Algoritmo para la identificación de usuarios usando características de tecleo

A. Hernández, J. Guadalupe, K. Martín y G. González

A. Hernández, J. Guadalupe, K. Martín y G. González.
Universidad Politécnica del Centro, Tabasco, México.
joselupe_2002@hotmail.com

M. Ramos.,V.Aguilera.,(eds.). Ciencias de la Ingeniería y Tecnología, Handbook -©ECORFAN- Valle de Santiago, Guanajuato, 2013.

Abstract

This paper presents a method for users authentication using as parameter the users keystroke dynamics. The authentication model is based on the comparison of templates, each template is made up of the times in which each user performs the events of releasing button and releasing press a key, these times are handled with an accuracy of four digits for the comparison of similarity of templates applied on statistical scatter functions, obtaining a percentage of acceptance (PA) compared with percentage of similarity (PS) decides to accept or reject a user. During the tests were calculated errors of false acceptance and false rejection first one, mentioned gaining 0.0%.

1 Introducción

En la actualidad la seguridad informática juega un papel importante, algunas veces hemos escuchado o leído frases como: la información es dinero ó la información es poder, estas nos hacen pensar en qué podría pasar si la información confidencial que tiene una empresa saliera a la luz pública, imaginarnos ésta situación nos hace reflexionar en lo valioso que es la información y las medidas de seguridad que se deben tomar en cuenta para que únicamente las personas autorizadas tengan acceso a ella.

De acuerdo a Davies (2002), un modo de acceso tradicional a los sistemas de cómputo es el basado en contraseña, el propósito de la contraseña es verificar que el usuario es quien dice ser, es decir la contraseña actúa como mecanismo que autentifica al usuario. Sin embargo este método de autenticación presenta algunos inconvenientes debido a su simplicidad como los siguientes: los usuarios adoptan como contraseñas palabras obvias como su nombre, sus iníciales, fecha de nacimiento, las cuales pueden ser robadas fácilmente; un intruso puede ver lo que teclea el usuario en el momento de autentificarse; ó mediante programas ejecutados en segundo plano grabar lo que el usuario teclea y así conocer su contraseña. De acuerdo a los puntos anteriores nos damos cuenta que la contraseña no es suficiente para tener la seguridad que el usuario es físicamente quien dice ser.

Los mecanismos de autenticación según Obaidat (2002) se dividen en tres grupos: algo que el usuario conoce: como una contraseña; algo que el usuario posee: como una tarjeta y algo que el usuario es: a través de técnicas biométricas. La técnica que podemos emplear para realmente saber si el usuario es físicamente quien dice ser es la biometría, esta se clasifica en biometría estática y biometría dinámica, la primera identifica a una persona por un rasgo físico que lo hace diferente de cualquiera y la segunda identifica a una persona midiendo su comportamiento. Dentro de la biometría dinámica existe una técnica para autentificar a un usuario en base a su dinámica de tecleo (según Marino (2010) se llama dinámica de tecleo a los patrones de tecleo asociados a la velocidad de tecleo y al tiempo de presión al teclear) llamada biometría de tecleo.

1.1 Metodología

La parte básica para la autenticación es una interfaz que sea capaz de recolectar los tiempos de tecleo de cada usuario al momento de autenticarse así como también al crear sus plantillas de tecleo, esta interfaz debe proporcionarnos un conjunto de tiempos pertenecientes a una secuencia de caracteres escritos.

Los elementos necesarios para el desarrollo de este interfaz y de la aplicación biométrica son: rutinas para la detección de eventos del teclado, un contador de tiempo con una precisión de cuatro cifras mínimo para la diferenciación de los tiempos en cada usuario y normalización de estos tiempos, para realizar una autenticación en red.

1.2 Detección de los eventos del teclado

La detección de los eventos del teclado en los lenguajes de programación de alto nivel no es una tarea difícil ya que éstos incorporan rutinas que se encargan del manejo de los eventos de teclado como son: pulsar tecla o soltar tecla. Deseamos medir el comportamiento del usuario ante el teclado para esto mediremos las características siguientes:

El tiempo que transcurre cuando el usuario presiona una tecla y suelta la misma tecla, a este evento llamaremos pulsar – soltar.

