Ciencias Matemáticas, Vol. 33, No. 1, Pag. 66-70, 2019

Debilidades de los métodos de discretización para contraseñas gráficas
Weakness of the discretization methods for graphic passwords

Ernesto A. Borrego Rodríguez[1]*, Pedro E. Navarro Sosa[1], Carlos M. Legón Pérez[1]

Resumen La autenticación gráfica surge como una alternativa a problemas presentados por las contraseñas alfanuméricas, pero los métodos de discretización de imágenes existentes hasta ahora aún presentan problemas de seguridad y usabilidad para los usuarios. En este trabajo se analizan y comparan tres métodos de discretización de imágenes: Discretización Robusta, Discretización Centrada y Discretización Optima; y se muestran sus principales diferencias y debilidades criptográficas mediante sus regiones de tolerancia y r-seguridad.

Abstract Graphic authentication emerges as an alternative to problems presented by alphanumeric passwords, but the existing methods of images discretization still present security and usability problems for users. In this work, three methods of images discretization are analyzed and compared: Robust Discretization, Centered Discretization and Optimal Discretization; and its main differences and cryptographic weaknesses are shown through its regions of tolerance and r-security.

Palabras Clave
Autenticación gráfica — Discretización — Región de tolerancia — r-seguridad

[1] *Instituto de Criptografía, Universidad de La Habana, La Habana, Cuba, ernesto.borrego@matcom.uh.cu, pedropepe3437@gmail.com, clegon58@gmail.com*
***Autor para Correspondencia**

Introduction

Los métodos de autenticación gráfica como Pure Recall, Recognition y Cued Recall Based Techniques surgen como una alternativa para solucionar problemas presentados por las contraseñas alfanuméricas. [1] El presente trabajo se enfoca en el estudio de tres de los principales métodos de discretización de imágenes utilizados en sistemas de autenticación del tipo Cued Recall [1], [2], [3]. Se comparan dichas discretizaciones en cuanto al tamaño de región de tolerancia que utilizan, el radio de seguridad que ofrecen, y los problemas de seguridad que pueden presentar.

Primeramente se verá cómo es el funcionamiento de las tres discretizaciónes dentro de los sistemas de autenticación [1], [2], [3]. A continuación se explican las deficiencias que presentan cada una de las discretizaciones.

1. Discretización

Todo tipo de autenticación gráfica del tipo Cued-Recall [4] se basa invariablemente en seleccionar al menos un punto de una imagen. Específicamente en el PassPoints y en el Cued Click Point [4] esto plantea una serie de requerimientos y problemas a tener en cuenta a la hora de crear un sistema de autenticación.

Para empezar, se necesita que un usuario seleccione un punto como su contraseña en la fase de registro para que, posteriormente lo reseleccione en la fase de autenticación. Como es de esperar, pedirle al usuario que vuelva a seleccionar exactamente el mismo punto dos veces volveria el proceso demasiado complejo, hay que tener en cuenta que requerir el mismo punto para la autenticación es pedir exactamente el mismo píxel dentro de una imagen de $m \times n$ pixeles. Lo que hace necesario definir un margen de error alrededor del punto original de la contraseña dentro del cual se pude seleccionar un punto y sea aceptado como válido.

Definición 1 (Región de Tolerancia) *Se llama Región de Tolerancia al conjunto de puntos de la imagen que son aceptados como válidos en la fase de autenticación para el punto original.*

Sea RT el conjunto de puntos de la región de tolerancia, I el conjunto de puntos de la imagen, y f una función indicadora que para todo punto de la imagen devuelve 1 *si el punto es aceptado como válido y* 0 *si no, entonces:*

$RT \subset I$ *tal que* $\forall p \in RT,\ f(p) = 1$

Esta región de tolerancia es la que determinará en todos los sistemas de autenticación cuales son los puntos alrededor de la contraseña aceptados por el sistema. Estrechamente relacionada tenemos la siguiente definición:

Definición 2 (Punto r-seguro) *Un punto p_0 se dice r-seguro en una RT para un radio r dado, si y solo si:*

$\forall p \in I$, *tal que* $\|p - p_0\| < r$ *entonces* $p \in RT$,

donde p_0 es el punto original de la contraseña. Al parámetro r se le denomina radio de seguridad.

