pseudoaleatorio [2].

Los generadores pseudoaleatorios producen secuencias finitas y periódicas de números empleando operaciones aritméticas y/o lógicas y lo único que se puede conseguir es que estas secuencias sean las mas largas posibles antes de comenzar a repetirse y que superen las pruebas estadísticas de aleatoriedad [2].

El generador de llaves utiliza la función h. Utilizando el algoritmo de un solo tiempo de la función h se crea el generador de secuencias pseudoaleatorias para enteros de 15 bits. La semilla inicial son los valores iniciales de x y y para la función $t = h(x, y)$. El algoritmo del generador se muestra en la figura 2.2 La secuencia generada se calcula siguiendo la fórmula [2]:

$$x_0^{k+2} = h(x_0^{k+1}, x_0^k) \tag{2.1}$$

En la figura 2.2 se muestra a grandes rasgos el funcionamiento del generador, inicialmente se programa la semilla x y y, y se genera el primer número de la secuencia x_0^1 a partir de este valor empieza la recursión, donde x_0^1 pasa a ser el siguiente valor de y y el valor inicial de x el cual es x_0^0 se convierte en el valor de y, con estos nuevos valores se calcula $x_0^2 = h(x_0^1, x_0^0)$ y así sucesivamente se va generando la secuencia pseudoaleatoria [2].

2.1.2 Algoritmos de un solo tiempo

La encriptación y desencriptación se realiza utilizando una familia de permutaciones ψ, ϕ las cuales son implementadas utilizando el fenómeno de

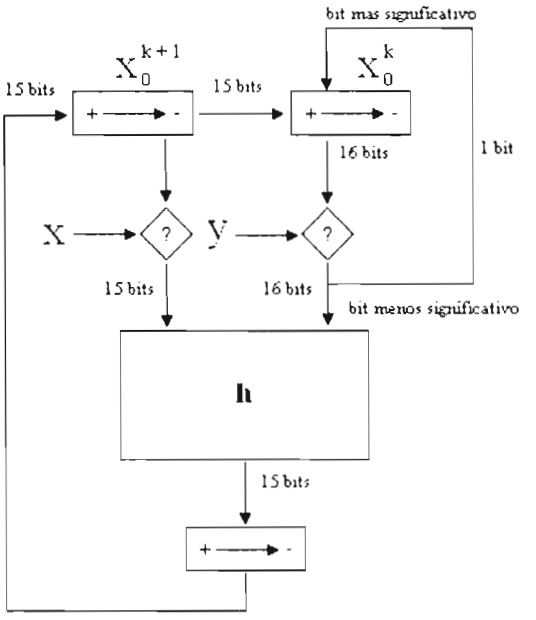

Figura 2.2: Algoritmo del generador de números aleatorios

sincronización en autómatas acoplados [2]. Utilizando las leyes y teoremas del álgebra booleana se redujeron las funciones ψ y ϕ a algoritmos de un solo tiempo, de manera que en cada ciclo de reloj se ejecute una función completa.

A continuación se muestra como hacer la reducción, tomando como ejemplo la unidad encriptadora básica de tamaño 15. Se va a obtener el primer bit, de la palabra **m** y se va a reducir por medio de las leyes del álgebra booleana. En la figura 2.3 se muestra la unidad encriptadora básica de tamaño 15 en donde se va a hacer la reducción del bit m_1.

La palabra **m** corresponde a la permutación $m = \phi_x(c)$ la cual se genera haciendo evolucionar el autómata hacia adelante, usando como entradas las palabras **c**, **x** y **t**.

En la figura 2.3 se muestran las dependencias para calcular el valor del bit m_1, estas dependencias son marcadas con una línea, siguiendo cada una de las dependencias se observa que finalmente el bit m_1 depende sólo de cuatro coordenadas de **x** y una de **c**, las cuales están indicadas con un círculo.

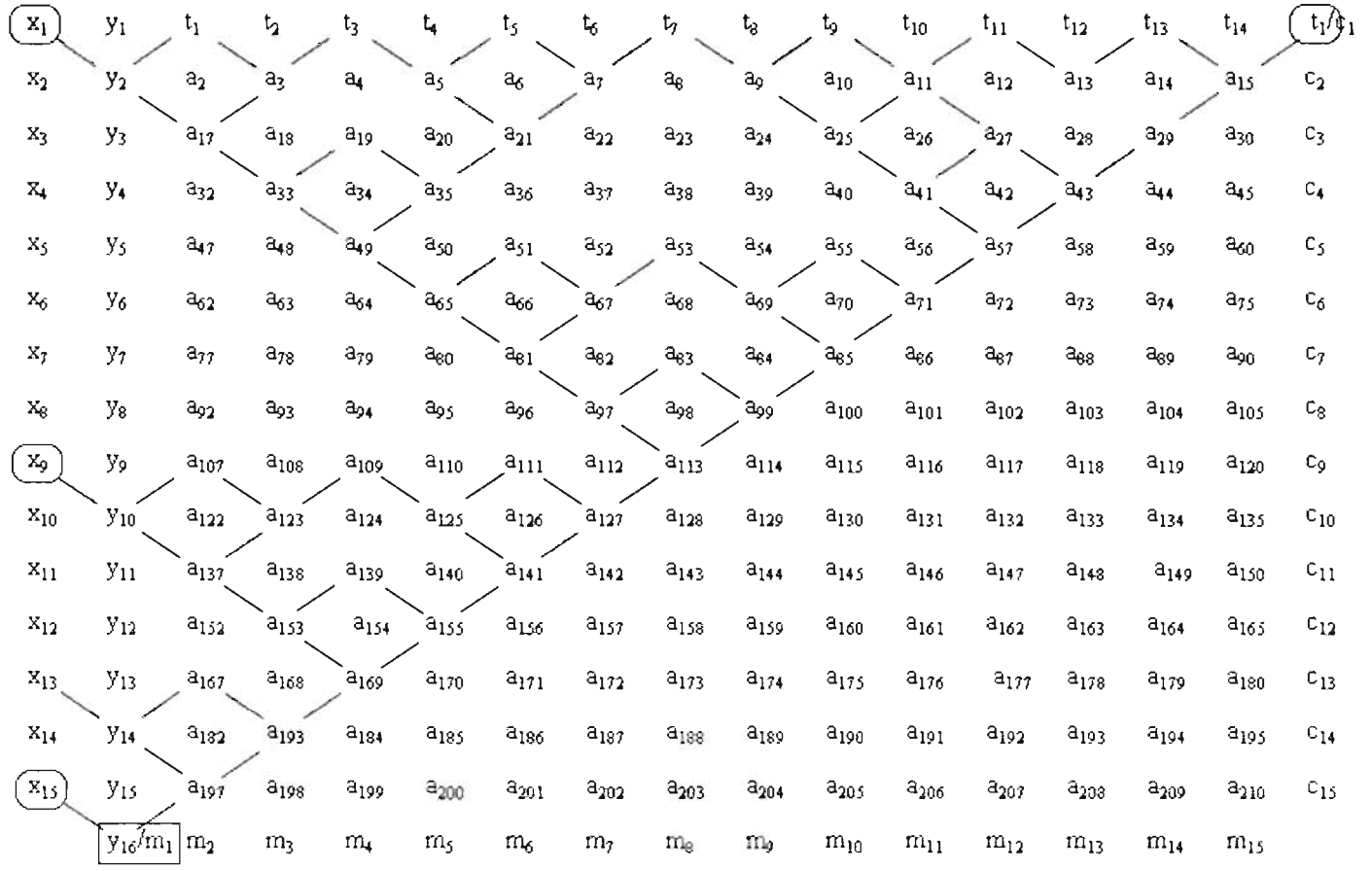

Figura 2.3: Reducción del primer bit m_1 de la palabra m

A continuación se muestra un ejemplo donde se calcula el valor del bit m_1 de la palabra **m** siguiendo cada uno de sus dependencias de abajo hacia arriba y se va reduciendo con álgebra booleana,

$$
\begin{aligned}
m_1 &= x_{15} \oplus a_{197} = x_{15} \oplus y_{14} \ominus a_{193} = x_{15} \oplus x_{13} \oplus a_{167} \oplus a_{167} \oplus a_{169} \\
&= x_{15} \oplus x_{13} \oplus a_{153} \ominus a_{155} = x_{15} \oplus x_{13} \ominus a_{137} \oplus a_{139} \oplus a_{139} \ominus a_{141} \\
&= x_{15} \oplus x_{13} \oplus y_{10} \oplus a_{123} \oplus a_{125} \oplus a_{127} \\
&= x_{15} \oplus x_{13} \oplus x_9 \oplus a_{107} \oplus a_{107} \ominus a_{109} \oplus a_{109} \oplus a_{111} \ominus a_{111} \oplus a_{113} \\
&= x_{15} \oplus x_{13} \oplus x_9 \oplus a_{97} \oplus a_{99} = x_{15} \oplus x_{13} \ominus x_9 \oplus a_{81} \oplus a_{83} \oplus a_{83} \ominus a_{85} \\
&= x_{15} \oplus x_{13} \oplus x_9 \oplus a_{65} \oplus a_{67} \oplus a_{69} \oplus a_{71} \\
&= x_{15} \oplus x_{13} \oplus x_9 \oplus a_{49} \oplus a_{51} \oplus a_{51} \oplus a_{53} \oplus a_{53} \oplus a_{55} \oplus a_{55} \oplus a_{57} \\
&= x_{15} \oplus x_{13} \oplus x_9 \oplus a_{33} \oplus a_{35} \ominus a_{41} \oplus a_{43} \\
&= x_{15} \oplus x_{13} \ominus x_9 \oplus a_{17} \oplus a_{19} \oplus a_{19} \oplus a_{21} \oplus a_{25} \oplus a_{27} \oplus a_{27} \oplus a_{29} \\
&= x_{15} \oplus x_{13} \oplus x_9 \oplus y_2 \oplus a_2 \oplus a_5 \oplus a_7 \oplus a_9 \oplus a_{11} \oplus a_{13} \oplus a_{15} \\
&= x_{15} \oplus x_{13} \oplus x_9 \oplus x_1 \oplus t_1 \oplus t_1 \oplus t_3 \oplus t_3 \ominus t_5 \oplus t_5 \oplus t_7 \oplus t_7 \oplus t_9 \ominus t_9 \\
&\quad \oplus t_{11} \oplus t_{11} \oplus t_{13} \oplus t_{13} \oplus c_1 \\
&= x_{15} \oplus x_{13} \oplus x_9 \oplus c_{11}
\end{aligned}
$$

Nótese que en el ejemplo anterior el resultado final no depende de la palabra **t**. Esta reducción se puede hacer con cada uno de los bits de la palabra **m**.

Para realizar la reducción de la palabra **c**, la cual es identificada como el bloque encriptado, que corresponde a la permutación $c = \psi_x(m)$ se hace evolucionar el autómata hacia atrás en el tiempo, usando como entradas las palabras **x** y **m**.

Para la reducción de la palabra **t** que corresponde a la función h, se hace evolucionar el autómata hacia atrás utilizando como entradas las palabras **x** y **y**.

Las expresiones para obtener las permutaciones ϕ, ψ y h, para la unidad encriptadora de 15 bits se muestran en el apéndice A.

