Memorias del XVI Congreso Latinoamericano de Control Automático, CLCA 2014
Octubre 14-17, 2014. Cancún, Quintana Roo, México

Cifrado caótico de plantilla de huella dactilar en sistemas biométricos

Murillo-Escobar M.A. * **Cruz-Hernández C.** **
Abundiz-Pérez F. * **López-Gutiérrez R.M.** *

* *Facultad de Ingeniería, Arquitectura y Diseño, Universidad Autónoma de Baja California (UABC), Ensenada, BC, México.*
** *Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), Ensenada, BC, México.*

Resumen: En las últimas décadas, la biometría ha ganado terreno en el campo de la autentificación e identificación de personas debido a propiedades como singularidad, accesibilidad, aceptabilidad, inalterabilidad, universalidad, etcétera. Sin embargo, la transmisión de datos a través de canales inseguros y almacenamiento de información biométrica es un problema de seguridad en la actualidad. Por otra parte, los sistemas caóticos poseen características y propiedades criptográficas muy útiles para el diseño de algoritmos de cifrado eficientes y seguros. En este trabajo, se presenta una aplicación del algoritmo de cifrado caótico presentado previamente en Murillo-Escobar *et al.* (2014), para cifrar plantillas de huella dactilar en base a secuencias caóticas optimizadas del mapa logístico y una ronda de permutación-difusión para proteger los datos y evitar robo de identidad. La plantilla de la huella dactilar (2072 bytes de tres muestras) se extrae por medio de un módulo lector de huella dactilar y un microcontrolador de 32 bits, para su análisis en computadora. Se presenta un análisis de seguridad basado en simulación en MatLab para verificar la eficiencia, seguridad del cifrado propuesto y su potencial uso en sistemas embebidos de autentificación biométrica.

Palabras clave: análisis de seguridad, autentificación, cifrado caótico, huella dactilar, sistemas biométricos.

1. INTRODUCCIÓN

En la actualidad, la autentificación juega un papel básico en nuestra vida diaria, ya sea en viajes, trabajos, universidades, fábricas, hospitales, computadoras, banca en línea, compras en línea, correo electrónico, entre muchos otros, donde se requiere que la persona se autentifique de alguna forma para poder acceder a sitios restringidos o realizar operaciones financieras. Los métodos tradicionales de autentificación que se basan en tarjetas de identificación, códigos de identificación personal, contraseñas, etc. están perdiendo terreno debido a que no son seguros y tampoco son prácticos.

Por otra parte, la biometría es capaz de medir mediante métodos automáticos las características fisiológicas humanas o los comportamientos personales, tales como rostro, iris, huella dactilar, geometría de mano, firma, voz, etc. (características universales, inalterables, únicas y medibles, de cada persona); muchos sistemas biométricos han sido utilizados para la identificación y autentificación de personas, siendo la huella dactilar el más utilizado, ya que es práctico, seguro, altamente aceptado y los costos de implementación son bajos. Los procesos básicos en un sistema biométrico son: *enrolamiento* que consiste en extraer una o más muestras del identificador biométrico con un método (por ejemplo, extracción de minucias en huella dactilar) y generar una plantilla para almacenarla en memoria (en sitio o remotamente) para futuras comparaciones; *verificación* que consiste en comparar el biométrico actual con el biométrico almacenado (para autentificación *uno a uno* y para identificación *uno a todos*) para determinar si concuerdan, Soutar *et al.* (1999).

Uno de los problemas de seguridad que presentan los sistemas de autentificación biométrica, como el robo de identidad, radica en el almacenamiento de las plantillas (remota o localmente) y una defensa es el cifrado de plantilla antes de su almacenamiento o transmisión a través de un canal inseguro, Soutar *et al.* (1999), Ratha *et al.* (2001), Roberts (2007) y Vielhauer *et al.* (2013); en base a este problema de seguridad, la primera solución fue propuesta por Ratha *et al.* (2001) a lo que llamaron biométrico cancelable. Las técnicas convencionales como TDEA (Triple Data Encryption Algorithm) y AES (Advanced Encryption Standard), son estándares de cifrado simétricos (una llave secreta para cifrar y descifrar) utilizados actualmente para cifrado de datos. Sin embargo, TDEA es un algoritmo lento comparado con AES, mientras que AES tiene desventajas en el número de rondas de cifrado; ademas, AES se puede expresar mediante una fórmula algebraica compacta, aunque actualmente no es posible

⋆ Agradecemos al CONACYT, por el apoyo económico brindado a través del proyecto de Grupos de Investigación en Ciencia Básica, Ref. 166654.
⋆⋆Autor de correspondencia: Cruz-Hernández C. Tel.: +52.646.1750500, Fax:+52.646.1750554, CICESE, Ensenada, B.C., México. (correo electrónico: *ccruz@cicese.mx*)

resolver ecuaciones algebraicas de este tipo, la seguridad del algoritmo podría caer en los próximos 10 años, cuando alguien desarrolle una solución a este tipo de ecuaciones, Ferguson *et al.* (2001).