Esta definición, plantea una propiedad indispensable a cumplir para todo región de tolerancia:

Propiedad 1 *Todo punto p_0 de la contraseña debe ser r-seguro en su región de tolerancia.*

Para lograr obtener que cada punto de la contraseña pertenezca a una región de tolerancia donde sea r-seguro en la cual un usuario legítimo pueda autenticarse, surge la discretización de la imagen. Una primera idea muy intuitiva sería particionar la imagen en cuadrículas, lo que plantea el segundo problema a tener en cuenta a la hora de discretizar una imagen para autenticar: el problema del vértice. [1]

El problema del vértice plantea que para una partición como la anterior, existe la posibilidad de que se seleccione un punto que se encuentre en uno de los vértices o aristas de la partición, o relativamente cerca($< r$). El primer caso plantea un problema de decisión para seleccionar cual es la región de tolerancia que le corresponde al punto, mientras que para ambos casos no se cumple la propiedad 1 de la región de tolerancia.

Una de las principales razones por la cual es necesario la discretización se basa en la seguridad. Si bien se necesita aceptar puntos relativamente cerca del punto original de la contraseña, el sistema no puede guardar en texto plano los puntos seleccionados. Para ello, una vez escogidos los puntos de la contraseña por el usuario, el sistema calcula su valor *hash* y lo guarda. Esto supone otro problema a tener en cuenta, ya que, para puntos cercanos en la imagen, sus valores *hash* son bastante diferentes, por lo que dado los valores *hash* de dos puntos, no es posible determinar la distancia entre ellos, por tanto, tampoco se puede asegurar que se encuentren a una distancia menor que r. Para resolver esto, es necesario cifrar todo la región de tolerancia una vez escogida y no exactamente el punto original de la contraseña.

En resumen, es necesario discretizar la imagen de forma tal que, para todo punto de la imagen exista una región de tolerancia a la cual pueda pertenecer en la que sea r-seguro y donde en el sistema de autenticación se cifre toda su región de tolerancia.

1.1 Discretización Robusta

Para evitar el problema del vértice, en [1] se propone un método llamado *Robust Discretization*(Discretización Robusta) que utiliza un conjunto de tres particiones diferentes de la imagen: G_0, G_1, G_2; para garantizar que todo punto sea *r-seguro* en al menos una de las tres particiones. Es bastante intuitivo que para una imagen bidimensional, 3 particiones son necesarias y suficientes para asegurar una separación de al menos r pixeles entre el punto y al menos una de las particiones.

En la figura 1 se observan tres particiones diferentes G_0, G_1, G_2, dos puntos A y B escogidos en la fase de registro y dos circunferencias de radio r centradas en dichos puntos. En este caso ocurre que el punto A está a una distancia menor que r de las particiones G_1 y G_2, por lo que en este caso para el punto A se escoge la partición G_0, mientras que B se encuentra demasiado cerca de G_0, pero es *r-seguro* en G_1 y G_2, por lo que para el punto B se pueden escoger cualquiera de estas dos particiones. Una vez escogida, se selecciona como RT el cuadrante de dicha partición al que pertenece el punto.

