

A novel security and authentication method for dorsal hand vein with discrete time chaotic systems [☆]

Omer Faruk Boyraz^a, Emre Guleryuz^b, Akif Akgul^c, Mustafa Zahid Yildiz^d, Harun Emre Kiran^e,
Jawad Ahmad^f

Anadolu Isuzu Automotive Industry and Trade Inc., R&D Center, Kocaeli, Turkey

Protek Health Informatics R&D Center, Kocaeli, Turkey

*Department of Computer Engineering
Faculty of Engineering, Hitit University
19030, Corum, Turkey*

*Department of Electrical and Electronics Engineering
Faculty of Technology, Sakarya University of Applied Sciences
54050, Sakarya, Turkey*

*School of Computing, Merchiston Campus, 10 Colinton Road,
Edinburgh Napier University, UK.*

^aomer.boyraz@isuzu.com.tr

^bemreguleryuz61@gmail.com

^cakifakgul@hitit.edu.tr

^dmustafayildiz@subu.edu.tr

^eharunemrekiran@hitit.edu.tr

^fJ.Ahmad@napier.ac.uk

Abstract

Hand vein images have become important biometric signs used for identification systems. Also, dorsal hand vein images have noteworthy advantages in terms of reliability and contactless procedure. Surgically changing the vascular pattern under the skin is extremely difficult. Therefore, the use of such patterns in identity recognition applications is increasing day by day. In this context, many studies have been carried out in essential areas such as image acquisition for different textures, image preprocessing, data security, image feature extraction and recognition. In this article, a new dorsal hand vein identification application has been carried out with the chaos-based security mechanism to protect personal data and the Improved SURF method to reduce the error matches of traditional SURF method in the current application. There are two improvements in the presented system. The first one finds the features representing only the vein segments, not the remaining tissue or hairy parts. The second one eliminates the mismatching points to increase the accuracy.

Both encryption and data hiding techniques are preferred together to protect person data. In addition, dorsal hand vein images taken with a camera for matching and recorded in the database of the people have been improved with the newly developed I-SURF feature extraction method. Thereby,

with the developed application, in addition to keeping the personal identity information securely in the database with chaos-based methods, higher accuracy rate has been achieved in matching the new image taken with the camera.

Keywords:

Chaos, Chaotic systems, Data security, Vein images, Interface design, Improved SURF.

1. Introduction

Biometrics; it allows to identify people according to human physical characteristics that vary from person to person, such as face, fingerprint, iris, gait. Surgically changing the vascular pattern under the skin, which is another physical characteristic, is extremely difficult [1, 2]. Thus, a biometric system created using dorsal hand vein patterns that are unique from person to person is extremely safe [3]. The fact that the vein patterns are largely hidden under the skin and difficult to steal or view under visible light makes it advantageous to be preferred in identification applications. The fact that vascular patterns are largely hidden under the skin and that they are difficult to get illegally or scan under visible light make it advantageous to be preferred in identification applications.

Performing verification and recognition from vein patterns on the dorsal hand has many advantages over other pattern recognition systems (such as fingerprint, palm print, finger vein). One of the most important advantages of this is the contactless and sterile identification and verification processes during the acquisition of the images [4]. In the literature, vein recognition technology includes image acquisition, image preprocessing, feature extraction and recognition, respectively [5].

Most of the proposed methods for feature extraction use tissue or pattern information extracted from vein images to characterize the vein model. Based on vein recognition studies, it can be said that there are three different feature extraction groups. The first of these is a method based on geometry. These methods use vascular structure information. Geometry-based feature extraction methods such as mean curvature [6], extreme graph of directional fields (EGDF) [7] Gabor filter [8, 9, 10, 11] and the maximal intra-neighbor difference (MIND) vectors [12] are affected by rotation and scaling of vein images. Methods such as Local Binary Pattern (LBP) [13] and the Local Derivative Pattern (LDP) [14, 15], the local triple model Local Ternary Patterns (LTP) [16] extract various statistical data from vein images. These statistical methods can show a gray histogram distribution of the vein image. However, these methods lose positional information on the vascular tissue. In addition, these methods are sensitive to rotation and scale changes. Therefore, they are not suitable methods for contactless vein recognition. The third feature extraction methods are local invariant feature based methods. Some of these are the SIFT [17] and SURF [18] methods, which are not affected by both rotation scaling and rotation.

