INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS MONTERREY

PROGRAMA DE GRADUADOS EN TECNOLOGÍAS DE
INFORMACIÓN Y ELECTRÓNICA

T E S I S

MAESTRÍA EN CIENCIAS EN SISTEMAS INTELIGENTES

Generación de Claves Criptográficas basadas en el "FingerCode" de Huellas Dactilares

por

José Abdón Ramírez Ruiz

Monterrey, N.L., Diciembre de 2005

©José Abdón Ramírez Ruiz, 2005.

Generación de Claves Criptográficas basadas en el "FingerCode" de Huellas Dactilares

por

José Abdón Ramírez Ruiz

T e s i s

Presentada al Programa de Graduados en Tecnologías de Información y Electrónica

del

Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey

como requisito parcial para obtener el grado académico de

Maestro en Ciencias

en

Sistemas Inteligentes

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

Monterrey, N.L., Diciembre de 2005

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

División de Graduados en Tecnologías de Información y Electrónica
Programa de Graduados en Tecnologías de Información y Electrónica

Los miembros del comité de tesis recomendamos que la presente tesis del M.C. José Abdón Ramírez Ruiz sea aceptada como requisito parcial para obtener el grado de **Maestro en Ciencias en Sistemas Inteligentes**.

Comité de Tesis

Dr. Carlos Pfeiffer Celaya
Asesor principal

Dr. Juan Arturo Nolazco F.
Sinodal

Dr. Carlos Mex Perera
Sinodal

Dr. David A. Garza Salazar
Director del Programa de Graduados en Tecnologías de Información y Electrónica

Diciembre de 2005

Resumen

El presente trabajo fue motivado por posibilidad actualmente abierta en la comunidad científica de utilizar biométricas para cifrar y descifrar información de una manera segura y confiable. Así, nos enfocamos en investigar la posibilidad de hacerlo utilizando como biométrica a la huella dígital. Por lo tanto la idea fue obtener con algún generador clásico de claves criptográficas, claves que sean confiables y posteriormente asignárselas a los patrones de las huellas pertenecientes a los individuos correspondientes utilizando para dicho fin algún clasificador de patrones.

En la huella dígital nos concentramos básicamente en sus patrones de textura mismos que fueron extraídos por un proceso conocido como FingerCode que utiliza un banco de 8 filtros de Gabor. De tal manera que la huella primero es dividida en pequeñas regiones y luego cada región es filtrada con cada uno de los filtros y se le calculada su desviación absoluta promedio que da origen a los vectores de caractísticas utilizados.

Como clasificador se utilizó una Máquina de Vectores de Soporte (SVM) ideada por Vapnik que ha mostrado tener buenos resultados en trabajos dirigidos en la misma línea hechos con voz.

Así, se propusieron varios modelos para resolver el problema que utilizaron un grupo de SVM's en una configuración muy especifica y que trabajan sobre el espacio de características de textura de la huella dígital mapeandolas al espacio de claves criptográficas.

Con la utilización de los modelos propuestos se lograron eficiencias mínimas del 70 %, que nos pareció bastante bueno como un comienzo en esta dirección, sin embargo, existen aún vulnerabilidades importantes en cuestión de seguridad. Por lo tanto, consideramos que esto es apenas el comienzo y que falta mucho trabajo por hacer.

A ese grandioso sueño que me trajo hasta aquí
Que ha sido la fuente de mi pasión e impulso
Que ha cobrado la vida de varios hombres
Que espera cobrar la vida de varios más
En la espera de que alguien al fín la descubra.

La Inteligencia Artificial.

Agradecimientos

A mis padres por el apoyo incondicional que siempre me han brindado.

A mis hermanas que aunque estaban lejos siempre me enfocaban por el mejor camino.

A mis amigos que me daban animos.

A mi amigo Iván López M. cuyos conocimientos en Latex fueron decisivos para la culminación de esta tesis, por la gran cantidad de movies que vimos para distraer la mente

A mi asesor el Dr. Carlos Pfeiffer Celaya cuyas ideas a este trabajo fueron claves.

A mis sinodales el Dr. Juan Arturo Nolazco y el Dr. Carlos Mex por realizar una revisión del trabajo y contribuir al mejoramiento de este.

A mis perros el Maxino y el Tomate que cuando las dudas abrumaban a mi mente, ellos venian y me explicaban.

Contenido

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En la actualidad el incremento en la manipulación de la información de manera electrónica se ha convertido en una manera muy eficiente de tratar con datos. Esto se ha remarcado aún más con el desarrollo de internet que permite la conectividad entre grandes cantidades de ordenadores en diferentes partes del mundo. Así, ha sido utilizado como un medio en el cual no sólo se presenta información como lo fue desde sus inicios, sino que también permite la transmisión de datos entre clientes y prestadores de servicios como lo son los bancos, supermercados, librerías, etc. Este tipo de información en la mayoría de los casos es información confidencial por lo cual se ha motivado a la utilización de la criptografía, que se encarga de codificar y decodificar información, para mantener dicha confidencialidad. Para que tal codificación y decodificación sea llevada a cabo se requiere de la generación y utilización de llaves que son conocidas como claves criptográficas. Así, existen dos tipos de criptografía que difieren en la manera en como son generadas y usadas tales claves, que son:

Clave Secreta. Estos sistemas se basan en la generación de una clave que sirve para cifrar como para descifrar información.

Clave Pública. Estos sistemas se basan en una clave llamada privada que se utiliza para generar otra clave llamada pública. Con la clave pública se cifra el documento y con la clave privada se descifra o viceversa. Por esta razón se conocen como sistemas asimétricos.

Las claves que se utilizan son en general largas cadenas de bits que regularmente tienen que ser memorizados por los usuarios, puestos en tarjetas o almacenados en discos digitales. Por esta razón son susceptibles de violación porque pueden ser extraviadas, robadas u olvidadas.

Así, en los últimos años se han estado haciendo esfuerzos en la posibilidad de asignar dichas claves a patrones que sean abstraídos de las características físicas o de comportamiento propias de los individuos y que son conocidas como Biométricas ya que se sabe presentan patrones que caracterizan a cada individuo como único, de entre las cuales, por mencionar algunas tenemos: la Huella Digital, el Iris, la Retina, la Voz,

el Rostro, la Geometría de la mano, las Firmas, las Venas de la Mano, los Oídos, la Forma de Caminar, el Olor, etc. [15]. Las biométricas solventan el hecho de que no pueden ser olvidadas porque en principio nunca son aprendidas, pero si pueden ser duplicadas o robadas dado que no son secretas [16] ya que dichas biométricas están al alcance de todos, es decir, cualquiera podría obtener una muestra de voz grabándola, o de la huella digital a partir de los objetos que tocamos a diario, etc., de otra persona. Por esta razón, tales biométricas tienen que ser combinadas junto con algo que si sea secreto. Sin embargo, a pesar de la no confidencialidad de las biométricas, existe aún, un problema más grave que es difícil de resolver y que tiene que ver con la manera de lidiar de forma eficiente con las variaciones que presentan las biométricas obtenidas de dichos individuos, es decir, la abstracción de patrones que es obtenida de las biométricas no siempre es igual, lo cual es debido tanto a errores causados por los aparatos de medición (como imprecisión en la medición debido a la resolución, entrada incorrecta de las biométricas al aparato, etc.) como a variaciones que el organismo sufre con el tiempo tales como deformaciones, accidentes, enfermedades, etc. Estas variaciones pueden ser pequeñas o grandes, y en el caso en particular en el que son grandes, el problema se vuelve más complejo. Esto complica las cosas, porque es difícil asignar una clave precisa a un individuo cuando existe dificultad para identificarlo, de tal manera que, esto implicaría que se pudiera cifrar pero no descifrar debido a no poder recuperar la llave con la precisión necesaria ya sea porque se cifro o se desea descifrar con una llave errónea o ambas.

De todas las biométricas posibles las más estudiadas son iris y huella digital. La huella digital es un problema más complejo que la utilización del iris, ya que el iris presenta una gran estabilidad a las variaciones [8], misma que no sucede con la huella. Así, nuestro enfoque está dirigido hacia la huella dígital.

Han habido varios trabajos que tratan este problema con huella digital de entre los cuales se puede citar a Soutar et al. [19] quienes desarrollaron un sistema que asigna claves criptográficas basadas en la huella digital, sin embargo, tiene el inconveniente de que primero identifica al usuario y después de dicha identificación se libera una clave criptográfica ligada a la persona que esta almacenada en una base de datos que permite codificar o decodificar según sea el caso, lo cual lo hace inseguro debido a que terceros podrían obtener la clave de la base de datos y utilizarlas con otros fines. Clancy et al [5] propusieron la generación de claves a través de la identificación de minucias en la huella. Estas minucias son identificadas por comparación con las minucias de la huella almacenada del usuario por cercanía y se genera la clave criptográfica almacenada en un polinomio oculto generando con ello resultados con una variabilidad del 20 al 30 por ciento.

En este trabajo se generan claves criptográficas que son asignadas mediante un clasificador a patrones que están basados en la textura extraída de pequeñas regiones de la huella digital, mismas que caracterizan a un individuo a través de un código vectorial conocido como FingerCode. Para obtener este código se utilizan filtros

pasabanda direccionales. Los filtros utilizados aquí son los denominados filtros de Gabor [2] que se describirán con detalle mas adelante. El clasificador utilizado fue una Máquina de Vectores de Soporte (SVM) ideada por Vapnik [20], ya que el proceso de entrenamiento garantiza encontrar una solución única puesto que está planteado como un problema de programación cuadrática. Además, fue implementada con la obtención de buenos resultados en trabajos dirigidos en esta área pero utilizando como biométrica los patrones de voz por el grupo de investigaciones en seguridad del ITESM [13].

En esta investigación se pretendió explorar la posibilidad de utilizar tanto al SVM como al FingerCode para la asignación de claves criptográficas en huella digital.

1.1 Definición del Problema

El problema es poder asignar claves criptográficas, construidas a partir de un generador de claves, a individuos utilizando para dicho fin sus correspondientes patrones de textura de sus huellas digitales, de tal manera que, cuando tales individuos deseen cifrar o descifrar información privada utilicen únicamente su huella digital.

1.2 Objetivos

El objetivo general es investigar la posibilidad de utilizar los patrones de textura obtenidos de la huella digital (FingerCode) para la asignación de claves criptográficas a individuos mediante un clasificador de patrones conocido como Máquina de Vectores de Soporte (SVM). Los objetivos particulares a cumplir en este trabajo de investigación son los siguientes:

a. Implementar y determinar los parámetros adecuados del algoritmo para la extracción de características de textura de la huella digital (FingerCode) propuesto en [14]. Luego, aplicarlo sobre un conjunto de huellas obtenidas bajo ciertas condiciones de las siguientes bases de datos [3]:

> La número 4 de Nist, y las utilizadas en los concursos organizados por Maltoni en la universidad de Bolonia (2001 and 2002 Finger Print Verification Contest (FPVC) Data Bases).

b. Encontrar un modelo adecuado para la asignación de claves criptográficas a los códigos de textura obtenidos utilizando SVM's y aplicarlo.

c. Caracterizar el desempeño del sistema, definir sus vulnerabilidades y proponer modelos que mejoren el desempeño.

d. Proponer una aplicación de dichos modelos a Digital Ridge Managment.

1.2.1 Alcances

En el presente trabajo no se pretende resolver el problema en su totalidad, cosa que sería demasiado ambiciosa debido a la complejidad del mismo. Tampoco se pretende brindar solución total a todos los problemas de vulnerabilidad existentes en los modelos propuestos, sino que más bien, deseamos explorar el espacio del problema para entenderlo mejor e ir mejorándo paso a paso, atacando una parte del problema a la vez. De esta manera, se comienza por lidiar con la variabilidad de los patrones de textura con un primer modelo y se avanza atacando una de las vulnerabilidades descubiertas en dicho modelo construyendo otro.

1.3 Hipótesis

Es posible construir sistemas basados en Maquinas de Vectores de Soporte (SVM‘s) que sean capaces de asignar claves criptográficas a patrones de textura de huellas digitales conocidos como FingerCode’s.

1.3.1 Justificación

Se ha visto que utilizar clasificadores de patrones en problemas diversos donde la variabilidad juega un papel preponderante ha resultado ser muy satisfactorio. La razón de porque utilizar SVM‘s en lugar de otros clasificadores como por ejemplo redes neuronales, etc., es porque los SVM‘s convierten los problemas de clasificación en problemas de optimización cuadrática que son fáciles de resolver y que tienen una solución única. Además de que han tenido mucho éxito en aplicaciones muy diversas y de que en particular han sido utilizados para resolver el mismo problema planteado en este trabajo pero utilizando voz como biométrica. Finalmente, la utilización de este tipo de clasificadores sobre los FingerCodes obtenidos de huellas digitales es algo que no se ha explorado antes y podría ser beneficioso.

1.3.2 Preguntas de Investigación

¿Es posible construir sistemas que asignen claves criptográficas a FingerCodes utilizando modelos que contienen SVM‘s?

¿Es clasificable el FingerCode a través de SVM‘s y en que Kernel?

¿Si el FingerCode es clasificable a través de SVM‘s entonces que porcentajes de error existen?

¿Qué nivel de seguridad se puede lograr?

1.4 Contribución

La contribución del presente trabajo consiste en implementar y probar nuevas ideas desarrollando experimentos que permitan identificar características nuevas en el problema y guíen hacia nuevos posibles caminos en la búsqueda de dicha solución utilizando para tal fin clasificadores (SVM's) de patrones y las características de textura de la huella digital que es extraída a través de la utilización de filtros de Gabor que han mostrado ser muy útiles en este tipo de análisis.

1.5 Organización de la Tesis

A continuación se presenta una guía general del contenido de la presente tesis:

En el capitulo 2 se describen los antecedentes teóricos utilizados en el desarrollo de esta investigación, que describen: la estructura de la huella digital, los filtros de Gabor que fueron utilizados para la obtención del FingerCode, como se obtiene el FingerCode, la criptografía de clave secreta que fue la que se utilizó como modelo para cifrar, la estructura de un criptosistema biométrico y finalmente el funcionamiento del SVM que fue utilizado para la asignación de claves criptográficas a patrones de textura provenientes de la huella digital de los individuos en cuestión.

En el capitulo 3 se presentan los modelos propuestas para los sistemas biométricos describiendo a detalle su funcionamiento y sus vulnerabilidades.

En el capitulo 4 se describen las características de implementación para la obtención del FingerCode y de los modelos propuestos. También se detallan las características de los experimentos realizados y el tipo de mediciones realizadas.

En el capitulo 5 se esbozan los resultados obtenidos en los experimentos descritos en el capitulo 4 así como un análisis de los mismos.

En el capítulo 6 se describe como los modelos aquí descritos pueden ser utilizados en administración de derechos digitales (DRM).

Finalmente se presentan las conclusiones del trabajo realizado y sus resultados. Además se mencionan las pautas a seguir en trabajos futuros.

Capítulo 2

Marco Teórico

Para poder realizar la obtención de claves criptograficas a través de la huella digital es necesario primero entender cuales son la características que identifican a un individuo como único, como lo son las minucias y la textura, también es necesario saber como extraerlas, y con que grado de precisión sucede esta extracción, y finalmente entender como se puede asignar una clave criptográfica a partir de la información obtenida. A continuación se presentan una serie de teorías necesarias que nos permitieron desarrollar los objetivos de este trabajo.

2.1 Características de la Huella Dígital

La huella digital esta formada por un patrón de valles y crestas [1] representadas por líneas negras y blancas respectivamente como se muestra en la figura 2.1.

Figura 2.1: Patrón de valles y crestaso.

El centro de la huella se define como el punto donde existe la máxima curvatura de valles y crestas [14] con concavidad cóncava cuando la huella se encuentra vertical perfectamente alineada. En la figura 2.2 se muestra el centro de dos huellas marcados con cruces.

Figura 2.2: Las cruces blancas determinan los centros de las huellas.

La identificación de individuos debido a las características de las huellas esta basada en dos tipos de patrones diferentes que la huella presenta: Minucias [10] y Textura [9]. Las Minucias consisten en ciertas anormalidades que aparecen en las crestas: terminación y bifurcación. Una terminación es cuando una cresta se corta y una bifurcación es cuando se separa en dos. Estos se puede observar en la figura 2.3.

Figura 2.3: Minucias.

Para identificar a un individuo utilizando minucias primero se localizan en una huella como se muestra en la Figura 2.4 y posteriormente por comparación de posiciones se determina si es o no el individuo correcto.

Figura 2.4: Minucias de una huella digital.

Con respecto a la Textura, se puede ver que si se observan pedazos lo suficientemente pequeños del patrón de la huella, se podrá notar que se identifican estructuras periódicas que consisten generalmente de una secuencia de líneas paralelas con cierta orientación como se muestra en la figura 2.5. La combinación de todas estas pequeños texturas se utilizan también para extraer características.

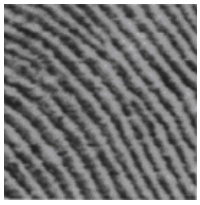

Figura 2.5: Crestas en una pequeña región de una huella digital.

En realidad no se utiliza toda la huella completa para caracterizar su textura, sino que en lugar de ello se definen ciertas regiones que son de interés. La manera de hacerlo es como se muestra en la Figura 2.6 donde se muestra una pequeña porción de la huella dividida en sectores concéntricos alrededor del centro. Cada región de estas tiene una textura aproximadamente uniforme.

Figura 2.6: Regiones de interés para caracterizar la textura de una huella dígital

La identificación de una persona se logra comparando la similaridad de los coeficientes obtenidos al aplicar un banco de filtros de Gabor sensibles a los cambios de textura de la huella en todas las regiones de interés.

Para poder hacer identificación automática de las características de la huella mencionadas anteriormente lo que se hace es obtener una escaneo de la huella digital. En dicho escaneo lo que se tiene es una imagen digital que contiene los patrones de la huella. Esta imagen es monocromática, es decir, cada píxel solo puede tener un valor entre 0 y 255 que indican blanco y negro respectivamente. Así, la imagen luego es tratada para la extracción de las minucias o texturas, según sea el caso, a través de técnicas de visión artificial que determinan al individuo en cuestión.