Por otra parte, los sistemas caóticos poseen características como ergodicidad, mezcla de datos, sensibilidad a condiciones iniciales, sensibilidad a parámetros de control, que son comparadas con propiedades criptográficas como permutación, difusión, secuencias pseudoaleatorias y complejidad del sistema fuente, lo que ha generado trabajos en cifrado de imagen biométrica, Inzunza-González y Cruz-Hernández (2013), Liu (2012), Cui (2010), Alghamdi y Ullah (2010), Alghamdi *et al.* (2009), Zhao *et al.* (2008), Khan *et al.* (2007), y Han *et al.* (2007). Recientemente, los sistemas caóticos también han sido utilizados para la protección de plantillas biométricas en George (2013), Moujahdi *et al.* (2012), Wang *et al.* (2011), Arjunwadkar y Kulkarni (2010), pero carecen de análisis de seguridad para verificar y validar la seguridad del proceso de protección propuesto.

En este trabajo, se presenta una aplicación de un algoritmo simétrico de cifrado caótico basado en mapa logístico, una ronda de permutación-difusión y en una llave secreta de 128 bits para la protección de plantilla de huella dactilar y evitar robo de identidad. Una plantilla de 2072 bytes es leída de un módulo de huella dactilar conectado con un microcontrolador de 32 bits y extraída con una memoria USB para el análisis en computadora. Se presenta un análisis de seguridad basado en simulaciones en MatLab y se muestra que el esquema de protección propuesto es práctico, seguro y eficiente para su uso en sistemas embebidos de autentificación biométrica.

La organización del trabajo es de la siguiente forma: una breve descripción del esquema de autentificación biométrica se muestra en la Sección 2, así como también, el esquema de protección de plantilla biométrica que se utiliza en este trabajo. En la Sección 3 se muestran los detalles del algoritmo de cifrado basado en trabajo previo, Murillo-Escobar *et al.* (2014). Los resultados experimentales y análisis de seguridad se presentan en la Sección 4 en base a simulaciones en MatLab. En la Sección 5, se mencionan las consideraciones para que el esquema propuesto sea llevado a nivel de sistema embebido. Finalmente, las conclusiones son mencionadas en la Sección 6.

2. ESQUEMA DE AUTENTIFICACIÓN BIOMÉTRICA

El proceso de autentificación biométrica tradicional se basa en la lectura del biométrico mediante un sensor, procesamiento-extracción de características y comparación con plantilla previamente almacenada en la base de datos. Inicialmente, en el proceso de *enrolamiento* se determina un biométrico físico o de comportamiento, como iris, rostro, huella dactilar, palma de la mano, voz, firma, etc., para utilizase como identificador; después, un sensor realiza la adquisición de datos; posteriormente, un algoritmo procesa los datos y extrae características biométricas (plantillas); finalmente, la plantilla es almacenada en una memoria local o remota para futuras comparaciones. Una vez que el *enrolamiento* se ha concluido con éxito, se puede realizar el proceso de *verificación* para autentificar al usuario con los siguientes pasos: leer el biométrico, extraer características y comparar *uno a uno* con la plantilla previamente almacenada (fig. 1), Luis-García (2003).

En la figura 2, se muestra el esquema de protección de plantilla biométrica implementada en este trabajo. Antes del almacenamiento de plantillas en el proceso de *enrolamiento*, la plantilla es cifrada mediante un algoritmo en base a secuencias caóticas y una llave secreta de 128 bits para evitar un ataque por fuerza bruta exitoso (Sec. 4.1) y para determinar la condición inicial-parámetro de control del mapa logístico (Tabla 1); una vez que se ha cifrado la información, esta se puede almacenar local o remotamente. En el proceso de autentificación, la plantilla es descifrada con el mismo algoritmo caótico y comparada con una plantilla reciente (del mismo biométrico) del usuario para validar su identificación.