When the previous studies on dorsal hand vein recognition applications are examined, it is seen that the extraction of local invariant features is more suitable for contactless recognition. Because local invariant properties are not affected by changes such as rotation, scale and depth. However, many challenges were encountered in practice. Because of various noises, hairy areas in tissue, and low-contrast pattern, feature points obtained from vein images may be relatively low. For these reasons, such challenges can be overcome by increasing the image contrast and applying noise canceling filters.

Irrelevant areas need to be removed in order to speed up image processing processes and obtain crucial key-points. After extraction of the target Region of Interest (RoI), various image processing steps are performed and the images are ready for key-point detection. While these processes are necessary for key-point extraction, traditional local feature extraction methods cannot fully detect crucial key-points. One of the traditional and popular feature extraction methods, SURF does not only detect key-points on the relevant vein pattern. In addition, the method finds key-points in hairy areas and tissues other than the vein area. This causes both a decrease in accuracy and an increase in time for feature matching.

Chaos is an important discipline that is seemingly disordered but has a special order to it, and studies non-linear events. The discipline is used in many fields such as communication [19], security [20], medicine [21, 22], control [23], random number generators [24, 25, 26], and its usage areas are becoming increasingly common.

Today, chaos theory has been applied to secure communication with many new methods [27], and chaotic systems are used in many areas such as encryption, data hiding, and random number generators. For example, Datcu et al. [28] carried out an encryption application with pseudo-random number generator design. Ahmad et al. [29] carried out a study on confusing and diffusing the image pixels by utilizing the chaotic system. Liao et al. [30], on the other hand, proposed a chaos-based random number generator design.

The main contributions made in this study are: (1) We encrypt identity data taken from individuals for identification randomly with a chaotic system in the preprocessing, and then hide it more than once in random coordinates in the image using the LSB method, again chaotically; (2) in the process of matching images, we develop Improved SURF (I-SURF) method to increase the accuracy of matching features obtained from dorsal vein images.

In the article's scope, after the introduction section, discrete-time chaotic systems used in encryption and data hiding applications, their dynamical analyzes and the designed security and authentication method are explained. In the third section, the chaos-based random number generator design, analysis and text encryption application, and in the fourth section, the methods developed to recover the data with data hiding application without corruption are mentioned. In the fifth section, advantages of I-SURF method are shown by matching data and applying identification. In the sixth section, an interface design is made so that the application can be easily used by users. In the last section, the conclusion and future work are mentioned.

2. The Chaotic Maps Used in the Study and A Novel Security Method Design

2.1. Discrete-Time Chaotic Maps

In this article study, two one-dimensional different discrete-time chaotic systems are used for encryption and data hiding. Cubic Map, the first of these chaotic systems, is used to encrypt identity data received from a person who registered in the application database. Ricker's Population Model, which is another preferred chaotic system, is preferred for determining coordinates of dorsal vein images of the person in order to hide the encrypted data within the images. The reason for using the one-dimensional discrete-time chaotic systems is that such systems have much faster processing times than continuous-time systems.

The equation for Cubic Map is given in Equation 1, and the equation for Ricker's Population Model Map is given in Equation 2.

$$X_{n+1} = AX_n(1 - X_n^2) \quad (1)$$

Parameter A in Equation 1 is taken as 3 in this article study. If the initial condition is X_0 , it is set to 0.1. For Equation 2, if the value of parameter A is 20 and its initial condition is X_0 , it is set to 0.1.

$$X_{n+1} = AX_n e^{-X_n} \quad (2)$$

There are many preferred analysis methods to understand whether a system is chaotic. Behaviors of the system in a certain time (time series), phase portraits, examination of bifurcation diagrams are some of these analysis methods. With these methods, it can be decided whether the system is chaotic or not. These analyzes will be discussed in the next subsection of the section.

2.1.1. Time Series Analysis

A time series is a graph of data showing how the parameters of a system behave over time [31]. The time series of values in the range $X_0 - X_{200}$ calculated using Equation 1 is shown in Figure 1. In addition, the time series of values in the range of $X_0 - X_{200}$ calculated using Equation 2 is given in Figure 2.