2.2 Filtros de Gabor

La textura de las imágenes actualmente juega un papel muy importante en una gran diversidad de tareas, de las cuales por mencionar algunas tenemos: a las imágenes médicas, que muchas veces son utilizadas para el diagnóstico de alguna enfermedad, a las imágenes de huella digital, que son utilizadas para identificación de personas, etc. Así, el análisis de la textura es un factor importante para clasificar, segmentar y reconocer las características más significativas de una imagen. Los filtros de Gabor han probado tener gran éxito en la identificación de dichas texturas debido al manejo del filtrado espacial y a la consideración de bajas y altas frecuencias que son de gran importancia en el análisis de texturas.

Un filtro de Gabor [2] es un filtro paso banda que selecciona un cierto rango de longitudes de onda. Los filtros vienen dados por la siguiente expresión:

$$G(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\mathbf{x}^t\mathbf{A}\mathbf{x}} e^{i\mathbf{k}^t\mathbf{x}} \tag{2.1}$$

Donde A está compuesta por una matriz de parámetros P diagonal y una matriz de rotación R:

$$\mathbf{A} = \mathbf{RPR}^t = \begin{bmatrix} \cos\varphi & -sen\varphi \\ sen\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} \sigma_x^{-2} & 0 \\ 0 & \sigma_y^{-2} \end{bmatrix} \begin{bmatrix} \cos\varphi & sen\varphi \\ -sen\varphi & \cos\varphi \end{bmatrix} \tag{2.2}$$

Con x^t = (x,y). Eligiendo $k^t = [k_0\ cos\ \phi\ ,\ k_0\ sen\ \phi\ ]$ se obtiene un conjunto de filtros con diferentes frecuencias. Además se deben seleccionar los parámetros de la matriz P para definir el ancho de banda. La exponencial compleja dada en 2.1 puede descomponerse en una parte real y una imaginaria como sigue:

$$\begin{aligned} G(x,y) &= \tfrac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\mathbf{x}^t\mathbf{A}\mathbf{x}} \cos(\mathbf{k}^t\mathbf{x}) \\ G(x,y) &= \tfrac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\mathbf{x}^t\mathbf{A}\mathbf{x}} sen(\mathbf{k}^t\mathbf{x}) \end{aligned} \tag{2.3}$$

Que origina las correspondientes oscilaciones del filtro. Variando el valor de k_0 se obtienen diferentes frecuencias de oscilación y con diferentes valores de ϕ se obtienen diferentes orientaciones del filtro. En la figura 2.7 se muestra una representación 3D de una función de Gabor.

Estas funciones de onda se discretizan en un reticulado bidimensional generando una matriz de números que es conocida como mascara que regularmente son matrices de 3x3, sin embargo, pueden ser mas grandes. Esta máscara es entonces pasada localizando su centro en cada punto de la imagen a la que se le desea aplicar el filtro haciendo un barrido de izquierda a derecha y de arriba hacia abajo. Cada punto de la imagen es remplazado por la suma ponderada de los valores de la imagen con los valores de la matriz correspondientes a la máscara, operación conocida como correlación. Así, se obtiene una nueva imagen que es la correspondiente filtrada a la original. Estos filtros se utilizan especialmente porque eliminan las partes de la imagen de una huella digital

que no presentan líneas paralelas con cierta separación y orientación a las definidas por el filtro.

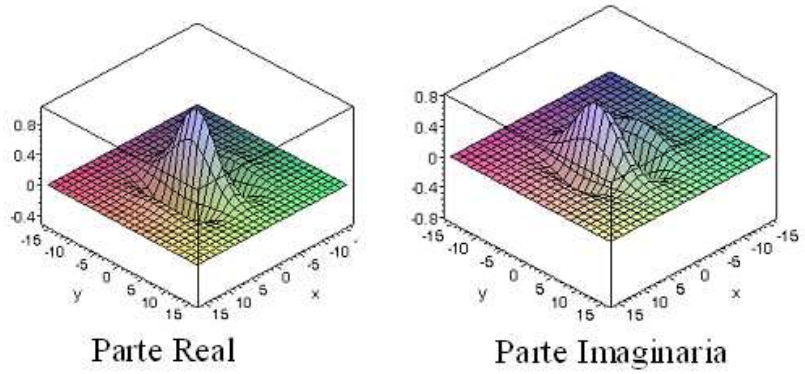

Figura 2.7: Una onda de Gabor con $\phi = 0$, k = $2\pi/10$ y σx = σy = 4

2.3 Extracción de Textura de Huella(FingerCode)

Para caracterizar la textura de la huella primero se debe de contar con una imagen digital de la huella deseada en una escala de grises. Luego, se localiza su centro. A partir de ese punto, tomado como referencia, se definen las pequeñas regiones de interés que denominaremos sectores Si con i = 1,2,. . .,n como se muestra en la figura 2.8. Obsérvese que los sectores son concéntricos al centro de la huella.

Figura 2.8: Regiones de interés para caracterizar la textura de una huella dígital

Cada sector debe ser normalizado por separado para poder compensar distorsiones generadas por el filtro de Gabor. La normalización se lleva a cabo de la siguiente manera: dejemos que I(x,y) denote el valor de gris en el píxel (x,y), M_i y V_i sean el promedio y la varianza respectivamente en el sector S_i y N(x,y) el valor de gris normalizado en el píxel (x,y). Entonces para todos los píxeles en el sector Si la imagen normalizada esta definida como:

$$N(x,y) = \begin{cases} Mo + \sqrt{\frac{Vo(I(x,y)-M_i)^2}{V_i}}, \text{si } I(x,y) > M_i \\ Mo - \sqrt{\frac{Vo(I(x,y)-M_i)^2}{V_i}}, \text{de lo contrario.} \end{cases} \quad (2.4)$$

Donde M_o y V_o son los valores deseados del promedio y la varianza respectivamente. El efecto de esta normalización es ajustar los valores de gris de la imagen para que tengan un valor promedio y varianza por sector iguales a M_o y V_o respectivamente. En la figura 2.9 se muestra una huella junto a su respectiva normalización.

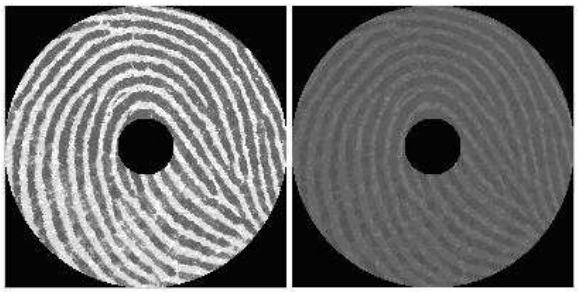

Figura 2.9: Regiones de interés para caracterizar la textura de una huella dígital

continuación, se filtra la huella normalizada con 8 filtros de Gabor cada uno con una dirección diferente $\phi = 0°$,22.5° ,45° ,67.5° ,90° ,112.5° ,135° ,157.5°. En la Figura 2.10 se muestran las 8 funciones de Gabor utilizadas en una escala de grises. Como se puede observar cubren todas las posibles orientaciones debido a su simetría en pasos discretos de 22.5°. También es de observarse que solo se utiliza la parte real del filtro de Gabor.

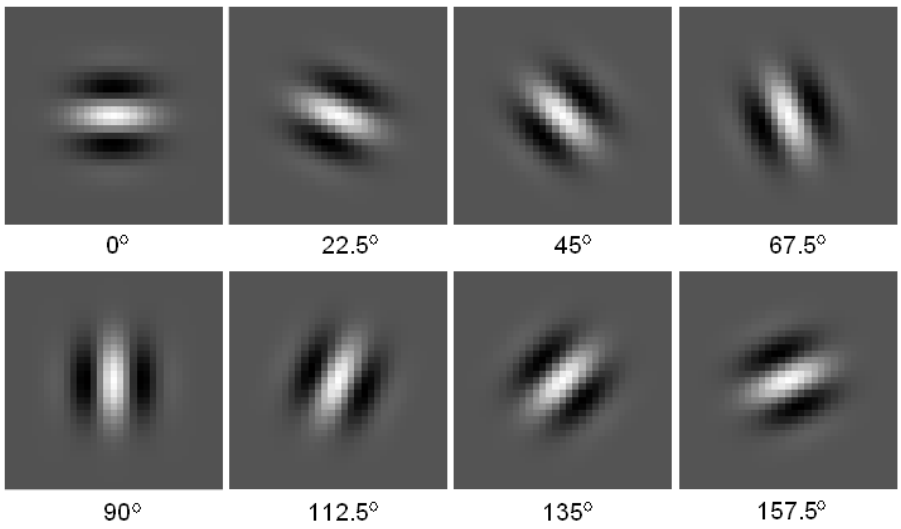

Figura 2.10: Funciones de los filtros de Gabor.

Estas funciones aquí mostradas están rotadas 90° con respecto a la expresión mostrada en la ecuación 2.3 y multiplicada por la constante $a = 2\pi\sigma_x\sigma_y$, por lo tanto realizando dicha rotación y multiplicando por la constante a, se obtiene la siguiente forma funcional para los filtros de Gabor utilizados aquí:

$$\begin{aligned} G(x,y) &= e^{-\frac{1}{2}\left[\frac{x'^2}{\sigma_x^2}+\frac{y'^2}{\sigma_y^2}\right]}\cos(kx') \\ \text{con}: x' &= xsen\varphi + y\cos\varphi \\ y' &= x\cos\varphi - ysen\varphi \end{aligned} \tag{2.5}$$

Estos filtros son aplicados a la imagen de la huella normalizada por convolución, de tal manera que al final de dicha aplicación se obtienen 8 imágenes de la huella correspondientes al resultado de cada filtro aplicado como se muestra en la Figura 2.11.

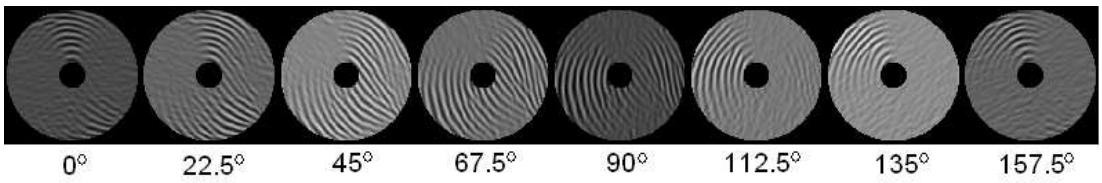

Figura 2.11: Resultado de aplicar los filtros a la huella normalizada.

Finalmente, para calcular el vector de características, dejemos que $F_{i\phi}$ *(x,y)* denote el valor de gris del píxel (x,y) de la imagen filtrada con el filtro en la dirección ϕ para el sector S_i. Así, para $\forall_i$, i $\in$ { 1,2,. . .,n } y toda $\forall\,\phi$, $\phi \in$ { 0° ,22.5° ,45° ,67.5° ,90° ,112.5° ,135° , 157.5° } tenemos que el valor de la componente del vector de característica $V_{i\phi}$ es la desviación absoluta promedio (A.A.D. por sus siglas en ingles) del promedio definida como sigue:

$$V_{i\varphi} = \frac{1}{n_i}\left[\sum_{n_i} |F_{i\varphi}(x,y) - P_{i\varphi}|\right] \tag{2.6}$$

Donde n_i es el número de píxeles en S_i y $P_{i\phi}$ es el promedio de los valores del píxel de $F_{i\phi}$*(x,y)* en el sector S_i. El resultado de esta operación se muestra en la Figura 2.12 donde cada nivel de gris representa el valor de la componente obtenida para cada sector correspondiente.

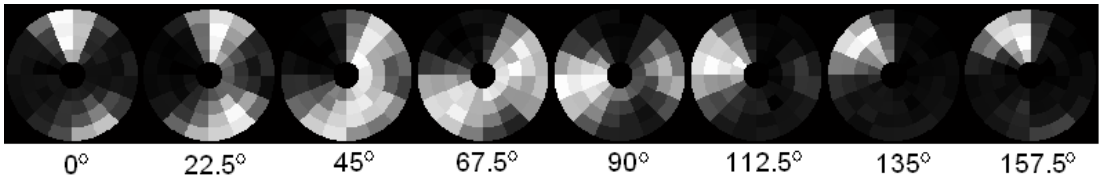

Figura 2.12: Vector de características.

A las 8 secuencias, donde cada secuencia es un círculo con varios sectores, de valores obtenidos se le conoce como el código de la huella (FingerCode) y forma un vector. Este código es luego comparado con una base de códigos de huellas de diferentes individuos para identificar el patrón y por ende al individuo poseedor de dicha huella. Esta comparación se hace a través del cálculo de la distancia euclidiana entre vectores del patrón obtenido con el centroide de los cúmulos cercanos que representan a las personas registradas y así la identificación se logra asociando el nuevo vector con el cúmulo más cercano a él. En la Figura 2.13 se resumen los pasos realizados mostrados aquí para abstraer el vector de características y hacer la identificación correspondiente.

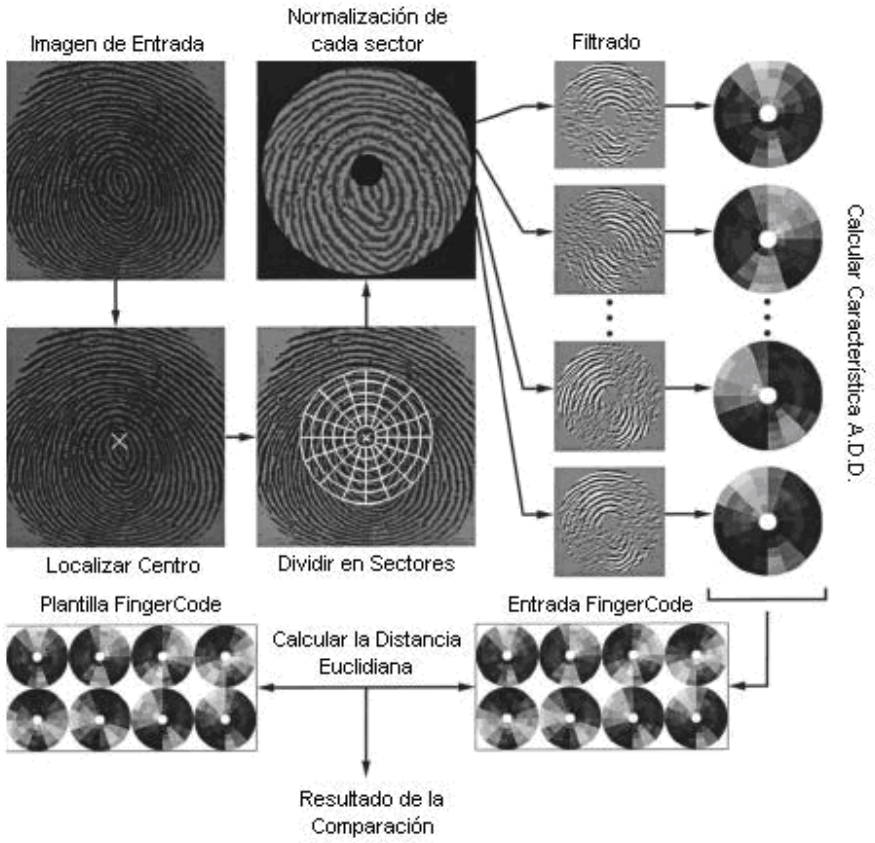

Figura 2.13: Procedimiento para abstraer características y comparar una huella.

En el trabajo realizado no se requiere hacer la comparación para hacer identificación de la persona, puesto que esta ya esta implícita, sino que por el contrario se necesita sólo el vector obtenido para a partir de ahí obtener una clave criptográfica.

2.4 Identificación a través del FingerCode

Una implementación del algoritmo para extraer el FingerCode debe de responder con las mismas características que fueron reportadas en [6] donde dicho algoritmo fue

propuesto y utilizado para identificación. En el se describe que si se mide la distancia euclideana como variable aleatoria entre pares de huellas y se construyen dos histogramas uno para cada par de huellas que pertenecen al mismo dedo y el otro para cada par que no, entonces se deberían de obtener los histogramas mostrados en la parte izquierda de la Figura 2.14.

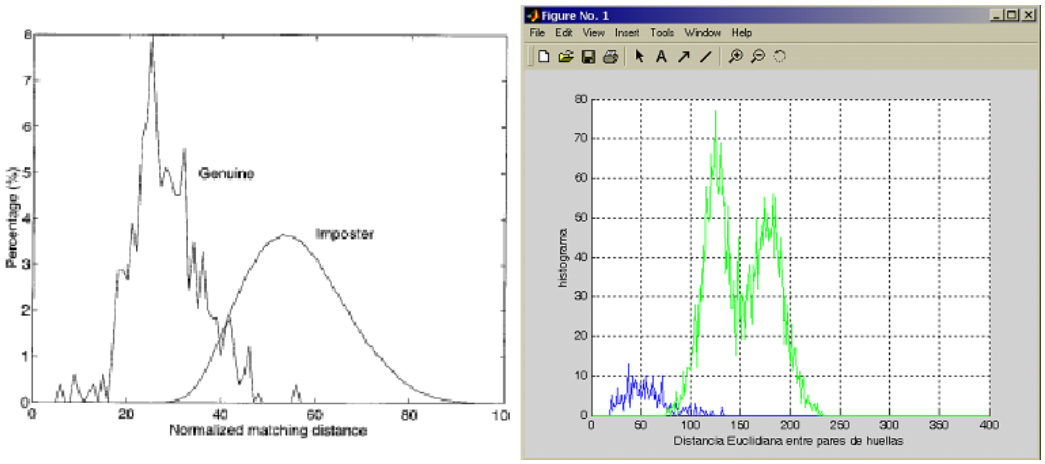

Figura 2.14: Histogramas de la distancia euclidiana entre pares de huellas. Izquierda: Histogramas reportados en [14]. Derecha: Histogramas obtenidos. Azul: huellas del mismo usuario Verde: huellas de diferentes usuarios.

