Authentication Based on the Image Encryption using Delaunay Triangulation and Catalan Objects

Faruk Selimović[1], Predrag Stanimirović[1], Muzafer Saračević[2], Aybeyan Selimi[3], Predrag Krtolica[1]

[1]Faculty of Science and Mathematics, University of Nis, Višegradska 33, 18106 Niš, Serbia, faruk.selimovic@pmf.edu.rs, pecko@pmf.ni.ac.rs, krca@pmf.ni.ac.rs

[2]Department of Computer Sciences, University of Novi Pazar, Dimitrija Tucovića bb, 36300 Novi Pazar, Serbia, muzafers@uninp.edu.rs

[3]Faculty of informatics, International Vision University, Major C. Filiposki 1, 1230 Gostivar, North Macedonia, aybeyan@vizyon.edu.mk

Abstract: This paper presents the authentication method using the Delaunay triangulation incremental algorithm and the Catalan objects. The proposed method is a combination of computational geometry and cryptography. This method presents a new step towards encoding the triangle coordinates using the Catalan-key. We provided specific suggestions for the application of this method in the authentication for bank clients by the image encryption. Client authentication verification is performed by asking the client to enter the (x,y) coordinate values of randomly selected indices of an array. If the entered coordinates match the index values in the banking system array, then the transaction or other operation is approved. If the matching fails, it means that we have an unidentified person who has followed the whole process and wants to break into the banking system. There are many advantages arising from a scenario for the user authentication by the assigned Catalan object and the stack permutation method. Also, we provided concrete examples for the Delaunay encryption of image with an authentication scenario and experimental results for the proposed method.

Keywords: Authentication; Cryptography; Delaunay triangulation; Catalan objects; Image encryption

1 Introduction

The modern electronic age and the devices that accompany it, bring many opportunities. One of their main capabilities is to store data and protect them from unauthorized access. The science behind data protection methods is called cryptography. Along with cryptography, cryptanalysis is also being developed and

always strives to find out the secret message received by one of the cryptographic methods. At its core, cryptography is based on mathematical models of algorithms.

In these days of the modern technology, there is a growing need to create secure, i.e., reliable user authentication systems. Special emphasis is placed on banking transactions and systems that are often the target of hacker attacks. This paper will deal with Delaunay triangulation and image encryption using the Catalan object as the basic method of authentication. A scenario that includes a potential attacker will be presented. Implementation of operations of such models of algorithms over valuable (publicly available) information is called encryption. Through this process, we obtain modified information that is not understandable. The reverse process, when the ciphered text again receives intelligible information, is called decryption. One of the indispensable input parameters in the chosen encryption algorithm is a Cryptographic key. It is a binary string of 0 and 1 whose length depends on the cryptographic algorithm used.

The main contribution of this paper is a novel encryption method, stated using the Delaunay Triangulation incremental algorithm and the Catalan objects. The proposed method is a combination of computational geometry and cryptography. This method presents a new step towards encoding the triangle coordinates using the Catalan-key. We provided specific suggestions for the application of this method in the authentication for bank clients by the image encryption. There are many advantages arising from a scenario for the user authentication by the assigned Catalan object and the stack permutation method.

The rest of the paper is organized as follows. Similar research from the field of computational geometry application in cryptography is discussed and surveyed in the second section. A special focus is given to some applications of the combinatorial problems based on the Catalan objects (such as Lattice Path combinatorics, Stack permutations, Balanced Parentheses, and Ballot problem) in the file encryption and decryption. The third section discusses the basic properties of the Voronoi - Delaunay triangulation of the image. Examples for the encryption by Delaunay triangulation of image and authentication scenario are given in the fourth section. The fifth section contains experimental results and a detailed analysis of the encryption method. Concluding remarks and suggestions for further research are given in the sixth section.

2 Review of Related Research

Authenticating through the means of encrypted images is not a new idea. In [1], Luan Guangyu proposed a new encryption scheme and authentication of asymmetric images based on equal decay modules in the Fresnel transform domain. The benefits of this scheme are multiple; first, the open-spectrum Fresnel

is rarely sampled. Then, the rare presentation of the Fresnel spectrum is divided into two complexly valuable masks with the same decay modulus, both of which are required for decryption and authentication. Lin Yuan in [2] developed an authentication scheme through image, based on the encryption of double images and partial phase decryption in the fractional Fourier transform domain. Only part of the information of the encrypted result phase is stored for the decryption, while the rest of the phase and all other amplitude information is discarded. In [3], Huaqian Yang proposed fast encryption of image authentication. In particular, a key hash function was introduced to generate a 128-bit hash value from an ordinary image and secret keys. The hash value plays a role of encryption and decryption keys, while the secret hash keys are used to authenticate decrypted images.

When it comes to the Delaunay triangulation application, there are many ideas of authentication by fingerprint image triangulation. In this regard, in the paper [4], Zanobya N. Khan developed the idea of separation of palm lines. Then, the endpoints of these lines are determined and a link between them is created using Delaunay triangulation to generate a distinct topological structure for each palm imprint. After that, different geometric and quantitative characteristics are distinguished from the triangles of Delaunay triangulation that aids the identification of different individuals. In [5], a Delaunay triangulation technique for fingerprint-based image was developed on the grounds of matching to avoid authentication mistake, caused by incorrect entries and OTP (One-Time Password) submissions to ensure maximum security in authentication verification. This triangulation method allows unobstructed access to authorized users at the ATMs even with modified fingerprints and also improves security.