El tiempo que transcurre cuando el usuario suelta una tecla y presiona la tecla siguiente, a este evento le llamaremos soltar - pulsar.

Figura 1 Medición de tiempos Pulsar – Soltar

Cada uno de los eventos mencionados anteriormente se medirá por cada uno de los caracteres de la cadena que el usuario establezca como nombre de usuario o contraseña. Es decir, si la cadena fuese por ejemplo “CASA” entonces tendríamos cuatro tiempos del evento *pulsar – soltar* y tres tiempos del evento *soltar – pulsar*.

Figura 1.1 Número de muestras de tiempo para el evento pulsar – soltar

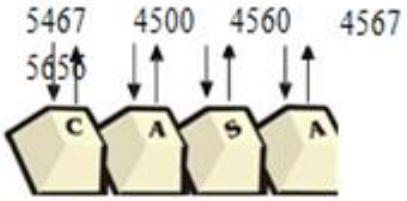
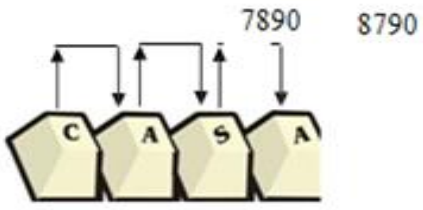


Figura 1.2 Número de muestras de tiempo para el evento soltar - pulsar



Entonces tendríamos lo siguiente:

$$\#\text{ Tiempos eventos pulsa – soltar} = n$$

$$\#\text{ Tiempos eventos } \textit{soltar} – \textit{pulsar} = n - 1 \qquad (1)$$

Donde n es el número de caracteres de la cadena.

1.3 Contador para la medición de tiempos de tecleo

El siguiente paso para el modelo es la implementación de un contador que indicará el tiempo que transcurre en cada uno de los eventos del teclado, este contador es deseable que se incremente con suficiente rapidez de tal manera que por ejemplo para el evento pulsar – soltar el tiempo que transcurren entre pulsar la tecla y soltar la tecla tenga cuatro cifras como mínimo. Entre más rápido se incremente el contador, existirá mayor diferenciación en la dinámica de tecleo de un usuario a otro, ya que los intervalos de tiempo estarán más separados.

Así entonces, la velocidad de tecleo puede ser una característica importante para la diferenciación de los usuarios.

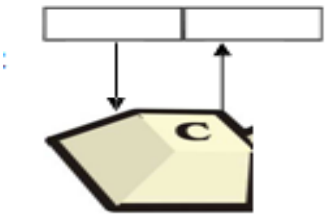
El manejo e implementación de estos contadores depende en gran medida del sistemaoperativo, para ésta aplicación trabajaremos sobre plataforma Windows, en la tabla 1 se muestra un comparación entre distintos contadores indicando el número de cifras que se proporcionan en la medición de cada uno de los eventos.

Tabla 1 Comparación de funciones de tiempo (número de cifras)

Dimensión	Pulsar - Soltar	
Timer	2	2 a 3
GetTickCount	2	2 a 3
System.currecntTimeMillis	2	2 a 3
QueryPerformanceCounter	5	4 a 5

En la figura 1.3 se muestra un ejemplo para calcular el corte de tiempo del evento pulsar soltar tecla, se intenta mostrar que de acuerdo a la rapidez con la que se incremente el contador será el número de cifras que obtengamos.

Figura 1.3 Cálculo del tiempo para el evento pulsar – soltar



Tiempos eventos pulsa – soltar = n

Tiempos eventos *soltar – pulsar* = n -1

Donde n es el número de caracteres de la cadena.

(1.1)

El contador que elegiremos para nuestro trabajo es la primitiva del API de Windows QueryPerformanceCounter, esta accede a un reloj de alta precisión del hardware del sistema.

La precisión de este reloj depende del hardware específico por lo tanto para saber cuantos ciclos marca nuestro sistema en un segundo se obtiene a través de la primitiva Query Performance Frequency.