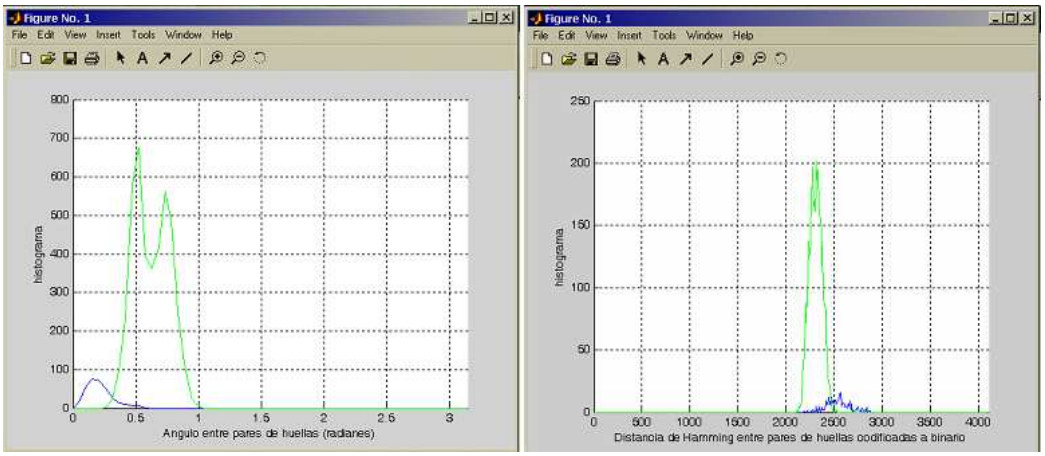

Figura 2.15: Izquierda: Histogramas del ángulo entre pares de huellas. Derecha: Histogramas de la distancia de Hamming entre pares de huellas. Azul: huellas del mismo usuario Verde: huellas de diferentes usuarios.

A la derecha de la misma figura se muestran los histogramas obtenidos por una implementación de este. Como se puede ver, existe una distancia más grande entre pares de huellas de diferentes dedos que entre pares del mismo dedo. Aparte, es notorio observar que existe una región de traslape entre los histogramas, misma en

la que se confunden las huellas de diferentes dedos con las de mismos dedos. Por lo tanto, para construir un sistema de verificación con umbral bajo estas ideas se debe de tener un compromiso entre la probabilidad de aceptar a personas falsas (APF) y la probabilidad de rechazar a personas autenticas (RPA). En la Figura 2.15 se muestran otras dos métricas con las que se probó también y que en esencia refuerzan los mismos resultados obtenidos para la distancia euclidiana, el traslape sigue existiendo.

La conversión a binario para poder medir las distancias de hamming es hecha pasando directamente a binario el valor de cada componente del FingerCode y el ángulo es el mismo que el que existe entre pares vectores del FingerCode.

Las graficas anteriores no reflejan ni dicen nada acerca de la cantidad de variación existente entre los patrones del FingerCode entre pares de huellas del mismo dedo y mucho menos de si son o no variables sobre dichos patrones. Ya que de existir zonas que tuvieran la menor variabilidad, estas serían preferidas. Así, al medir la cantidad de variación que existe entre componentes de FingerCode de huellas del mismo dedo con la siguiente ecuación:

$$\sigma_{relativa} = \sqrt{\sum_i \sum_j (\mathbf{x}_{ij} - \overline{\mathbf{x}}_i)^2}$$

donde x_i el promedio de las componentes de todos los vectores del mismo usuario i y x_{ij} es el vector j-ésimo del usuario i, se obtiene la existencia de uniformidad de dicha variación sobre todos los componentes del FingerCode y no nada más de eso, sino que también se puede obtener un valor determinado que la caracteriza.

Los resultados de dicha medición se muestran en la Figura 2.16 donde se grafica la dispersión relativa contra las posiciones del FingerCode. Nótese que se puede apreciar como la dispersión se mantiene casi prácticamente constante alrededor de 25. Esto quiere decir que existe una variación de las componentes de un vector de FingerCode de aproximadamente 50 unidades de ancho.

Así, puesto que cada componente corresponde a un pedacito de área sobre la huella digital, concluimos que no existen áreas que presenten mayor o menor variabilidad que otras. De tal manera, que no hay forma de explotar ciertos segmentos que presenten menor variabilidad que otros.

Por lo tanto, como se puede observar a partir de los resultados presentados aquí, para construir un sistema que asigne claves criptográficas al FingerCode se requiere que este sea capaz de hacer una especie de verificación implícita. Y esta es de antemano difícil debido a dos cosas: la primera, el gran valor de variabilidad de las componentes del vector de características y la segunda debido al traslape existente entre vectores de diferentes huellas.

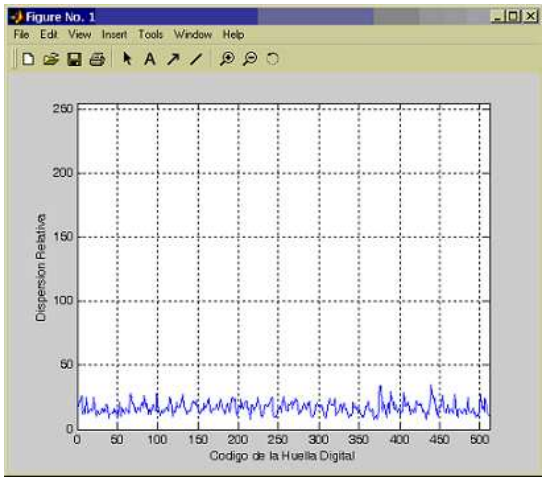

Figura 2.16: Dispersión relativa de las componentes de los FingerCodes pertenecientes a pares de huellas del mismo dedo

2.5 Criptografía

La criptografía se ocupa del diseño de procedimientos para cifrar, es decir, para enmascarar una determinada información de carácter confidencial. El esquema fundamental de un proceso criptográfico (cifrado/descifrado) puede resumirse del modo en que se muestra en la Figura 2.17.

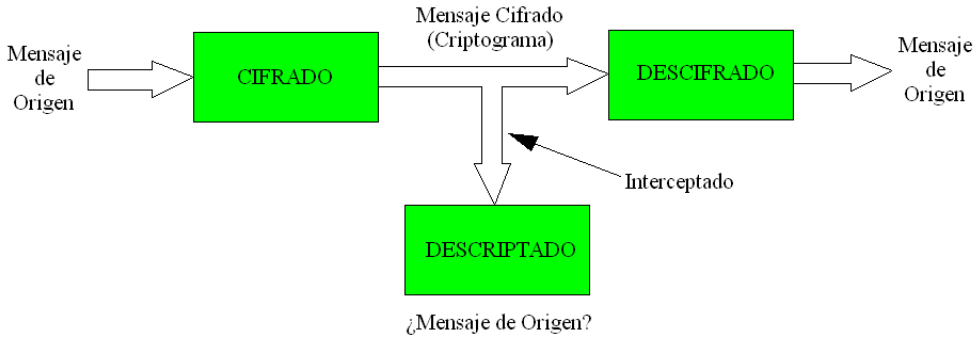

Figura 2.17: Proceso general de Cifrado/Descifrado.

A y B son respectivamente, el usuario y el receptor de un determinado mensaje. A transforma el mensaje original mediante un determinado procedimiento de cifrado controlado por una clave, en un mensaje cifrado (criptograma) que se envía por un canal público. En recepción, B con conocimiento de la clave transforma ese criptograma en el texto fuente, recuperando así la información original.

En el proceso de transmisión, el criptograma puede ser interceptado por un enemigo que lleva a cabo una labor de descriptado; es decir, intenta, a partir del criptograma y sin conocimiento de la clave, recuperar el mensaje original. Un buen sistema criptográfico será, por lo tanto, aquel que ofrezca un descifrado sencillo pero un descriptado imposible o, en su defecto, muy difícil.

El tipo particular de transformación aplicada al texto claro o las características de las claves utilizadas marcan la diferencia entre los diversos métodos criptográficos. Una primera clasificación en base a las claves utilizadas puede desglosarse como sigue:

a. MÉTODOS SIMÉTRICOS. Son aquellos en los que la clave de cifrado coincide con la de descifrado. Lógicamente, dicha clave tiene que permanecer secreta, lo que presupuse que emisor y receptor se han puesto de acuerdo previamente en la determinación de la misma, o bien que existe un centro de distribución de claves que se la ha hecho llegar a ambos por un canal seguro.

b. MÉTODOS ASIMÉTRICOS. Son aquellos en los que la clave de cifrado es diferente a la de descifrado. En general, la clave de cifrado es conocida libremente por el público, mientras que la de descifrado es conocida únicamente por el usuario.

Los métodos simétricos son propios de la Criptografía clásica o Criptografía de clave secreta, mientras que los métodos asimétricos corresponden a la Criptografía de clave pública.

Una de las diferencias fundamentales entre Criptografía clásica y Criptografía moderna radica en el concepto de seguridad. Antes, los procedimientos de cifrado tenían una seguridad probable; hoy, los procedimientos de cifrado han de tener una seguridad demostrable. Esto lleva a una primera clasificación de seguridad criptográfica:

a. SEGURIDAD INCONDICIONAL (Teórica). El sistema es seguro frente a un atacante con tiempo y recursos computacionales ilimitados.

b. SEGURIDAD COMPUTACIONAL (Práctica). El sistema es seguro frente a un atacante con tiempo y recursos computacionales limitados.

c. SEGURIDAD PROBABLE. No se puede demostrar su integridad pero el sistema no ha sido violado.

d. SEGURIDAD CONDICIONAL. Todos los sistemas son seguros en tanto el enemigo carece de medios para atacarlos.

2.5.1 Condiciones de Secreto Perfecto

Shannon definió sus condiciones de secreto perfecto partiendo de dos hipótesis básicas:

1. La clave secreta se utilizará una sola vez.

2. El enemigo tiene acceso solo al criptograma.

Basadas en estas dos hipótesis, Shannon enunció sus condiciones de secreto perfecto, que pueden sintetizarse tal y como sigue.

Un sistema criptográfico verifica las condiciones de secreto perfecto si el texto claro X es estadísticamente independiente del criptograma Y, lo que en lenguaje probabilístico puede expresarse como:

$$P(X = x|Y = y) = P(X = x) \tag{2.7}$$

Para todos los posibles textos fuente x = { x_1, x_2, x_3, ..., x_M } y todos los posibles criptogramas y = { y_1, y_2, y_3, ..., y_N }; es decir, la probabilidad de que la variable aleatoria X tome el valor x es la misma con o sin el conocimiento del valor tomado por la variable aleatoria Y. En términos más sencillos, esto equivale a decir que la información sobre el texto claro aportada por el criptograma es nula. Por lo tanto, el enemigo no puede hacer una mejor estimación de X con conocimiento de Y que la que haría sin su conocimiento, independientemente del tiempo y recursos computacionales de los que disponga para el procesado del criptograma.

Asimismo, y basado en el concepto de entropía, Shannon [18] determinó la menor cantidad de clave necesaria para que pudieran verificarse las condiciones de secreto perfecto. En efecto, la longitud de la clave k tiene que ser, tan larga como la longitud del texto claro M, es decir, k≥M.

2.6 Criptosistemas Biométricos

Como se mencionó antes, para que una clave sea segura se requiere que por lo menos sea tan larga como los datos a encriptar y debe ser mantenida en secreto. Esta clave tiene que ser memorizada, susceptible a ser olvidada, o guardada de alguna manera, susceptible a ser robada o perdida. Así que, seria deseable mejor asociar dichas claves a biométricas de los usuarios como se muestra en la Figura 2.18.

En una primera aproximación, la clave criptográfica y la biométrica pertenecientes a un mismo usuario se encontrarían almacenadas en una base de datos y para liberar dicha clave se necesitaría hacer primero un paso de identificación biométrica que significaría comparar la biométrica del usuario con la que se encuentra almacenada.

Hacerlo así no es muy bueno porque el sistema de autentificación y el de liberación de clave criptográfica están desacoplados, lo que quiere decir que el sistema es susceptible al ataque del Caballo de Troya (El ataque del Caballo de Troya puede remplazar al subsistema de identificación biométrica y simplemente inyectar un bit de información para hacer aceptación o rechazo al subsistema de liberación de claves). Por otro lado,

si se tiene acceso a la base de datos, entonces la biométrica se puede comprometer y aun peor, esta podría ser utilizada en otros sistemas biométricos.

Figura 2.18: a. Sistema criptográfico y b. Sistema criptográfico usando biométricas.

Por lo tanto, para no comprometer la señal biométrica, en lugar de guardar en la base de datos la plantilla biométrica x se almacena una versión transformada de ella a través de una función no invertible H(x), donde la nueva representación puede contener exactamente la misma, menos o más información que la representación original. Esta función H(x) es conocida como función Hash. Esta idea se muestra en la Figura 2.19.

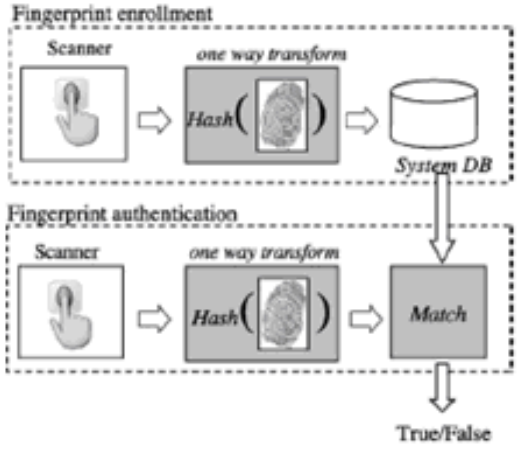

Figura 2.19: Utilización de una función Hash.

De esta manera, un usuario podría usar diferentes funciones Hash en diferentes sistemas, de tal manera que si una biométrica hasheada es comprometida, la biométrica original no lo estará y por consecuente los demás sistemas que tampoco lo estarán.

Pero aún siguen estando desacoplados los sistemas de identificación y liberación de clave criptográfica. Existen trabajos que utilizan funciones Hash, por ejemplo: Davida et al. [12], [11] propuso un algoritmo basado en la biométrica del Iris. Ellos consideran una representación binaria de la textura del Iris, llamada IrisCode [7], que es un vector de 1048 bits de longitud. El comparador biométrico calcula la distancia de Hamming entre la entrada y la plantilla en la base de datos y la compara contra un umbral para determinar si las dos muestras biométricas son de la misma persona o no. Durante el proceso de identificación ellos utilizan un código de corrección de errores para lidiar con las variaciones de la biométrica. También Soltar et al [4], [17] propuso un algoritmo para enlazar claves en un sistema de comparación de huellas basadas en correlación óptica. Este algoritmo liga una clave con la imagen de la huella del usuario en el momento del enrolamiento utilizando una función filtro de correlación. La clave es entonces recuperada solamente sobre una autentificación exitosa utilizando códigos de corrección de error para lidiar con la variabilidad. Esta función filtro es en si una función Hash.

El problema del desacoplamiento de los cirptosistemas biométricos sería resulto si la clave estuviera intrínsecamente ligada con la plantilla biométrica en forma tal que la clave no sería liberada sin una autentificación exitosa como se muestra en la Figura 2.20.

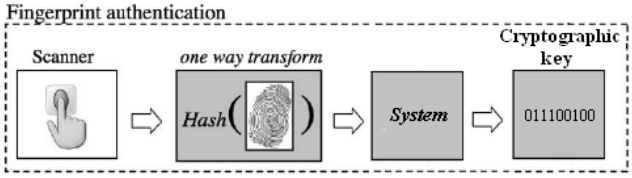

Figura 2.20: Criptosistema Biométrico.

En esta dirección Paola et. al. [13] propone la utilización de clasificadores de patrones SVM's (Suport Vector Machine) no lineales para la asignación de claves a patrones de voz. En sus experimentos describen eficiencias por encima del 90 % para 139 usuarios con claves binarias de longitud 30 bits.

La generación de claves criptográficas a través de biométricas aún tiene varios problemas. A diferencia de una clave criptográfica, representaciones biométricas específicas de una misma persona varían dramáticamente. Consecuentemente, no es obvio como señales biométricas con variaciones inherentes pueden ser usadas para la generación de claves criptográficas. Además, los sistemas de autentificación actuales no son perfectos. Y es deseable que en esquemas de generación de claves criptográficas, no deba de haber una correlación entre la identidad y la clave criptográfica que pudiera ser explotada por el atacante (Condición de secreto perfecto).

2.7 Máquina de Vectores de Soporte (SVM)

Supongamos que tenemos un conjunto de vectores x_i en R^n con i=1,2,3,...,N que queremos clasificar dentro de dos clases de tal manera que m de los N vectores pertenecen a una clase que denominaremos sin perdida de generalidad A con valor 1 y el resto de los vectores (N-m) pertenecen a la otra clase denominada B con valor -1. Así, tenemos entonces que, se quiere encontrar una función *f(x)* que satisfaga que:

$$f(\mathbf{x}_i) = \begin{cases} 1 & \text{si } \mathbf{x}_\text{i} \in \text{A} \\ -1 & \text{si } \mathbf{x}_\text{i} \in \text{B} \end{cases} \tag{2.8}$$

para todos los posibles vectores x en R^n. Esto quiere decir que la función *f(x)* es un clasificador robusto, dado que, se espera que siempre responda correctamente a valores de x que no se especificaron en la condición definida por la ecuación 2.8.

Si *f(x)*= θ (w·x+b) donde:

$$\theta(x) = \begin{cases} 1 & \text{si x} > 0 \\ -1 & \text{si x} \leq 0 \end{cases} \tag{2.9}$$

entonces se dice que los vectores x_i en R^n son linealmente separables. Esto quiere decir que *f(x)* es un clasificador lineal si existe un hiperplano en R^n que separe a las dos clases por completo como se muestra en la Figura 2.21.