2.1.3 Unidad Encriptadora

Haciendo uso de las expresiones booleanas de los algoritmos de un solo tiempo de las funciones $m = \phi_x(c)$ y $c = \psi_x(m)$, se implementan la unidad encriptadora y la unidad desencriptadora.

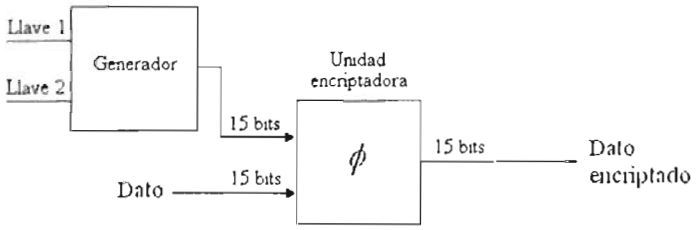

Figura 2.4: Unidad encriptadora

La figura 2.4 muestra un esquema del funcionamiento de la unidad encriptadora. Para realizar la encriptación es necesario programar la semilla inicial que genera una llave para cada bloque de 15 bits que se quiere encriptar.

En la figura 2.5 se puede ver a grandes rasgos el funcionamiento de la unidad desencriptadora. Para llevar a cabo la desencriptación se programa la misma semilla inicial que se utilizó para realizar la encriptación, que sirve para generar una llave para cada bloque de 15 bits que se quiere desencriptar.

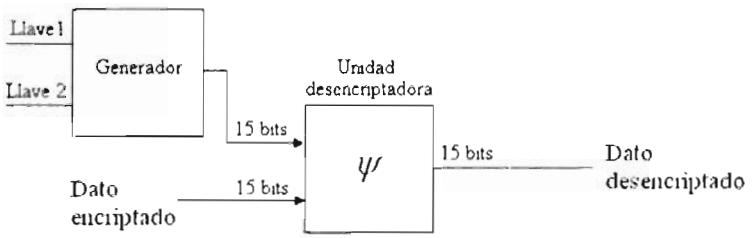

Figura 2.5: Unidad desencriptadora

Finalmente se integran todas las funciones de encriptación, desencriptación y generación de llaves en el ESAC.

La implementación del ESAC se lleva a cabo utilizando las versiones de un solo tiempo de las permutaciones $m = \phi_x(c)$, $c = \psi_x(m)$ y la función h mostradas en el apéndice A.

CAPÍTULO 3

Aplicaciones

Con el fin de probar la versatilidad del ESAC se presentan las siguientes aplicaciones:

- Envío de datos en la red GSM. Se introducen las características básicas de los sistemas celulares y el envío de información mediante el servicio de mensajes cortos SMS (Short Message Service)en la red GSM esto como base para aplicar el algoritmo de encriptación y desencriptación en el módulo GT48 de Sony Ericsson.
- Autentificación con huellas digitales y su validación. Se hace una descripción de las etapas o módulos en los que se divide el proceso de autentificación por medio del reconocimiento de huellas digitales.

3.1 Sistemas celulares

Los sistemas celulares se basan en la división del área de cobertura de un operador en lo que se denomina células, estas células se caracterizan por su tamaño que viene determinado por la potencia del transmisor pero de un modo muy particular ya que lo que se persigue siempre en los sistemas celulares es que la potencia de transmisión sea lo mas baja posible a fin de poder reutilizar el mayor número de frecuencias. El porque de tener el mayor número de frecuencias disponibles tiene que ver con que a mayor número de frecuencias libres mayor es el número de usuarios que pueden hacer uso del

sistema ya que cada uno puede usar una frecuencia sin interferir en la de otro usuario (realmente no se utiliza una frecuencia por usuario pero la idea general es esa). De este modo todas las bandas de frecuencias se distribuyen sobre las células a lo largo del área de cobertura del operador de manera que todos los canales de radio se encuentran disponibles para ser usados en cada grupo de células lo cual no sucedería si se produjese una emisión de la señal con una potencia superior ya que se podría interferir en otras células adyacentes interfiriendo en las frecuencias disponibles. Como es de suponer la distancia que debe existir entre dos células debe ser lo suficientemente grande como para que no se produzca interferencia entre ellas, hay que decir también que hay determinados canales que se reservan para labores de señalización y control de toda la red. La figura 3.1 muestra un esquema de una red celular.

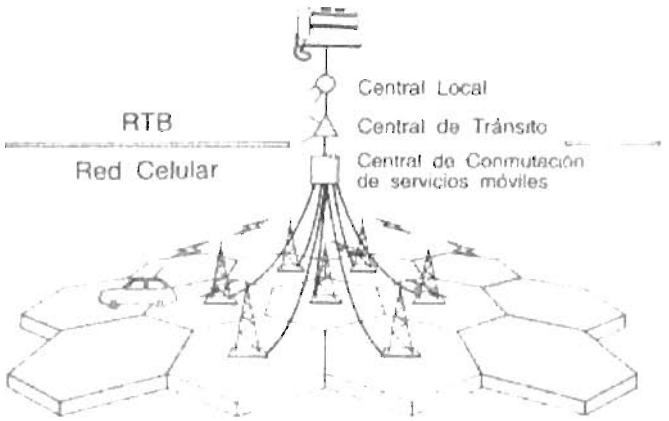

Figura 3.1: Red celular

Hay dos condiciones que las células deben de verificar para que un sistema celular funcione:

1. Por un lado el nivel de potencia del transmisor debe ser el mínimo para reducir las interferencias con los transmisores de las células vecinas.

2. Las células vecinas no pueden compartir los mismos canales, el motivo es similar al anterior, reducir el número de interferencias.

Las células se unen unas con otras mediante cable o bien mediante radio enlaces así como con la red telefónica fija.

3.1.1 GSM

GSM es un sistema de comunicación basado en el uso de células digitales que se desarrollo para crear un sistema para móviles único que sirviese de estándar para Europa y que fuese compatible con los servicios existentes y futuros sobre ISDN (Integrated Services Digital Network) o RDSI (Red Digital de Servicios Integrados).

En GSM se distinguen cuatro tipos diferentes de células, son las siguientes:

- Macrocélulas (Macrocells): son células de gran tamaño utilizadas en áreas de terreno muy grandes y donde la distancia entre áreas pobladas es muy distante entre sí.

- Microcélulas (Microcells): se utilizan en áreas donde hay una gran densidad de población, a mayor número de células mayor número de canales disponibles que pueden ser utilizados por mas usuarios simultáneamente.

- Células selectivas (Selective cells): en muchas ocasiones no interesa que una célula tenga cobertura de 360º sino que interesa que tenga un alcance y un radio de acción determinado, en este caso es donde aparecen las células selectivas, el caso más típico de células de este tipo son aquellas que se disponen en las entradas de los túneles en los cuales no tiene sentido que la célula tenga un radio de acción total sino un radio de acción que vaya a lo largo del túnel.

- Células sombrilla (Umbrella cells): este tipo de células se utilizan en aquellos casos en los que se tiene un elevado número de células de tamaño pequeño y continuamente se están produciendo cambios del terminal de una célula a otra (handovers) para evitar que suceda esto lo que se haces agrupar conjuntos de microcélulas de modo que se aumenta la potencia de la nueva célula formada y se puede reducir el número de handovers que se producen.

Hoy en día GSM es un estándar que no es utilizado solamente en Europa ya que actualmente es usado en casi cien países en todo el mundo y el número de usuarios que hacen uso de él se ha venido duplicando de año en año. De

igual modo el número de servicios que se han ido desarrollando sobre GSM han ido evolucionando con el paso del tiempo. se han definido tres tipos de categorías de servicios que pueden ofrecerse sobre una red GSM.

Las tres categorías de servicios sobre GSM son: teleservicios que englobaría a los servicios básicos de telefonía; los servicios portadores que son los usados para la transmisión y recepción de datos; y los servicios complementarios generalmente extensiones de los teleservicios y que proporcionan nuevas características a la red GSM.

Estos son algunos de los servicios disponibles para cada categoría.

Teleservicios

- Telefonía
- Llamadas de emergencia: posibilita hacer llamadas de emergencia pulsando un botón aún sin contar con la tarjeta SIM.
- Servicio de mensajes cortos (SMS): es posible enviar un mensaje de hasta 160 caracteres desde y hacia un terminal móvil. Si el móvil no esta conectado o fuera de cobertura, el mensaje se almacena en la central de mensajes hasta que el abonado se conecte, avisándoles de la existencia de dicho mensaje.
- Servicios de fax y voz: capacidad de recibir y enviar llamadas hacia o desde todo el mundo tanto con teléfonos fijos como con teléfonos móviles. Permite al usuario recibir mensajes de fax en cualquier máquina a través de su móvil.

Servicios Portadores

- Transmisión síncrona y asíncrona de datos

Servicios Complementarios

- Llamada en espera
- Llamadas múltiples
- Identificación de llamadas

3.1.2 Envío de datos en la red GSM

En algunos casos el envío de información en la red celular se hace mediante el servicio de mensajes cortos ó SMS. SMS es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos, también conocidos como mensajes de texto, entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano. SMS fue diseñado originalmente como parte del estándar de telefonía móvil digital GSM, pero en la actualidad esta disponible en una amplia variedad de redes. SMS permite enviar y recibir mensajes escritos de hasta 160 caracteres.

3.1.3 Modulo GSM/GPRS GT48

El modulo GT48 de Sony - Ericsson es una terminal de control GSM/GPRS inteligente que incluye todo lo necesario para quienes desean seguir, controlar y obtener información operacional de máquinas, sistemas y equipos mediante la comunicación inalámbrica. El diseño compacto y autocontenido admite opciones múltiples de *input/output* haciendo de este el dispositivo ideal para el seguimiento de instalaciones o áreas con opciones limitadas de *input/output* [6]. En la figura 3.2 se puede observar el módulo GT48.

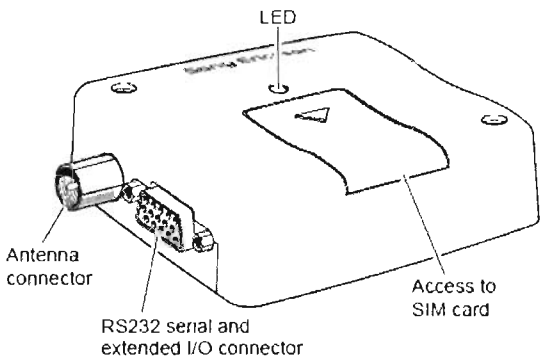

Figura 3.2: Modulo GT48

El GT48 cuenta además con su propio lector de tarjeta SIM. El SIM es una pequeña tarjeta inteligente que sirve para identificar las características de nuestro terminal. Esta tarjeta se inserta en el interior del móvil y permite al usuario acceder a todos los servicios que haya disponibles por su operador, sin

la tarjeta SIM el terminal no nos sirve de nada por que no podemos hacer uso de la red. El SIM esta protegido por un número de cuatro dígitos que recibe el nombre de PIN o Personal Identification Number. La mayor ventaja de las tarjetas SIM es que proporcionan movilidad al usuario ya que puede cambiar de terminal y llevarse consigo el SIM. Una vez que se introduce el PIN en el terminal, el terminal va a ponerse a buscar redes GSM que estén disponibles y va a tratar de validarse en ellas, una vez que la red (generalmente la que tenemos contratada) ha validado el terminal el teléfono queda registrado en la célula que lo ha validado.