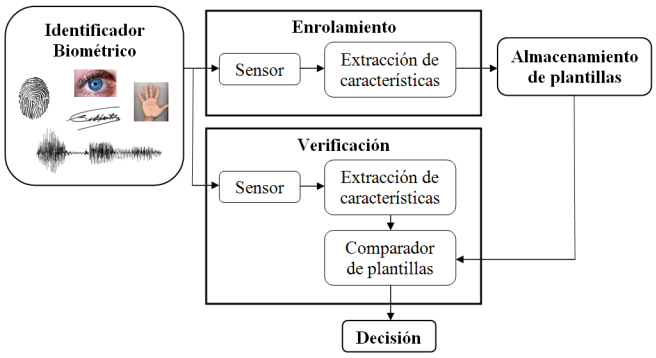

Figura 1. Esquema de autentificación biométrica.

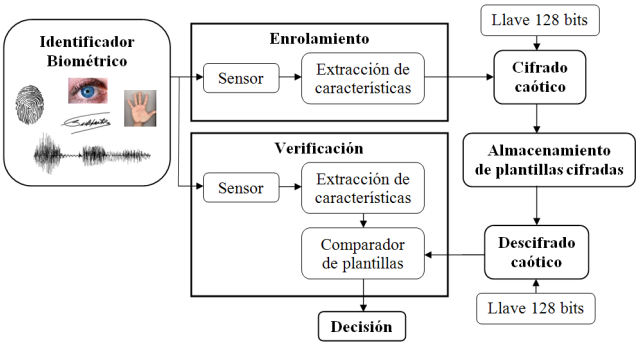

Figura 2. Esquema de autentificación biométrica con protección de plantilla por cifrado caótico antes de su almacenamiento.

3. ALGORITMO DE CIFRADO CAÓTICO

El proceso de cifrado consiste en generar una plantilla cifrada de una plantilla clara, mediante un algoritmo que utiliza la arquitectura de permutación-difusión (modificar posición-valor) mediante secuencias caóticas del mapa logístico. El proceso de cifrado se basa en un trabajo previo Murillo-Escobar *et al.* (2014); se considera como plantilla clara $P \in [0, 255]$ con una longitud de $\ell = 2072$ (bytes) datos extraídos del módulo lector de huella dactilar. El mapa logístico es el sistema caótico unidimensional más conocido, May (1976); el mapa es de simple estructura y se describe con la siguiente ecuación en diferencias no lineal

$$x_{i+1} = a x_i (1 - x_i), \tag{1}$$

donde $x_i \in (0,1)$ es el estado del sistema, $x_0 \in (0,1)$ es la condición inicial y $a \in (3.999{,}4)$ es el parámetro de control para generar caos y evitar ventanas periódicas. Los sistemas caóticos como el mapa logístico tienen desventajas para su implementación en sistemas de cifrado como rangos caóticos discontinuos, distribución no uniforme, periodicidad y espacio de llaves pequeño; mientras que las ventajas son la simple estructura, su fácil implementación y alta tasa de salida.

En el cifrado caótico utilizado en este trabajo se hacen consideraciones para evitar las desventajas y mantener las ventajas del mapa logístico, generando un sistema de cifrado altamente eficiente y seguro que consiste de los siguientes pasos:

Paso 1 (llave secreta): Se define como 128 bits (32 caracteres hexadecimales) $K \in [0-9, A-F]$ para generar secuencias caóticas de dos mapas logísticos de acuerdo a la Tabla 1 y con ello, se evita un espacio de llaves pequeño. Además, se tienen suficientes llaves para evitar un ataque por fuerza bruta.

Paso 2 (valor Z): Todos los valores de la plantilla clara se suman con datos caóticos del mapa logístico 2 iterado $I = 2172$ con a_2 y x_{2_0} de la Tabla 1. Los elementos de la plantilla y datos caóticos se suman como sigue

$$S = \left\{S + [P_i * x^{L2}_{2173-i}] + x^{L2}_{2173-i}\right\} \ mod\ 1, \qquad (2)$$

donde $i = 1, 2, 3, \ldots, 2072$, P_i son los elementos de la plantilla clara, S es una variable inicializada en cero, x^{L2} es la secuencia caótica por el mapa logístico 2 y mod es la operación de módulo. Para evitar la posible cancelación de Z se escala entre $1-255$ como sigue

$$Z = (round(S * 254) + 1)\,/256, \qquad (3)$$

donde $Z \in (0,1)$ con precisión decimal de 10^{-15} y $round$ es la operación de redondeo al valor mas cercano.