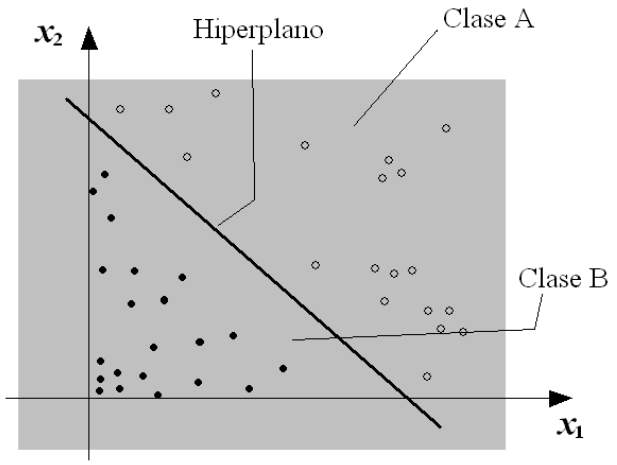

Figura 2.21: Vectores linealmente separables.en R^2.

2.7.1 SVM lineal

Un SVM lineal es un clasificador lineal con margen máximo. Esto significa que un SVM asigna un hiperplano a la función *f(x)* tal que exista la mayor distancia posible entre él y los vectores xi pertenecientes a la clase A en un lado y los vectores x_i

pertenecientes a la clase B del otro como se muestra en la Figura 2.22 donde a los vectores xi que están marcados con círculos y que definen este margen se les conoce como vectores de soporte. Esta forma de obtener el hiperplano garantiza que este separador tenga propiedades deseables en términos de generalización robusta para nuevos ejemplos.

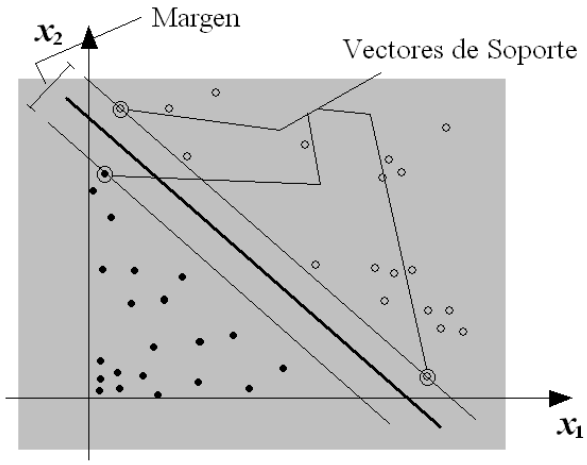

Figura 2.22: Margen y Vectores de Soporte.

Encontrar dicho hiperplano es un problema de optimización de programación cuadrática (PC). Así, el problema consiste en encontrar los valores α_i que maximizan la siguiente expresión (Mayores detalles consultar el apéndice A):

$$\max_{\alpha} J_D(\alpha) = \max_{\alpha} \left[\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \bullet \mathbf{x}_j) \right] \tag{2.10}$$

$$\text{con restricciones } \alpha \cdot \text{y} = 0 \text{ y } 0 \leq \alpha_i \leq \text{c}.$$

con la siguiente forma funcional del clasificador:

$$f(\mathbf{x}) = \theta \left(\sum_{i=1}^{N} \alpha_i y_i (\mathbf{x}_i \bullet \mathbf{x}) + b \right) \tag{2.11}$$

Este problema tiene tres propiedades importantes. Primero, la expresión tiene un único máximo global que puede ser encontrado de manera eficiente. Segundo, los datos entran en la expresión sólo en forma de pares de productos punto. Esta segunda propiedad es también verdadera para la ecuación del propio separador (2.11) una vez que ya han sido calculados los α_i. Y tercera, los pesos α_i asociados con cada punto de los datos son 0 excepto para los vectores de soporte. Por esta razón, debido a que normalmente existen menos vectores de soporte que de datos el número de parámetros que define al hiperplano es normalmente más pequeño que N evitando el sobreajuste.

2.7.2 SVM no lineal

El SVM no lineal es precisamente un SVM lineal que funciona sobre los datos ϕ(x) donde x son los vectores de entrada. Esta función ϕ(x) es una función vectorial que mapea a los vectores x a un espacio vectorial de dimensión mayor o infinita. Entonces el producto punto estará dado por K(x,y)=ϕ(x)·ϕ(y) donde K es conocida como Kernel. Así puesto que el problema de PC (9) del SVM lineal requiere ϕ(x)·ϕ(y) no es necesario conocer la forma exacta de ϕ(x), basta con introducir el Kernel. Esto es posible gracias al siguiente resultado:

> Para cualquier función continua y simétrica K(x,y) que satisfaga las condiciones de Mercer, ahí existe un espacio de Hilbert H, un mapeo ϕ(x): R^n →H y número positivo λ_i tal que se puede escribir:
>
> $$\mathbf{K}(\mathbf{x},\mathbf{y}) = \sum_{i=0}^{n_H} \lambda_i \phi_i(\mathbf{x})\,\phi_i(\mathbf{y}) \tag{2.12}$$
>
> donde x, y $\in R^n$ y n_H es la dimensión de H (que puede ser infinitamente dimensional). La condición de Mercer requiere que:
>
> $$\int \mathbf{K}(\mathbf{x},\mathbf{y})\, g(\mathbf{x})\, g(\mathbf{y})\, d\mathbf{x} d\mathbf{y} \geq 0 \tag{2.13}$$
>
> para cualquier función g(x) de cuadrado integrable. La integral es tomada sobre el subconjunto compacto de R^n.

Así, el hiperplano encontrado en el espacio de ϕ(x) puede corresponder en el espacio original x, a hipercurvas como se muestra en la Figura 2.23.

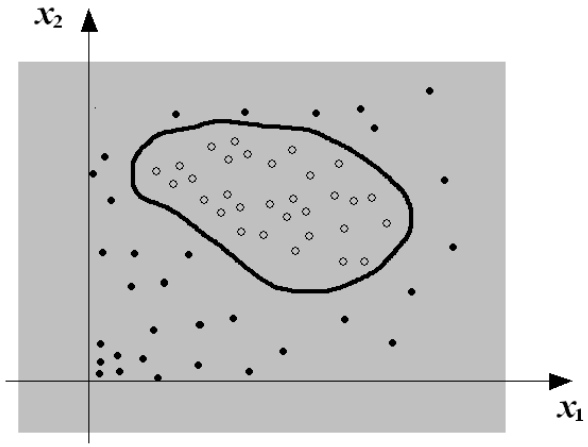

Figura 2.23: Hipercurvas de separación generadas por el SVM no lineal.

Por lo tanto el problema de PC estaría dado de la siguiente forma:

$$\max_{\alpha} J_D(\alpha) = \max_{\alpha} \left[\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) \right] \tag{2.14}$$

$$\text{con restricciones } \alpha \cdot \text{y} = 0 \text{ y } 0 \leq \alpha_i \leq \text{c}.$$

con función de clasificador:

$$f(\mathbf{x}) = \theta \left(\sum_{i=1}^{N} \alpha_i y_i \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) + b \right) \tag{2.15}$$

A continuación se muestran algunas funciones Kernel conocidas:

NOMBRE	KERNEL
Lineal	K(x,y) = (x • y)
Polinomial de grado d	K(x,y) = $(\gamma + (x \bullet y))^d$
Funciones Base Radiales (RBF)	K(x,y) = $e^{-\left(\frac{\|x-y\|^2}{\sigma^2}\right)}$
Sigmoide (MLP)	K(x,y) = tanh (k_1 (x • y) + k_2)

Tabla 2.1: Momentos centrales y ángulos de orientación de c/segmento.

2.8 Resumen

En este capitulo se trataron los aspectos teóricos necesarios para el desarrollo de este trabajo. Se describió la estructura de la huella digital y sus características, así como los elementos principales de ésta que son aprovechados para identificación y asignación de claves criptográficas. Se explicó que es un filtro de Gabor y la utilización en el procedimiento para obtener el FingerCode. Se describieron las características que conforman a un criptosistema clásico y biométrico. Y finalmente se mostró el funcionamiento de los clasificadores SVM's.

Capítulo 3

Modelos Propuestos

Para poder generar una clave criptográfica binaria a través del código de textura (FingerCode) de la huella digital presentamos 4 modelos diferentes mostrando las ventajas y desventajas de cada uno. Cada modelo en el orden consecutivo presentado aquí ha pretendido ser una mejora de los anteriores. En todos los modelos se utiliza el mismo clasificador de patrones conocido como Support Vector Machine (SVM) descrito anteriormente. Este clasificador ha mostrado funcionar con razones de eficiencia alrededor del 90 % en la generación de claves criptográficas a partir de patrones de voz para 139 usuarios con claves binarias de alrededor de 30 bits.

En si, los modelos no pretenden generar claves criptográficas, sino que se supone de antemano un generador de claves, el cual es utilizado para generar una clave que se le quiere asignar a un usuario. Obtenido lo anterior, se procede entonces, a entrenar al sistema clasificador para que sea capaz de relacionar la clave asignada con la huella digital que dicho persona autorizada posea.

3.1 Modelo A

En este modelo se supone la existencia de n usuarios, cada uno con un conjunto de m huellas digitales del mismo dedo y una clave criptográfica binaria de k dígitos como se muestra en la Figura 3.1 La clave criptográfica es generada de manera aleatoria con distribución de probabilidad uniforme.

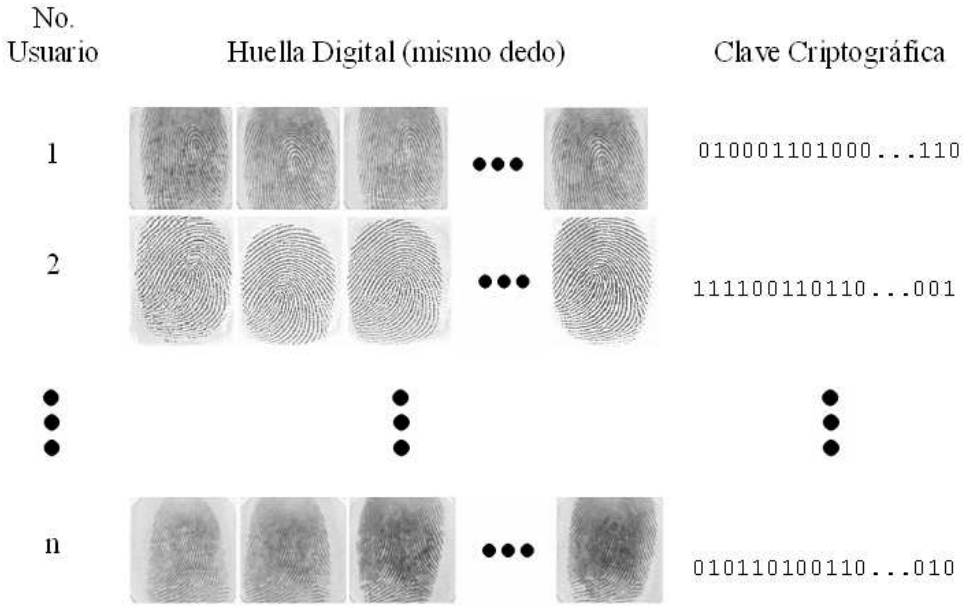

Figura 3.1: m huellas dígitales del mismo dedo y clave criptográfica de k dígitos asignada por usuario.

Así, el primer paso es entonces obtener los códigos de textura de las huellas digitales (FingerCode). De esta manera se tendrán un conjunto de vectores de p componentes cuyos valores estarían dados en el conjunto de los números enteros entre 0 y 255. Lo dicho hasta aquí se representa en la Figura 3.2 donde el elemento a_{ijl} representa a la componente i-ésima del vector de características correspondiente a la huella j-esima del i-ésimo usuario y el elemento c_{iq} representa al q-ésimo bit del i-ésimo usuario.

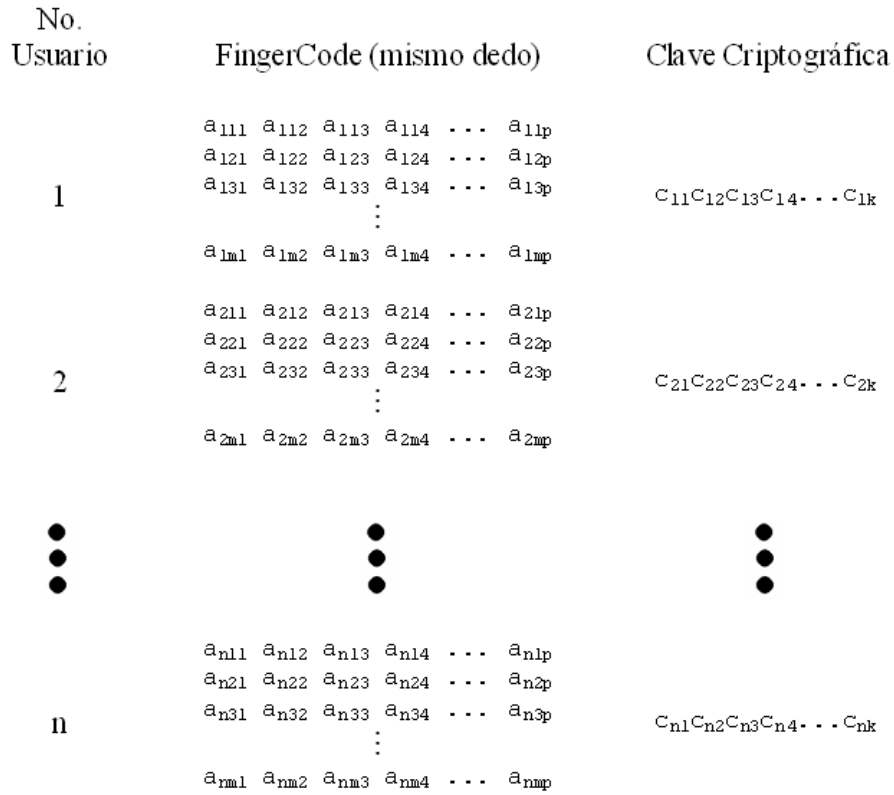

Figura 3.2: m códigos de huellas dígitales del mismo dedo y clave criptográfica de k dígitos asignada por usuario.

El siguiente paso consiste en entrenar al sistema para que sea capaz de asociar, a los vectores de características pertenecientes a una sola persona, su clave criptográfica asignada. Puesto que se tienen m huellas por persona, entonces, en el proceso de entrenamiento, se utilizan m' de ellas y las restantes (1-m') son usadas para verificar el nivel de aprendizaje que se tuvo en el entrenamiento, regularmente estos últimos se conocen como datos de prueba.

El sistema que se utiliza en este modelo tiene la arquitectura mostrada en la Figura 3.3 donde se puede observar que contiene k clasificadores SVM. Para entrenar a este sistema se tiene que entrenar a cada SVM independientemente, lo cual muestra que la arquitectura de este sistema esta constituida por múltiples subsistemas independientes con las mismas características que estan constitudos por un SVM que actua sobre todo los datos de entrada y que genera un posible bit de la clave. Así el sistema completo que contiene múltiples SVM's es equivalente a varios sistemas idénticos con un solo SVM. La forma de entrenar a cada SVM se muestra en la Figura 3.4 donde se puede apreciar la relación que deben de tener los datos de entrada y salida que se encuentran en la Figura 3.2.

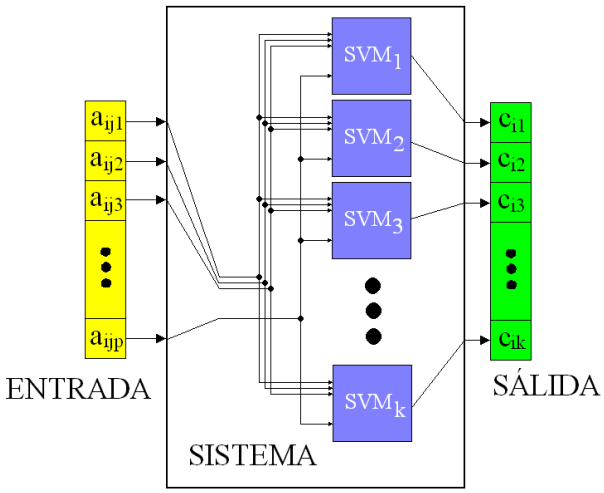

Figura 3.3: Arquitectura del Modelo A.

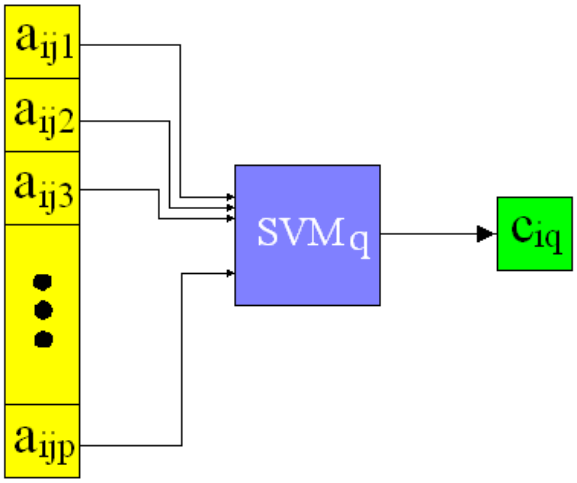

Figura 3.4: Entrenamiento de los SVM's.

No se conoce con exactitud como se agrupan los vectores de FingerCode en el espacio de FingerCodes, sin embargo, es posible describir al menos intuitivamente lo que esta sucediendo con este sistema.

Primero, se sabe que los vectores pertenecientes a un mismo individuo están más cercanos entre ellos que los pertenecientes a individuos diferentes [14]. Por lo tanto, es razonable suponer que los códigos de textura pertenecientes a una persona forman cúmulos de puntos, de tal manera, que cada persona tendría su propio pedazo de espacio asignado en el espacio de FingerCodes como se muestra en la Figura 3.5.