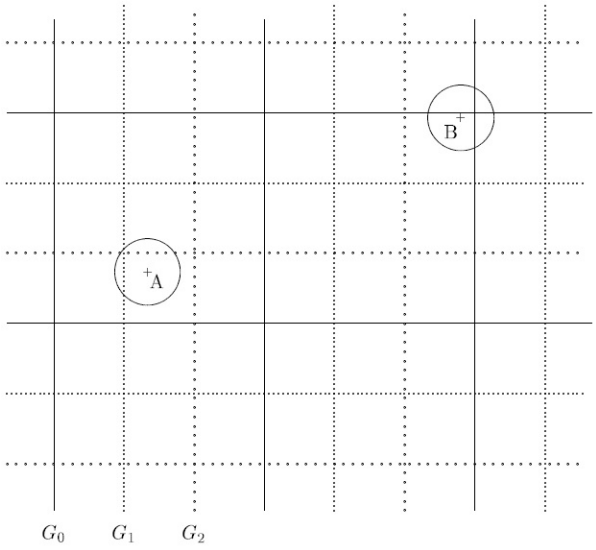

Figura 1. Particiones

En la *Discretización Robusta* para lograr la r-seguridad del punto, las cuadrículas de las particiones son de $6r$x$6r$(tolerancia) y cada partición esta a una distancia de $2r$ de las otras. Por construcción, en la fase de autenticación, un punto a una distancia menor o igual a r del punto original pertenecerá al mismo cuadrante, lo que asegura la validez del usuario ya que la imagen de la función *hash* será la misma. Por otro lado, cualquier punto a una distancia mayor a $5\sqrt{2}r$ pertenecera a otro cuadrante, por lo que se garantiza la no autenticación.

La figura 2 muestra un cuadrante de una de las tres particiones y una linea de puntos azules que delimita la región exacta de los puntos dentro del cuadrante que de ser seleccionados como contraseña se seleccionaría dicho cuadrante. Se debe notar que en el peor de los casos de la linea de puntos azules (esquina superior derecha o esquina inferior izquierda) el punto seleccionado permanece r-seguro dentro del cuadrante, y además, cualquier punto fuera de la linea azul devolvería otro cuadrante a seleccionar.

Pero esta forma de discretizar implica una tolerancia demasiado grande con respecto al círculo definido por la r-seguridad del punto. Lo que conlleva a un área de aceptación completamente por fuera del círculo.

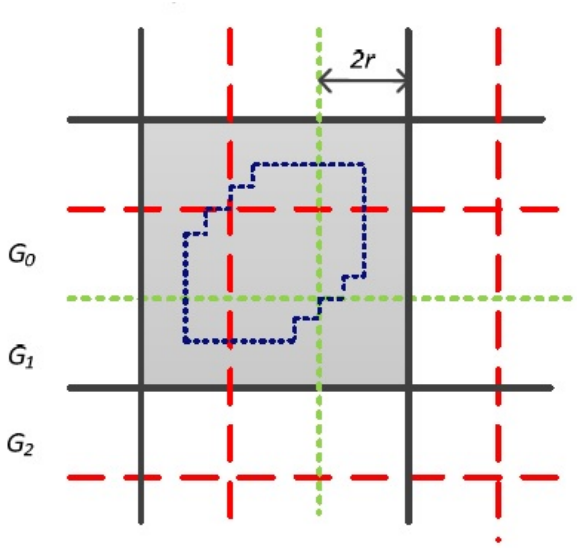

Figura 2. Región de tolerancia

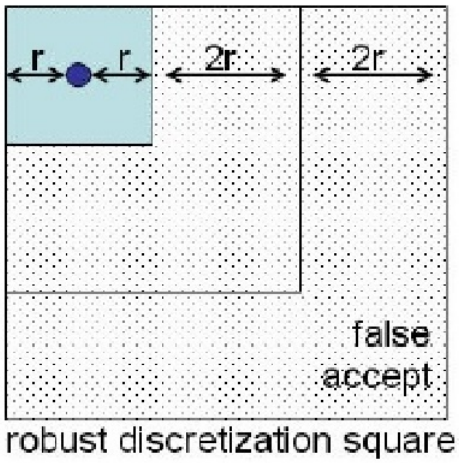

Figura 3. Falsos aceptados

En la figura 3 se muestra que a la izquierda del punto original, la tolerancia solo acepta puntos en una distancia menor que r, mientras que a la derecha se aceptan hasta una distancia de $5r$.