Paso 3 (cifrado): El mapa logístico 1 es iterado $T = 5{,}000$ con a_1 y x_{1_0} como se muestra en Tabla 1; se determina una secuencia para permutación y otra para difusión con (4) y (5), respectivamente.

$$Q_i = \ round\left(x^{L1}_{2928+i} * 2071\right) + 1, \qquad (4)$$

donde $i = 1, 2, 3, \ldots, 2072$, $Q \in [1, 2072]$ y $round$ es la operación de redondeo al valor mas cercano; Q contiene valores repetidos que son determinados por software y reemplazados por los faltantes de forma automática para generar una permutación optimizada, es decir, todas y cada una de las posiciones del vector de la plantilla clara se permutadas sin excepción.

$$F_i = \left\{\left(x^{L1}_{2928+i} * 1000\right) + Z\right\} \ mod\ 1 \qquad (5)$$

donde $i = 1, 2, 3, \ldots, 2072$; $F_i \in (0,1)$; la distribución del mapa logístico 1 (figura 3(a), donde se tienen muchos valores cercanos a 0 y a 1, que afectan en el proceso de cifrado) es mejorada al multiplicarse por 1000 para generar un proceso de difusión optimizado, es decir, generar una secuencia caótica con una mejor distribución de datos caóticos (figura 3(b), con distribución uniforme para generar un mejor cifrado). Después, el vector F_i es transformado con la siguiente igualdad

$$Y_i = \ round\left(F_i * 255\right), \qquad (6)$$

donde $i = 1, 2, 3, \ldots, 2072$ con $Y_i \in [0, 255]$. Finalmente, el proceso de cifrado se realiza con la siguiente operación

$$E_i = (P(Q_i) + Y_i) \ mod\ 256, \qquad (7)$$

donde $i = 1, 2, 3, \ldots, 2072$ y $E \in [0, 255]$ es la plantilla cifrada (criptograma). El valor de Z es agregado al final del criptograma $E_{2073} = (S * 254) + 1$.

Descifrado: El proceso de descifrado consiste en invertir los pasos de cifrado. Calcular Q_i y Y_i con la misma llave secreta del cifrado y con $Z = E_{2073}/256$.

Para recuperar la plantilla clara se realiza el siguiente cálculo

$$D(Q_i) = (E_i - Y_i) \ mod\ 256, \qquad (8)$$

donde $i = 1, 2, 3, \ldots, 2072$, Q y Y es el vector de permutación y difusión, respectivamente, E es el criptograma y D es la plantilla descifrada.

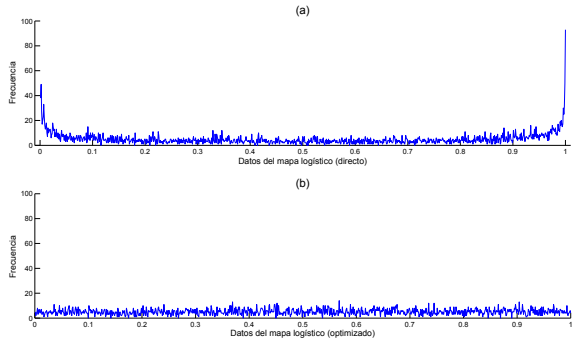

Figura 3. Distribución de 5000 datos caóticos del mapa logístico: (a) datos directos y (b) datos optimizados.

4. RESULTADOS EXPERIMENTALES

Es esta sección, se analiza la robustez del cifrado de la plantilla biométrica. El módulo de huella dactilar Futronic FS83 se utiliza para leer y extraer la plantilla clara (2072 bytes en base al método de extracción de minucias) con especificaciones por fabricante de tasa de aceptación falsa 10^6 (uno en un millón) y tasa de rechazo falso 10^2 (uno en cien); también, se utiliza un microcontrolador Freescale 32 bit COLDFIRE M52259 como dispositivo maestro y con una memoria USB se extraen los datos de la huella dactilar para realizar los respectivos análisis en computadora con MatLab (fig. 4).

Tabla 1. Llave secreta y su distribución en dos mapas logísticos, Murillo-Escobar *et al.* (2014).

Llave secreta	Parámetro de control	Condición inicial
32 dígitos hexadecimal	$H_1, H_2, \ldots, H_{32}$ donde $H \in [0-9, A-F]$	
cálculos	$A = \frac{(H_1, H_2, \ldots, H_8)_{10}}{2^{32}+1}$	$C = \frac{(H_{17}, H_{18}, \ldots, H_{24})_{10}}{2^{32}+1}$
	$B = \frac{(H_9, H_{10}, \ldots, H_{16})_{10}}{2^{32}+1}$	$D = \frac{(H_{25}, H_{26}, \ldots, H_{32})_{10}}{2^{32}+1}$
logístico 1	$a_1 = 3{,}999 + (\alpha_1 * 0{,}001)$	$x_{1_0} = \beta_1 \ mod\ 1$
	$\alpha_1 = (A + B + Z) \ mod\ 1$	$\beta_1 = C + D + Z$
logístico 2	$a_2 = 3{,}999 + (\alpha_2 * 0{,}001)$	$x_{2_0} = \beta_2 \ mod\ 1$
	$\alpha_2 = (A + B) \ mod\ 1$	$\beta_2 = C + D$