We will state some applications of the combinatorial problems based on the Catalan objects (such as Lattice Path combinatorics, Stack permutations, Balanced Parentheses, and Ballot problem) in the file encryption and decryption. The paper [6] analyzes the properties of the Catalan numbers and their relation to the Lattice Path combinatorial problem in cryptography, i.e., in the files and plain text encryption and decryption. A procedure for the application of one computational geometry algorithm in the process of generating hidden cryptographic keys from the 3D image segment was presented in [7]. In this paper, a combination of polygon triangulations, Catalan numbers and cryptography is made. The paper [8] examines the possibilities of applying appropriate combinatorial problems (Ballot Problem, Stack permutations, and Balanced Parentheses) in the files and plain text encryption and decryption. Catalan numbers play an important role in data hiding and steganography. The purpose of paper [9] is related to investigating the properties of the Catalan numbers and their possible application in the procedure of data hiding in a text, more specifically in the area of steganography. The authors of [10] provided some straightforward information, such as how much spurious and missing minutiae can influence on a Delaunay triangulation. Their research is supported by the results of experiments carried out on two common

variants of the Delaunay triangulation in four different cases. The experimental results show that Delaunay triangulation based structures are more sensitive to missing minutiae than spurious minutiae. In the paper [11] authors proposed a 3D Delaunay triangulation based fingerprint authentication system as an improvement to the authentication performance without adding extra sensor data. From the experimental results, it is observed that the 3D Delaunay triangulation based fingerprint authentication system outperforms the 2D based system in terms of matching performance by using the same feature representation.

3 Voronoi - Delaunay Triangulation of Images

The authentication can be based on the Delaunay triangulation of a selected image and coloring of triangles through the process of triangulation. Thus triangulated and colored image should be triangulated again in such a manner that the coordinates of the triangle vertices are coded by the Catalan key.

Definition 3.1. *By Catalan key we assume a Catalan object having balanced parenthesis property, i.e., it is a bit string with exactly n* 1 *bits and n* 0 *bits where, in every prefix, the number of* 1*'s is greater or equal to the number of* 0*'s.*

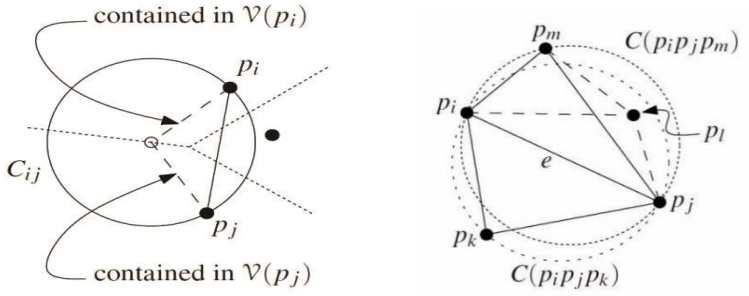
We will state some properties of the Delaunay triangulation: uniqueness and independence from the starting point, the formed triangles are in the shape of equilateral triangles, there is no other point in the circle of triangles (circle property), the convex hull is triangulated, the segment obtained from the closest pair of points is in triangulation, the segment derived from the point and its closest points is the side of the triangle in the triangulation.

Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of points in the plane. These points are called centers or sites. Delaunay triangulation of set P is a unique triangulation where no triangle circumscribed circle contains sites in its interior. On the other hand, the dual of Delaunay triangulation is Voronoi diagram. Voronoi diagram for some set of sites also partitions a plane into the regions, where every region consists of all points closer to a site p_i than to any other site.

Now, let us explain the term of the edges of a Delaunay diagram. For the edge $\overline{p_i\, p_j}$ we say it belongs to the Delaunay diagram if there is a circle C_{ij} to which p_i and p_j belong and no other site lies inside the circle, and the center of the circle C_{ij} lies on the edge of the Voronoi diagram defined with $V(p_i)$ and $V(p_j)$. The three points p_i, p_j ,p_k are vertices of the Delaunay diagram of the set P if and only if the circle through the points p_i, p_j and p_k does not contain other points from P within the circle. The following definition of Delaunay triangulation arises from this statement.

Definition 3.2. *For a triangulation $\mathcal{T}$ of a set of points in the plane P, we say it is a Delaunay triangulation if and only if the circumscribed circle of any triangle in $\mathcal{T}$ does not contain the points from P inside the circle.*

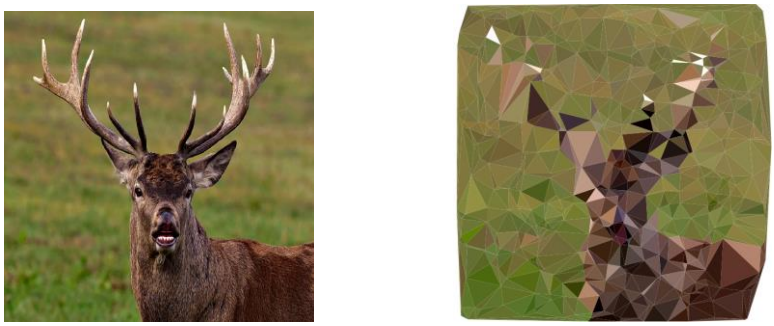
Thus defined triangulation is also called *Legal Delaunay triangulation*. Figure 1 presents Delaunay triangulation. It can be noticed that the legal triangulations $\Delta p_i\ p_j\ p_m$ and $\Delta p_i\ p_j\ p_k$ are formed inside the circles $C(p_i\ p_j\ p_m)$ and $C(p_i\ p_j\ p_k)$. If it happens that some point (vertex) suddenly appears, as in our case the vertex p_l, then these triangulations would be illegal Delaunay triangulations.