A continuación se mencionan las características principales del modulo GT48:

- Diseño autocontenido.
- Doble banda GSM 850/1900 MHz.
- GSM/GPRS Clase 8.
- Conectividad 2x UART.
- Pila TCP/IP integrada (TCP/IP stack).
- Diseños embedidos.

RS232 Serial Connector

El modulo GT48 cuenta con un conector de alta densidad de 15 pines que cuenta con entradas y salidas configurables, el conector soporta el estándar RS232 DB9 para comunicación serial, además incluye una interfaz *4-wire* RS232, figura 3.3.

Para la comunicación serial el GT48 soporta el formato estándar de 1 bit de parada, 8 bits de datos, paridad ninguna, en total 10 bits por caracter. Por default el GT48 tiene una velocidad de transmisión de 9600 bps, sin embargo el rango de transmisión puede llegar hasta 230.4 kbps [6].

Las siguientes son la señales utilizadas por el GT48 para llevar a cabo la comunicación serial.

RD

RD es la señal de salida que el GT48 utiliza para enviar datos a una aplicación externa.

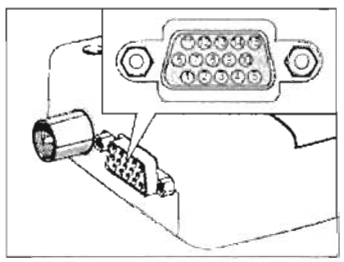

Figura 3.3: 15 pin high density connector

TD
TD es la señal de entrada, utilizada por la aplicación externa para mandar datos al GT48.

Request to Send (RTS)
Condiciona al GT48 para el envío de datos. Por default su estado es inactivo. Para permitir la transmisión de datos la señal debe ponerse en alto.

Clear to Send (CTS)
CTS indica que el GT48 esta listo para transmitir datos. Por default esta señal se encuentra en alto.

Data Terminal Ready (DTR)
DTR indica que DTE (data terminating equipment) esta listo para recibir y mandar datos. La señal se activa en alto.

Data Set Ready (DSR)
Una señal activa DSR se envía del GT48 a la aplicación (DTE) para indicar que la comunicación ha sido establecida.

Data Carrier Detect (DCD)
DCD indica cuando el GT48 esta recibiendo un portador válido cuando la señal esta en alto.

Ring Indicator (RI)

RI indica que una señal se sonido esta siendo recibida por el GT48 cuando la señal esta en alto.

El modulo tiene la capacidad de almacenar y correr códigos escritos por el usuario para realizar un sinfín de tareas. La escritura de los aplicaciones esta basada en lenguaje C. El módulo tiene hasta 44 kbytes de espacio disponible para el almacenamiento de dos aplicaciones.

El software M2mpower IDE permite al usuario escribir, simular, verificar errores y descargar las aplicaciones al modulo GT48. El software incluye librerías y funciones intrínsecas necesarias para desarrollar las aplicaciones necesarias.

3.1.4 Envío de datos encriptados en la red GSM

A continuación se da una explicación de las etapas en las que se divide el proceso de envío de información encriptada a través de la red GSM utilizando el módulo GT48 de Sony - Ericsson. La figura 3.4 muestra un esquema del proceso. Se utilizan dos módulos GT48 cada uno conectado a una PC mediante el puerto serial respectivamente.

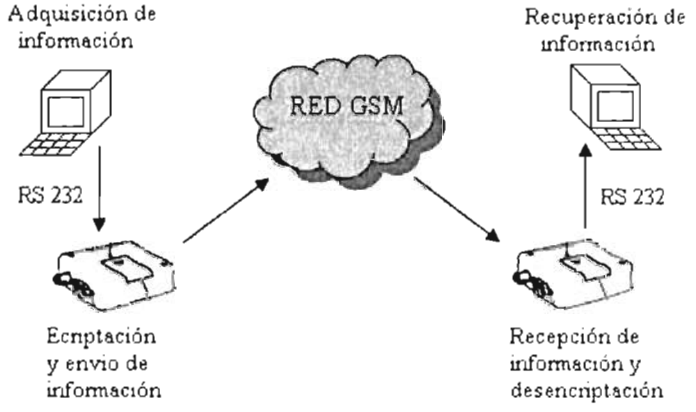

Figura 3.4: Envío de información a través de la red GSM

El primer módulo ó módulo 1 realiza las siguientes funciones:

1. Lee la información enviada desde la PC a través del puerto serial

2. Encripta la información.

3. Envía la información encriptada utilizando un SMS.

La figura 3.5 muestra el diagrama de bloques del algoritmo programado en el módulo 1. Se comienza con la declaración de variables y funciones. Las funciones definidas son la función generadora de secuencias pseudoaleatorias y la función que contiene el algoritmo de encriptación, en ambas funciones se utilizan las expresiones booleanas de los algoritmos de un solo tiempo de las funciones $c = \psi_x(m)$ y h. El primer paso es abrir el puerto serial para leer los bytes enviados desde la PC al módulo, cuando el número de bytes leídos es mayor que 1 el programa pasa al siguiente paso, encriptar los datos, los datos son encriptados en bloques de 16 bits, utilizando para cada bloque una llave generada por la función generadora de secuencias pseudoaleatorias, cuando los datos son encriptados el módulo manda un SMS cuyo contenido es la información encriptada. Después de mandar el SMS el programa vuelve al inicio y espera nuevamente a leer bytes del puerto serial.

El segundo módulo ó módulo 2 realiza las siguientes funciones:

1. Recibe la información enviada desde el módulo 1.

2. Desencripta la información.

3. Envía la información desencriptada a la PC a través del puerto serial.

La figura 3.6 muestra el diagrama de bloques del algoritmo programado en el módulo 2. Inicialmente se realiza la declaración de variables y funciones. Las funciones definidas son la función generadora de secuencias pseudoaleatorias y la función que contiene el algoritmo de desencriptación, en ambas funciones utilizan las expresiones booleanas de los algoritmos de un solo tiempo de las funciones $m = \varphi_x(c)$ y h. En el primer paso el módulo 2 se encuentra en espera de leer el SMS que fue enviado desde el módulo 1, cuando recibe el SMS el siguiente paso es desencriptar los datos, los datos son desencriptados en bloques de 16 bits, utilizando para cada bloque una llave generada por la función generadora de secuencias pseudoaleatorias, en seguida se envían los datos desencriptados a través del puerto serial a la PC. Finalmente el programa vuelve al inicio y espera a leer un nuevo SMS.

En el apéndice D aparece el código de los programas para el módulo 1 y el módulo2.

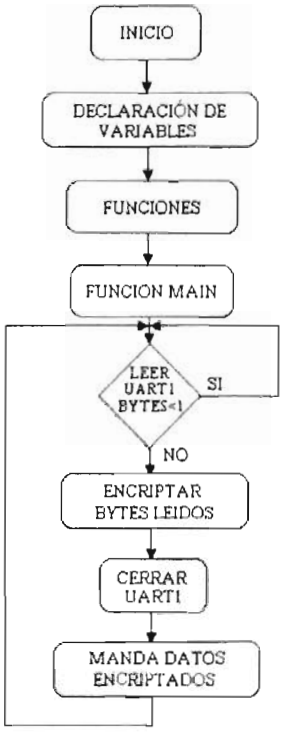

Figura 3.5. Diagrama de bloques del algoritmo programado en el módulo 1

3.2 Autentificación con huellas digitales y su validación

A continuación se presenta un protocolo eficiente y seguro para transmitir imágenes encriptadas de un sensor de huellas digitales a un servidor que desencripta y valida las imágenes.

3.2.1 Autentificación biométrica por huellas digitales

Entre todas las técnicas biométricas, la identificación basada en las huellas digitales es el método mas antiguo, el cual ha sido usado en numerosas aplicaciones.

El uso de huellas digitales para la identificación y autentificación de personas se ha popularizado en los últimos años gracias a las nuevas tecnologías

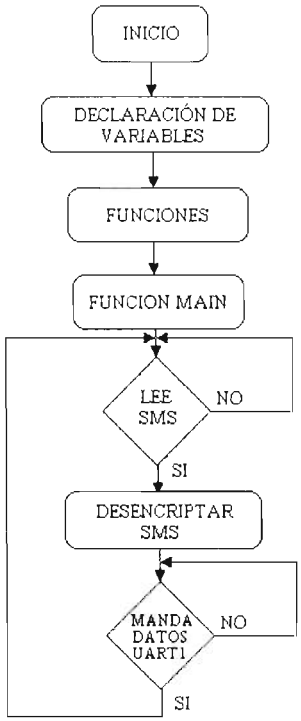

Figura 3.6: Diagrama de bloques del algoritmo programado en el módulo 2

de sensores baratos y confiables para la captura de huellas y al desarrollo de algoritmos para hacer el análisis y comparación de huellas automáticamente

Actualmente la identificación basada en huellas digitales tiene errores de identificación reportados con menos del 1 %, y los sistemas están basados en dos etapas: el procesamiento de la huella y la identificación de la huella

El propósito del preprocesamiento es extraer características diferenciales entre individuos. Se ha encontrado que las minucias (localización de las terminaciones y bifurcaciones) es una buena característica diferencial. El propósito de la segunda etapa, como su nombre lo indica es la identificación de la huella, es decir, comparar las minucias obtenidas en el preprocesamiento con las minucias almacenadas en una base de datos previamente creada. Esta base de datos contiene las minucias de las personas que pueden acceder a un sistema.

La identificación por medio de huellas digitales se basa en cuatro patrones

que toda huella tiene que son:

- Presilla derecha. Una curva que va de la parte baja de la huella digital a casi la mitad y regresa a la parte superior de la misma.
- Arco. Como lo dice su nombre un arco que cruza por la mitad de la huella digital de izquierda a derecha.
- Verticilo. Es una "S"que se forma en la huella digital repetidas veces en la parte superior de la huella.
- Presilla izquierda. Lo mismo que la presilla derecha pero este patron comienza de izquierda al centro y de regreso.

En los cuales existen 8 patrones característicos que se muestran en la figura 3.7.

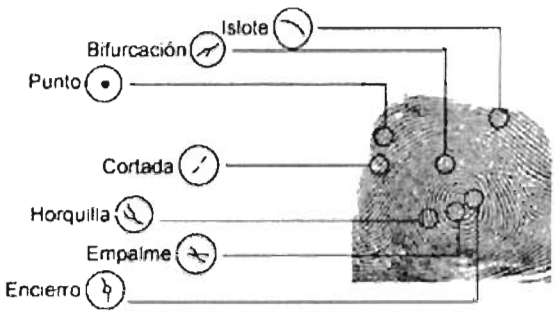

Figura 3.7: Patrones características de la huella digital

Algunas desventajas del reconocimiento biométrico de huellas digitales son que generalmente requiere una cantidad considerable de recursos computacionales y que los patrones sí pueden cambiar debido a causas radicales.

3.2.2 Procedimiento

En seguida se hace una descripción de las etapas o módulos en los que se divide el proceso del sistema propuesto de autentificación biométrica por medio de reconocimiento de huella digital.