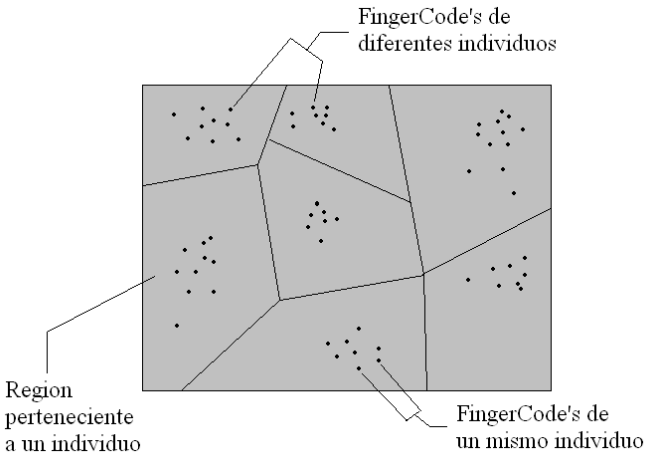

Figura 3.5: Representación abstracta del espacio de FingerCode's.

Entonces, debido a la forma de entrenamiento, cada clasificador tendría que asignar una función binaria discretizada sobre regiones del espacio de FingerCode's pertenecientes a sus respetivos usuarios, como se puede observar en la Figura 3.6 y por lo tanto el espacio se vería dividido en regiones, que más o menos tendrían las formas complementarias a la distribución de los datos de los usuarios en el sistema para cada clasificador. Ahí se puede ver, que la línea continua gruesa determina la hipersuperficie de separación de regiones a las cuales cada clasificador le asigna 0 ó 1. También se observan los cortes dibujados con líneas punteadas que junto con las líneas continuas gruesas dividen los espacios pertenecientes a cada individuo. Las líneas punteadas no las genera el clasificador y fueron puestas solo para representar las regiones de los usuarios del sistema. Como se puede observar entonces, las regiones de los usuarios se encuentran dividas aproximadamente igual por todos los clasificadores. Así, cuando superponemos los clasificadores el resultado es una función como la que se muestra en la Figura 3.7 donde existen entonces regiones del espacio en las cuales la respuesta del sistema no es la adecuada, es decir existe el traslape, por lo tanto, cuando se introduzca a un usuario que el sistema nunca ha visto antes, si este individuo se encuentra dentro de la región de traslape el sistema no responderá con alguna clave correcta, mientras que por el contrario lo hará. En los experimentos se mostrara que este sistema es muy tolerante a intrusos, es decir, que existe una alta probabilidad que agentes extraños caigan en la región de traslape. Las claves generadas en la región de traslape no corresponden a ninguna clave de los usuarios del sistema.

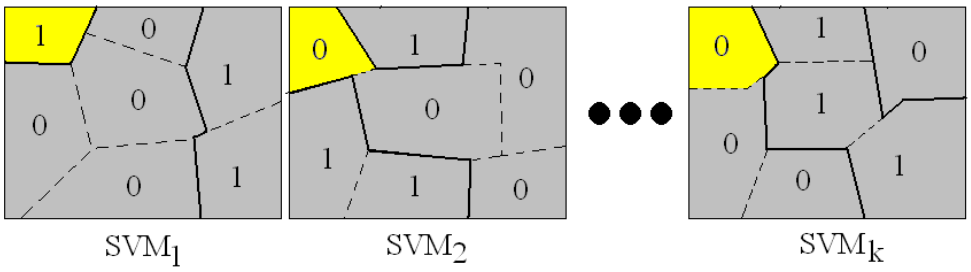

Figura 3.6: La región en color amarillo pertenece al mismo usuario pero cada clasificador le asigna un valor diferente.

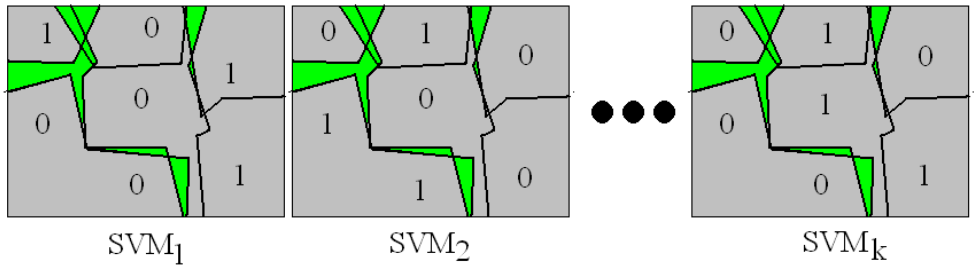

Figura 3.7: La región en color verde pertenece al error generado por el sistema.

Ataques a este sistema serían:

a. Hacer una búsqueda en el espacio de FingerCode's, lo cual sería difícil porque se requiere una búsqueda en 255^p valores y para FingerCode p estaría del orden de 500.

b. Hacer una búsqueda en la clave, lo cual sería difícil porque las claves son por lo regular muy grandes lo cual involucraría una búsqueda de un valor en 2^b valores, donde b es del tamaño de los datos a ocultar. Para un documento b sería muy grande, alrededor de 8d donde d = (numero de caracteres).

c. A través de los vectores de soporte. Si los vectores de soporte son conocidos, y la función kernel es invertible entonces son conocidas las hipersuperficies de separación de cada clasificador, de tal manera que es conocida la respuesta del sistema y por superposición de las hipersuperficies se pueden rastrear las regiones del espacio que generan claves validas, permitiendo así el conocimiento no solo de la clave de un usuario sino también su biométrica.

d. Puesto que la biométrica no es confidencial, entonces se podría conseguir la huella, e introducirla al sistema.

El problema que se describirá con más detalle en el capitulo 5 es que cuanto más grande es el número de usuarios, este sistema se va haciendo cada ves más ineficiente. Por otro lado, el sistema podría generar una clave criptográfica valida para un individuo falso que nunca había visto antes durante su entrenamiento.

3.2 Modelo B

En este modelo se supone la existencia de n individuos dentro de los cuales (n-1) funcionan como agentes de confusión y uno es el usuario del sistema. Cada individuo tiene un conjunto de m huellas digitales del mismo dedo, sin embargo, solo el usuario posee una clave criptográfica binaria de k dígitos mientras que a los demás se les asigna una cadena de k 0's como se muestra en la Figura 3.8 La clave criptográfica es generada de manera aleatoria con distribución de probabilidad uniforme.

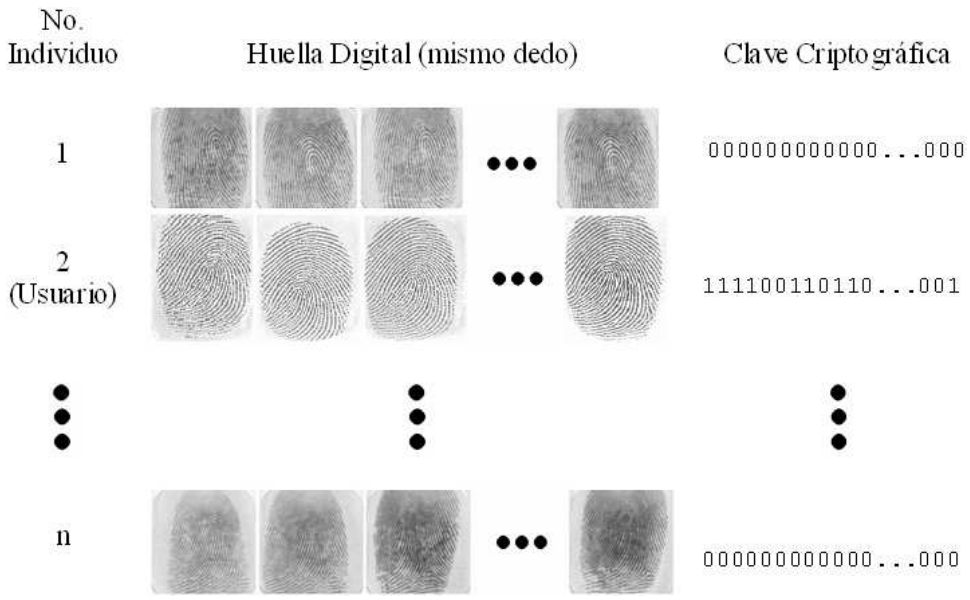

Figura 3.8: m huellas digitales del mismo dedo dentro de las cuales existe solo una persona que posee una clave criptográfica, el resto poseen 0's.

El proceso de obtener los FingerCode's a partir de las huellas de la Figura 3.8, las definiciones empleadas en la Figura 3.2, la arquitectura y el proceso de entrenamiento son el mismo que en el Modelo A.

Este sistema funciona aislando al individuo de interés del resto por lo cual el espacio de FingerCode's se vería dividido como se muestra en la Figura 3.9.

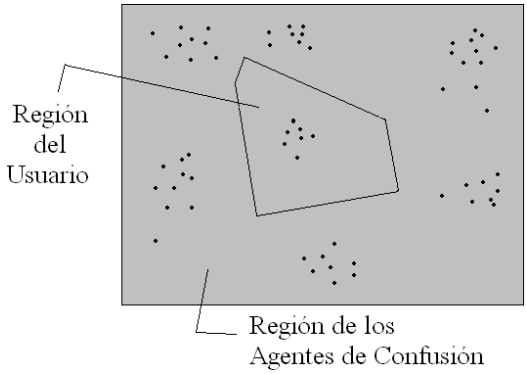

Figura 3.9: Espacio de FingerCode's.

Entonces los clasificadores del sistema tendrían la tarea de asignar una función binaria a las dos únicas regiones del espacio, ya que el espacio estaría dividido exactamente por la misma frontera para todos los SVM's. No es difícil ver esto, basta con advertir que todos los clasificadores SVM_q que son entrenados bajo la condición $c_{iq} = 0$ con i=usuario, son equivalentes y todos los SVM_q que son entrenados bajo la condición $c_{iq} = 1$ con i=usuario también son equivalentes, por lo tanto, solo se tienen dos clasificadores diferentes arreglados según la clave criptográfica.

Ataques a este sistema serían:

a. Hacer una búsqueda en el espacio de FingerCode's lo cual sería difícil porque se requiere una búsqueda en 255^p valores y para FingerCode p estaría del orden de 500.

b. Hacer una búsqueda en la clave, lo cual sería difícil porque las claves son por lo regular muy grandes lo cual involucraría una búsqueda de un valor en 2^b valores, donde b es del tamaño de los datos a ocultar. Para un documento b sería muy grande, alrededor de 8d donde d = (numero de caracteres).

c. Descifrar el significado de los vectores de soporte, que en este caso, el sistema, sería demasiado vulnerable ya que como solo se tienen dos clasificadores diferentes, los clasificadores del mismo tipo tendrán los mismo vectores de soporte, por lo tanto, solo vasta con asignar a ambos una combinación complementaria de 0's y 1's y si no funciona probar con el complemento de dicha secuencia de 0's y 1's asignada.

d. Puesto que la biométrica no es confidencial, entonces se podría conseguir la huella, e introducirla al sistema.

Este sistema funciona bastante bien, pero tiene la vulnerabilidad de solo estar formado por dos clasificadores distintos, por lo tanto, este modelo solo puede ser considerado como un modelo de prueba o modelo extremo.

3.3 Modelo C

Este modelo solventa la dificultad presentada por el Modelo B, que, después del entrenamiento, contenía solo dos clasificadores diferentes. La idea aquí, es introducir a las claves del Modelo B un pequeño ruido aleatorio. Cuando este ruido aleatorio se va incrementando, en el límite como caso extremo, cuando el número de bits con valor 1 de ruido forman en promedio la mitad de los bits existentes, este y el Modelo A, son equivalentes y por el otro extremo es equivalente con el Modelo B, es decir, cuando no hay existencia de ruido.

Para describir la idea del ruido aleatorio consideremos que tenemos los mismos datos que en la Tabla 3.1 y entonces prestémosle atención solo a la primera y tercera columna que contienen al No. de individuo y su asignación de clave criptográfica donde se ha definido un ejemplo especifico para evitar confusión debido a la generalización matemática.

No. Individuo	Clave Criptográfica
1	0000000000000000
2	0000000000000000
3	0000000000000000
Usuario = 4	1111001101100010 01
5	0000000000000000
6	0000000000000000
7	0000000000000000
8	0000000000000000
9	0000000000000000

Tabla 3.1: Claves criptográficas de 9 individuos.

No. Individuo	Clave Criptográfica
1	0000000001010000 00
2	1000000010000000 10
3	0000000000000001 00
Usuario = 4	1111001101100010 01
5	0010010000000000
6	0000001000100100 01
7	0100100000001000 00
8	0000000100000010 00
9	0001000000000000

Tabla 3.2: Claves cript. con pequeño ruido aleatorio en 9 individuos.

Así pues, el ruido aleatorio consiste entonces en cambiar en cada columna de las claves criptográficas un 0 por un 1, evitando los bits de la clave del usuario. La selección del bit a cambiar en cada columna es seleccionado al azar con una distribución de probabilidad uniforme. De esta manera, aplicando el ruido aleatorio a las claves de la Tabla 3.1 nos queda lo que se muestra en la Tabla 3.1.

De esta manera, el sistema funciona muy bien y se ha eliminado la existencia de solo dos tipos de clasificadores diferentes. Como se puede observar, en este modelo, los clasificadores aíslan a dos personas del resto, dentro de las cuales una es el usuario. Esto se puede ver en la Figura 3.10 También se puede observar que la función binaria asignada por los clasificadores SVM solo es aleatoria dentro del espacio de estas dos personas, mientras que en el resto se vuelven 0.

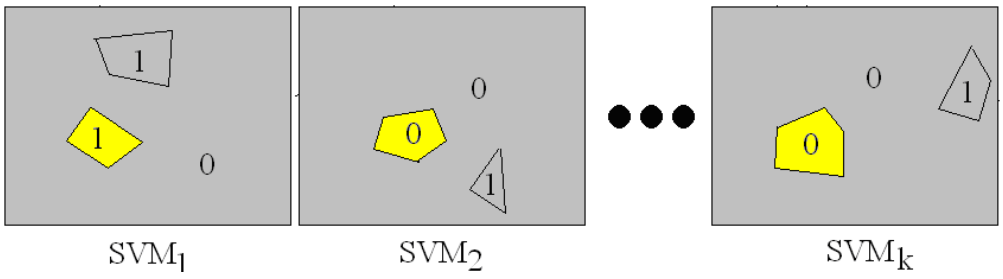

Figura 3.10: El polígono de color amarillo representa al usuario y el color gris al usuario que le correspondió el ruido aleatorio.

Ataques a este sistema serían:

a. Hacer una búsqueda en el espacio de FingerCode's lo cual sería difícil porque se requiere una búsqueda en 255^p valores y para FingerCode p estaría del orden de 500.

b. Hacer una búsqueda en la clave, lo cual sería difícil porque las claves son por lo regular muy grandes lo cual involucraría una búsqueda de un valor en 2^b valores, donde b es del tamaño de los datos a ocultar. Para un documento b sería muy grande, alrededor de 8d donde d = (número de caracteres).

c. A través de los vectores de soporte. Si los vectores de soporte son conocidos, y la función kernel es invertible entonces son conocidas las hipersuperficies de separación de cada clasificador, de tal manera que es conocida la respuesta del sistema y por superposición de las hipersuperficies se puede rastrear la región del espacio a la que el sistema responde con 1. Después, no se sabe que subregión de la región de la respuesta del sistema igual a 1 es la correspondiente al usuario, de tal manera que, entonces, se seleccionan una secuencia de puntos uniformemente distribuidos en el espacio $\phi(x)$ con clasificación 1 y se retornan al espacio original, así puesto que, en la región del usuario este responde más veces con 1 que el resto de las regiones se tendrá una densidad de puntos mayor sobre dicha

región permitiendo el conocimiento de su clave y de los patrones de textura de FingerCode de su biométrica. Sin embargo como ya se mencionó anteriormente dicha biométrica no es confidencial por lo tanto no es un factor a atacar.

d. Puesto que la biométrica no es confidencial, entonces se podría conseguir la huella, e introducirla al sistema.

Este sistema funciona bastante bien. La cantidad de ruido aleatorio se podría incrementar a 2 bits por columna, pero el riesgo que se corre es que conforme este se incrementa, la eficiencia del sistema disminuye.

3.4 Modelo D

Este modelo es el mismo que el descrito en el Modelo C con la única diferencia que cambia la manera de operar y entrenar el sistema incorporando una permutación secreta. El punto es que los diferentes clasificadores del sistema se entrenen con permutaciones aleatorias diferentes del FingerCode y operen con dichas permutaciones también. Cada una de estas permutaciones sería aplicada a todo el conjunto de entrenamiento por igual para cada clasificador. Tales permutaciones permitirían que el conjunto de FingerCode's del usuario estuvieran en diferentes regiones del espacio de FingerCode's para cada clasificador, como se muestra en la Figura 3.11.

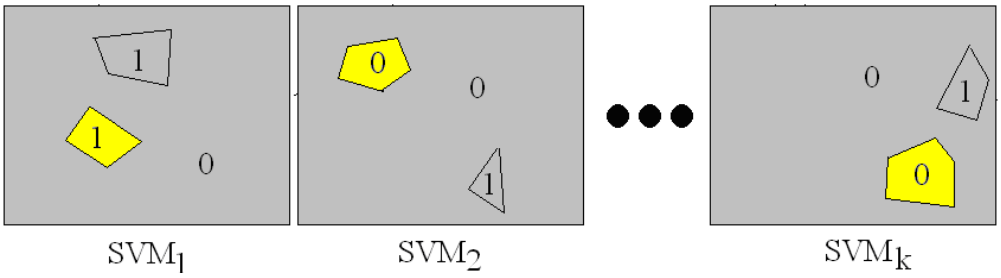

Figura 3.11: El polígono de color amarillo representa al usuario y el color gris al usuario que le correspondió el ruido aleatorio.

Esto difumaría el ataque c del Modelo C descrito anteriormente puesto que la densidad de puntos sobre todos los posibles individuos sería uniforme. Estas regiones diferentes estarían ligadas solo por una permutación, y romper dicha permutación significaría buscar un valor dentro de p donde p esta alrededor de 500. También eliminaría el ataque por obtención no permitida de la biométrica ya que se esta combinando con algo que es confidencial del usuario (La secuencia de permutaciones). El problema es que es necesario almacenar o memorizar tal secuencia de permutaciones.