(a) The edge of the Delaunay diagram (b) Delaunay triangulation

Figure 1
Example of Delaunay triangulation

Figure 2 gives an example of a Delaunay triangulated image with colored triangles. The coding scenario will be explained later. In order to more clearly describe the process of Delaunay image triangulation, let us first explain the process of obtaining Delaunay triangulation.



(a) Original picture (b) Delaunay triangulated picture

Figure 2
Original and Delaunay triangulated picture

If we want to legalize them it is necessary to remove the closed line segment $\overline{p_m p_j}$ and to form a new one, i.e. $\overline{p_l p_t}$. The process of legalization of the edges is defined by the Algorithm 3.1 [12].

Algorithm 3.1 LegalizeEdge ($p_r, \overline{p_l p_j}$, $\mathcal{T}$)

Require: The point being inserted is p_r, and $\overline{\mathrm{p_l\ p_j}}$ is the edge of $\mathcal{T}$ that may need to be flipped.

1: **if** $\overline{p_l p_j}$ is illegal
2: **then** For $\Delta p_i p_j p_k$ adjacent to $\Delta p_i p_j p_r$ along $\overline{p_l p_j}$, flip $\overline{p_l p_j}$, i.e. replace $\overline{p_l p_j}$ with $\overline{p_r p_k}$.
3: LegalizeEdge ($p_r, \overline{p_l p_k}$, $\mathcal{T}$)
4: LegalizeEdge ($p_r, \overline{p_k p_j}$, $\mathcal{T}$)

In our method, we randomly select a set of points P within the given image, providing that their coordinates contained in vectors (X,Y) are within the given image resolution.

In general, locating the points begins by forming a large triangle $\Delta p_{-1} p_{-2} p_{-3}$ containing all points of the set P. It is necessary that these points are far enough away so they do not interfere with the Delaunay triangulation of set P. These points are chosen on the principle $p_{-1} = (3M,0)$, $p_{-2} = (0,3M)$ and $p_{-3} = (-3M,-3M)$, where M is the maximal absolute value among the coordinates of points in P. This ensures that P is contained within the triangle $\Delta p_{-1} p_{-2} p_{-3}$.

The number of triangles created by the Delaunay triangulation algorithm is at most $9n + 1$, and the time complexity of this algorithm is $O(n \log n)$ using $O(n)$ memory locations. This legalization process best describes the incremental Delaunay triangulation construction algorithm [12] which is also the basis of our presented image encryption model.

Suppose now that the set of sites P is given within the image of the desired resolution. It is necessary to determine for each image point which site is closest to it.

Definition 3.3. *Voronoi diagram V or (P) of a set of points $P = \{p_1, p_2, \ldots, p_n\}$ is a division of plane into areas (regions or cells) such that point X belongs to point area p_i if and only if:*

$$d(X, p_i) < d(X, p_j), \quad \forall i \neq j. \tag{1}$$

The area of the point $\mathrm{p_i}$ is called the Voronoi cell of p_i and denoted by $V(p_i)$. In Figure 3, we present the Voronoi diagram that is dual to the Delaunay diagram (triangulation).

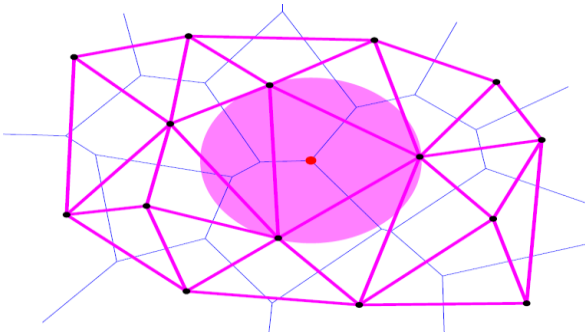


Figure 3
Voronoi diagram (blue) and Delaunay triangulation (pink)

It is important to note that the nodes of the Voronoi diagram are in fact the centers of the circles at which the mentioned Delaunay triangles already lie.

4 The Encryption by Delaunay Triangulation of Image and Authentication Scenario

Our image encryption method is based on the application of the Delaunay triangulation incremental algorithm on encrypting the (*x*,*y*) coordinates of points in *P*. The vertex coordinates encryption process is based on the use of Stack permutation method for binary representations of Catalan objects. As known, Catalan numbers C_n appear in many combinatorial problems counting so called Catalan objects, and they are given by formula [13]:

$$C_n = \frac{(2n)!}{(n+1)!\,n!} = \frac{1}{n+1}\binom{2n}{n}, \quad \text{n} \geq 0 \tag{2}$$

Particularly, for chosen *n*, C_n is the total number of bit strings containing exactly *n* '1' bits and *n* '0' bits and satisfying balanced parenthesis property. Such a bit string, as presented in *Definition 3.1*, is termed as Catalan key. A simple example of generating $C_3 = 5$ Catalan keys for $n = 3$ is shown in Figure 4.

For example, for $n = 4$, according to equation (2), we have a set of $C_n = 14$ values that satisfy the balanced parenthesis property. Decimal equivalents of these bit strings are 170, 172, 178, 180, 184, 202, 204, 210, 212, 216, 226, 228, 232, 240, while these bit strings are the following: 10101010, 10101100, 10110010, 10110100, 10111000, 11001010, 11001100, 11010010, 11010100, 11011000, 11100010, 11100100, 11101000, 11110000.