4.1 Ataque por fuerza bruta

Un ataque por fuerza bruta consiste en probar cada una de las posibles llaves secretas hasta descifrar la plantilla clara. Para que el sistema criptográfico pueda resistir a este tipo de ataque, en la actualidad se requiere de por

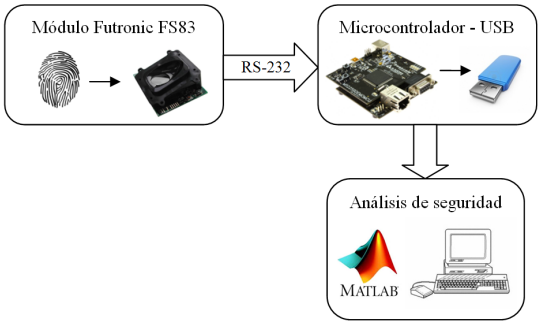

Figura 4. Extracción de plantilla dactilar para su análisis en MatLab.

lo menos 2^{100} posibles combinaciones de llaves secretas, Alvarez y Li (2006); en el esquema propuesto se utilizan 2^{128} posibles combinaciones de llaves secretas, por lo que se considera que el esquema propuesto puede resistir un ataque por fuerza bruta.

4.2 Sensibilidad a la llave secreta

En el proceso de cifrado, cuando la LLAVE secreta es ligeramente modificada se debe producir un criptograma muy diferente incluso con el uso de la misma plantilla clara; también, el proceso de descifrado debe generar una plantilla descifrada muy diferente al variar ligeramente la llave secreta. Para este análisis, se utiliza una plantilla clara de 25 elementos fijos en un valor de 100 y tres 3 llaves secretas muy parecidas (diferentes en un bit); las figuras 5 y 6, muestran la alta sensibilidad a llave secreta en proceso de cifrado y descifrado, respectivamente. Por lo tanto, el esquema presentado es altamente sensible a la llave secreta.

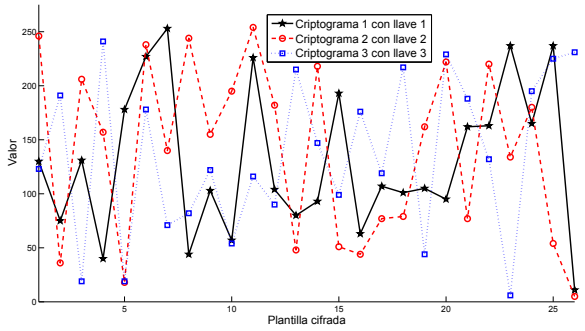

Figura 5. Cifrado de 25 elementos fijos en 100 con tres llaves diferentes en un bit.

4.3 Sensibilidad a la plantilla clara

Además de la sensibilidad del cifrado a la llave secreta, también debe presentar sensibilidad a pequeños cambios en la plantilla clara. NPCR (tasa de cambio de pixel neto) y UACI (intensidad de cambio promedio unificado) determinan la sensibilidad del cifrado a la plantilla clara. En este análisis, con el uso de la misma LLAVE secreta K_1 se generan dos plantillas cifradas E_1 y E_2, de dos plantillas claras A y B muy parecidas (diferentes en un bit). El valor

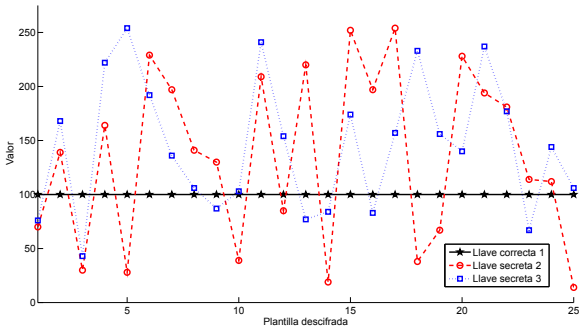

Figura 6. Descifrado de 25 elementos fijos en 100 con tres llaves diferentes en un bit.

de NPCR y UACI es determinado con la ecuación (9) y (11), respectivamente.

$$NPCR = \frac{\sum_{i=1}^{i=2073} W(i)}{2073} \times 100 \tag{9}$$

donde

$$W(i) = \begin{cases} 0 & \text{si } E_1(i) = E_2(i) \\ 1 & \text{si } E_1(i) \neq E_2(i) \end{cases} \tag{10}$$

y

$$UACI = \frac{100}{2073} \sum_{i=1}^{i=2073} |E_1(i) - E_2(i)| \tag{11}$$

En este análisis, se considera como plantilla clara A, a la plantilla clara de la fig. 7 y como plantilla clara B, a la misma plantilla clara pero con un bit diferente. Las dos plantillas claras A y B, son cifradas con la misma LLAVE secreta K_1 para generar E_1 y E_2. En base a lo anterior, se determinan valores de $NPCR = 99.2\,\%$ y $UACI = 88.3\,\%$, por lo que se comprueba la sensibilidad del cifrado a la plantilla clara.