Por esta razón surgieron otros métodos de discretización que resuelven el problema de los falsos aceptados centrando la región de tolerancia en el punto original. [2]

1.2 Discretización Centrada

Para resolver los problemas presentes en la Discretización Robusta, en [2] proponen la Discretización Centrada que ofrece considerables mejoras de seguridad y usabilidad.

Se analizará primero el caso unidimensional de dicha discretización, luego solo se extenderá el razonamiento a dos dimensiones para aplicarlo al caso de imágenes.

Sea un punto x sobre la semirrecta numérica $[0,\infty)$, en realidad $x \in [0,m]$ donde m es el tamaño de la imagen(ancho o largo). El objetivo es que la región de tolerancia este centrada en el punto x, por lo que se construye el segmento de longitud $2r$ centrado en el punto y a partir de este segmento se particiona el intervalo $[0,m]$ en segmentos de igual longitud. Es evidente que en la mayoria de los casos existirá un sobrante de longitud $d \in [0,2r)$ al inicio del intervalo $[0,m]$ por lo que si se guarda esta longitud d, es posible construir la partición en segmentos de longitud $2r$ comenzando en d donde uno de los segmentos tendrá por centro al punto x.

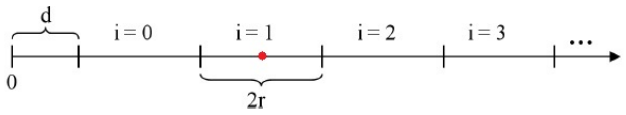

Figura 4. Discretización centrada unidimensional

Una vez fijado r y establecido el punto x en la fase de registro, se puede calcular la distancia sobrante inicial de la siguiente forma $d = (x-r) \bmod 2r$, mientras que si se enumeran todos los intervalos de la partición, el intervalo exacto donde se encuentra x se puede hallar: $i = \lfloor \frac{x-r}{2r} \rfloor$.

Una vez seleccionado un punto en la fase de autenticación, se calcula el valor del intervalo $i' = \lfloor \frac{x'-d}{2r} \rfloor$. Notese que x' no está necesariamente centrado en el intervalo i', pero si se cumple que $|x-x'| < r$ entonces $i = i'$, y por lo tanto sus valores *hash* tambien lo serán.

Por ejemplo, sean $x = 13$ y $r = 5,5$, entonces:

$$i = \lfloor \frac{x-r}{2r} \rfloor = \lfloor \frac{13-5,5}{11} \rfloor = 0$$

Por otro lado $d = (x-r) \bmod 2r = 7,5$.

Sean ahora $x' = 10$, entonces:

$$i' = \lfloor \frac{x'-d}{2r} \rfloor = \lfloor \frac{10-7,5}{11} \rfloor = 0 = i$$

Para realizar la extensión a dos dimensiones solo hay que considerar para cada punto de la contraseña el par (d_x, d_y), el cual se guarda en texto claro en el sistema, y el valor *hash* del vector de los i correspondientes a cada punto. [2]

Este método soluciona los problemas de la Discretización Robusta presentados en el epígrafe anterior, ya que centra la región de tolerancia en el punto de la clave.

1.3 Discretización Optimal

Otro método llamado *Optimal Discretization* es descrito en [3]. Este método es bastante similar a la Discretización Centrada en cuanto a que logra mantener la filosofía de una región de tolerancia centrada en el punto original de la clave, pero utiliza propiedades de la aritmética modular para construirla. Una vez más se empezará describiendo la idea en una dimensión para mayor comprensión, ya que es facilmente extendible.