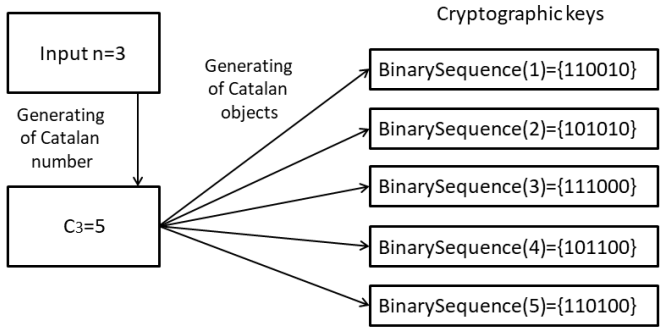


Figure 4
Generating Catalan keys for $n = 3$

In the process of encryption of the site coordinates in the Delaunay triangles, we use the Stack permutation method of chosen Catalan key.

1) If the current bit in the Catalan key is '1', then *push* the current bit of string, which is about to be encoded.

2) If the current bit in the Catalan key is '0', then *pop* one bit from the stack and send it to the output.

Example 4.1. *We present an example of encoding one of the* (x,y) *coordinates of the Delaunay triangle vertices in the image by applying the Stack permutations. Let* $x = 1430$ *with binary equivalent* $1430_{10}=10110010110_2$ *containing* $n = 11$ *bits. So, we need Catalan key with 22 bits. One of the possible choices is* $K = 2816098$. *Its binary equivalent is* $2816098_{10}=10101011111000011000010_2$.

Figure 5 illustrates details.

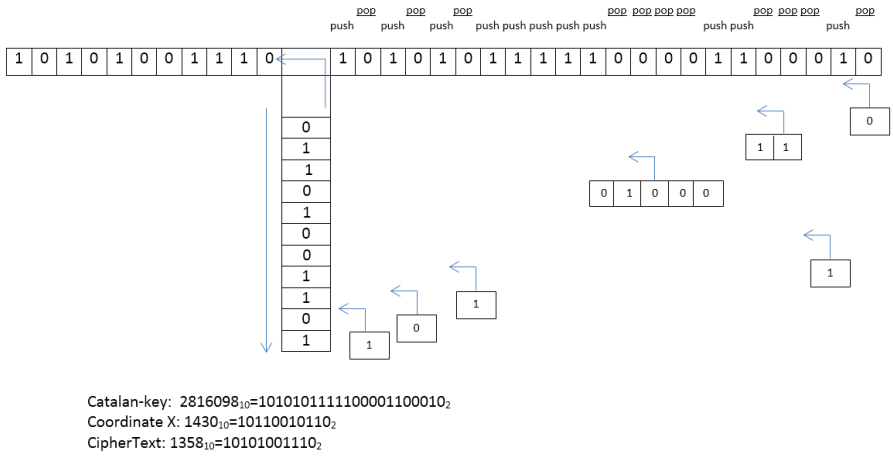


Figure 5
Coordinates encryption example based on Stack Permutation principle

In Example 4.1, the coordinate x has an integer value. Generally, its value can also be a real number. In that case, we encode integer and fraction part separately. Let us, now, explain the image encryption algorithm. The input elements are randomly selected sites, that is, their (x,y) coordinates. We start with initial triangulation $\mathcal{T}$ containing the triangle $\Delta p_{-1}p_{-2}p_{-3}$. This is, in fact, an auxiliary triangle from which we begin and later it loses on its importance because we do not need it. In other words, its vertices are removed as well as all the sides of the triangles contained therein.

In the next step, the initial triangle of the image $\Delta p_i p_j p_k$ is formed. It should be emphasized that the (x,y) coordinates of each vertex of this triangle, as well as all the subsequent ones that will be created, are in the range of the image resolution. This is a pre-condition needed in order to assure a correct triangulation of the image. The process of forming of this triangle ends after 3 count loops.

In each count, the coordinates of the vertices p_r are added to the array K which we need in the further (x,y) coordinates encryption process. After forming an initial triangle in the process of triangulation, the function *size()* is called to find a pixel with (x,y) coordinates located at the gravity center of the triangle (the center pixel of the triangle).

The way this function finds the pixel mentioned is represented by the following standard formula for calculating coordinates of the gravity center of a triangle:

$$T = \left(\frac{x_1 + x_2 + x_3}{3}, \frac{y_1 + y_2 + y_3}{3}\right) \tag{3}$$

It should be noted that function *size*() is called for each newly created triangle. Then, the RGB function, coloring the triangle with the color of the center pixel, is called. It should be emphasized that for the 4th and every subsequent entry of a random vertex p_r, one can expect two cases about its positions with respect to the triangle $\Delta p_i p_j p_k$ and within the "large" triangle $\Delta p_{-1}p_{-2}p_{-3}$. In the first case, p_r is inside the triangle $\Delta p_i p_j p_k$. In the other case, p_r is on the edge of $\Delta p_i p_j p_k$.

In the first case, three lines will be drawn from the vertex p_r to the adjacent sites, while in the second case two lines will be drawn from p_r to the opposite vertices p_l and p_k (it is supposed here vertex p_l occurred outside the triangle $\Delta p_i p_j p_k$). In case that a vertex p_r appears outside the triangle $\Delta p_i p_j p_k$, then a triangle is formed by drawing lines towards all vertices visible from the vertex p_r. In each of these steps, Algorithm 3.1 is called to verify each edge of an adjacent triangle with respect to p_r, taking care that the Delaunay triangulation condition is satisfied.

Figure 6 presents in detail the ways of the accidental appearance of the vertex p_r.