4.4 Histogramas

El proceso de difusión tiene como objetivo uniformizar el criptograma para evitar un ataque estadístico exitoso, por lo que se debe generar un criptograma con la mayor parte de los elementos y que cada elemento se genere con una probabilidad similar (histograma uniforme). En las figuras 7 y 8, se muestran los histogramas de la plantilla clara con 2072 bytes y la plantilla cifrada con 2073 bytes, respectivamente; por lo tanto, el cifrado genera un histograma uniforme y además, cubre todo el rango de posibles elementos.

4.5 Entropía de la información

La entropía de la información mide que tan impredecible es un mensaje y en este caso, la utilizaremos para determinar que tanto desorden genera el algoritmo de cifrado. Si el proceso de difusión es bueno, este generará una plantilla cifrada con mucho desorden, ruido, etc., y se tendrá una entropía alta; caso contrario, si la entropía del mensaje es baja, esto indica que el cifrado no es suficientemente pseudoaleatorio y puede generar datos predecibles que son utilizados en un ataque de entropía.

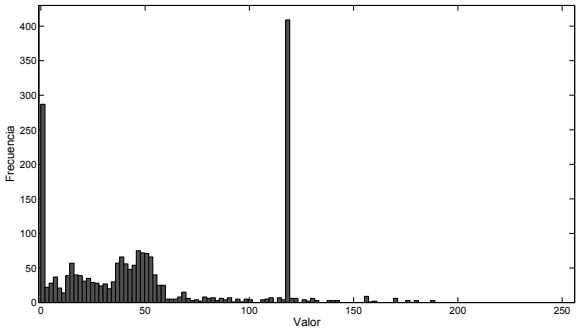

Figura 7. Histograma de plantilla clara.

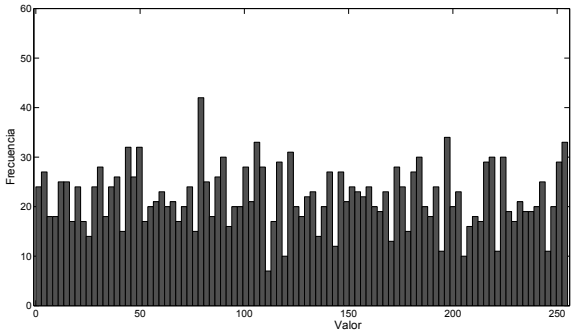

Figura 8. Histograma de plantilla cifrada.

La entropía de un mensaje se calcula con la siguiente expresión

$$H(T) = \sum_{i=0}^{2^N-1} p\left(T_i\right) log_2 \left(\frac{1}{p\left(T_i\right)}\right) \qquad (12)$$

donde $H(T)$ es la entropía en bits, N es el número de bits que representa a un elemento de la secuencia T, 2^N son las posibles combinaciones y $p(T_i)$ es la probabilidad del elemento T_i. Cada elemento de la plantilla clara o cifrada es representado por 8 bits (0-255), por lo que la entropía máxima es $H = 8$; mayor la entropía, más impredecible resulta el cifrado. Los resultados de la entropía de la plantilla clara es $H = 5.49$, mientras que la entropía de la plantilla cifrada es $H = 7.93$; por tanto, los datos generador por el cifrado se consideran impredecibles, es decir, los valores generados en el proceso de cifrado no tienden a valores específicos, al contrario, genera cada posible valor con una probabilidad similar.

4.6 Ataque de plantilla clara elegida/conocida

El criptoanalista puede atacar un sistema criptográfico para encontrar la llave secreta bajo el principio de Kerckhoffs: el algoritmo de cifrado se conoce excepto la llave secreta, Petitcolas (2011). El ataque de plantilla clara elegida y ataque de plantilla clara conocida, se basan en la suposición de que una llave secreta se utiliza para generar varios criptogramas, por lo que el criptoanalista puede implementar este tipo de ataques sobre el algoritmo de cifrado para determinar la llave secreta que se ha utilizado durante un tiempo y con ello descifrar otros criptogramas que tiene a su alcance. Un ataque de plantilla clara elegida/conocida es muy poderoso y puede quebrantar el sistema criptográfico si no se hace una consideración importante en el proceso de cifrado caótico: las secuencias caóticas utilizadas para el cifrado deben ser diferentes para cada plantilla clara considerando el uso de la misma llave secreta.