La primera etapa consiste en capturar la huella digital, para posteriormente enviarla a través del puerto serial hacia la estación base (PC). En la figura 3.8 podemos ver la interfaz creada en LabVIEW, cuando corremos la aplicación el programa automáticamente carga la imagen y la envía a través del puerto serial.

Figura 3.8: Adquisición y envío de la imagen

Las huellas dactilares se adquieren con el sensor biométrico FDC-FPC1010 montado en el DSP (Digital Signal Processing) C6713, del kit de desarrollo de herramientas para la autentificación de huellas dactilares de Texas Instruments (Fingerprint Authentification Development Tool). El tamo de las imágenes obtenidas es de 152 x 200 pixels, codificándose cada pixel con una escala de grises de 8 bits [3].

Procesamiento de la imagen

Al obtener la imagen digitalizada de la huella dactilar, esta es sometida a un proceso de compresión y encriptación para ser enviada via inalámbrica utilizando el transceiver AC5124C de Aerocomm, el cual trabaja en el ancho de banda de 2.4 GHz.

El transceiver AC5124C tiene una interface de tipo serial, la cual permite al Host enviar y recibir comunicación desde el transceiver. Todas las señales de entrada y salida son de nivel TTL de 5V DC, excepto para RSSI (Received Signal Strength Indicator), la cual es una salida analógica. El RSSI

es utilizado por el Host para determinar la intensidad de la señal al instante de recibirla en le receptor. La figura 3.9 muestra la imagen de la huella encriptada.

Figura 3.9: Huella encriptada

El proceso de encriptación de la imagen se lleva a cabo en la estación base, que cuenta con con un programa desarrollado en LabVIEW para realizar tal propósito. La figura 3.10 muestra el proceso que se lleva a cabo para realizar la encriptación de la imagen. La imagen es procesada como una secuencia de bloques de 16 bits. En este mismo proceso, la imagen es comprimida un 25 por ciento para reducir su tamaño con el fin de reducir el tiempo de transferencia a través del transceiver utilizado.

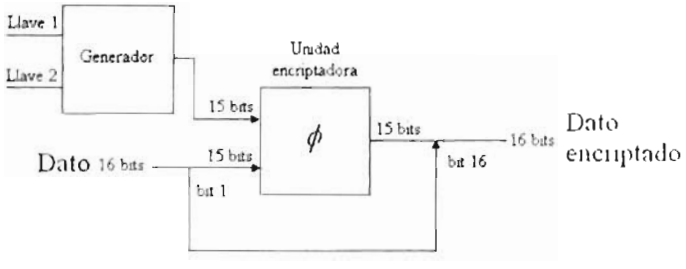

Figura 3.10: Proceso de encriptación

La adquisición de imágenes y voz genera una secuencia de bloques de 16 bits. Es necesario modificar las expresiones booleanas de los algoritmos

de un solo tiempo de las funciones $m\phi_x(c)$ y $c = \psi_x(m)$ de 15 a 16 bits (ESAC). En la figura 3.11 aparece un ejemplo de como es procesado cada bloque de la imagen. En la encriptación el bit 1 pasa a ser el bit 16 del bloque encriptado. En la desencriptación el bit 16 pasa a ser el bit 1 del bloque desencriptado (original). De esta manera el bit que contiene la menor cantidad de información de la imagen no es alterado por el proceso de encriptación y desencriptación.

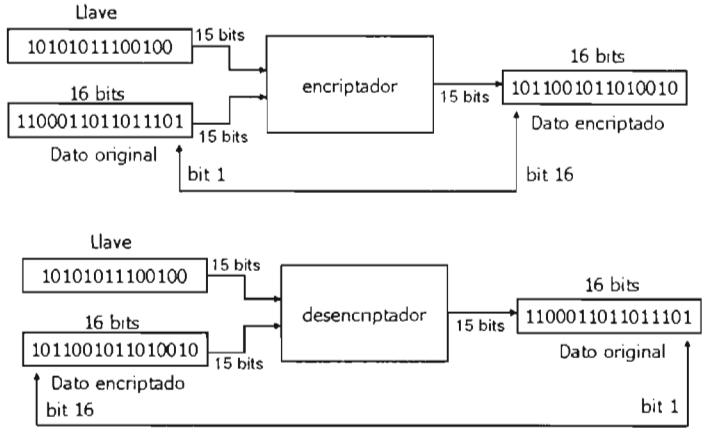

Figura 3.11: Encriptación y Desencriptación: 16 bits

Luego la imagen es enviada desde la estación base al servidor para su análisis. El servidor es capaz de desencriptar la imagen recibida y validarla. La figura 3.12 muestra el proceso llevado a cabo para desencriptar la imagen recibida.

Proceso de identificación

La identificación consiste en que, a partir del elemento que se esta utilizando para verificar (la huella digital, en este caso) uno pueda llegar a los datos de la persona que pone su dedo sobre el sensor biométrico. En éste módulo, la imagen digital de la huella del usuario, previamente registrada en la base de datos, se convierte en el template a buscar, el sistema se encarga de realizar la búsqueda verificando con todas las imágenes almacenadas en la base de datos. Cuando se logra una búsqueda satisfactoria nos muestra la

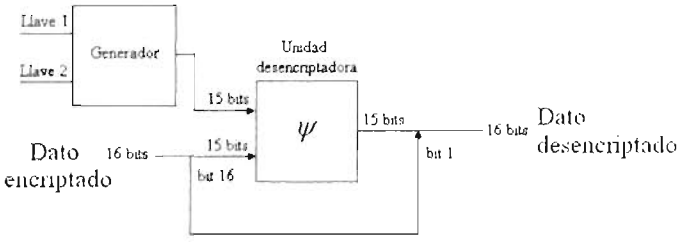

Figura 3.12: Proceso de desencriptación

identidad del usuario, caso contrario, nos indica que la huella dactilar no se encuentra en nuestra base de datos [3]. En la figura 3.13 se muestra el panel frontal de la identificación de por medio de la huella digital.

Figura 3.13: Panel frontal de identificación

CAPÍTULO 4

Pruebas experimentales y resultados

En este capítulo se presentan las pruebas realizadas con las aplicaciones mencionadas en el capítulo 2, así como los resultados obtenidos.

4.1 Pruebas experimentales

Se realizaron las pruebas necesarias para determinar específicamente el área de cobertura de los transceiver. Esto con el fin de determinar la confiabilidad en el envío y recepción de las huellas encriptadas en el sistema de autentificación biométrica.

En el caso de los MODEM GT48 la cobertura varía dependiendo del servicio de telefonía celular con que se cuente.

4.1.1 Procedimiento

La figura 4.1 muestra la estación remota que es donde se hace la adquisición de la huella digital, se encripta y se envía por medio de los transceiver hacia la estación base. La estación base recibe la imagen encriptada de la huella digital, la desencripta y se encarga de validarla.

Se realizaron dos tipos de pruebas. La primera prueba consiste en agregar ruido a la imagen encriptada de la huella digital y la segunda prueba en determinar el área de cobertura de los transceiver. En ambas pruebas es necesario contar obviamente con una base de datos. En la base de datos se

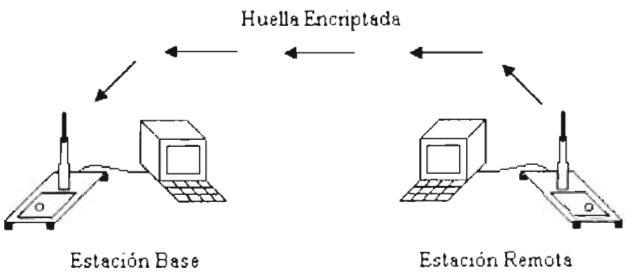

Figura 4.1: Estación base y estación remota

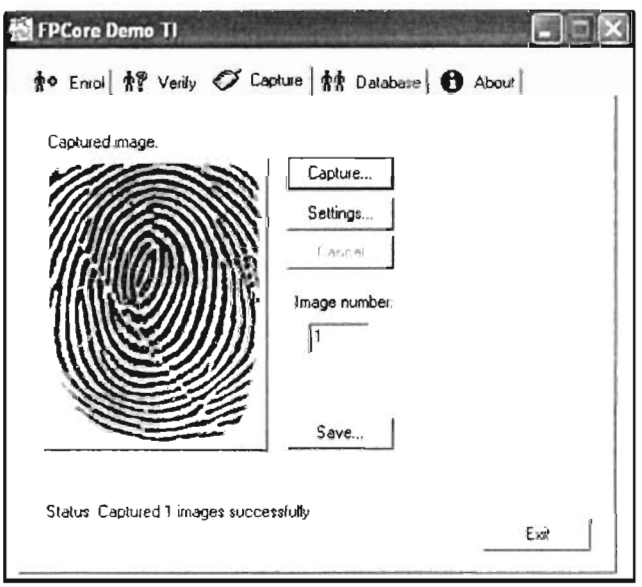

Figura 4.2: Adquisición de la huella digital

Figura 4.3: Huella desencriptada con ruido aleatorio

encuentran registradas siete personas, contiene cinco huellas digitales de cada persona.

Prueba 1

La primera prueba consiste en dos partes: en la primera se agrega ruido aleatorio; en la segunda se agrega ruido gaussiano.

Se entiende por ruido en imágenes cualquier pixel de una imagen que no corresponde exactamente con la realidad. Cuando se adquiere una imagen, ésta esta contaminada por ruido. El ruido se debe, la mayoría de las veces al equipo electrónico utilizado en la captación de las imágenes y al ruido añadido en los tramos de transmisión (posibles interferencias ó errores al transmitir los bits de información.)

Ruido Aleatorio

Como se menciono en el capítulo 2 la imagen se divide en una secuencia de bloques de 16 bits cada uno, el ruido aleatorio consiste en *sumarle 1 ó 0* al bit de un bloque ambos aleatoriamente seleccionados. Se comienza agregando un bit erróneo después 2, 3, 4 y así sucesivamente.

En la imagen 4.3 en el inciso (a) se muestra la imagen desencriptada de una huella digital, en el inciso (b) se agregan 5 bits erróneos, en el inciso (c) 10, finalmente en el inciso (d) se agregan 20 bits erróneos. Los bits erróneos se agregan antes de desencriptar la imagen de la huella digital.

En la figura 4.4 se muestra un ejemplo de como generar ruido aleatorio. Se elige aleatoriamente el bit 12 del bloque 5 de la secuencia de la imagen encriptada, y se le *suma* un 1 ó un 0. Cuando se *suma* un 1, el bit elegido

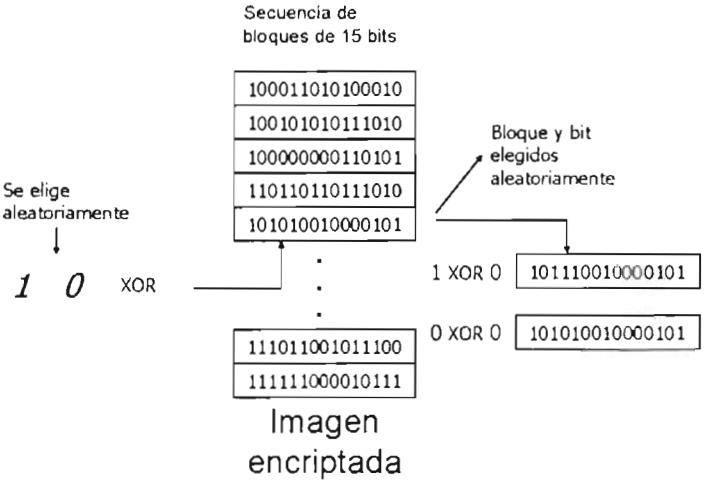

Figura 4.4: Ruido aleatorio

cambia de 0 a 1; cuando se *suma* un 0 el bit no cambia.