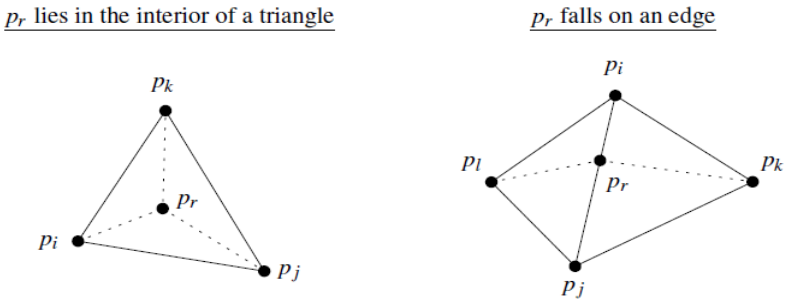


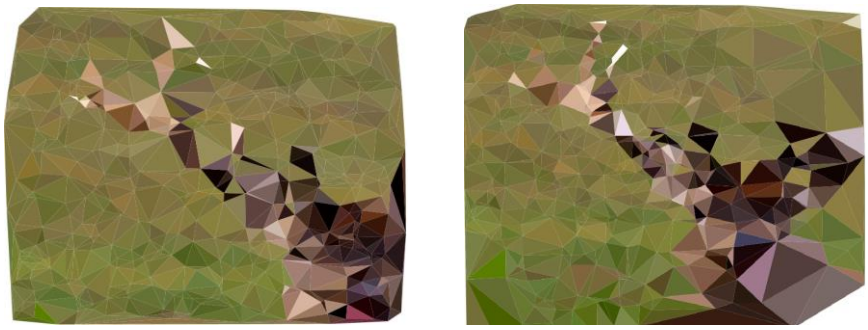
Figure 6
Ways of appearance of the vertex p_r

After the edges have been legalized, that is, the process of input of the triangle vertices is completed; the vertices and edges of the large triangle $\Delta p_{-1}\, p_{-2}\, p_{-3}$ are removed because they were needed only to create the initial image triangulation. In the following step, as a return value, we get triangulated image $\mathcal{D}$, that is, $\mathcal{D}$ is the set of all triangles obtained by the triangulation process. Now, it should be emphasized that $\mathcal{D}$ (triangulated image) is in fact a set of triangles that is a subset of the set, where $\Delta p_{-1}p_{-2}p_{-3}$ and incident edges are removed. In this way, the portions of the image outside the convex hull (i.e., outside the boundary edges of $\mathcal{D}$) are removed.

Now, the elements of the array K, that is, the (x,y) coordinate values of the vertices, are converted into their binary equivalents. This is the operating condition of the *Stack Permutation*) method. Then, we choose corresponding Catalan key and, using Stack Permutation method, encrypt the coordinates and put them into the array K_s.

From now on, the image triangulation process is repeated as in the previous steps, except that the tags are changed ("*s*" added) due to clarity. The decryption process, i.e., returning the image to its original triangulated form, is obtained so that the encrypted coordinates of the vertices change place with the original (originally chosen at random) in methods called due to decryption. In this way, the originally triangulated image is obtained.

In Figure 7, one can clearly see the difference between the original and the encrypted image.



(a) Delaunay triangulated image (b) Delaunay triangulated-encrypted image

Figure 7
Delaunay triangulated and Delaunay triangulated-encrypted image

For the purpose of the authentication, for each user, in addition to username and password, the original image, the Delaunay triangulated image, and the Delaunay triangulated-encrypted image are kept in Table 1.

Table 1
User table in Bank system (*A*=Random image; *B*=Delaunay triangulation image; *C*=Encrypted-Delaunay triangulation image)

Authentication data	**Catalan Key Decimal -Binary**	**Input / output files**	**The coordinate string of the Delaunay image triangulation**	**The coordinate string of the Decode-Delaunay image triangulation**	**Index of array Code-Delaunay image triangulation**
ID=1 username = pera password= peric2816098	2816098= 10101011 11100001 11100010	A=image.jpg B=del_image.jpg C=code_image.jpg	vertex 1: X=124 Y=439, vertex 2:X=397 Y=114, vertex 3: X=26 Y=2, vertex 4: X=144 Y=510, vertex 5: X=408 Y=265, vertex 6: X=491 Y=194, vertex 7: X=322 Y=14, vertex 8: X=344 Y=26, vertex 9: X=268 Y=157, vertex 10: X=349 Y=477	vertex 1: X=244 Y=367, vertex 2: X=391 Y=120, vertex 3: X=200 Y=8, vertex 4: X=66 Y=510, vertex 5: X=450 Y=385, vertex 6: X=443 Y=26, vertex 7: X=280 Y=140, vertex 8: X=464 Y=200, vertex 9: X=388 Y=199, vertex 10: X=469 Y=471	vertex 4: X=66 Y=510, vertex 7: X=280 Y=140, vertex 2: X=391 Y=120

Now we present two algorithms. Algorithm 4.1 performs the image triangulation, while Algorithm 4.2 encrypts the image. It should be emphasized that the result of Algorithm 4.1 is presented in Figure 7(a). After triangulating the image, we launch the second algorithm. The result of Algorithm 4.2 is presented in Figure 7(b).

Algorithm 4.1 Delaunay triangulate picture (p_r, $\overline{p_1 p_J}$, $\mathcal{T}$)