Ataques a este sistema serían:

e. Hacer una búsqueda en el espacio de FingerCode's lo cual sería difícil porque se requiere una búsqueda en 255^p valores y para FingerCode p estaría del orden de 500.

f. Hacer una búsqueda en la clave, lo cual sería difícil porque las claves son por lo regular muy grandes lo cual involucraría una búsqueda de un valor en 2^b valores, donde b es del tamaño de los datos a ocultar. Para un documento b sería muy grande, alrededor de 8d donde d = (numero de caracteres).

Como se vera en los resultados más adelante, el hecho de usar un conjunto de permutaciones del FingerCode para entrenar al modelo C no cambia el desempeño obtenido por el sistema y así la respuesta de los SVM's ante permutaciones se mantiene invariante.

3.5 Seguridad de los Modelos ante la no confidencialidad de la Biométrica

Como ya se observó, el modelo D proviene de cambiar la distribución de los datos que se utilizan para entrenar al modelo C utilizando para ello permutaciones aleatorias diferentes. Con esto se logra evitar la vulnerabilidad del sistema ante el conocimiento de la biométrica pudiendo combinarla con algo que es secreto del usuario del sistema. De esta manera, el usuario tendría que crear su propia secuencia de permutaciones y posteriormente utilizarla para entrenar al sistema para guardar su clave criptográfica y para poderla recuperar después. Entonces no sería mala idea utilizar estas permutaciones en los modelos A, B y C para dicho fin conservando sus mismas eficiencias ya que los SVM's son invariantes ante tal transformación como se verá en los resultados experimentales.

Sin embargo, el problema, como ya se mencionó antes, es que el usuario tendría que aprenderse las secuencias de permutaciones aplicadas a su sistema particular y dichas permutaciones serían demasiadas. Por lo tanto, en lugar de hacer esto, sería mejor que dichas secuencias estuvieran ocultas de alguna otra manera con alguna clave más corta (NIP) y fueran desencadenadas al momento de que dicho usuario proporcionara tal clave.

Para hacer esto, se propone la utilización de un generador de números aleatorios donde tal NIP se correspondería con la semilla del mismo. Por lo tanto, la secuencia de números aleatorias generada se utilizaría para reacomodar los elementos de FingerCode en su nueva posición en cada permutación como se muestra en la Figura 3.12.

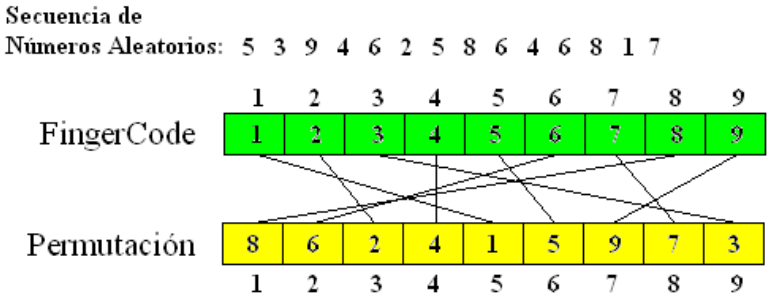

Figura 3.12: Los números dentro de los recuadros representan al número de componente del FingerCode mientras que los que están por encima y por debajo representan a la posición.

En ella se muestra una secuencia de números aleatorios que es utilizada tal como se van generando para indicar la nueva posición de cada componente del FingerCode empezando de izquierda a derecha. Por ejemplo, la primera componente se pone en la posición del primer número aleatorio que es 5, la segunda componente se pone en la posición del segundo número aleatorio que es 3 y así sucesivamente.

Aplicar esta idea a los modelos B, C y D es directo, ya que estos tienen ya la arquitectura necesaria, sin embargo, para el modelo A no es así. La razón es que, generar una permutación diferente en los FingeCodes por cada usuario podría dispersar los patrones tanto que podría ya no ser un problema fácil para los clasificadores y por lo tanto con mal funcionamiento. En lugar de esto, es preferible que existan n sistemas idénticos uno por cada usuario del sistema. Y en cada uno de estos sistemas el resto de los usuarios de los otros sistemas tienen la función de ser agentes distractores como sucede en los modelos B, C y D. De esta manera, cada usuario tendría su propio sistema al cual se aplicaría su propia secuencia de permutaciones.

Cabe hacer notar que en la seguridad impresa en las permutaciones del FingerCode para estos modelos es igual a la seguridad que otorga la semilla y no al número de permutaciones, lo cual significa que si la semilla es pequeña entonces los sistemas serían demasiados vulnerables.

3.6 Resumen

En este capitulo se describieron básicamente los cuatro modelos propuestos que fueron ideados para enfrentar el problema, se mostró su arquitectura, su funcionamiento, la manera de entrenarlos y sus vulnerabilidades, así como la manera de utilizar NIP's para ocultar permutaciones de los FingerCode's que fueron generadas aleatoriamente para lidiar con la confidencialidad de las biométricas.

Capítulo 4

Desarrollo Experimental

Primero se desarrollo el sistema que obtiene el vector de características de una huella digital basado en las especificaciones presentadas en [14] y se monto un experimento pequeño para reproducir dichos resultados. Después, se implementaron los 4 modelos mencionados en el capitulo 3 junto con sus variantes para la seguridad con biométricas y se caracterizó el desempeño de cada uno. A continuación se describen los detalles de diseño y experimentales utilizados.

4.1 Implementación del Sist. para la obtención de características de textura

Se hizo necesario diseñar por completo un programa escrito en Matlab Ver. 6 Release 13 para la extracción de características de la huella digital. Dicha etapa estuvo dividida en dos partes, la del diseño del algoritmo con sus apropiados parámetros y la de la prueba del funcionamiento del mismo que debía de reproducir los resultados publicados en [14].

4.1.1 Parametros de la implementación

Se implementó un sistema para la extracción de características de textura de la huella digital como el descrito en la sección 2.3. Dicho sistema contó con 4 bandas circulares divididas en 16 partes iguales cada una sumando un total de 64 sectores con la distribución mostrada en la Figura 4.1.

Figura 4.1: Area de interés dividida en 64 sectores

Cada banda concéntrica mide 20 píxeles de ancho. El sector central no es considerado como una pequeña región de interés por la curvatura demasiado pronunciada que presenta.

La normalización de la imagen de la huella se llevo a cabo con los siguientes valores para $M_o = 100$ y $V_o = 100$. Los parámetros de los filtros de Gabor se pusieron a los siguiente valores k $= 2\pi/\lambda$, $\lambda = 10$ y $\sigma_x = \sigma_y = 4$, donde λ es la longitud de onda del filtro de Gabor medida en píxeles que fue ajustada de tal manera que aproximadamente diera la longitud transversal de un valle y una cresta, puesto que el ancho de una banda mide 20 píxeles y en el ancho caben aproximadamente dos valles y dos crestas para huellas escaneadas a 500 dpi.

Las mascaras para implementar los filtros de Gabor fueron matrices cuadradas de 33x33 pixeles similares a las mostradas en la Figura 2.10 pero sin el ajuste a la escala de grises que se les aplicó para poder muestrearlas gráficamente.

4.1.2 Prueba de la implementación

Para probar el buen funcionamiento del sistema que extrae las características de textura de la huella digital se diseñó un experimento con 12 usuarios diferentes con 8 huellas digitales del mismo dedo por usuario. Estos 12 usuarios fueron elegidos de una base de datos de huellas digitales, de tal manera que, sus huellas estuvieran lo mejor alineadas verticalmente posible y con los centros de las huellas lo suficientemente centrados en las imágenes como para no truncar las texturas de las áreas de interés. También se cuido que la calidad de la imagen de las huellas fuera lo mejor posible. De esta manera se contó con un experimento con 96 huellas en total. Los centros de las huellas digitales fueron obtenidos manualmente.

Este experimento dio como resultado un conjunto de 96 vectores de características de 512 componentes donde el valor de cada componente es un número entero entre 0

y 255, es decir, tiene el tamaño de un byte.

A los 96 vectores de código resultantes del experimento mencionado se les hicieron mediciones para caracterizar la variabilidad existente en el FingerCode y el desempeño del sistema para identificación. Las mediciones aplicadas fueron las siguientes:

Dispersión Relativa

La primera medición realizada fue la dispersión relativa que se utilizó para poder cuantificar la variabilidad que existe en las componentes de los vectores de FingerCode para diferentes huellas del mismo usuario. Sin embargo, no se disponen de suficientes datos por persona como para hacer una medición de este tipo así que la manera de hacerlo fue calculando la dispersión de todos los usuarios con respecto a sus promedios respectivos, con la siguiente ecuación:

$$\sigma_{relativa} = \sqrt{\sum_i \sum_j \left(\mathbf{x}_{ij} - \overline{\mathbf{x}}_i\right)^2} \tag{4.1}$$

Donde x_i denota el promedio de las componentes de todos los vectores del mismo usuario i y x_{ij} es el vector j-ésimo del usuario i.

Distancia Euclidiana

La distancia euclidiana se utilizó para cuantificar las distancias existentes entre pares de huellas del mismo usuario y mismo dedo contra pares de huellas de diferentes usuarios y diferentes dedos. Por lo tanto, esta se midió de la siguiente manera:

$$d_{ij} = \sqrt{\sum_k (x_{ik} - x_{jk})^2} \tag{4.2}$$

Donde i y j denotan dos vectores diferentes o iguales y k va sobre las componentes de dichos vectores.

Ángulo

El ángulo entre dos vectores esta dado por:

$$\phi = \cos^{-1}\left(\frac{\mathbf{x}_i \circ \mathbf{x}_j}{|\mathbf{x}_i|\,|\mathbf{y}_j|}\right) \tag{4.3}$$

Distancia de Hamming

La distancia de Hamming se define como el número de bit en que son similares dos cadenas de bit.

4.2 Implementación de Criptosistemas Biométricos

Se implementaron las 4 arquitecturas descritas en el capitulo 3 utilizando el clasificador ya existente conocido como SVM Light Ver 6.01. En todos los experimentos realizados aquí, este clasificador conservó los parámetros por default excepto los siguientes:

Learning options:	
-c float	- C: trade-off between training error
Kernel options:	- type of kernel function:
-t int	0: linear (default)
	1: polynomial $(sa * b + c)^d$
	2: radial basis function exp(-gamma $\|$a-b$\|^2$
	3: sigmoid tanh(s a*b + c)
	4: user defined kernel from kernel.h

que fueron puestos a los siguientes valores: c=9 y t=0, así, todos nuestros experimentos fueron probados con SVM Lineales. Como generador de claves criptográficas y generador de permutaciones se utilizó el generador de números aleatorios de Matlab Ver 6 Release 13. Se utilizo también un conjunto de datos pertenecientes a 32 individuos distintos (Agentes Extraños) a los usados en los experimentos descritos en los siguientes apartados para verificar la respuesta de los sistemas a personas que nunca habían visto durante su entrenamiento. A continuación se describe con detalle las características de los experimentos realizados y las mediciones utilizadas.

4.2.1 Modelo A

Se realizaron experimentos con número de usuarios $n = 12$, 24 y 31, con $m = 8$ huellas del mismo dedo por usuario y con m'=4 y 6 para el grupo de entrenamiento y de prueba. En todos los experimentos se usaron claves criptográficas de tamaño $k = 10$ y la dimensión de los vectores de características fue de $p = 512$.

4.2.2 Modelo B

Por cada experimento para diferentes valores de n se realizaron n subexperimentos del modelo B, uno para cada persona del grupo de n individuos considerada como usuario del sistema. Así, cada experimento contó con un total de individuos $n = 12$, 24 y 31, con $m = 8$ huellas del mismo dedo por usuario y con m'=4 y 6 para el grupo

de entrenamiento y de prueba. El tamaño de las claves que se usaron fue de $k = 10$ y la dimensión de los vectores de características de $p = 512$.

4.2.3 Modelo C

Los parámetros y características de este experimento fueron las mismas que las del apartado anterior, anexando únicamente la característica de ruido aleatorio.

4.2.4 Modelo D

Los parámetros y características de este experimento fueron las mismas que las del Modelo C, sin embargo, primero se obtuvieron todas las k permutaciones necesarias del grupo de FingerCodes utilizando el generador de números aleatorios para cada uno de los $n = 12$, 24 y 31 usuarios utilizados.

4.2.5 Modelos A, B Y C con seguridad ante biométrica no confidencial

Se repitieron todos los experimentos de todos los modelos como se describió en la sección 3.5. utilizando permutaciones aleatorias ocultadas en el generador de números aleatorios de Matlab Ver. 6 Release 13 con diferentes semillas (NIP's), una por cada usuario en cada experimento, obtenidas al azar.

4.2.6 Mediciones Aplicadas

A continuación se describen las características que fueron medidas en los experimentos de criptosistemas biométricos.

Porcentaje de Eficiencia

Esta medida fue calculada como la proporción de claves criptográficas recuperadas por el sistema a partir de los datos de huellas digitales de prueba que fueron acertadas a sus correspondientes claves asignadas. Es decir:

$$\text{Porcentaje de eficiencia} = 100 \left[\frac{\text{No. de claves acertadas}}{\text{No. de claves totales} = \text{nm}'} \right]$$

donde n y m' son el No. de personas y el No. de escaneos por huella dígital de la misma persona en los datos de prueba respectivamente. Así, obtenemos una buena medida del desempeño del sistema.

Porcentaje de claves reproducidas

Se mide como la cantidad de veces que cada clave perteneciente al grupo de huellas de Agentes Extraños coincide con las claves de los usuarios del sistema

$$\text{Porcentaje de Claves Reproducidas} = 100 \left[\frac{\text{No. de claves coincidentes}}{nn'} \right]$$

donde n y n' son el número de usuarios en el sistema y el número total de claves de Agentes extraños respectivamente.

Promedio de bits erróneos por clave

Para cada clave generada por el sistema para los datos de prueba se contó la cantidad de bits erróneos. Y luego, se le saco el promedio sobre todas las claves generadas, es decir:

$$\overline{\text{Bits erroneos por Clave}} = \left[\frac{\sum_{i=1}^{k} \text{No. de Bits erroneos en la } i-\text{esima clave}}{nm'} \right]$$

donde n, m' y k son el No. de personas, el No. de escaneos por huella digital de la misma persona y tamaño de la clave criptográfica respectivamente.

4.3 Resumen

En este capítulo se describieron las características de implementación de los algoritmos necesarios que en este caso estuvieron divididos en dos etapas, la primera que extrae el FingerCode de una Huella Digital y la segunda que se encarga de asignarle una clave criptográfica a dicho FingerCode a través de clasificadores SVM's. Además hemos hecho una descripción del tipo de experimentos que fueron realizados tanto para la extracción del FingerCode como para los modelos propuestos. También se especificó, la forma en la que fueron obtenidas las huellas y de donde, cuantas se utilizaron y como.

Capítulo 5

Resultados

Como ya se mencionó anteriormente el trabajo de implementación consistió en dos partes: en la primera, donde se implemento el algoritmo que nos permitió obtener las características de las huellas digitales (FingerCode's) y en la segunda en la que se implementaron y probaron todas las arquitecturas de sistemas ideadas que permitieron asignar y recuperar claves criptográficas a un grupo de personas dadas sus huellas digitales y sus NIP's en los casos correspondientes. A continuación se describen resultados y análisis de resultados obtenidos en los experimentos planteados en el sección 4.2.

5.1 Criptosistemas Biométricos

Los clasificadores como el SVM llegan a ser bastante robustos como para resolver el problema de la variabilidad entre patrones similares, sin embargo, para el traslape el clasificador es el responsable del compromiso que existe entre la probabilidad de aceptación de personas falsas y la probabilidad de rechazo de personas autenticas. Para ello, este construye tal compromiso al momento de ser entrenado y por lo tanto, depende en gran medida del tipo de huellas que son utilizadas durante el entrenamiento. Esto es importante, porque aunque nosotros no estamos interesados en identificación sino asignación de claves criptográficas, para poder llevar a cabo dicha asignación mediante nuestros modelos se esta haciendo de antemano una verificación de identidad implícita. Por ello nuestros modelos podrían servir para dicho fin. Así, los factores de probabilidad RPA y APF se ven disfrazados dentro de la No Recuperación de Claves para Personas Autenticas (NRCPA) y en la Recuperación de Claves Autenticas para Personas Falsas (RCAPF).

En la Figura 5.1 se muestra una grafica que compara las eficiencias mínimas determinadas con un nivel de confianza del 95 % de los 4 modelos propuestos A, B, C y D obtenidas para los experimentos descritos en el apartado 4.2 utilizando 4 huellas por persona para el entrenamiento y 4 para prueba. Como se puede observar en dicha gráfica el Modelo A presenta la más alta probabilidad de NRCPA seguida por el Modelo C y D y finalmente por el Modelo B. Así, la eficiencia tiende a disminuir conforme incremente el número de usuarios para el Modelo A, manteniéndose

prácticamente constante para el Modelo B e incrementando para los Modelos C y D. La razón de porque los Modelos C y D conservan la misma curva de comportamiento, es porque el desempeño de los SVM's se mantiene invariante ante permutaciones simultaneas de las componentes de los vectores de entrada. Esta conclusión fue obtenida experimentalmente al probar los mismos datos utilizados en C en D.

En la Figura 5.2 se muestra la misma gráfica de eficiencia de los mismo 4 modelos pero incrementando la cantidad de huellas por persona utilizadas en el entrenamiento de 4 a 6 y por lo tanto disminuyendo la cantidad de huellas por persona en los datos de prueba de 4 a 2. Así, se puede observar que fue incrementada drásticamente solo en el Modelo A y en el resto se disminuyó ligeramente por lo cual se puede concluir que no hubo un gran cambio significativo.