En un esquema de cifrado donde se utiliza la arquitectura de permutación-difusión, si se elige una plantilla clara para cancelar el proceso de difusión (por ejemplo, una plantilla clara definida en ceros), el criptograma puede representar la secuencia caótica que se utilizó para cifrar (llave secreta) y puede ser utilizada por el criptoanalista para descifrar otros criptogramas que fueron previamente cifrados con esta llave secreta. El el proceso de cifrado de este trabajo, las secuencias caóticas para permutación y difusión, son determinadas de la llave secreta, de la plantilla clara y de datos caóticos del mapa logístico 2. Por tanto, en cada cifrado se generan secuencias caóticas diferentes, incluso con el uso de la misma llave secreta y con una plantilla clara definida en ceros; este proceso evita un ataque de plantilla clara elegida/conocida exitoso. En las figuras 9 y 10, se muestran la sensibilidad de los vectores de permutación y difusión, respectivamente, donde tres plantillas muy parecidas (que varían en un bit entre ellas) son cifradas con la misma llave secreta.

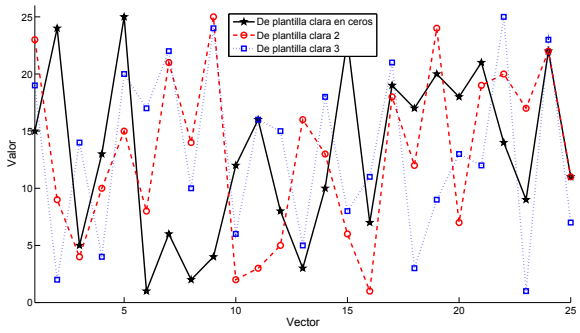

Figura 9. Sensibilidad del vector de permutación a llave secreta y plantilla clara.

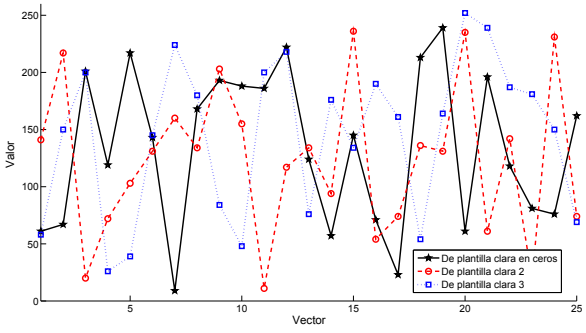

Figura 10. Sensibilidad del vector de difusión a llave secreta y plantilla clara.

4.7 Tiempo de cifrado

Los sistemas de autenticación se implementan en tiempo real y el tiempo de procesamiento de datos es una característica importante. En este trabajo, el proceso de lectura

y extracción de plantilla de huella dactilar se realiza con microcontrolador en base a programación en lenguaje C y un reloj de 80 MHz, lo que genera un tiempo de lectura de 0.9 segundos; mientras que, el proceso de cifrado de 2073 bytes es de 0.25 segundos basado en simulación en MatLab. Por lo tanto, el proceso de protección de plantilla dactilar puede implementarse en aplicaciones de tiempo real.

5. IMPLEMENTACIÓN EN SISTEMA EMBEBIDO

En esta Sección, se mencionan las consideraciones para que el proceso de cifrado propuesto sea trasladado a un sistema embebido de bajo costo, sin que se tenga dependencia ni de una computadora ni de MatLab. En general, un sistema embebido es un sistema para controlar una aplicación específica (alarma, televisor, lavadora, etc.) que consta de un módulo central programado (microcontrolador, FPGA (Arreglo de compuertas lógicas programables), ASIC (Circuito integrado de aplicación específica), etc.), sensores (presión, humedad, temperatura, huella digital, etc.) y periféricos de entrada-salida (teclado, LCD, memoria USB, etc.). Por otra parte, la seguridad y desempeño del esquema propuesto en este trabajo fue analizado y verificado a nivel de simulaciones en MatLab; la lectura de la huella ya se tiene implementado en el sistema embebido, de tal manera que, se requiere realizar la programación (lenguaje C) del algoritmo de cifrado y algoritmo de descifrado en microcontrolador, además de incluir un teclado matricial para controlar los distintos procesos (enrolamiento y verificación) y una pantalla LCD como interfaz humana.