Require: Randomly selected image and set P.
1: Make initial set of triangles $\mathcal{T}$ containing Δp_{-1}, p_{-2} and p_{-3}.
2: **for** $r = 1$ to n do (Put p_r in $\mathcal{T}$)
Find $\Delta p_i p_j$pk $\in \mathcal{T}$, which contains p_r
Put the p_r into array K.
if p_r in the interior of the $\Delta p_i p_j p_k$

then
LegalizeEdge (pr, $\overline{p_i\, p_j}$, $\mathcal{T}$)
Call ***size()*** to find central pixel of $\Delta p_r\, p_i p_j$
Call **RGB** function and color $\Delta p_r\, p_i\, p_j$
LegalizeEdge (p_r, $\overline{p_j\, p_k}$, $\mathcal{T}$)
Call ***size()*** to find central pixel of $\Delta p_r\, p_j\, p_k$
Call **RGB** function and color $\Delta p_r\, p_j\, p_k$
LegalizeEdge (p_r, $\overline{p_k\, p_i}$, $\mathcal{T}$)
Call ***size()*** to find central pixel of $\Delta p_r\, p_k\, p_i$
Call **RGB** function and color $\Delta p_r\, p_k\, p_i$
else (p_r on an edge of $p_i p_j p_k$, say the edge $\overline{p_i\, p_j}$)
LegalizeEdge (p_r, $\overline{p_i\, p_l}$, $\mathcal{T}$)
Call ***size()*** to find central pixel of $\Delta p_r\, p_i\, p_l$
Call **RGB** function and color $\Delta p_r\, p_i\, p_l$
LegalizeEdge (p_r, $\overline{p_l\, p_j}$, $\mathcal{T}$)
Call ***size()*** to find central pixel of $\Delta p_r\, p_l\, p_j$
Call **RGB** function and color $\Delta p_r\, p_i\, p_l$
LegalizeEdge (p_r, $\overline{p_j\, p_k}$, $\mathcal{T}$)
Call ***size()*** to find central pixel of $\Delta p_r\, p_j\, p_k$
Call **RGB** function and color $\Delta p_r\, p_j\, p_k$
LegalizeEdge (p_r, $\overline{p_k\, p_i}$, $\mathcal{T}$)
Call ***size()*** to find central pixel of $\Delta p_r\, p_k\, p_i$
Call **RGB** function and color $\Delta p_r\, p_k\, p_i$
3: Discard p_{-1}, p_{-2} and p_{-3} with all their incident edges from $\mathcal{T}$.
4: **Output:** D ($\mathcal{D}$ is triangulated picture or subset triangles of the set $\mathcal{T}$).

Algorithm 4.2 Delaunay triangulate-encryption picture (p_r, $\overline{p_i\, p_j}$, $\mathcal{T}$)

Require: Triangulation $\mathcal{D}$ resulting from Algorithm 4.1 and array K.
1: **for** r = 1 to n do (Access p_r in the array K)
2: Convert p_r in binary record
3: Call **Stack permutation** method on the basis of chosen Catalan key.
4: Convert p_s in decimal record (after permutation, bit p_r becomes p_s)
5: Put the p_s in array K_S.
6: Make initial set of triangles $\mathcal{T}_s$ containing Δp_{-1}, p_{-2} and p_{-3}..
7: **for** s = 1 to n do (Put p_s in $\mathcal{T}_s$)
Find $\Delta p_i p_j p_k \in \mathcal{T}_s$, which contains p_s
Put the p_r into array K.
if p_s in the interior of the $\Delta p_i p_j p_k$
then
LegalizeEdge (p_s, $\overline{p_i\, p_j}$, $\mathcal{T}_s$)
Call ***size()*** to find central pixel of $\Delta p_s p_i p_j$
Call **RGB** function and color $\Delta p_s p_i p_j$
LegalizeEdge (p_s, $\overline{p_j\, p_k}$, $\mathcal{T}_s$)
Call ***size()*** to find central pixel of $\Delta p_s p_j p_k$
Call **RGB** function and color $\Delta p_s p_j p_k$

LegalizeEdge (p_s, $\overline{p_k\, p_i}$, $\mathcal{T}_s$)
Call ***size()*** to find central pixel of $\Delta p_s p_k p_i$
Call **RGB** function and color $\Delta p_s\ p_k\ p_i$
else (p_s on an edge of $p_i\ p_j\ p_k$, say the edge $\overline{p_i\, p_j}$)
LegalizeEdge (p_s, $\overline{p_i\, p_l}$, $\mathcal{T}_s$)
Call ***size()*** to find central pixel of $\Delta p_s p_i p_l$
Call **RGB** function and color $\Delta p_s\ p_i p_l$
LegalizeEdge (p_s, $\overline{p_l\, p_j}$, $\mathcal{T}_s$)
Call ***size()*** to find central pixel of $\Delta p_s\ p_l\ p_j$
Call **RGB** function and color $\Delta p_s\ p_i\ p_l$
LegalizeEdge (p_s, $\overline{p_j\, p_k}$, $\mathcal{T}_s$)
Call ***size()*** to find central pixel of $\Delta p_s p_j p_k$
Call **RGB** function and color $\Delta p_s p_j p_k$
LegalizeEdge (p_s, $\overline{p_k\, p_i}$, $\mathcal{T}_s$)
Call ***size()*** to find central pixel of $\Delta p_s p_k p_i$
Call **RGB** function and color $\Delta p_s p_k p_i$
8: Discard p_{-1}, p_{-2} and p_{-3} with all their incident edges from $\mathcal{T}_s$.
9: **Output:** $\mathcal{D}_s$ ($\mathcal{D}_s$ is triangulated picture or subset triangles of the set $\mathcal{T}_s$).

Figure 8 presents the authentication process in 5 steps, which we will explain in details.