Sea el valor X el punto original escogido por el usuario en la fase de registro como su contraseña, x el valor dado al sistema para la autenticación, y r el radio de tolerancia. Se calcula un valor φ de la siguiente forma:

- si $(X \bmod 2r \geq r)$, entonces $\varphi = X \bmod r$
- $(X \bmod 2r < r)$, entonces $\varphi = (X \bmod 2r) - r$

El valor φ se guarda en texto claro en el sistema junto con el valor *hash* de $S_X = \left\lfloor \frac{X-\varphi}{2r} \right\rfloor$. Una vez el sistema obtiene el valor x seleccionado, calcula el valor *hash* de $S_x = \left\lfloor \frac{x-\varphi}{2r} \right\rfloor$ y lo compara con el ya guardado. Notese que:

$$S_X = S_x \iff x - r \leq X < x + r$$

Por ejemplo, sean $X = 38$ y $r = 5$, entonces:

$$\varphi = X \bmod r = 38 \bmod 5 = 3$$

Por otro lado $S_X = \left\lfloor \frac{X-\varphi}{2r} \right\rfloor = \left\lfloor \frac{38-3}{10} \right\rfloor = 3.$
Sea ahora $x = 40$, entonces:

$$S_x = \left\lfloor \frac{x-\varphi}{2r} \right\rfloor = \left\lfloor \frac{40-3}{10} \right\rfloor = 3 = S_X$$

Otro caso, $X = 83$ y $r = 5$, entonces $\varphi = -2$, por otro lado $S_X = \left\lfloor \frac{X-\varphi}{2r} \right\rfloor = 8$. Sea ahora $x = 80$, entonces:

$$S_x = \left\lfloor \frac{x-\varphi}{2r} \right\rfloor = \left\lfloor \frac{80+2}{10} \right\rfloor = 8 = S_X$$

Extendiendo a dos dimensiones, en el sistema se guardan en texto claro los pares (φ_x, φ_y) para cada punto, y el valor *hash* del vector de pares (S_x, S_y), donde (x, y) son las coordenadas del punto selecionado por el usuario.

2. Ventajas y desventajas

Estas tres discretizaciones cumplen con el objetivo inicial planteado anteriormente para discretizar una imagen, pero difieren entre si en espectos escenciales como el tamaño de la región de tolerancia, el radio de seguridad que ofrecen y la información revelada por el sistema. En el epígrafe 1.1 se mostró como en la Discretización Robusta es necesaria una tolerancia relativamente grande con respecto al circulo de r-seguridad del punto. En [2] se analiza como para tamaños de tolerancia normales en la Discretización Robusta, el radio de r-seguridad del punto es pequeño. (Cuadro 1)

RT	9×9	13×13	19×19
r (en pixeles)	1,50	2,17	3,17

Cuadro 1. Radio de seguridad fijando RT

Por otro lado, para radios de r-seguridad aceptables se requieren grandes regiones de tolernacia. (Cuadro 2)

r (en pixeles)	4	6	9
RT	24×24	36×36	54×54

Cuadro 2. RT fijando el radio de seguridad

Sin embargo, la Discretización Centrada y la Optima, al centrar la Región de Tolerancia alrededor del punto, no presentan problemas en la relación entre esta y el radio; pero, en ambos casos, como en la Discretización Robusta, se aceptan y rechazan puntos que se encuentran a la misma distancia del punto original de la contraseña. Esto se debe a que en los tres casos la Región de Tolerancia tiene forma cuadrada(figura 5), mientras que para evitar puntos aceptados y rechazdos con distancias equivalentes, es necesaria una tolerancia circular. [2], [5]