En un Pentium IV a 2.79 GHz llamando esta función devuelve la cantidad de 3579545 es decir, que en cada ciclo transcurre aproximadamente la tercera parte de una millonésima de segundo. Query Performance Counter puede ser invocada desde cualquier lenguaje de programación bajo la plataforma Windows y devuelve una cantidad numérica de alrededor de once cifras que son los ciclos del procesador que han transcurrido desde que se encendió la computadora.

Debido a que el contador depende directamente del hardware la velocidad con la que se incrementa es variable de una computadora a otra, como la base para la dinámica de tecleo es la velocidad con la que el usuario teclea entonces el tiempo debe ser más o menos normal cada vez que se autentifique. En Marino (2010) se proponen dos técnicas para normalizar los tiempos que se obtengan de cada usuario, normalización por min/max y normalización por la media, dejando ver claro que la normalización por la media es mejor para atacar este problema.

1.4 Normalización por la media

Consideremos el caso de medir tiempos generados por el evento soltar – pulsar tecla, dada una secuencia de caracteres S = {s1, s2, … sn,}, le corresponde un tiempo de tecleo T = {t1, t2, … tn-1} donde n es el número de caracteres de la secuencia, ahora denotemos a xTs como el patrón de tecleo de usuario normalizado por un factor (x) correspondiente a la velocidad del procesador de la computadora X, y lo mismo para yTs en otra computadora Y con distinta velocidad de procesador. Entonces tenemos:

Computadora X = xTs = {xt1, xt2, ...,xtn-1}
Computadora Y = yTs = {yt1, yt2, ...,ytn-1 }

(1.2)

Ahora llamamos m = (t1 + t2 + ... + tn-1) / (n – 1) media de los tiempos de tecleo. De forma que tendremos en la máquina X, xm = media de los tiempos de tecleos en la computadora X. y lo mismo para la computadora Y. Así denotamos:

xT's = { xTs normalizado por la media } .:

xT's = xTs / xm = { xt1 / xm , xt2 / xm, ... , xtn-1 / xm }

Y del mismo modo;

$$yT's = yTs / ym = \{ yt1 / ym, yt2 / ym, ... , ytn-1 / ym \} \quad (1.3)$$

Entonces xT's ≅ yT's independientemente de cuáles sean las velocidades de las computadora X y Y. Para el caso de los tiempos generados por los eventos pulsar – soltar la demostración es análoga. Para ilustrar el proceso de esta técnica se realizó un sencillo experimento en dos computadoras, la primera con procesador Pentium ® IV a 2.79 GHz, y la segunda computadora con procesador Pentium ® IV a 1.8 GHz. La secuencia caracteres utilizada fue BIOMETRIA.

Tabla 1.1 Muestra de tiempos en computadoras con procesadores diferentes, para la palabra BIOMETRIA

	B-I	**I-O**	**O-M**	**M-E**	**E-T**	**T-R**	**R-I**	**I-A**	**Media**
2.79 GHz.	654	753	970	563	614	418	770	559	662.63
1.8 GHz.	264	454	408	409	285	275	420	261	347

La diferencia de velocidad en los dos procesadores es de 0.99 GHz. se obtuvieron tiempos mayores en el procesador a 2.79 como se observa en la tabla 1.1. Observando los tiempos podríamos decir que las muestras no son del mismo usuario, a continuación aplicaremos el proceso de normalización por la media. Calculamos la media de cada muestra teniendo 662.63 para el procesador a 2.79 GHz y 347 para el procesador a 1.8 GHz, ahora dividiendo cada tiempo t_n entre su respectiva media.

Tabla 1.2 Resultados del proceso de normalización

	B-I	**I-O**	**O-M**	**M-E**	**E-T**	**T-R**	**R-I**	**I-A**	**Media**
2.79 GHz.	0.98	1.13	1.46	0.84	0.92	0.63	1.16	0.84	0.99
1.8 GHz.	0.76	1.30	1.17	1.17	0.82	0.79	1.21	0.75	0.99

En la tabla 1.2 se puede observar que los tiempos de los dos procesadores muestran menor diferencia, y la media general tiende a ser aproximadamente igual.