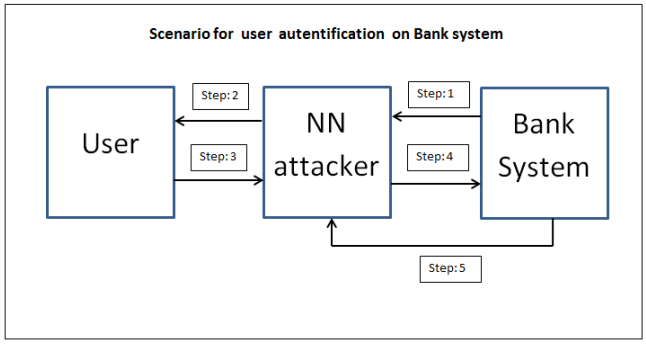


Figure 8
Scenario for user authentication on Bank system

Step 1: When the user requests the authentication, the banking system firstly identifies the user by the randomly assigned Catalan key. This is the numerical part of the password. For example, if the username is: *"pera"* and the password is *"peric2816098"*, the Catalan key is 2816098 in decimal. In binary, it is 10101011111000001100010. Then, the system randomly selects an image and a number of sites. The incremental algorithm for Delaunay image triangulation is performed and the vertex coordinates are stored in an array. For example, if an image is selected and triangulated with 10 random points, the vertex coordinates

that would be stored in Delaunay array can be: p_1 = (124,439), p_2 = (397,114), p_3 = (26,2), p_4 = (144,510), p_5 = (408,265), p_6 = (491,194), p_7 = (322,14), p_8 = (344, 26), p_9 = (268,157), *and* p_{10} = (349,477).

Step 2: The triangulated image is encrypted by the numerical part of the password, i.e. the Catalan key. We now get the encrypted triangulated image, with a Delaunay array of encrypted coordinates with 10 vertices: p_{s1} = (244,367), p_{s2} = (391,120), p_{s3} = (200,8), p_{s4} = (66,510), p_{s5} = (450, 385), p_{s6} = (443, 26), p_{s7} = (280,140), p_{s8} = (464,200), p_{s9} = (388,199), *and* p_{s10} = (469, 471). This image and array are kept in the database because they are later very important in the user authentication process. Now, the originally triangulated image, with the original array of 10 vertices, is sent by the banking system to the user. At this point, a potential attacker has the opportunity to capture a file with a triangulated image and 10 vertices coordinates array. Since no confirmation is being sought at this point, the attacker simply passes the file to reach the user.

Step 3: In this step, the user accepts the file with a triangulated image, 10 vertices of the coordinates array and by the same Catalan key (because only he and the Bank know that the numeric part of the password or Catalan key) encrypts the resulting image and gets the same Delaunay encrypted image and encrypted vertex coordinates as in Step 2. At this point, the user sends the encrypted image to the banking system but does not send the array with encrypted coordinates. Again, the attacker captures the submitted image and forwards it to the Banking system. It should be emphasized that the attacker has no information about the encrypted coordinates of the vertices.

Step 4: This step is reserved for verifying the authenticity of a Delaunay triangulated image sent by the user and the one that has been encrypted and stored by the banking system. Since the images are the same, there is user authentication verification.

Step 5: Given the fact that the attacker, even if he had captured and forwarded the encrypted image to the banking system, does not have information on the value of the encrypted coordinates, the banking system sends an input request for 3 random index coordinates of the encrypted Delaunay array, e.g., values for (x,y) coordinates with indices 4,7 and 2. In this case, the attacker should enter the values p_{s4} = (66,510), p_{s7} = (280,140), and p_{s2} = (391,120), which is almost impossible. If the system does not receive feedback on the correct coordinates, it understands that the attacker is involved in the process of authentication and suspends further actions. Yet, if it receives the correct values of the requested coordinates, then the transaction is approved and that means that the attacker did not participate in the authentication process.

5 Experimental Results

The time of the Delaunay image triangulation and its encryption was tested on the triangle vertices for n = {20,40,80,160,320,640}. Our Delaunay incremental algorithm, which is modified by the Stack permutation method, is implemented in *Java NetBeans environment* (see encryption time for proposed method in Table 2).

Table 2

Encryption time for proposed method

N Vertex	Delaunay Triangulation in "ms"	Code Delaunay Triangulation in "ms"
10	70	70
20	83	90
40	173	119
80	325	326
160	1370	1237
320	8261	10064
640	84469	92571

If we present the tabular results in a graph, we will notice that the encryption time is not directly proportional to the number of triangle vertices. We may also notice that the encryption time to a large extent is not much different from the needed time for sole Delaunay triangulation, which points to the efficiency of the stack permutation method (see Figure 9).

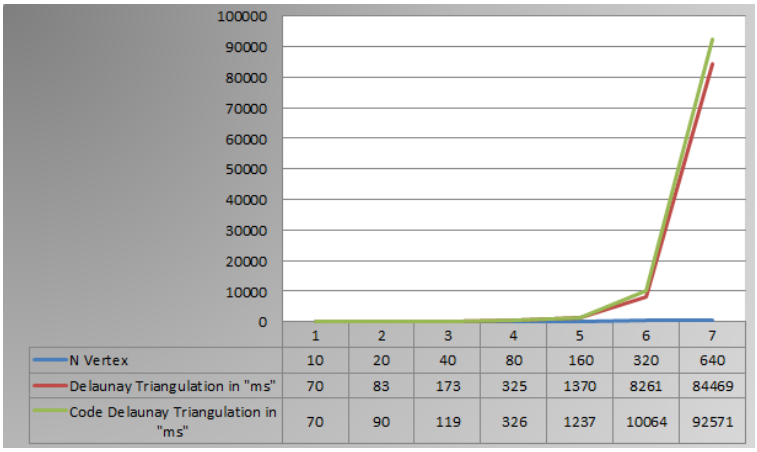


Figure 9

Graphical illustration of the encryption time

Testing was done on a computer with the following features: *Intel Core i5-CPU 2.6 GHz, RAM 8GB, Operating system: Windows 7 Microsoft -64 bits.*

Conclusion and Further Research

The main contribution of this proposed method is novel encryption using the Delaunay Triangulation incremental algorithm and the Catalan objects. The proposed method is a combination of computational geometry, authentication, and cryptography. This method presents a new step towards encoding the triangle coordinates using the Catalan-key. The presented method consists of several stages. Catalan object is assigned to the user by a random selection from the corresponding database. Afterward, a random 2D image is selected and triangulated by the Delaunay Triangulation Incremental algorithm. During the triangulation process, we enter n randomly selected triangle vertices whose *(x,y)* coordinates are stored in an array. The proposed method belongs to both computational geometry and cryptography. It presents a new step towards encoding the triangle coordinates using the Catalan key.