Ruido Gaussiano

El ruido gaussiano tiene un efecto general en toda la imagen de la huella digital, es decir, la intensidad de cada pixel de la imagen se ve alterada en cierta medida con respecto a la intensidad en la imagen original.

En la figura 4.5 podemos ver la distribución de ruido gaussiano que viene dada por la siguiente función:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \tag{4.1}$$

Donde μ es el valor medio y σ es la varianza. En las pruebas realizadas $\mu = 0$ y σ varia de 0,01 a 0.20.

La figura 4.6 muestra la imagen desencriptada de una huella digital. En el inciso a $\sigma = 0{,}00$; en b $\sigma = 0{,}05$; en c $\sigma = 0{,}10$ y en d $\sigma = 0.20$.

El procedimiento que se siguió en las dos pruebas, ruido aleatorio y ruido gaussiano, fue enviar cada una de las 5 huellas registradas de cada persona agregando ruido (aleatorio y gaussiano) hasta llegar al punto en el que

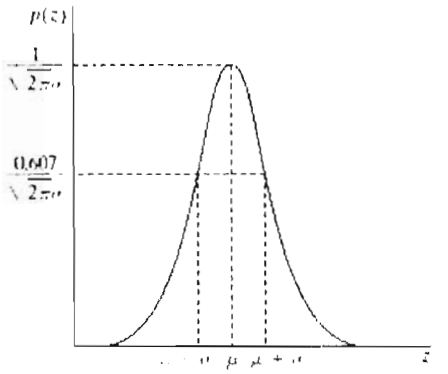

Figura 4.5: Distribución gaussiana

no fuera posible identificar y validar la huella digital, esto como base para determinar que tanto afecta el ruido en el proceso de envío y recepción de imágenes encriptadas.

Figura 4.6: Ruido gaussiano en imágenes

Prueba 2

La segunda prueba se realizo dentro y fuera de las instalaciones del Instituto de Investigación en Comunicaciones Ópticas (IICO), en esta prueba la estación remota se fue alejando de la estación base esto con el fin de determinar la distancia en la cual había pérdida total de comunicación entre los transceiver.

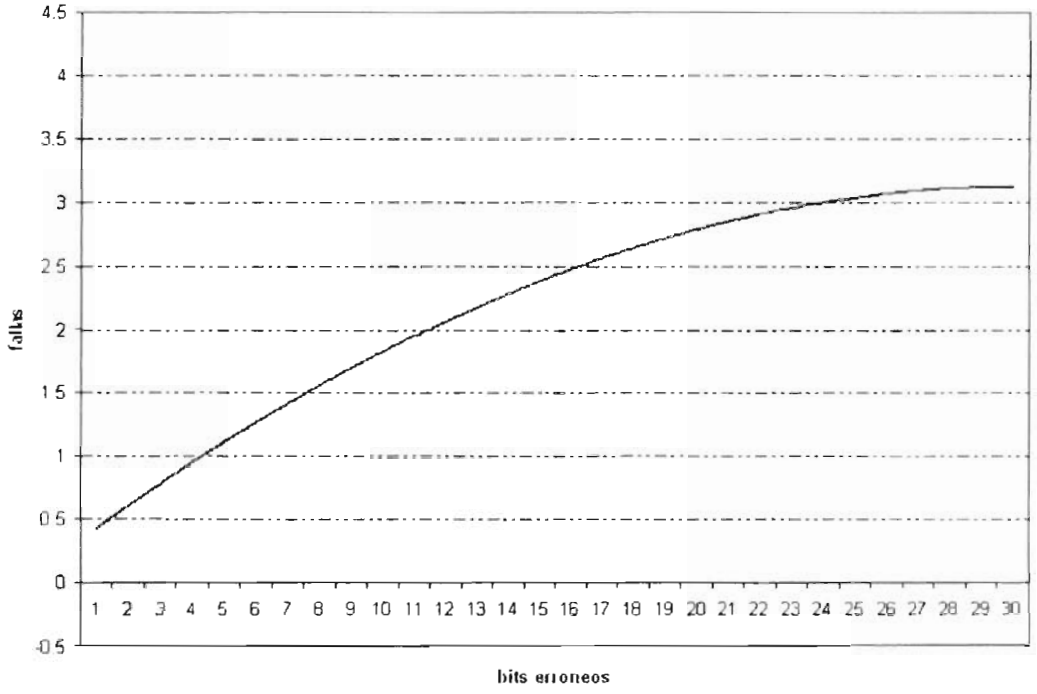

Figura 4.7: Fallas en la validación de la huella digital

4.2 Resultados

Se presentan los resultados obtenidos en las pruebas mencionadas en la sección anterior. Con el uso de gráficas se pretende esquematizar el número de fallas en el envío y recepción de las imágenes en ambas pruebas.

4.2.1 Ruido en envío y recepción de imágenes

El procedimiento que se siguió para cada persona registrada en la base de datos, para ruido aleatorio y ruido gaussiano, fue el siguiente:

1. Adquirir la huella de la persona. Se pide a la persona que coloque la misma huella digital que se utilizo para registrarla en la base de datos.

2. Se encripta la imagen de la huella digital y se envía.

3. La estación base recibe la imagen de la huella digital encriptada le agrega ruido y la desencripta. Para el caso de ruido gaussiano se comienza agregando 1 bit erróneo.

4. Se repiten los dos paso anteriores hasta agregar a 30 bits erróneos, después de esta cantidad ya no es posible validar la huella lo cual no permite identificar a la persona. Para el caso de del ruido gaussiano se varía σ desde 0,01 hasta 0,13.

La figura 4.7 muestra la gráfica realizada con los resultados obtenidos en la prueba realizada a una persona. Se grafica cantidad de bits erróneos contra fallas, las fallas significan que no se pudo validar la huella.

En la gráfica de la figura 4.8 se puede ver la cantidad de fallas totales para el caso en que se agrega ruido aleatorio. Para el caso en el que se agrega ruido gaussiano los resultados se muestran en la gráfica de la figura 4.9.

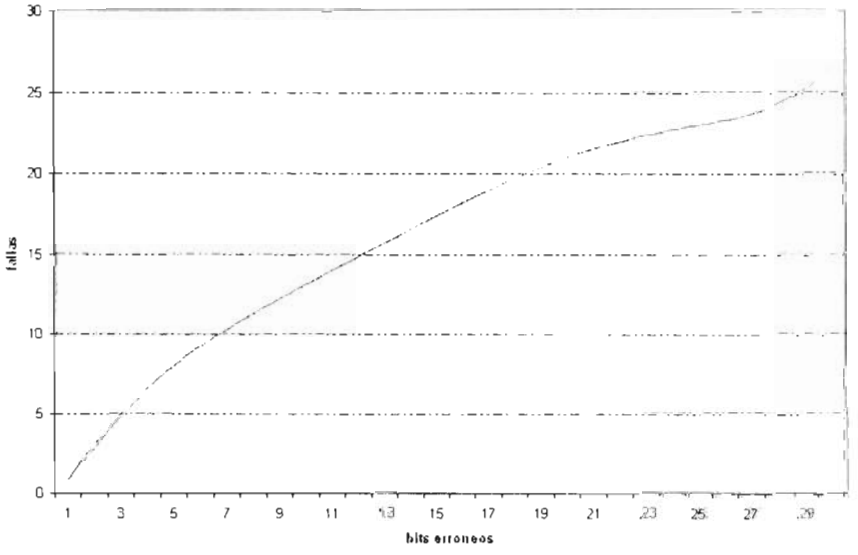

Figura 4.8: Número de fallas - Ruido Aleatorio

4.2.2 Área de cobertura

Se pudo determinar que con el transceiver AC5124C utilizado se logran distancias de cobertura de alrededor de los 40 m, con lo cual el área a cubrir sería una circunferencia de aproximadamente 80 metros de diámetro. Para incrementar la cobertura sería necesario reemplazar los transceivers AC5124C utilizados por otros de mayor alcance.

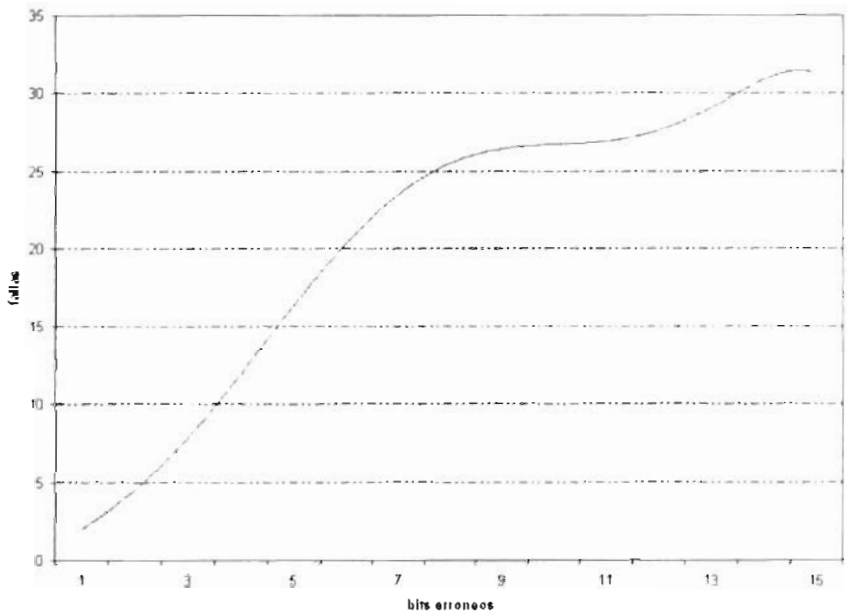

Figura 4.9: Número de fallas - Ruido Gaussiano

Durante las pruebas se observó una pérdida por trayectoria dependiendo de la distancia. A distancias cortas de la antena del transceiver AC5124C las obstrucciones como paredes y árboles no interactúan significativamente con las que se propagan. Sin embargo, a medida que las ondas electromagnéticas se ven más obstruidas por las paredes y árboles de diferentes edificios de las instalaciones del instituto, la pérdida se incrementa significativamente.

CAPÍTULO 5

Conclusiones

Se implemento el ESAC (encriptación por sincronización en autómatas celulares) el cual es un cifrador que encripta y desencripta bloques de 15 bits, utilizando para cada bloque una llave generada por un generador de números pseudoaleatorio. Fue necesario modificar el ESAC para encriptar y desencriptar bloques de 16 bits debido a que la información se dividió en secuencias de bloques de 16 bits.

Con el fin de probar la versatilidad del ESAC se presentaron dos aplicaciones: envío de datos encriptados en la red GSM y un sistema de validación y autentificación con huellas digitales.