Grafico 1 Gráfica de los tiempos no normalizados

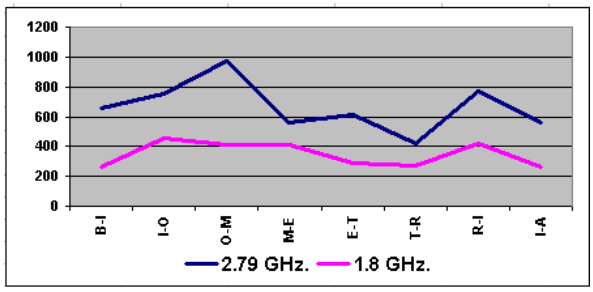
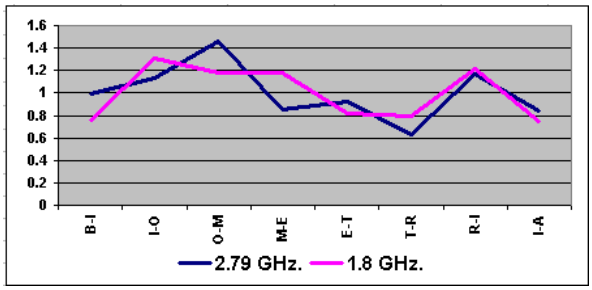


Grafico 1.1 Gráfica de los tiempos normalizados



Observando las gráficas anteriores, se muestra mejor en qué consiste el proceso de normalización de las muestras. En la fig. 5 las líneas generadas por los tiempos tn no normalizados se encuentran separadas aunque se puede observar cierta similitud en cuanto a las ondas o curvaturas; ahora bien cuando se aplica el proceso de normalización el valor de los tiempos cambia para las dos muestras pero al estar normalizados por la media no se pierden las ondas de las curvas por lo que cada línea generada es similar a la original pero con tiempos normalizados como se muestra en la grafico 1.1.

Tabla 1.4 Ejemplo de una nueva muestra de tiempo

	W-O	O-O	O-D	D-Y	Y-S	S-A	A-R	R-G	G-E
Nueva	0.835	1.456	1.783	1.435	0.627	1.216	1.519	0.915	0.515

Buscamos una función que nos indique si el tiempo nuevo comparado en una columna de la plantilla es similar, entonces necesitamos saber que tan parecidos son todos los tiempos de cada columna de la plantilla para así tener un punto de comparación con el tiempo nuevo.

La función nos indicará de manera numérica el grado de desviación de los tiempos de cada una de las muestras almacenadas en la planilla, la función que cumple con estos objetivos es la desviación estándar S.

La desviación estándar nos dice cuánto tienden a alejarse los puntos del promedio ver eq (1.4).

$$S = \sqrt{\frac{\sum (x - \bar{x})^2}{N}} \tag{1.4}$$

x: Denota cada uno de los tiempos de cada columna de la plantilla. $\bar{x}$: es la media de cada columna.
n : el número de muestras de tecleo que forman la plantilla.

Si sacamos la desviación estándar de cada columna, tendríamos un número que nos indicará cuanto fue el grado de desviación del usuario al momento de capturar las muestras para la plantilla en dicha tecla o intervalo de tecla específico. Una vez teniendo la desviación estándar de cada columna de la plantilla, como segundo paso necesitaríamos saber cuánto se desvió la nueva muestra, para esto tomaremos como referencia la media de cada columna de la plantilla ya que la media es la que toma la desviación estándar en el proceso anterior. Entonces en este paso sacaremos una nueva desviación estándar a la que llamaremos S' en base a la media de cada columna de la plantilla y el nuevo tiempo de la columna correspondiente. Si se tratase de un usuario autentico entonces suponemos que la mayoría de las columnas S sería mayor a S'.

Otra de las funciones estadísticas que utilizamos para el modelo es el coeficiente de variación este nos indica cual es la desviación de los puntos pero en términos de porcentajes como mencionamos anteriormente lo que nos interesa es un porcentaje que nos indique el grado de similitud entre la muestras nuevas de tiempo y las muestras de tiempo almacenadas en la platilla eq (1.5).

$$C.V. = \frac{S}{\bar{X}} * 100 \tag{1.5}$$

S: Es la desviación estándar de cada columna. $\bar{x}$:Es la media de cada columna.