From testing the proposed method, it can be concluded that the best authentication is performed by asking the client to enter the *(x,y)* coordinate values of randomly selected 3 indices of an array. If the entered coordinates match the index values in the banking system array, then the transaction or other operation is approved. If the matching fails, it means that we have an unidentified person who has followed the whole process and wants to break into the banking system. The triangulated image is encrypted by the assigned Catalan object and the Stack Permutation method. For encryption purposes, in our cryptosystem, we use $n>256$. Considering given computational and memory limits, it is virtually impossible to generate a complete set of all Catalan keys (or objects). In fact, creating a large space of Catalan keys ensures the security of the described cryptosystem.

Directions for further development of our method can be related to personal authentication using hand vein triangulation, modeled on the work of Kumar and Prathyusha [14, 15]. Their method is a novel approach to authenticate individuals using triangulation of hand vein images and simultaneous extraction of knuckle shape information. Also, this approach is fully automated and employs palm dorsal hand vein images acquired from the low-cost, near-infrared, contactless imaging.

Furthermore, some issues in applying 3D Delaunay triangulation in fingerprint authentication can be discussed. From our previous research, according to the model [16] where we have shown the possibilities of applying the triangulation method in the biometric identification process, 3D Delaunay triangulation in fingerprint and face print authentication research can be used in further work.

Acknowledgments

Predrag Stanimirović gratefully acknowledge support from the Ministry of Education, Science and Technological Development, Republic of Serbia, Grant No. 174013.

References

[1] Luan, G., Li, A., Zhang, D., Wang, D. Asymmetric image encryption and authentication based on equal modulus decomposition in the Fresnel transform domain, IEEE Photonics Journal Open Access, 2019, Vol. 11, No. 1, Art. No. 8572731

[2] Yuan, L., Ran, Q., Zhao, T. Image authentication based on double-image encryption and partial phase decryption in nonseparable fractional Fourier domain, Optics Laser Technology, 2017, Vol. 88, pp. 111-120

[3] Yang, H., Wong, K., Liaoc, X., Zhang, W., Wei, P. A fast image encryption and authentication scheme based on chaotic maps, Communications in Nonlinear Science and Numerical Simulation, 2010, Vol. 15, No. 11, pp. 3507-3517

[4] Zanobya, N. K., Qureshi, R. J., Ahmad, J. On Feature based Delaunay Triangulation for Palmprint Recognition, Journal of Platform Technology, 2015, Vol. 3, No. 4, pp. 9-18

[5] Vijaya Ranjini, S., Rajarajan, S. Enhanced Fingerprint Recognition with OTP using Delaunay Triangulation to Improve ATM Security, Indian Journal of Science and Technology, 2016, Vol. 9, No. 1, pp. 1-6

[6] Saračević, M., Adamović, S., Biševac, E. Application of Catalan Numbers and the Lattice Path Combinatorial Problem in Cryptography, Acta Polytechnica Hungarica, 2018, Vol. 15, No. 7, pp. 91-110

[7] Saračević, M., Aybeyan, S., Selimović, F. Generation of cryptographic keys with algorithm of polygon triangulation and Catalan numbers, Computer Science AGH, 2018, Vol. 19, No. 3, pp. 243-256

[8] Saračević, M., Korićanin, E., Biševac, E. Encryption based on Ballot, Stack permutations and Balanced Parentheses using Catalan-keys, Journal of Information Technology and Applications, 2017, Vol. 7, No. 2, pp. 69-77

[9] Saračević, M., Adamović, S., Miškovic, V. A novel approach to steganography based on the properties of Catalan numbers and Dyck words, Future Generation Computer Systems, 2019, Vol. 100, pp. 186-197

[10] Yang, W., Hu, J., Wang, S. The effect of spurious and missing minutiae on Delaunay triangulation based on its application to fingerprint authentication, 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Xiamen, 2014, pp. 995-999, doi: 10.1109/FSKD.2014.6980975

[11] Hu, J., Khalil, I., Tari, Z., Wen, S. Application of 3D Delaunay Triangulation in Fingerprint Authentication System, Lecture Notes of the Institute for Computer Sciences Social Informatics and Telecommunications Engineering, 2018, Vol. 235, pp. 291-298

[12] De Berg, M., Kreveld, M., Overmars, M., Schwarzkopf, O. Computational Geometry Algorithms and Applications, Springer-Verlag, Berlin, Hidelberg, 1997

[13] Koshy, T. Catalan Numbers with Applications, Oxford University Press, New York, 2009

[14] Kumar, A., Prathyusha, K. V. Personal Authentication Using Hand Vein Triangulation and Knuckle Shape, IEEE Transactions on Image Processing, 2009, Vol. 18, No. 9, pp. 2127-2136

[15] Kumar, A., Prathyusha, K. V. Personal authentication using hand vein triangulation, Biometric Technology for Human Identification, 2008, Vol. 6944, doi: 10.1117/12.779159

[16] Saračević M., Elhoseny, M., Selimi, A., Lončarević Z. Possibilities of applying the triangulation method in the biometric identification process, in Springer: Biometric Identification Technologies Based on Modern Data Mining Methods, 2020