Se presentaron las característica básicas y de funcionamiento de la red celular, en particular, la red GSM. Se implemento de manera práctica el algortimo de encriptación sobre el MODEM GT48 de Sony - Ericsson, el cual es una terminal de control GSM/GPRS inteligente que incluye todo lo necesario para controlar y obtener información operacional de máquinas, equipos, sistemas y equipos mediante la comunicación inalámbrica.

En el sistema de validación y autentificación, para la adquisición de las huellas digitales, se utilizó el sensor FPC1010-FPC de Texas Instruments. Éste sensor tiene una gran variedad de aplicaciones: teléfonos móviles, computadoras portátiles, **sistemas de seguridad**, etc. En ésta aplicación el algoritmo de encriptación se implementó utilizando el software LabVIEW.

En ambas aplicaciones se encontró que el algoritmo de encriptación utilizado es extremadamente versátil, confiable y seguro

Debido a que el ruido afecta a la señal cuando se envía información a través de cualquier medio de transmición, se simuló ruido que se añade a los datos encriptados, que son los que se envian. Se consideraron dos tipos de ruido: aleatorio y gaussiano.

El ruido aleatorio afecta el bit de un bloque de la secuencia de información, ambos elegidos aleatoriamente, mientras que el ruido gaussiano tiene un efecto general en toda la información. Se encontró que la corrupción con éste tipo de ruidos es muy significativa al momento de realizar la desencriptación, esto debido a que existe un cambio en la imagen desencriptada, que aumenta gradualmente cuando se va agregando mas ruido. A causa de éste cambio hay un momento donde el sistema es incapaz de autentificar y validar la huella digital.

Finalmente se realizaron las pruebas necesarias para determinar específicamente el área de cobertura de los transceivers. La cobertura que se alcanzó fue de aproximadamente 40 m, con lo que se puede cubrir una circunferencia de alrededor de 80 m. Si se desea tener una mayor área de cobertura es necesario sustituir los tranceivers utilizados por otros de mayor alcance.

Para el caso donde se quiera enviar una gran cantidad de información a través de la red celular y en donde se tenga restringido el coste para GSM/GPRS/EDGE. una opción es la Internet ó la red telefónica convencional, en éste caso se propone implementar el algorítmo de encriptación en MODEM´s inteligentes. Por ejemplo, el MODEM TMS320VC54CST de Texas Instruments.

CAPÍTULO A

Algoritmos de un solo tiempo

A continuación se muestran las expresiones booleanas de los algoritmos de un solo tiempo de la función $m = \phi_x(c)$ para el caso de tamaño 15 de la unidad encriptadora.

$$
\begin{aligned}
m_1 &= x_{15} + x_{13} + x_9 + x_1 + c_1 \\
m_2 &= x_{14} + x_{10} + x_2 + c_1 \\
m_3 &= x_{13} + x_{11} + x_9 + x_3 + x_1 + c_1 + c_3 \\
m_4 &= x_{12} + x_4 + c_4 \\
m_5 &= x_{11} + x_9 + x_5 + x_3 + x_1 + c_1 + c_3 + c_5 \\
m_6 &= x_{10} + x_6 + x_2 + c_2 + c_6 \\
m_7 &= x_9 + x_7 + x_5 + x_1 + c_1 + c_5 + c_7 \\
m_8 &= x_8 + c_8 \\
m_9 &= x_7 + x_5 + x_1 + c_1 + c_5 + c_7 + c_9 \\
m_{10} &= x_6 + x_2 + c_2 + c_6 + c_{10} \\
m_{11} &= x_5 + x_3 + x_1 + c_1 + c_3 + c_5 + c_9 + c_{11} \\
m_{12} &= x_4 + c_4 + c_{12} \\
m_{13} &= x_3 + x_1 + c_1 + c_3 + c_9 + c_{11} + c_{13} \\
m_{14} &= x_2 + c_2 + c_{10} + c_{14} \\
m_{15} &= x_1 + c_1 + c_9 + c_{13} + c_{15}
\end{aligned}
$$

Enseguida se muestran las expresiones booleanas del algoritmo de un solo tiempo de la función $c = \psi_m$ para el caso de tamaño 15 de la unidad encriptadora.

$$
\begin{aligned}
c_1 &= x_1 + x_9 + x_{13} + x_{15} + m_1 \\
c_2 &= x_2 + x_{10} + x_{14} + m_2 \\
c_3 &= x_3 + x_{11} + x_{15} + m_1 + m_3 \\
c_4 &= x_4 + x_{12} + m_4 \\
c_5 &= x_5 + x_{13} + m_3 + m_5 \\
c_6 &= x_6 + x_{14} + m_2 + m_6 \\
c_7 &= x_7 + x_{15} + m_1 + m_3 + m_5 + m_7 \\
c_8 &= x_8 + m_8 \\
c_9 &= x_9 + m_7 + m_9 \\
c_{10} &= x_{10} + m_6 + m_{10} \\
c_{11} &= x_{11} + m_5 + m_7 + m_9 + m_{11} \\
c_{12} &= x_{12} + m_4 + m_{12} \\
c_{13} &= x_{13} + m_3 + m_5 + m_{11} + m_{13} \\
c_{14} &= x_{14} + m_2 + m_6 + m_{10} + m_{14} \\
c_{15} &= x_{15} + m_1 + m_3 + m_5 + m_7 + m_9 + m_{11} + m_{13} + m_{15}
\end{aligned}
$$

Y finalmente se muestran las expresiones booleanas h para el caso de tamaño 15 de la función h.

$$
\begin{aligned}
l_1 &= x_1 + y_2 \\
l_2 &= y_1 + x_2 + y_3 \\
l_3 &= x_1 + x_3 + y_4 \\
l_4 &= y_1 + y_3 + x_4 + y_5 \\
l_5 &= x_1 + y_2 + x_3 + x_5 + y_6 \\
l_6 &= y_1 + x_2 + y_5 + x_6 + y_7 \\
l_7 &= x_1 + x_5 + x_7 + y_8 \\
l_8 &= y_1 + y_5 + y_7 + x_8 + y_9 \\
l_9 &= x_1 + y_2 + x_5 + y_6 + x_7 + x_9 + y_{10} \\
l_{10} &= y_1 + x_2 + y_3 + y_5 + x_6 + y_9 + x_{10} + y_{11} \\
l_{11} &= x_1 + x_3 + y_4 + x_5 + x_9 + x_{11} + y_{12} \\
l_{12} &= y_1 + y_3 + x_4 + y_9 + y_{11} + x_{12} + y_{13} \\
l_{13} &= x_1 + y_2 + x_3 + x_9 + y_{10} + x_{11} + x_{13} + y_{14} \\
l_{14} &= y_1 + x_2 + y_9 + x_{10} + y_{13} + x_{14} + y_{15} \\
l_{15} &= x_1 + y_{16} + x_9 + x_{13} + x_{15}
\end{aligned}
$$

CAPÍTULO B

Sensor Biométrico FPC1010

B.1 Especificaciones

- Sensor capacitivo de huellas dactilares de 200 x 152 pixeles.
- 3.3 Volts de operación.
- Mode selection switch.
- Use selection switch.
- Seis LEDs indicadores de status.

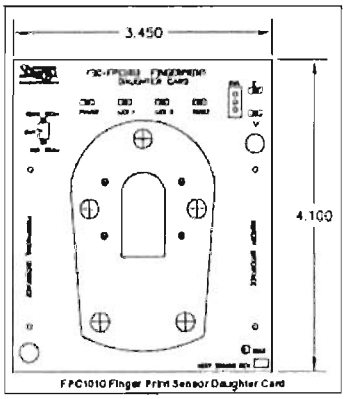

Figura B.1: Dimensiones del sensor

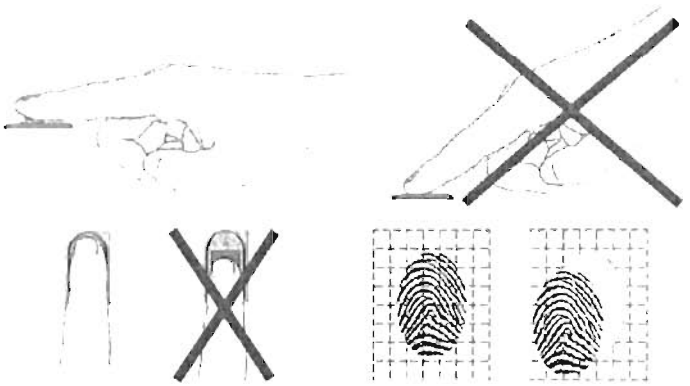

Figura B.2: Posición correcta del dedo sobre el sensor

Color space:	**RGB**
Interleaved data:	**YES**
Start address	**image** (image array defined in code)
Lines per display:	**200**
Pixels per line	**152**
Byte packing to fill 32 bits	**NO**
Bits per pixel	**8**
Palette Option:	**Gray scale of 256 colors**
Image Origin	**Top Left**
Everything else:	**Default**

Figura B.3: Configuración de imágenes digitales

CAPÍTULO C

Programas en LabVIEW

C.1 Envío de datos en la red GSM

En la figura C.1 se puede ver el programa que permite enviar la información de la PC1 hacia el módulo 1 a través del puerto serial.

La figura C.2 muestra el programa con el cual la PC2 lee la información proveniente del módulo 2 a través del puerto serial.

C.2 Autentificación con huellas digitales y su validación

La figura C.3 muestra el programa en LabVIEW que realiza la compresión y la encriptación de imágenes, la parte que realiza la encriptación esta formada por la funciones T y C.

En la figura C.4 se puede ver el programa que realiza la desencriptación de la imagen, incluye las funciones T y M para la generación de llaves y la desencriptación respectivamente.

La figura C.5 muestra el programa para enviar la imagen a través del puerto serial hacia el transceiver, contiene la configuración del puerto serial, y debido a que la información que se puede enviar por medio del transceiver es limitada, también contiene una función que divide la imagen para que pueda ser enviada en bloques.