Como la comparación es entre los tiempos de la plantilla y los nuevos tiempos entonces se aplicará la misma fórmula al igual que la desviación estándar, solo que los nuevos tiempos con relación a la media es decir, la media de la nueva muestra será la suma del nuevo tiempo más la media de la plantilla dividida entre dos a esta le llamaremos media',esto para cada columna.

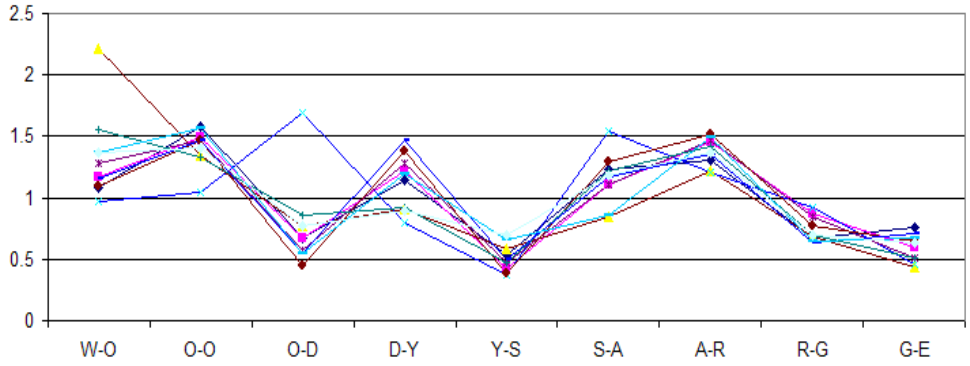
Tabla 1.5 Cálculo aplicado a la plantilla de la tabla 1.3

	W-O	O-O	O-D	D-Y	Y-S	S-A	A-R	R-G	G-E
M1	1.086	1.580	0.666	1.149	0.540	1.228	1.307	0.676	0.764
M2	1.166	1.488	0.676	1.213	0.423	1.107	1.458	0.868	0.598
M3	2.211	1.342	0.765	0.910	0.585	0.844	1.224	0.685	0.429
M4	0.965	1.043	1.696	0.790	0.374	1.540	1.210	0.915	0.463
M5	1.281	1.464	0.57	1.281	0.474	1.101	1.467	0.842	0.515
M6	1.091	1.462	0.444	1.379	0.387	1.293	1.519	0.775	0.646
M7	1.560	1.334	0.859	0.918	0.470	1.220	1.419	0.700	0.515
M8	1.154	1.461	0.544	1.469	0.496	1.174	1.358	0.635	0.705
M9	1.369	1.563	0.550	1.191	0.658	0.863	1.490	0.643	0.669
M10	1.364	1.411	0.766	0.889	0.696	1.178	1.372	0.691	0.629
S	0.356	0.153	0.354	0.229	0.109	0.201	0.108	0.999	0.109
Media	1.325	1.415	0.753	1.119	0.510	1.155	1.382	0.743	0.593
C. Var	26.8	10.84	46.97	20.54	21.39	17.43	7.83	13.44	18.44
Nueva	0.835	1.456	1.783	1.435	0.627	1.216	1.519	0.915	0.515
S'	0.346	0.029	0.727	0.223	0.082	0.043	0.096	0.121	0.055
Media'	1.080	1.436	1.268	1.277	0.569	1.186	1.451	0.829	0.554
C. Var'	32.07	2.05	57.38	17.48	14.47	3.65	6.65	14.63	0.554

Hasta aquí hemos calculado de manera porcentual que tanto están agrupados o desagrupados los puntos de la plantilla cuando el usuario teclea, de la misma manera el porcentaje que se desvió al autenticarse nuevamente en relación con la media de la plantilla. Entre más grande es el porcentaje, entonces mayor es la posibilidad de variación en esa columna en particular.