6. CONCLUSIONES

Los datos biométricos requieren ser cifrados antes de su almacenamiento o de su transmisión para garantizar la confidencialidad de datos personales y evitar robo de identidad. En este trabajo se presentó un esquema de protección de plantilla de huella dactilar basado en cifrado caótico; los análisis de simulaciones en MatLab muestran la seguridad, efectividad y desempeño del esquema propuesto. Además, el esquema presentado puede ser utilizado en la protección de plantillas de iris, rostro, palma de mano, etc., y puede ser implementado en sistemas embebidos de autenticación biométrica basados en microcontroladores, arreglo de compuertas lógicas programables (FPGA), circuito integrado de aplicación especifica (ASIC), etc., para aumentar la confidencialidad de los datos biométricos almacenados o transmitidos a través de canales inseguros.

REFERENCIAS

Alghamdi A.S. y Ullah H. (2010). A Secure Iris Image Encryption Technique Using Bio-Chaotic Algorithm. *International Journal of Computer Applications*, 3, 10-12.

Alghamdi A.S., Ullah H., Mahmud M. y Khan M.K. (2009). Bio-Chaotic Stream Cipher-Based Iris Image Encryption. *International Conference on Computational Science and Engineering*, 2, 739-744.

Alvarez G. y Li S. (2006). Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems. *International Journal of Bifurcation and Chaos*, 16(8), 2129-2151.

Arjunwadkar M. y Kulkarni R.V. (2010). Robust Security Model for Biometric Template Protection using Chaos Phenomenon. *International Journal of Computer Applications*, 3, 10-12.

Cui D. (2010). A Novel Fingerprint Encryption Algorithm Based on Chaotic System and Fractional Fourier Transform. *International Conference on Machine Vision and Human-machine Interface*, 168-171.

Ferguson N., Schroeppel R. y Whiting D. (2001). A simple algebraic representation of Rijndael. *Lecture Notes in Computer Science*, 2259, 103-111.

George R.M. (2013). Facial Template Protection Using Extended Visual Cryptography And Chaotic Encryption. *International Journal of Technology Enhancements and Emerging Engineering Research*, 1(4), 94-96.

Han F., Hu J., Yu X. y Wang Y. (2007). Fingerprint images encryption via multi-scroll chaotic attractors. *Applied Mathematics and Computation*, 185, 931-939.

Inzunza-González E. y Cruz-Hernández C. (2013). Double Hyperchaotic Encryption for Security in Biometric Systems. *Nonlinear Dynamics and Systems Theory*, 13(1), 55-68.

Khan M.K., Zhang J. y Tian L. (2007). Chaotic secure content-based hidden transmission of biometric templates. *Chaos, Solitons & Fractals*, 32, 1749-1759.

Liu R. (2012). Chaos-Based Fingerprint Images Encryption Using Symmetric Cryptography. *9th International Conference on Fuzzy Systems and Knowledge Discovery*, 2153-2156.

Luis-García R. de, Alberola-López C., Aghzout O. y Ruiz-Alzola J. (2003). Biometric identification systems. *Signal Processing*, 83 (2003), 2539-2557.

May R.M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261(5560), 459-467.

Moujahdi C., Ghouzali S., Mikram M., Rziza M. y Bebis G. (2012). Spiral Cube for Biometric Template Protection. *Lecture Notes in Computer Science*, 7340, 235-244.

Murillo-Escobar M.A., Abundiz-Pérez F., Cruz-Hernández C. y López-Gutiérrez R.M. (2014). A novel symmetric text encryption algorithm based on logistic map. *International Conference on Communications, Signal Processing and Computers*, 49-53.

Ratha N.K., Connell J.H. y Bolle R.M. (2001). Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3), 614-634.

Petitcolas F.A.P. (2011). Kerckhoffs' Principle. *Encyclopedia of Cryptography and Security*, 675.

Roberts C. (2007). Biometric attack vectors and defences. *Computers & Security*, 26, 14-25.

Soutar C., Roberge D., Stoianov A., Gilroy R. y Kumar V. (1999). Biometric Encryption. *ICSA Guide to Cryptography*, chapter 22.

Vielhauer C., Dittmann J. y Katzenbeisser S. Design Aspects of Secure Biometric Systems and Biometrics in the Encrypted Domain. *Security and Privacy in Biometrics*, 25-43, (2013).

Wang X., Xu T. y Zhang W. (2011). Chaos-Based Biometrics Template Protection and Secure Authentication. *State of the art in Biometrics*, chapter 15.

Zhao S., Li H. y Yan X. (2008). A secure and efficient fingerprint images encryption scheme. *The 9th International Conference for Young Computer Scientists*, 2803-2808.