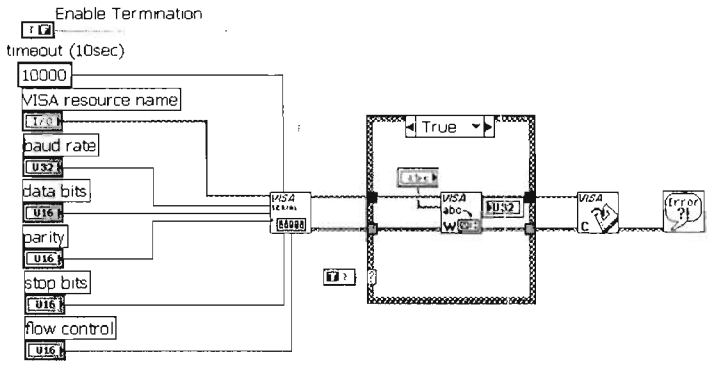

Figura C.1: Envío de información a través del puerto serial

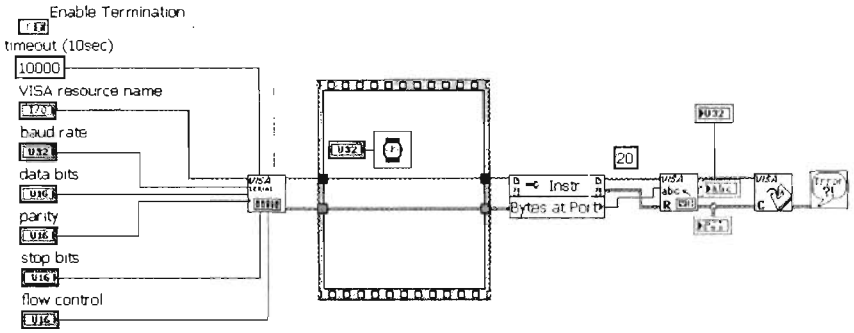

Figura C.2: Lectura de información del puerto serial

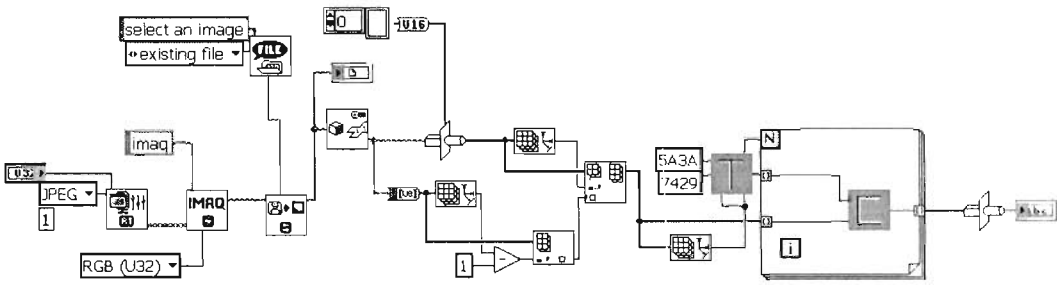

Figura C.3: Compresión y encriptación de la imagen

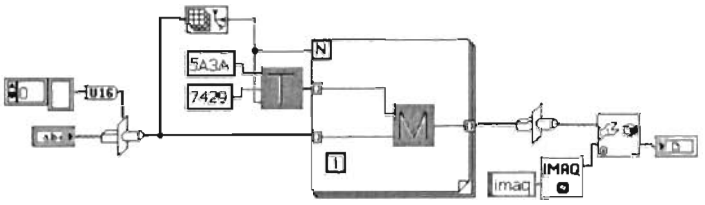

Figura C.4: Desencriptación de la imagen

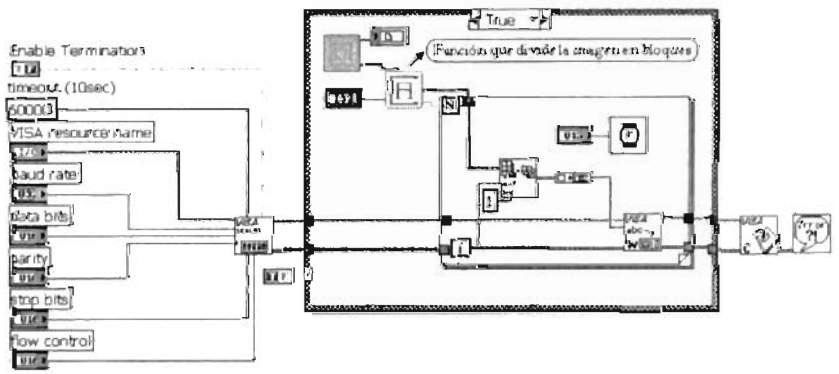

Figura C.5: Encriptación y envío de la imagen

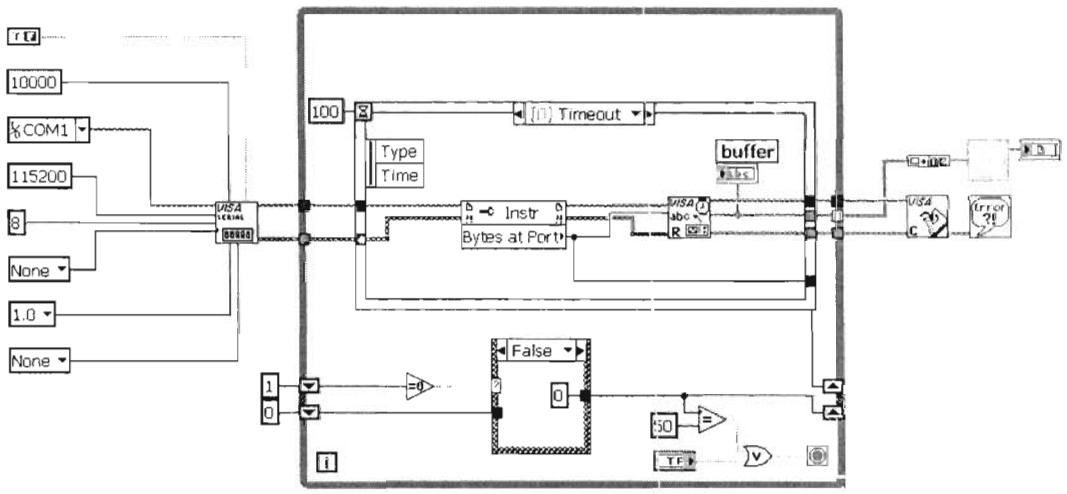

Figura C.6: Recepción y desencriptación de la imagen

La figura C.6 muestra el programa que recibe la imagen a través del puerto serial y la desencripta, ya que la imagen encriptada fue enviada en bloques, el ciclo permite juntar los bloques para poder desencriptar y obtener la imagen original.

CAPÍTULO D

Programas M2mpower IDE

A continuación se presenta el código de las tareas programadas en los módulos GT48.

Programa de encriptación y envío de SMS

Este programa lee datos del puerto serial, los encripta y los envía como un SMS.

```
/*Declaración de variables globales*/
char string[1000] = 0;
char array[1000] = 0;
int sx;
int sy;
int temp;
int i;
int a[160];
int sem1 = 75;
int sem2 = 35;

/*Función generadora de secuencias pseudoaleatorias*/
llaves(int x,int y)
{
 int t = 0;
 int x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15;
```

```
int y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15,y16;
int t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15;

x1 = x&1;
x2 = (x&2)>>1;
x3 = (x&4)>>2;
x4 = (x&8)>>3;
x5 = (x&16)>>4;
x6 = (x&32)>>5;
x7 = (x&64)>>6;
x8 = (x&128)>>7;
x9 = (x&256)>>8;
x10 = (x&512)>>9;
x11 = (x&1024)>>10;
x12 = (x&2048)>>11;
x13 = (x&4096)>>12;
x14 = (x&8192)>>13;
x15 = (x&16384)>>14;

y1 = y&1;
y2 = (y&2)>>1;
y3 = (y&4)>>2;
y4 = (y&8)>>3;
y5 = (y&16)>>4;
y6 = (y&32)>>5;
y7 = (y&64)>>6;
y8 = (y&128)>>7;
y9 = (y&256)>>8;
y10 = (y&512)>>9;
y11 = (y&1024)>>10;
y12 = (y&2048)>>11;
y13 = (y&4096)>>12;
y14 = (y&8192)>>13;
y15 = (y&16384)>>14;
y16 = (y&32768)>>15;

/*Expresiones booleanas de la función h*/
t1 = x1∧y2;
t2 = y1∧x2∧y3;
t3 = x1∧x3∧y4;
t4 = y1∧y3∧x4∧y5;
```

```
 t5 = x1^y2^x3^x5^y6;
 t6 = y1^x2^y5^x6^y7;
 t7 = x1^x5^x7^y8;
 t8 = y1^y5^y7^x8^y9;
 t9 = x1^y2^x5^y6^x7^x9^y10;
 t10 = y1^x2^y3^y5^x6^y9^x10^y11;
 t11 = x1^x3^y4^x5^x9^x11^y12;
 t12 = y1^y3^x4^y9^y11^x12^y13;
 t13 = x1^y2^x3^x9^y10^x11^x13^y14;
 t14 = y1^4x2^y9^x10^y13^x14^y15;
 t15 = x1^y16^x9^x13^x15;

 t = t|t1;
 t = t|(t2<<1);
 t = t|(t3<<2);
 t = t|(t4<<3);
 t = t|(t5<<4);
 t = t|(t6<<5);
 t = t|(t7<<6);
 t = t|(t8<<7);
 t = t|(t9<<8);
 t = t|(t10<<9);
 t = t|(t11<<10);
 t = t|(t12<<11);
 t = t|(t13<<12);
 t = t|(t14<<13);
 t = t|(t15<<14);

 return t;
}

/*Función que contiene el algoritmo de encriptación*/
C(int m,int t)
{
 int c = 0;
 int m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,m16;
 int c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16;
 int t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15;

 m1 = m&1;
 m2 = (m&2)>>1;
```

```
m3 = (m&4)>>2;
m4 = (m&8)>>3;
m5 = (m&16)>>4;
m6 = (m&32)>>5;
m7 = (m&64)>>6;
m8 = (m&128)>>7;
m9 = (m&256)>>8;
m10 = (m&512)>>9;
m11 = (m&1024)>>10;
m12 = (m&2048)>>11;
m13 = (m&4096)>>12;
m14 = (m&8192)>>13;
m15 = (m&16384)>>14;
m16 = (m&32768)>>15;

t1 = t&1;
t2 = (t&2)>>1;
t3 = (t&4)>>2;
t4 = (t&8)>>3;
t5 = (t&16)>>4;
t6 = (t&32)>>5;
t7 = (t&64)>>6;
t8 = (t&128)>>7;
t9 = (t&256)>>8;
t10 = (t&512)>>9;
t11 = (t&1024)>>10;
t12 = (t&2048)>>11;
t13 = (t&4096)>>12;
t14 = (t&8192)>>13;
t15 = (t&16384)>>14;

/*Expresiones boolenas de la función c*/
c1 = t1^t9^t13^t15^m2;
c2 = t2^t10^t14^m3;
c3 = t3^t11^t15^m2^m4;
c4 = t4^t12^m5;
c5 = t5^t13^m4^m6;
c6 = t6^t14^m3^m7;
c7 = t7^t15^m2^m4^m6^m8;
c8 = t8^m9;
```

```
 c9 = t9^m8^m10;
 c10 = t10^m7^m11;
 c11 = t11^m6^m8^m10^m12;
 c12 = t12^m5^m13;
 c13 = t13^m4^m6^m12^m14;
 c14 = t14^m3^m7^m11^m15;
 c15 = t15^m2^m4^m6^m8^m10^m12^m14^m16;
 c16 = m1;

 c = c|c1;
 c = c|(c2<<1);
 c = c|(c3<<2);
 c = c|(c4<<3);
 c = c|(c5<<4);
 c = c|(c6<<5);
 c = c|(c7<<6);
 c = c|(c8<<7);
 c = c|(c9<<8);
 c = c|(c10<<9);
 c = c|(c11<<10);
 c = c|(c12<<11);
 c = c|(c13<<12);
 c = c|(c14<<13);
 c = c|(c15<<14);
 c = c|(c16<<15);

 return c;
}

/*Función principal*/
main()
{
 char e[1000] = 0;
 int en[160] = 0;
 int enc[160] = 0;
 int aterr;
 int smserr;
 int j;
 int k;
 int I = 0;
 int OM = 0;
```

```
int S = 0;
int mn = 0;
int pdest;
int result;
int intero[160];
int INTERO[160] = 0;

/*Ciclo principal*/
while(1)
{
 u1o(0);  /*Open UART1*/

I = u1a();  /*Read number of bytes availables to be read from the UART1*/

 while(I < 1)
 {
  I = u1a();
 }

 u1r(c,1000);  /*Read data from the UART1 interface*/

/*Ciclo que checa si número de caracteres leídos del UART1 es par ó impar*/
 if(I%2 != 0)
 {
  OM = I+1;
  sncpy(c + (OM-1)," ",1);  /*Copy characters of one string to another*/
 }
 else
 {
  OM = I;
 }

 sx = sem1;
 sy = sem2;
 mn = 0;

 /*Ciclo que encripta los datos leídos del UART1*/
 for(i=0;i<OM/2;i++)
 {
  j = 2*i;
  k = 2*i+1;
```

```
        en[i] = (e[k]<<8)|(e[j]);

        a[i] = llaves(sx,sy);
        temp = sy&1;
        sy = sx|(temp<<15);
        sx = a[i];
        enc[i] = C(en[i],a[i]);

        itoa(enc[i], string, 100);  /*Convert an integer to string*/

        scat(array,string);  /*Append a string*/
        scat(array,"-");
      }

      u1c();  /*Close UART1*/
      mn = slen(array);  /*Get the length of a string*/

      aterr = atcrt();  /*Create an AT command channel*/
      smserr = smsi(0,0);  /*Initiailise Short Message Service (SMS)*/

      /*Send message*/
      smserr = smss("+524444266054",array,145,13,mn);  /*Send Short Message
Service (SMS) Message*/

      aterr = atdst();  /*Destroy an AT command channel*/
      }
  }
```