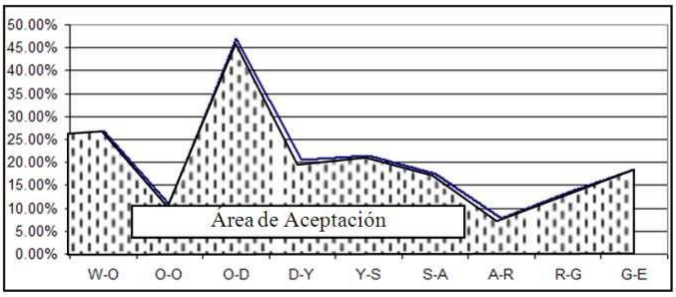
En la tabla anterior en la columna A-R es la que tiene menos porcentaje de variación esto quiere decir que este usuario cuando suelta la tecla A y presiona la R su tiempo es muy regular, sin embargo la columna O-D que es la de mayor porcentaje indica que no existe regularidad al soltar la O y presionar la D, que a veces lo hace rápido y a veces lo hace muy lento.

Grafico 1.2 Gráfica de los tiempos de la plantilla de la tabla 6 y su respectivo porcentaje de variación



En la gráfica 1.2 se puede observar el comportamiento de tecleo del usuario, donde las líneas indican cada una de las muestras y la curvatura el tiempo que tarda en realizar el evento de tecleo. Se observa que las líneas son iguales en su mayoría con excepciones en ciertos puntos los que denotaremos como puntos en el cual el usuario no está familiarizado o tiene complicaciones al pasar por esas teclas. Por lo general los coeficientes de variación que tienden a ser más grandes para está gráfica lo ocasiona uno o dos puntos fuera del rango, las columnas como Y-S, A-R, RG y GE estas tienen la característica de que sus puntos están agrupados en un rango pequeño de ahí que su porcentaje se encuentra entre los menores.

Grafico 1.3 Gráfica del coeficiente de variación de la plantilla de la tabla 6 esta muestra el límite de aceptación del usuario.



Hasta ahora, tenemos un modelo el cual nos indica los porcentajes de variación en los puntos de la plantilla, y el porcentaje de variación de los nuevos tiempos con respecto a la media de la plantilla, En la fig. 8 se muestra el área de aceptación para la plantilla que hemos venido manejando de ejemplo.

De acuerdo a la figura 8 para que un usuario sea autentico los nuevos tiempos deben estar por debajo de la línea límite, no como el de máximos y mínimos que eran dos líneas las limitantes.

Si graficáramos el coeficiente de variación de la nueva muestra podríamos darnos cuenta que como son del mismo usuario esta se encuentra por debajo del límite mostrado en la gráfica anterior con excepción de tres puntos que están por encima, es aquí donde surge la pregunta ¿éstos tres puntos que sobrepasan el límite son factor suficiente para rechazar al usuario? ó ¿Si los siete puntos que están por debajo del límite son suficiente para decir que el usuario es aceptado?, el criterio inicial de aceptación es el 60% dicho valor representa más de la mitad con respecto al 100% y fue tomado de manera arbitraria.

$$PS = \frac{\#Cv > Cv'}{\#\ Columnas} \qquad (1.6)$$

PS: Es el porcentaje de similitud de las nuevas muestras de tiempos comparadas con los tiempos de la plantilla del usuario.

#Cv>Cv': es el número de porcentajes en los que el coeficiente de variación (Cv) de la plantilla fue mayor al coeficiente de variación (Cv') de la nueva muestra de tiempo con respecto a la media de la plantilla.

#Columnas: son el número de columnas de la plantilla (para el caso de una plantilla del evento pulsar – soltar el numero de columnas es igual a la longitud de la frase tecleada y para el caso del eventos soltar pulsar es igual a la longitud de la frase tecleada menos uno.)

El cálculo de este porcentaje es el resultado final de este modelo en el cual nos dirá cual es el porcentaje de similitud de las muestras de este porcentaje, y de la manera de interpretarlo o compararlo depende la aceptación o rechazo del usuario como mencionamos anteriormente primero tomamos como condición que el porcentaje fuera mayor que 60% para aceptar al usuario, ya que en base a observaciones y pruebas del modelo nos dimos cuenta que el porcentaje de similitude los usuarios impostores estaba por debajo del50%, sin embargo, algunos usuarios auténticos no alcanzaban el 60%.