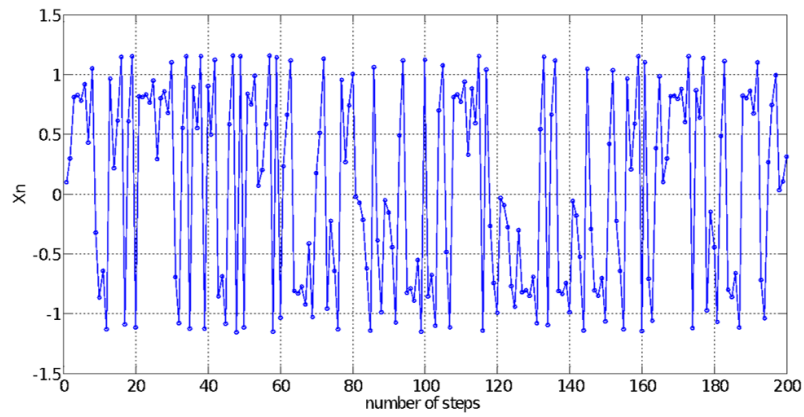


Figure 1: The time series of Cubic Map for X_n

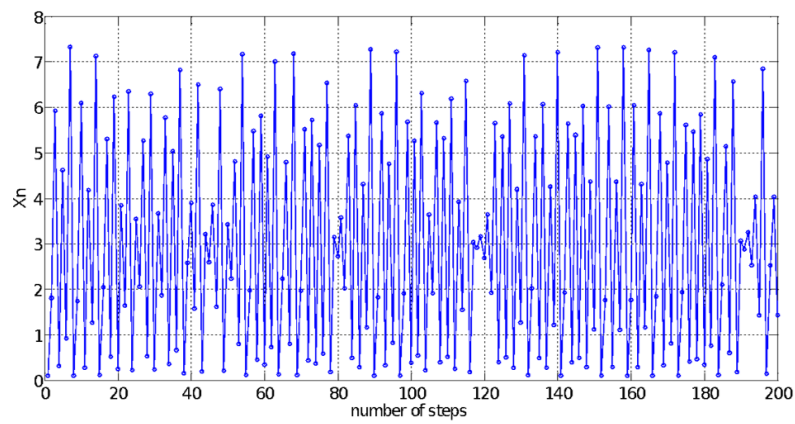


Figure 2: The time series of Ricker's Population Model for X_n

The time series shown in Figure 1 and Figure 2 are observed as a non-periodic signal with random behavior. It is also shown in Figure 3 and 4 that they are sensitively dependent on initial conditions. Because of these features, the two systems show chaotic behavior and are suitable for security applications.