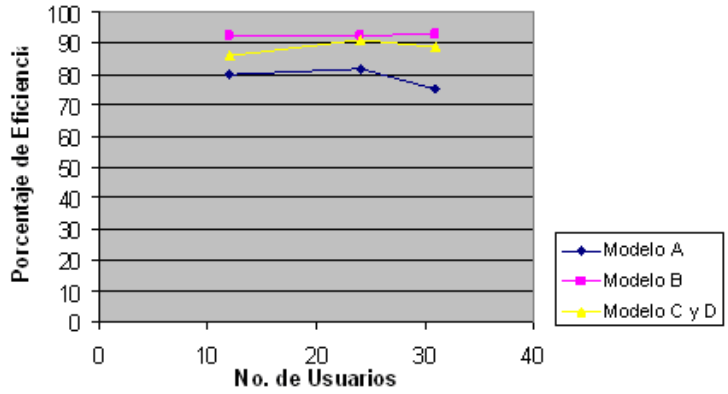

Figura 5.1: Mínimo valor de los intervalos de confianza para la Eficiencia

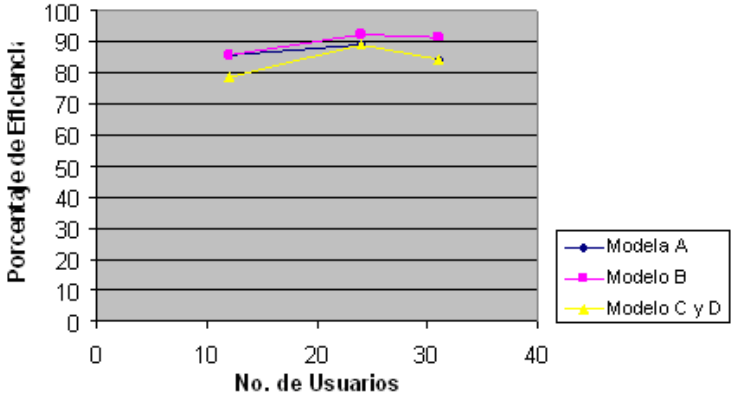

Figura 5.2: Mínimo valor de los intervalos de confianza para la Eficiencia

Ahora, la respuesta RCAPF fue relativamente baja. Para dicho fin se expusieron a los sistemas a 32 individuos distintos que nunca habían visto durante su entrenamiento. Los resultados de dichos experimentos se muestran en la Figura 5.3 donde se presentan los máximos porcentajes posibles con una confianza del 95 %.

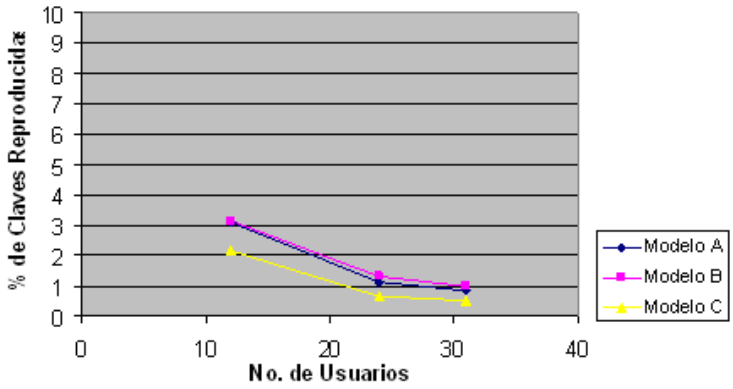

Figura 5.3: Máximo valor de los intervalos de confianza para RCAPF

Como se puede observar es muy difícil que se reproduzcan claves validas a huellas no autorizadas o desconocidas y es aún mas difícil cuando el número de usuarios del sistema incrementa. Para el modelo D no fue probada esta parte ya que como permuta las claves de entrada dependiendo de cierta secuencia generada por el usuario en cuestión, es muy difícil que una huella no autorizada sea reconocida como tal y por lo tanto sea liberada una clave valida.

Como nos pudimos percatar, los SVM's se ajustan de tal manera que se requiere un mayor número de intentos para personas autorizadas para poder generar sus claves criptográficas validas contra una mayor seguridad de reproducir claves a individuos no autorizados. Por otro lado, se pudo ver que en todos estos experimento solo se utilizaron clasificadores lineales, por lo cual, se puede concluir que los FingerCode's tienen la característica de ser linealmente separables. Mismo que no sucede con voz donde el grupo de investigaciones en seguridad del ITESM han tenido que utilizar el kernel RBF (Base de Funciones Radiales) para obtener buenos resultados.

Como se describió en la sección 3.5. los modelos A, B, y C no tienen la capacidad de ser seguros ante la no confidencialidad de las biométricas, por lo tanto nos vimos en la necesidad de combinarlos con algo que si fuera confidencial, y por lo tanto, se repitieron todos los experimentos para los modelos A, B y C utilizando permutaciones que fueron ocultadas en las semillas de un generador de números aleatorios para los mismos FingerCode's utilizados anteriormente. Los resultados encontrados coinciden con los obtenidos en los experimentos anteriores y que son mostrados en las Figuras 5.1 y 5.2 confirmando que los SVM's son invariantes ante dichas permutaciones como se describió anteriormente. Las tolerancias ha individuos no autorizados no fueron probadas aquí ya que las claves correctas responden a permutaciones del FingerCode que son secretas y privadas de cada usuario, por lo tanto, no se espera que alguna persona extraña introdujera su dedo, determinara un NIP cualquiera y se generaran claves correctas de usuarios. En las Figuras 5.1 y 5.2 se muestran los resultados numéricos obtenidos en los experimentos.

5.2 Resumen

En este capitulo presentamos los resultados numéricos obtenidos en los experimentos con los modelos propuestos. Mostramos como las eficiencias de los modelos estaban por encima de un 70 %. Comparamos las eficiencias entre los diferentes modelos y mostramos la alta tolerancia a intruso que estos presentan misma que crece cuando el número de usuarios del sistema incrementa.

Modelo	m'	No. de Usuarios	Porcentaje Eficiencia mínimo	Porcentaje Eficiencia	Porcentaje Eficiencia máximo
A	4	12	80.02	91.6667	97.68
		24	81.7	89.583	94.89
		31	75.3	83.065	89.2
B	4	12	92.61	100	100
		24	92.68	97.9167	99.72
		31	93.1	97.5806	99.498
C	4	12	85.75	95.8333	99.491
		24	91.14	96.875	99.35
		31	88.72	94.3548	97.7
D	4	12	85.75	95.8333	99.491
		24	91.14	96.875	99.35
		31	88.72	94.3548	97.7
A	6	12	85.8	100	100
		24	88.94	97.9167	99.9472
		31	84.3	93.5484	98.21
B	6	12	85.8	100	100
		24	92.61	100	100
		31	91.34	98.3871	99.959
C	6	12	78.9	95.8333	99.894
		24	88.94	97.9167	99.947
		31	84.3	93.5484	98.21
D	6	12	78.9	95.8333	99.894
		24	88.94	97.9167	99.947
		31	84.3	93.5484	98.21

Tabla 5.1: Intervalo de Confianza del 95 % para el porcentaje de eficiencia y porcentaje de eficiencia medido.

Modelo	m'	No. de Usuarios	Porcentaje de claves Reproducidas mínimo	Porcentaje de claves Reproducidas	Porcentaje de claves Reproducidas máximo
A	4	12	1.70995768	2.4038	3.09764232
		24	0.53772994	0.82799	1.11825006
		31	0.41609986	0.64103	0.86596014
B	4	12	1.75597125	2.4573	3.15862875
		24	0.69393136	1.015	1.33606864
		31	0.50215458	0.74442	0.98668542
C	4	12	1.03374766	1.6026	2.17145234
		24	0.23870817	0.45406	0.66941183
		31	0.18472024	0.35153	0.51833976
A	6	12	1.89426577	2.6175	3.34073423
		24	0.45011752	0.72115	0.99218248
		31	0.55437174	0.80645	1.05852826
B	6	12	1.75597125	2.4573	3.15862875
		24	0.8071992	1.1485	1.4898008
		31	0.62458093	0.88916	1.15373907
C	6	12	1.25610375	1.8697	2.48329625
		24	0.23870817	0.45406	0.66941183
		31	0.18472024	0.35153	0.51833976

Tabla 5.2: Intervalo de Confianza del 95 % para el porcentaje de claves reproducidas y porcentaje de claves reproducidas medido a partir de los FingerCode's de Agentes Extraños.

Modelo	m'	No. de Usuarios	Promedio de Bits Erróneos por clave
A	4	12	0.083333
		24	0.14583
		31	0.29839
B	4	12	0
		24	0.10417
		31	0.12903
C	4	12	0.041667
		24	0.0625
		31	0.14516
D	4	12	0.041667
		24	0.0625
		31	0.14516
A	6	12	0
		24	0.041667
		31	0.080645
B	6	12	0
		24	0
		31	0.064516
C	6	12	0.125
		24	0.020833
		31	0.12903
D	6	12	0.125
		24	0.020833
		31	0.12903

Tabla 5.3: Promedio de la cantidad de bits erróneos encontrada en cada clave asignada

Capítulo 6

Aplicación en Administración de Derechos Digitales (ADD)

La Administración de Derechos Digitales (DRM por sus siglas en ingles) es una tecnología que proporciona protección a la información digital utilizando para dicho fin encriptación, certificados y autentificación. Los destinatarios o usuarios autorizados deben de obtener un permiso para utilizar los materiales protegidos, ya sean estos, documentos, música, películas, etc., de acuerdo con los derechos y reglas de negocio definidas por el propietario del contenido.

Esto surgió con el fin de controlar la piratería, pues de esta manera, se tiene un mejor control sobre lo que se le permite y como se le permite a un usuario, hacer uso de dicha información. Las aplicaciones de tal tecnología son muy diversas, por ejemplo, en la industria de la música, se evita la reproducción y distribución no autorizada de material no permitiendo la capacidad de copiado, o por otro lado, en información tal como investigación o de cualquier otro tipo pertenecientes a industrias, se puede permitir el acceso solo a las personas autorizadas a través de la red con ciertas restricciones colocadas en el material tales como fechas de caducidad, incapacidad de impresión y de copiado, etc.

Así como se ha explicado, los DRM's pueden ser software. Algunos, funcionan como virus que se introducen en las computadoras para cumplir con su cometido tal como es el caso de los programas introducidos por sony BMG en los CD's de audio que una vez que son utilizados en los lectores de las computadoras se instalan clandestinamente y su función es evitar manipulaciones no autorizadas. En otras ocasiones, dicho software es implementado en aplicaciones para la reproducción de audio y video. También se han diseñado aplicaciones que son como plataformas específicas que cumplen con esta función y que permiten manipular información muy diversa.

Los modelos propuestos en este trabajo podrían ser utilizados para asignar autorizaciones a individuos a los cuales se les permite utilizar el contenido, es decir, ya que estos sistemas transfieren la información codificada, se podrían asignar las claves de decodifi-

cación del material a las huellas digitales de los individuos en cuestión, de esta manera, las claves no permanecerían explícitamente almacenadas en los sistemas DRM, sino que, se encontrarían almacenados dentro de los clasificadores SVM's. Así, para la utilización de dicho material sería necesaria la identificación implícita a través de la huella digital que desencadenaría la clave y por lo tanto, la información. Se puede observar entonces que, si un individuo no autorizado intentara utilizar dicho contenido, simplemente no podría decodificarlo. La arquitectura del DRM tendría que ser la mostrada en la Figura 6.1 Nótese, que es necesaria la existencia del DRM, que funciona como una plataforma que controla los permisos, una vez liberada la información.

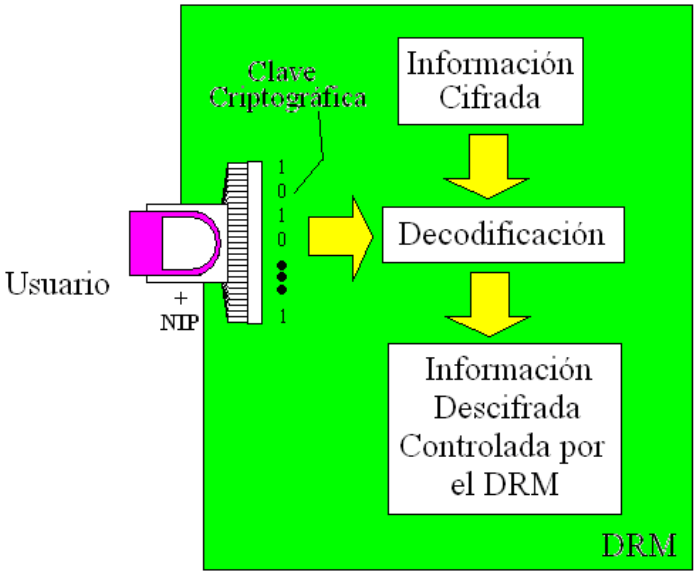

Figura 6.1: Arquitectura para la asignación de autorización en un DRM con huella digital.

Puesto que cada contenido esta codificado con la clave de su correspondiente usuario, a pesar de que, el DRM no conoce tal clave, identifica a los usuarios de acuerdo a los contenidos que les pertenecen y que se abren correctamente después del decodificado. Así, es capaz de aplicar los permisos de manipulación correspondientes a cada usuario. Esto obliga a que cada contenido sea codificado tantas veces como usuarios autorizados para el existan.

De esta manera, todo tipo de contenido estaría protegido contra manipulaciones no autorizadas del mismo.

6.1 Resumen

En este capítulo se describió la manera en la que pueden ser utilizados los modelos propuestos en esta tesis en la Administración de Derechos Dígitales.

Capítulo 7

Conclusiones

En este último capítulo se muestran las conclusiones obtenidas al final de la presente investigación y las contribuciones de éstas, así como el posible trabajo a futuro en favor de la extensión misma.

7.1 Conclusiones

Como se mostró en el transcurso de este trabajo, existen dos factores que complican el proceso de asignar claves criptográficas a las características de textura de la huella digital, la primera es la variabilidad tan grande que tienen las componentes de los vectores de FingerCode y la segunda el traslape que existe entre dichos vectores. Tratamos de solventar ambas dificultades utilizando un clasificador de patrones que ha mostrado ser muy exitoso en un sin fín de tareas conocido como Máquina de Vectores de Soporte ideado por Vapnik. Sin embargo, el desempeño obtenido fue apenas del 70 % de eficiencia, pero con una altísima tolerancia a impedir desencadenar claves validas a individuos no autorizados.

Una característica sobresaliente de esta investigación fue que los SVM's eran invariantes ante permutaciones aleatorias de los vectores de entrenamiento. Misma que nos permitió poder mezclar la biométrica con un NIP sin perder eficiencia que es una característica que se pierde cuando la seguridad incrementa, sin embargo, la seguridad es ahora responsabilidad del nip. Un ejemplo de esto se puede ver directamente sobre el modelo B que muestra niveles de eficiencia muy altos, pero que sin embargo, pierde seguridad al ser fácil de quebrantar.

También mostramos que utilizar SVM's lineales son suficientes para clasificar vectores de FingerCode para la recuperación de claves. Además cabe remarcar también que nuestro trabajo fue desarrollado bajo la suposición de que el usuario es cooperativo, condición que no siempre es valida.

cación del material a las huellas digitales de los individuos en cuestión, de esta manera, las claves no permanecerían explícitamente almacenadas en los sistemas DRM, sino que, se encontrarían almacenados dentro de los clasificadores SVM's. Así, para la utilización de dicho material sería necesaria la identificación implícita a través de la huella digital que desencadenaría la clave y por lo tanto, la información. Se puede observar entonces que, si un individuo no autorizado intentara utilizar dicho contenido, simplemente no podría decodificarlo. La arquitectura del DRM tendría que ser la mostrada en la Figura 6.1 Nótese, que es necesaria la existencia del DRM, que funciona como una plataforma que controla los permisos, una vez liberada la información.

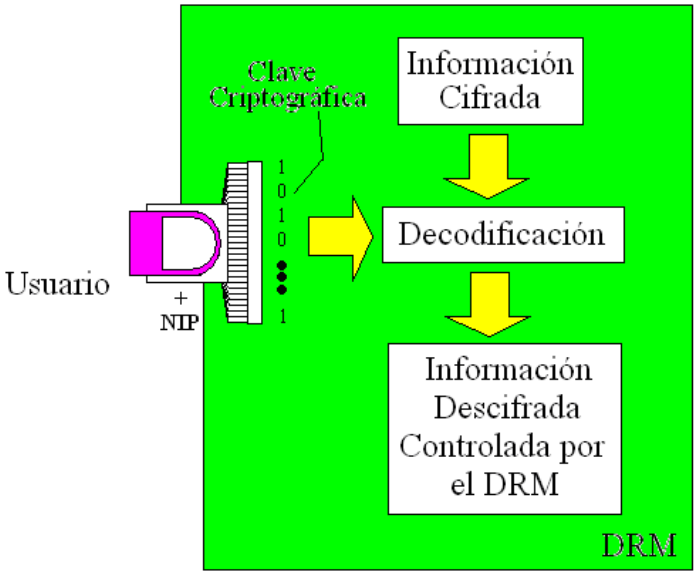

Figura 6.1: Arquitectura para la asignación de autorización en un DRM con huella digital.

Puesto que cada contenido esta codificado con la clave de su correspondiente usuario, a pesar de que, el DRM no conoce tal clave, identifica a los usuarios de acuerdo a los contenidos que les pertenecen y que se abren correctamente después del decodificado. Así, es capaz de aplicar los permisos de manipulación correspondientes a cada usuario. Esto obliga a que cada contenido sea codificado tantas veces como usuarios autorizados para el existan.

De esta manera, todo tipo de contenido estaría protegido contra manipulaciones no autorizadas del mismo.

6.1 Resumen

En este capítulo se describió la manera en la que pueden ser utilizados los modelos propuestos en esta tesis en la Administración de Derechos Dígitales.

7.2 Trabajo Futuro

El trabajo realizado en la presente tesis se relaciono principalmente con la experimentación de sistemas que generan claves criptográficas apartir de patrones de textura de la huella dígital utilizando clasificadores SVM. Existe aún mucho trabajo por realizar en esta línea. Primero podríamos tratar de explorar la utilización de una combinación de los dos tipos diferentes de patrones de la huella dígital como son: las minucias y la textura. También se podrían utilizar estas con uno o varios nips para generar permutaciones aleatorias de los patrones. Además de esto se podrían explorar la combinación de varias biométricas.