Programa de recepción y desencriptación de SMS

El siguiente programa lee un SMS, lo desencripta y lo envía a través del puerto serial.

```
/*Decalaración de variables globales*/
int sx;
int sy;
int temp;
int i;
int a[160];
int sem1 = 75;
int sem2 = 35;
```

```
int cont;
int CONT = 0;

/*Función generadora de secuencias pseudoaleatorias*/
llaves(int x,int y)
{
 int t = 0;
 int x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15;
 int y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15,y16;
 int t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15;

 x1 = x&1;
 x2 = (x&2)>>1;
 x3 = (x&4)>>2;
 x4 = (x&8)>>3;
 x5 = (x&16)>>4;
 x6 = (x&32)>>5;
 x7 = (x&64)>>6;
 x8 = (x&128)>>7;
 x9 = (x&256)>>8;
 x10 = (x&512)>>9;
 x11 = (x&1024)>>10;
 x12 = (x&2048)>>11;
 x13 = (x&4096)>>12;
 x14 = (x&8192)>>13;
 x15 = (x&16384)>>14;

 y1 = y&1;
 y2 = (y&2)>>1;
 y3 = (y&4)>>2;
 y4 = (y&8)>>3;
 y5 = (y&16)>>4;
 y6 = (y&32)>>5;
 y7 = (y&64)>>6;
 y8 = (y&128)>>7;
 y9 = (y&256)>>8;
 y10 = (y&512)>>9;
 y11 = (y&1024)>>10;
 y12 = (y&2048)>>11;
 y13 = (y&4096)>>12;
 y14 = (y&8192)>>13;
```

```
y15 = (y&16384)>>14;
y16 = (y&32768)>>15;

/*Expresiones booleanas de la función h*/
t1 = x1∧y2;
t2 = y1∧x2∧y3;
t3 = x1∧x3∧y4;
t4 = y1∧y3∧x4∧y5;
t5 = x1∧y2∧x3∧x5∧y6;
t6 = y1∧x2∧y5∧x6∧y7;
t7 = x1∧x5∧x7∧y8;
t8 = y1∧y5∧y7∧x8∧y9;
t9 = x1∧y2∧x5∧y6∧x7∧x9∧y10;
t10 = y1∧x2∧y3∧y5∧x6∧y9∧x10∧y11;
t11 = x1∧x3∧y4∧x5∧x9∧x11∧y12;
t12 = y1∧y3∧x4∧y9∧y11∧x12∧y13;
t13 = x1∧y2∧x3∧x9∧y10∧x11∧x13∧y14;
t14 = y1∧x2∧y9∧x10∧y13∧x14∧y15;
t15 = x1∧y16∧x9∧x13∧x15;

t = t|t1;
t = t|(t2<<1);
t = t|(t3<<2);
t = t|(t4<<3);
t = t|(t5<<4);
t = t|(t6<<5);
t = t|(t7<<6);
t = t|(t8<<7);
t = t|(t9<<8);
t = t|(t10<<9);
t = t|(t11<<10);
t = t|(t12<<11);
t = t|(t13<<12);
t = t|(t14<<13);
t = t|(t15<<14);

 return t;
}

/*Función que contiene el algoritmo de desencriptación*/
M(int c,int t)
```

```
{
 int m = 0;
 int c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16;
 int m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,m16;
 int t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15;

 c1 = c&1;
 c2 = (c&2)>>1;
 c3 = (c&4)>>2;
 c4 = (c&8)>>3;
 c5 = (c&16)>>4;
 c6 = (c&32)>>5;
 c7 = (c&64)>>6;
 c8 = (c&128)>>7;
 c9 = (c&256)>>8;
 c10 = (c&512)>>9;
 c11 = (c&1024)>>10;
 c12 = (c&2048)>>11;
 c13 = (c&4096)>>12;
 c14 = (c&8192)>>13;
 c15 = (c&16384)>>14;
 c16 = (c&32768)>>15;

 t1 = t&1;
 t2 = (t&2)>>1;
 t3 = (t&4)>>2;
 t4 = (t&8)>>3;
 t5 = (t&16)>>4;
 t6 = (t&32)>>5;
 t7 = (t&64)>>6;
 t8 = (t&128)>>7;
 t9 = (t&256)>>8;
 t10 = (t&512)>>9;
 t11 = (t&1024)>>10;
 t12 = (t&2048)>>11;
 t13 = (t&4096)>>12;
 t14 = (t&8192)>>13;
 t15 = (t&16384)>>14;

 /*Expresiones booleanas de la función m*/
 m1 = c16;
```

```
 m2 = t15^t13^t9^t1^c1;
 m3 = t14^t10^t2^c2;
 m4 = t13^t11^t9^t3^t1^c1^c3;
 m5 = t12^t4^c4;
 m6 = t11^t9^t5^t3^t1^c1^c3^c5;
 m7 = t10^t6^t2^c2^c6;
 m8 = t9^t7^t5^t1^c1^c5^c7;
 m9 = t8^c8;
 m10 = t7^t5^t1^c1^c5^c7^c9;
 m11 = t6^t2^c2^c6^c10;
 m12 = t5^t3^t1^c1^c3^c5^c9^c11;
 m13 = t4^c4^c12;
 m14 = t3^t1^c1^c3^c9^c11^c13;
 m15 = t2^c2^c10^c14;
 m16 = t1^c1^c9^c13^c15;

 m = m1;
 m = m|(m2<<1);
 m = m|(m3<<2);
 m = m|(m4<<3);
 m = m|(m5<<4);
 m = m|(m6<<5);
 m = m|(m7<<6);
 m = m|(m8<<7);
 m = m|(m9<<8);
 m = m|(m10<<9);
 m = m|(m11<<10);
 m = m|(m12<<11);
 m = m|(m13<<12);
 m = m|(m14<<13);
 m = m|(m15<<14);
 m = m|(m16<<15);

 return m;
}

/*Función principal*/
main()
{
 char array[1000] = 0;
 char strr[1000] = 0;
```

```
char arraytemp[1000] = 0:
char arreglo8[160] = 0;
int D[160];
int d[160];
int j;
int k;
int mn;
int S;
int SE;
int len;
int pdest;
int result;
int aterr;
int smsslot;
int smserr;
int smser;

/*Ciclo principal*/
while(1)
{
 aterr = atcrt();  /*Create an AT command channel*/

 /*Find sms message*/
  smsslot = smsrs();   /*Find first Un-read Short Message Service (SMS)
Message from memory*/

 /*Ciclo que checa si ya se encontro un SMS*/
 while(smsslot == 0)
 {
  smsslot = smsrs():
 }

 /*Read sms message data*/
 smserr = smsrm(array,160,smsslot);
 aterr = atdst();  /*Destroy AT channel*/

 mn = slen(array);  /*Get the length of a string*/

 sx = sem1;
 sy = sem2;
 cont = 0;
```

```
/*Ciclo que convierte el string recibido (SMS) en un entero*/
for(i=0;i<mn/2;i++)
{
 pdest = sstr( array,"");  /*Find a substring*/
 result = (pdest - array) + 1;

 sncpy(strr,array,mn-1);  /*Copy characters of one string to another*/

 D[i] = atoi(strr);  /*Convert string to integer*/

 while(D[i] != 0)
 {
  cont++;
  break;
 }

 sncpy(arraytemp,array + result,mn);

 array = arraytemp;
}

CONT = cont*2;

/*Ciclo que desencripta el SMS*/
for(i=0;i<cont;i++)
{
 j = 2*i;
 k = 2*i+1;

 a[i] = llaves(sx,sy);
 temp = sy&1;
 sy = sx-(temp<<15);
 sx = a[i];

 d[i] = M(D[i],a[i]);

 arreglo8[j] = d[i]&255;
 arreglo8[k] = (d[i]>>8)&255;
}

u1o(0);  /*Open UART1*/
u1cf(3,0);  /*Configure the UART1 interface*/
```

```
    len = u1f();   /*Reads the free space in the UART1 interface Tx buffer*/

    /*Ciclo que envia los datos a través del UART1*/
    if(u1s(arreglo8,CONT))
    {
     while(u1f() < len)
     {
      prtf(/nTransmiting");
     }
     prtf(/nTx Complete!");
    }
    u1c();   /*Close UART1*/
  }
}
```

Bibliografía

[1] Wayne Tomasi, *Advanced electronic communications systems*. Sixth Edition. Prentice Hall, New Jersey (2004).

[2] Marcela Mejía Carlos, *Encriptación por Sincronización en Autómatas Celulares*. (2001).

[3] María del Carmen Rodríguez Aranda, *Sistema de Desarrollo Inalámbrico para Monitoreo de Procesos e Identificación Personal*. (2005).

[4] Marc Van Droogenbroeck and Raphael Benedett, *Techniques for a Selective Encryption of Uncompressed and Compressed Images*. ACIVS, Ghent, Belgium, September 9-11 (1995).

[5] Mark Van Droogenbroeck, *Partial Encryption of Images for Real-Time Applications* (1995).

[6] Manual, *GT47/GT48 Technical Description*, June (2003)

[7] Manual, *M2mpoer Aplication Guide*, June (2003).

[8] Mikko Martsola, *Machine to machine communication in cellular network*, May. (2004).

[9] Martina Podesser, Hans-Peter Schmidt and Andreas Uhl, *Selective Bitplane Encryption for Secure Transmission of Image Data in Mobole Environments*, School of Telematics & Network Engineering Carinthia Tech Institute, Klagenfurt, AUSTRIA.

[10] J.A. Muñoz Rodríguez and R. Rodríguez-Vera, *Image encryption based on phase encoding by means of a fringe pattern and computational algorithms*, Revista Mexicana de Física (52) 53-63. Enero (2006).

[11] Randall K. Nichols, *ICSA Guide to Cryptography.*