Entonces, se optó porque la comparación de este porcentaje fuera de manera dinámica y dependiera directamente del comportamiento del usuario en el momento de crear su plantilla, lo que se tomó como referencia fue el promedio del coeficiente de variación de cada una de las columnas de la plantilla.

Si el coeficiente de variación es grande en promedio quiere decir, que las líneas de la plantilla no tienen un grado de similitud considerable por lo tanto cuando el usuario se autentique tendría un rango grande en el cual podría caer sus tiempos, entonces el porcentaje de aceptación se puede aumentar para así evitar el error de falsa aceptación.

Explicado todo lo anterior, entonces tenemos que el factor que nos indicará el porcentaje de aceptación es el promedio del coeficiente de variación, pero este promedio siempre está por debajo del 50% (tomando como base el estudio de muestreo), por lo tanto al promedio le sumaremos 50% y así cubriremos el 100% que se calcula en el promedio de similitud. Si calculamos el porcentaje de aceptación para el usuario que hemos venido utilizando como ejemplo, tendríamos un porcentaje de aceptación de 70.42% es decir para que este sea aceptado debe cumplir un 70.42% de porcentaje de similitud en relación a la media de la plantilla.

1.4 Mecanismo de adaptación

El mecanismo de aceptación que implementamos en este trabajo es muy sencillo y consiste únicamente en calcular para cada una de las muestras de la plantilla el coeficiente de variación con base a la media, como si se tratase de una nueva muestra, y sacar la de mayor coeficiente de variación reemplazándola con la nueva muestra de autenticación siempre y cuando esta tenga menor coeficiente de variación, en caso contrario la plantilla quedaría igual.

1.5 Pruebas

Las pruebas se realizaron sobre tres grupos de persona que se describen a continuación:

Grupo "Estudio Muestral":

Este grupo de diez personas fue resultado de un muestreo estratificado sobre trabajadores de la UJAT, oscilaban entre edades de 24 a 35 años, ocupaciones como programadores, contadores, secretarias y administradores.

Grupo "Universitarios":

Este grupo está constituido por doscientos estudiantes de la Universidad Juárez Autónoma de Tabasco de las carreras de Contaduría, Educación y Ciencias de la Comunicación.

Grupo "Varios":

Este grupo involucra veinte personas, de diferentes edades, ocupaciones y habilidades sobre el teclado.

Cada una de las personas con las que se probó la aplicación biométrica estableció dos frases; la primera como login y la segunda como contraseña, para la primera frase se recomendó se utilizará su nombre con apellidos, ya que suponemos que esta frase es fácil de escribir porque es lógico pensar que las personas estén familiarizados con su nombre. Para la extracción de las características de tecleo de las personas, teclearon diez veces la frase login y posteriormente diez veces la frase contraseña.

Los objetivos de las pruebas fueron establecer los errores de falsa aceptación (EFA) y falso rechazo (EFR) [10] en la autenticación local así como en red.

Tabla 1.6 Porcentajes de falsa aceptación y falso rechazo obtenidos en las pruebas

Grupo de Prueba	Tipo	Intentos	EFA.	EFR
Est. Muestral	Local	100	0%	5%
	Red	100	0%	19%
Universitarios	Local	2000	0%	26%
	Red	6000	0%	36%
Varios	Local	200	0%	25%

1.6 Análisis de resultados

De acuerdo a los experimentos realizados en los tres grupos de pruebas y a las observaciones realizadas se pudo determinar lo siguiente:

La frase con la que los usuarios se encuentran más familiarizados es su propio nombre y sus apellidos.

La captura de la plantilla de cada uno de los usuarios para su autenticación es la parte principal, por lo cual esta debe capturarse no intentando escribir más rápido de lo normal y no realizar pausas innecesarias entre teclas.

Si un usuario elegía una frase no familiarizada con él, se podía observar una disminución en la velocidad de tecleo, sin embargo, aún era posible extraer características únicas de tecleo. Cuando el usuario elegía una frase menor de diez caracteres, se incrementaba el error de falso rechazo de manera considerable.

La longitud adecuada de una frase, para el modelo planteado en este trabajo y con las cuales se disminuyó el error de falso rechazo fue entre quince y treinta caracteres.