Por otro lado también sería interesante probar si es posible obtener mejores rendimientos utilizando Kernel's no lineales en los SVM's, ya que, como se ha mostrado por otros miembros de equipo de investigaciones en seguridad del ITESM, la utilización del Kernel que usa funciones base radiales ha mostrado tener excelentes resultados sobre patrones de voz.

Además se podría probar con algunos otros clasificadores existentes como lo son: redes neuronales, etc.

Por ultimo, como se explico anteriormente, los resultados de este trabajo fueron obtenidos bajo la suposición de que los usuarios son cooperativos por lo tanto quearia abierta la posibilidad de explorar la rentabilidad bajo condiciones no cooperativas.

Apéndice A

SVM Lineal

Un SVM lineal es un clasificador lineal con margen máximo. Esto significa que un SVM asigna un hiperplano a la función *f(x)* tal que exista la mayor distancia posible entre él y los vectores x_i pertenecientes a la clase A en un lado y los vectores x_i pertenecientes a la clase B del otro como se muestra en la Figura A.1 donde a los vectores x_i que están marcados con círculos y que definen este margen se les conoce como vectores de soporte.

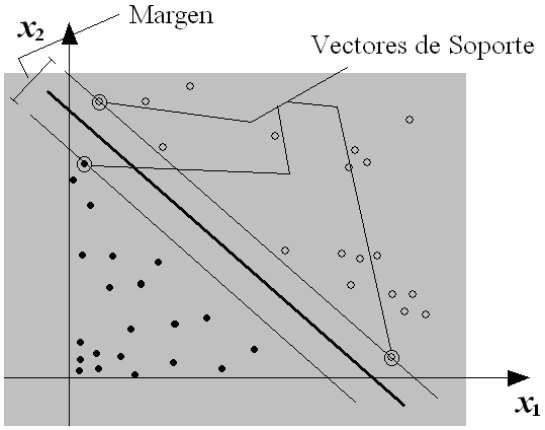

Figura A.1: Margen y Vectores de Soporte.

Entonces para encontrar el hiperplano óptimo primero vamos a considerar el caso en el que los datos de entrenamiento son linealmente separables propuesto por Vapnik & Lerner en 1963 [20] y luego el caso más general en el cual los datos no son del todo separables, pero que sin embargo, se le puede ajustar un hiperplano de separación publicado por Cortes y Vapnik en 1995 [6].

Caso separable

Supongamos nuevamente que tenemos un conjunto de entrenamiento xi, yi con i=1,2,. . .,N donde x_i son vectores en R^n con clasificación y en R^N cuyas componentes yi toman valores en el conjunto -1,1. Ahora, el hiperplano que separa a los vectores

de las dos clases satisface la siguiente ecuación w·x+b = 0 donde w es normal al hiperplano, |b|/‖w‖ es la distancia perpendicular del hiperplano al origen y ‖w‖ es la magnitud de w. Así, Vapnik consideró un problema reescalado tal que los puntos más cercanos al hiperplano satisfagan las siguientes condiciones:

$$\begin{aligned} \mathbf{x}_i \bullet \mathbf{w} + b \geq 1 \quad \text{para } y_i = 1 \\ \mathbf{x}_i \bullet \mathbf{w} + b \geq -1 \text{ para } y_i = -1 \end{aligned} \tag{A.1}$$

mismas que pueden ser reescritas en una sola desigualdad como sigue:

$$y_i \left(\mathbf{x}_i \bullet \mathbf{w} + b\right) \geq 1 \tag{A.2}$$

En este caso, el margen es igual a $2/\|\mathrm{w}\|$ y los puntos más cercanos al hiperplano tienen la distancia $1/\|\mathrm{w}\|$. Por lo tanto, maximizar el margen, equivale a minimizar la magnitud de w.

Así, buscamos minimizar la siguiente función:

$$J(\mathbf{w}, b) = \frac{1}{2}\mathbf{w} \bullet \mathbf{w} \tag{A.3}$$

$$\text{con restricciones } y_i \left(\mathbf{x}_i \bullet \mathbf{w} + b\right) - 1 = 0 \text{ para i=1,2,. . .,N.}$$

Este problema es resuelto utilizando los multiplicadores de Lagrange. Primero, definamos la siguiente función:

$$J_P(\mathbf{w}, b) = \frac{1}{2}\mathbf{w} \bullet \mathbf{w} - \sum_{i=1}^{N} \alpha_i \left[y_i \left(\mathbf{x}_i \bullet \mathbf{w} + b\right) - 1\right] \tag{A.4}$$

donde los α_i son multiplicadores de lagrange. También definamos el operador gradiente como:

$$\nabla = \frac{\partial}{\partial \mathbf{w}} + \frac{\partial}{\partial b}\hat{\mathbf{b}} \tag{A.5}$$

por lo tanto, al aplicar el operador gradiente a J_P(w,b) tenemos las siguientes condiciones que se deben de cumplir junto con las ecuaciones de restricción:

$$\begin{gathered} \nabla J_P(\mathbf{w}*, b*) = 0 \\ \left(\frac{\partial}{\partial \mathbf{w}} J_P(\mathbf{w}*, b*) + \frac{\partial}{\partial b} J_P(\mathbf{w}*, b*)\hat{\mathbf{b}}\right) - \sum_{i=1}^{N} \alpha_i \left(\frac{\partial}{\partial \mathbf{w}} \left[y_i \left(\mathbf{x}_i \bullet \mathbf{w} * + b*\right) - 1\right] + \frac{\partial}{\partial b} \left[y_i \left(\mathbf{x}_i \bullet \mathbf{w} * + b*\right) - 1\right] \hat{\mathbf{b}}\right) = 0 \\ \left(\frac{\partial}{\partial \mathbf{w}} \left(\frac{1}{2}\mathbf{w} * \bullet \mathbf{w}*\right) + \frac{\partial}{\partial b} \left(\frac{1}{2}\mathbf{w} * \bullet \mathbf{w}*\right) \hat{\mathbf{b}}\right) - \sum_{i=1}^{N} \alpha_i \left(\frac{\partial}{\partial \mathbf{w}} \left[y_i \left(\mathbf{x}_i \bullet \mathbf{w} * + b*\right) - 1\right] + \frac{\partial}{\partial b} \left[y_i \left(\mathbf{x}_i \bullet \mathbf{w} * + b*\right) - 1\right] \hat{\mathbf{b}}\right) = 0 \\ \mathbf{w} * - \sum_{i=1}^{N} \alpha_i \left(y_i \mathbf{x}_i + y_i \hat{\mathbf{b}}\right) = 0 \\ \mathbf{w} * - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i + \sum_{i=1}^{N} \alpha_i y_i \hat{\mathbf{b}} = 0 \\ \mathbf{w} * + (\alpha \bullet \mathbf{y})\, \hat{\mathbf{b}} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \end{gathered} \tag{A.6}$$

y finalmente la forma de w que satisface lo anterior es:

$$\mathbf{w}* = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \quad \text{con restricciones } \alpha \cdot \mathrm{y} = 0 \tag{A.7}$$

Para encontrar los valores óptimos de α_i reemplazamos la ecuación para w* de (A.7) en J_P(w,b) y la maximizamos con respecto a α, ya que se sabe que, la solución es un punto de silla en J_P(w,b,α), y encontrar dicho punto es equivalente a minimizar J(w,b) con las restricciones (A.2) lo cual da el siguiente problema de Programación Cuadrática (PC) como el problema dual en los multiplicadores de Lagrange:

$$\max_{\alpha} J_D(\alpha) = \max_{\alpha} \left[\sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \bullet \mathbf{x}_j \right) \right] \quad \text{con restricciones } \alpha \cdot \mathrm{y} = 0 \tag{A.8}$$

Por lo tanto el clasificador queda expresado como:

$$f(\mathbf{x}) = \theta \left(\sum_{i=1}^{N} \alpha_i y_i \left(\mathbf{x}_i \bullet \mathbf{x} \right) + b \right) \tag{A.9}$$

Una propiedad importantes del separador optimo es que los valores α_i de los vectores de entrenamiento diferentes de los de soporte son 0. Por lo tanto, debido a que normalmente siempre existen menos vectores de soporte que datos, el número de parámetros que definen al separador es normalmente más pequeño que N. También es notable el hecho de que, en el problema de PC los vectores xi aparecen como productos puntos de pares de ellos y lo mismo sucede con la función del clasificador en la cual el vector de entrada aparece como un producto punto con los vectores x_i.

Caso no separable

En la mayoría de los casos los datos no son linealmente separables como se muestra en la Figura A.2.

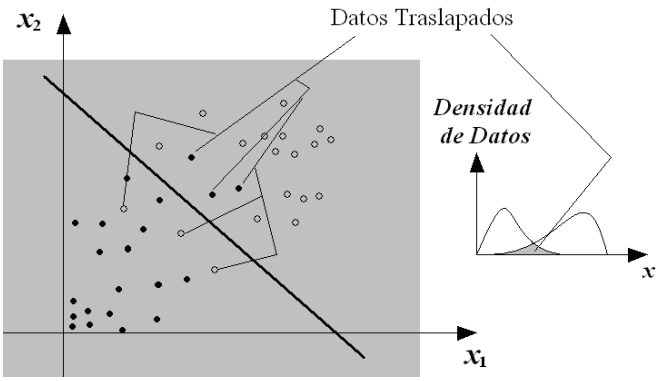

Figura A.2: Datos traslapados

Así la extensión del SVM Lineal al caso no separable se logra introduciendo variables adicionales de costo en la formulación del problema. Por lo tanto para tolerar el traslape se modifican los conjuntos de desigualdades (A.2) de la siguiente manera:

$$y_i \left(\mathbf{x}_i \bullet \mathbf{w} + b\right) \geq 1 - \xi_i \text{ con i} = 1, 2, ..., \text{N} \tag{A.10}$$

Con variables ξ_i>0. Cuando ξ_i>1, la i-ésima desigualdad es violada en comparación con la correspondiente desigualdad del caso linealmente separable. Por lo tanto, el problema de optimización se convierte en minimizar:

$$\begin{gathered} J(\mathbf{w}, b, \xi) = \tfrac{1}{2}\mathbf{w} \bullet \mathbf{w} + c\sum_{i=1}^{N} \xi_k \\ \text{con restricciones } y_i \left(\mathbf{x}_i \bullet \mathbf{w} + b\right) = 1 - \xi_i \text{ para i} = 1, 2, \ldots, \text{N y } \xi_\text{i} > 0 \end{gathered} \tag{A.11}$$

donde c es una constante real positiva que entre más grande sea, más alta es la penalidad a violaciones de las desigualdades. De manera similar al caso separable, se define la siguiente función:

$$J_P(\mathbf{w}, b, \xi) = J\left(\mathbf{w}, b, \xi\right) - \sum_{i=1}^{N} \alpha_i \left[y_i \left(\mathbf{x}_i \bullet \mathbf{w} + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{N} \nu_i \xi_i \tag{A.12}$$

donde los α_i y v_i son multiplicadores de lagrange. También definiendo el operador gradiente como:

$$\nabla = \frac{\partial}{\partial \mathbf{w}} + \frac{\partial}{\partial b}\hat{\mathbf{b}} + \frac{\partial}{\partial \xi} \tag{A.13}$$

por lo tanto aplicando el operador gradiente (A.13) a (A.12) tendremos el siguiente conjunto de condiciones que se deben de satisfacer junto con las ecuaciones de restricción (A.10):

$$\begin{gathered} \mathbf{w}* = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \\ \text{con restricciones } \alpha \cdot \text{y} = 0 \text{ y } 0 \leq \alpha_\text{i} \leq \text{c} \end{gathered} \tag{A.14}$$

que dan el siguiente problema de PC dual después de remplazar la ecuación para w* de (A.14) en (A.12):

$$\begin{gathered} \max_{\alpha} J_D\left(\alpha\right) = \max_{\alpha} \left[\sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \bullet \mathbf{x}_j\right)\right] \\ \text{con restricciones } \alpha \cdot \text{y} = 0 \text{ y } 0 \leq \alpha_\text{i} \leq \text{c} \end{gathered} \tag{A.15}$$

que es el mismo problema que el caso separable pero con la restricción adicional de que $0 \leq \alpha_i \leq$c y con la misma forma funcional del clasificador:

$$f\left(\mathbf{x}\right) = \theta\left(\sum_{i=1}^{N} \alpha_i y_i \left(\mathbf{x}_i \bullet \mathbf{x}\right) + b\right) \tag{A.16}$$

Apéndice B

Intervalo de Confianza para una Proporción

Durante este trabajo fue necesario formalizar estadísticamente las mediciones obtenidas, por lo tanto, en este apéndice se presenta a grandes rasgos la teoría para el cálculo de intervalos de confianza con muestras aleatorias grandes y pequeñas. Para mayores detalles se recomienda consultar algún texto de estadística.

Intervalo de Confianza para una Proporción con Muestras Grandes

Supongamos la existencia de una muestra aleatoria con n datos procedentes de alguna población P y supongamos también que x es la variable aleatoria que mide el número de éxitos de alguna propiedad en la muestra. Entonces el intervalo de confianza para la proporción p de éxitos de dicha propiedad en la población con un nivel de confianza de (1 - α) que es un valor entre 0 y 1 esta dado por:

$$\widehat{p} - z_{\alpha/2}\sqrt{\frac{\widehat{p}(1-\widehat{p})}{n}} < p < \widehat{p} + z_{\alpha/2}\sqrt{\frac{\widehat{p}(1-\widehat{p})}{n}} \tag{B.1}$$

donde $\hat{p} = \frac{x}{n}$ y $z_{\alpha/2}$ es el valor de la variable con distribución normal estándar z tal que:

$$\mathrm{P}(\mathrm{z} > \mathrm{z}_{\alpha/2}) = \alpha/2 \tag{B.2}$$

Intervalo de Confianza para una Proporción con Muestras Pequeñas

Supongamos la existencia de una muestra aleatoria con n datos provenientes de alguna población P que tiene una proporción p que satisface alguna propiedad m y supongamos también que x es la variable aleatoria que mide la cantidad de elementos

que satisfacen m en la muestra. Entonces, x sigue una Distribución Binomial f(x) que tiene la siguiente forma:

$$f(x) = \left(\begin{array}{c} n \\ x \end{array} \right) p^x \left(1-p\right)^{n-x} \tag{B.3}$$

con promedio np y varianza np(1-p). Llamemos entonces a F(x) la Distribución de Probabilidad Acumulada de f(x) que esta dada por:

$$F(x) = \sum_{x_0=0}^{x} \left(\begin{array}{c} n \\ x_0 \end{array} \right) p^{x_0} \left(1-p\right)^{n-x_0} \tag{B.4}$$

Por lo tanto, un intervalo de confianza con un nivel de confianza de (1-α) que es un valor entre 0 y 1 estará dado por los límites p_1 y p_2 que satisfacen:

$$\begin{array}{l} \sum\limits_{x_0=0}^{k-1} \left(\begin{array}{c} n \\ x_0 \end{array} \right) p_1^{x_0} \left(1-p_1\right)^{n-x_0} = 1 - \frac{\alpha}{2} \\ \sum\limits_{x_0=0}^{k} \left(\begin{array}{c} n \\ x_0 \end{array} \right) p_2^{x_0} \left(1-p_2\right)^{n-x_0} = \frac{\alpha}{2} \end{array} \tag{B.5}$$

Resolviendo estas ecuaciones numéricamente para p_i tenemos entonces que el intervalo de confianza para p que esta dado por:

$$\text{mín}\left(p_1, p_2\right) < p < \text{máx}\left(p_1, p_2\right) \tag{B.6}$$

Bibliografía

[1] Advances in fingerprint technology, 1991.

[2] Visión por computador: Imágenes digitales y aplicaciones, 2002.

[3] Handbook of fingerprint recognition, 2003.

[4] S. A. Stojanov R. Gilroy C. Soutar, D. Roberge and B. V. K. Vijaya Kumar. Biometric encryption using image processing. *SPIE*, 1998.

[5] N. Kiyavash T. C. Clancy and D. J. Lin. Secure smartcard-based fingerprint authentication. *ACM SIGMM*, 2003.

[6] Vapnik V. Cortes C. Support vector networks, machine learning. *Machine Learning*, 1995.

[7] J. G. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. Pattern Anal*, 1993.

[8] John G. Daugman. High confidence visual recognition of persons by test of statistical independence. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 1993.

[9] Nozha Boujemaa David Vitale Sylvain Bernard and Claude Bricot. Fingerprint segmentation using the phase of multiscale gabor wavelets. *The 5ht Asian Conference on Computer Vision*, 2002.

[10] FBI. The science of fingerprints. US Department of Justice.

[11] B. J. Matt G. I. Davida, Y. Frankel and R. Peralta. On the relation of error correction and cryptography to an offline biometric based identification scheme. *Workshop Coding and Cryptography*, 1999.

[12] Y. Frankel G. I. Davida and B. J. Matt. On enabling secure applications through off-line biometric identification. *IEEE Symp. Privacy and Security*, 1998.

[13] Nolazco Juan A. García, Perera L. Paola and Mex Perera Carlos. A phoneme-space-representation heuristic to improve the performance in a cryptographic-speech-key generation task. 2005.

[14] Anil K. Jain. Filterbank-based fingerprint matching. *IEEE transactions on Image Processing*, 2000.

[15] R. M. Bolle A. K. Jain and Eds. S. Pankanti. Biometrics: Personal identification is a networked society. *Norwell*, 1999.

[16] Stephen M. Matyas Allen Roginsky Mohammad, Peyravian and Nev Zunic. Generating user-based cryptographic keys and random numbers. 1999.

[17] R. K. Nichols. Biometric encryption. *in ICSA Guide to Cryptography*, 1999.

[18] C. E. Shannon. Communication theory of secrecy systems, bell. *Syst. Tech. J.*, 1949.

[19] D. Roberge Stojanov, R. Gilroy Soutar and Vijaya Kumar. Biometric encryption using image processing. *SPIE*, 1998.

[20] Lerner A. Vapnik V. Pattern recognition using generalized portrait method, automation and remote control. 1963.