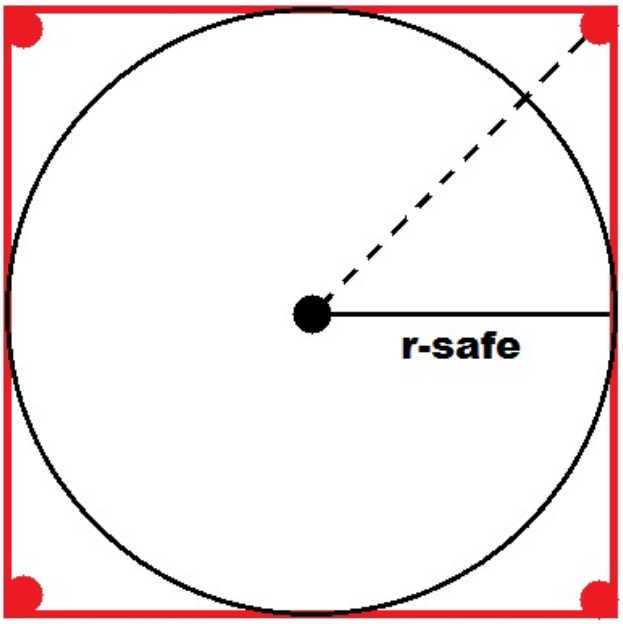

Figura 5. Región de tolerancia cuadrada y circular

Ademas, todas las discretizaciones antes vistas, guardan algún tipo de información en el sistema en texto claro, lo que puede ser aprovechado por un atacante para reducir el espacio de búsqueda y realizar ataques de diccionario. [6]

En la Discretización Robusta, se deja en texto claro cuales son las tres particiones que utiliza, incluso revela cuál es la partición exacta que le corresponde a cada punto. En la Centrada y en la Optima, lo guardado en texto claro es la distancia sobrante inicial d y el valor φ respectivamente, con los cuales se puede construir también la partición utilizada.

Aunque las dos últimas muestran menos información a un atacante, en todas es posible construir diccionarios utilizando esos datos e intersectándolos con los puntos que son más probables que el usuario utilice(*hotspot*), los cuales se calculan por técnicas de tratamiento de imágenes digitales como segmentación; para ello solo es necesario realizarle un trabajo previo de procesamiento de imágenes a la imagen utilizada

para autenticarse.

En los cuadros 3 y 4 se pueden apreciar las principales diferencias entre estas discretizaciones. Se muestra que la Discretización Robusta presenta serios problemas de usabilidad por la relación entre el radio de seguridad y el tamaño de la Región de Tolerancia. Mientras que en todas es revelada información aprovechable para comprometer su seguridad.

Tipo de discretización	**RT** $= 13$x13	**r** $= 6,5$
Robusta	$r = 2,17$	$RT = 39$x39
Centrada	$r = 6,5$	$RT = 13$x13
Optima	$r = 6,5$	$RT = 13$x13

Cuadro 3. Comparación de RT y r

Tipo de discretización	**Información en texto claro**
Robusta	Las tres particiones G_0, G_1, G_2
Centrada	La distancia sobrante inicial d
Optima	El valor φ

Cuadro 4. Comparación de RT y r

Conclusiones

Vistas las características principales de las discretizaciones Robusta, Optima y Centrada se puede concluir que las tres formas de discretización presentan probemas de usabilidad en la fase de registro debido a pixeles equidistantes que se encuentran tanto dentro como fuera de la región de tolerancia, mientras son aceptados otros mas lejanos.

Además, la investigación muestra una necesidad de crear un tipo de discretización con una región de tolerancia circular para resolver estos problemas de usabilidad, ya que el círculo es lo que se ajusta a la idea de distancia utilizada para aceptar o rechazar los puntos seleccionados.

Por otro lado, también es necesario crear una discretización que no revele ningun tipo de información en texto claro para dificultarle la construcción de diccionarios a un posible atacante y asi corregir los problemas de seguridad que presentan las tres discretizaciones analizadas en este trabajo.

Referencias

[1] Hong D. Memon N. Birget, J.C. Graphical passwords based on robust discretization. *IEEE Transactions on Information Forensics and Security*, 2006.

[2] Srinivasan J. Biddle R. Oorschot P. C. Chiasson, S. Centered discretization with application to graphical passwords. 2008.

[3] K. Bicakci. Optimal discretization for high-entropy graphical passwords. Technical report, TOBB University of Economics and Technology, Ankara, Turkey, 2007.

[4] Legón C. Socorro R. Navarro P. Rodriguez, O. Esquemas y técnicas de autenticación gráfica. 2018.

[5] P. Karmajit. Cued-click point graphical password using circular tolerance to increase password space and persuasive features. *Procedia Computer Science*, 2016.

[6] Bin B. Zhu. Security implications of password discretization for click-based graphical passwords. 2013.