El manejo de los tiempos de los dos eventos *pulsar – soltar* y *soltar – pulsar* en conjunto para la autenticación es recomendable cuando la plantilla del usuario fue creada en la misma computadora donde se está autenticando.

Para el caso de que un usuario se autentifique en otra computadora diferente a la donde creó su plantilla, es recomendable no incluir en el proceso de comparación los tiempos del evento *pulsar – soltar*, ya que éstos tienen una variación considerable que depende de la suavidad de las teclas. Sin embargo el evento *soltar – pulsar* es suficiente para realizar la comparación y obtener buenos resultados.

El promedio de porcentaje de similitud obtenido por usuarios impostores es de 35%, de ahí que el porcentaje general de falsa aceptación fue de 0% ya que por lo menos se necesitaría alcanzar un 60% de porcentaje de similitud para ser aceptado por el sistema.

El 35% de promedio de porcentaje de similitud para usuarios impostores que obtuvimos, nos da una holgura para poder bajar el porcentaje de aceptación base que fue de 60% a 40% lo que nos disminuiría el porcentaje de falso rechazo obtenido en las pruebas, teniendo la seguridad de que el error de falsa aceptación no subiría.

En general se obtuvo un error de falsa aceptación de 0%, lo que constituye la fortaleza de este método, el no aceptar a un usuario que no es el autentico, ya que el rechazar el correcto tiene como consecuencia únicamente que el usuario tenga que intentar autentificarse nuevamente.

1.7 Conclusiones

En este trabajo se planteó un modelo para la autenticación de usuarios a través de la biometría de tecleo, en trabajos anteriores se plantea la extracción de características de tecleo a través de la medición de los eventos pulsar – soltar y soltar – pulsar tecla, mismos que fueron utilizados en este trabajo, se utilizaron dos frases entre quince y treinta caracteres que actuarían como login y contraseña de los usuarios.

El modelo para la comparación de la dinámica de tecleo se basó en el uso de funciones de dispersión, se establecieron dos parámetros para la decisión de aceptación del usuario que son: el porcentaje de similitud y el porcentaje de aceptación, teniendo como condición final que el porcentaje de similitud fuera mayor o igual al porcentaje de aceptación.

Las pruebas nos dieron resultados satisfactorios en cuanto a la obtención de una tasa de error de 0% para el error de falsa aceptación y un 35% en promedio para el error de falso rechazo, éste porcentaje del 35% se puede disminuir considerablemente si se baja el porcentaje de aceptación, el cual en una implementación sería configurable.

Finalmente, esta técnica representa una tecnología de autentificación de bajo costo, ya que no requiere hardware adicional, actuando el teclado tradicional como dispositivo biométrico.

1.8 Referencias

A.M Montiel, F. Rius y F. J. Baron (2004), "Elementos básicos de estadística económica y empresarial", 3ª edición Prentice Hall.

Acevedo Daniel, Glemarys Hernández y Eugenio G. Scalise P (2000). "Identificación de Usuarios Basado en el Reconocimiento de Patrones de Tecleo" Universidad Central de Venezuela, Facultad de Ciencias.

Araújo Lizárraga, Sucupira Jr., Yabu-uti y Ling.(2008) "Autenticación personal por dinámica de tecleo basada en lógica difusa" Universidad Estatal de Campinas (UNICAMP).

D. Umphress and G. Williams (1995), "Identity Verification Through keyboard Characteristics" International Journal Man-Machine Studies, Academic Press.

Davies (2002) "Security for Computer Networks: An Introduction to Data Security" John Wiley and Sons, New York, 2002

Enzhe Yu, Sungzoon Cho (2004) "Keystroke dynamics identity verification problems and practical solutions" Department of Industrial Engineering, College of Engineering, Seoul National University.

Leonard Kazmier, Alfredo Diaz Mata (2002) , "Estadística aplicada a la administración y economía", 2ª edición Mc Graw Hill.

Marino Tapiador Mateo. (2010) "Biometría de tecleo, autenticación de usuarios" Ingeniería Informática, Universidad Autónoma de Madrid.

Obaidat (2002) M. S. "Keystroke dynamics based Authentication" Monmouth University Applied Science University.