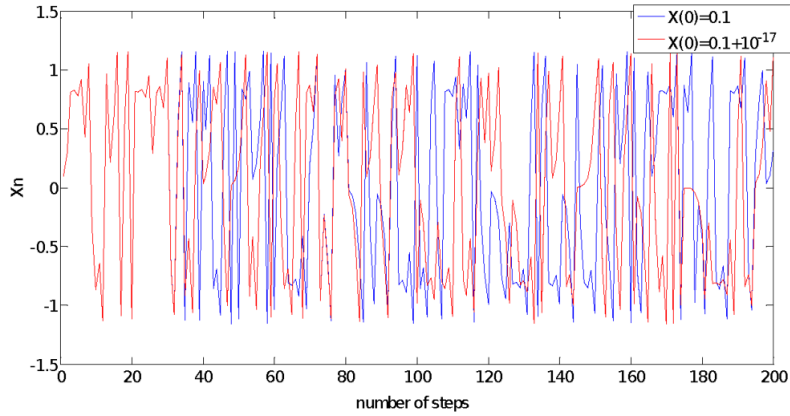


Figure 3: Initial condition sensitivity for Cubic Map

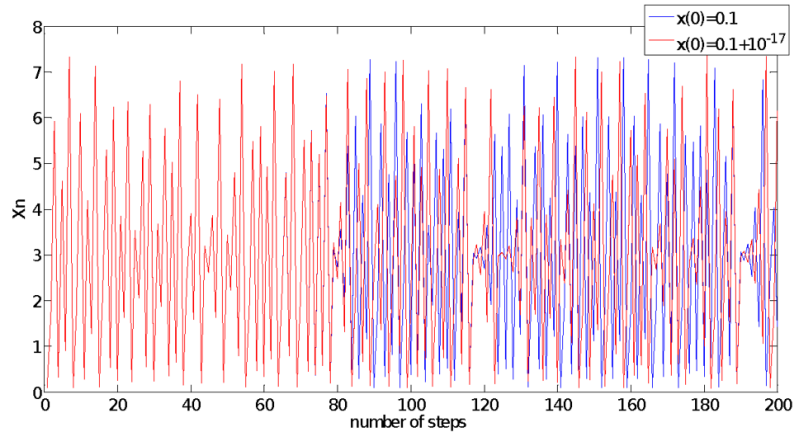


Figure 4: Initial condition sensitivity for Ricker's Population Model

Initial condition of the Cubic Map nonlinear system given in Equation 1 and Figure 3, and Ricker's Population Model nonlinear system given in Equation 2 and Figure 4 are shown the time series of X_n values found by calculating 200 steps at 0.1 and $0.1 + 10^{-17}$ values. According to the Figure 3, the first 30 steps are the same, but after 30 steps all the values are different. According to the figure, even a 10^{-17} change in the initial condition affects the result. Thus, the encrypted data cannot be decrypted properly, as the random numbers will not be the same as the random numbers used to encrypt the identity information of the person. In the Figure 4, the first 80 steps are the

same and all the next values are different. According to the figure, coordinates of encrypted identity data hidden inside dorsal vein images cannot be found properly and the encrypted data cannot be obtained properly.

2.1.2. Phase Portraits

Since both nonlinear systems used in this study are one-dimensional, one phase portrait analysis is performed. The x-axis of the phase portraits is X_n , and the y-axis is X_{n+1} . Phase portraits according to values calculated in 100 thousand steps are shown in Figure 5 and Figure 6.

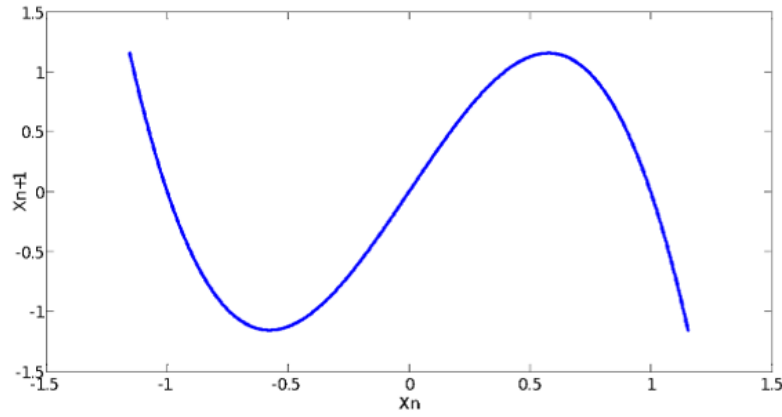


Figure 5: The Phase portrait of Cubic Map

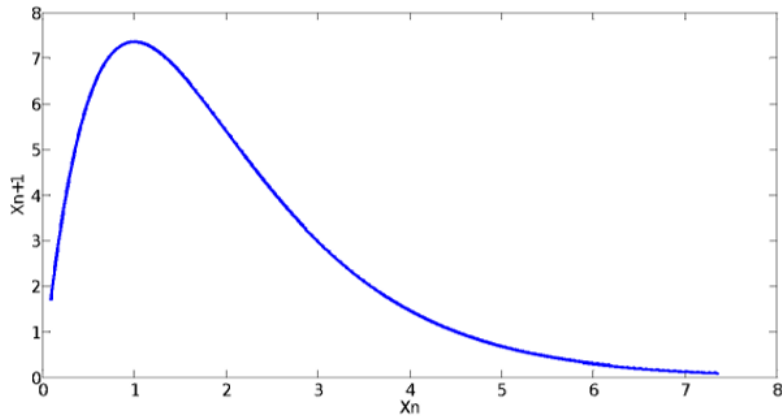


Figure 6: The Phase portrait of Ricker's Population Model

In Figure 5 and Figure 6, X_n and X_{n+1} take values in a certain order relative to each other. At the same time, it can be said that these nonlinear systems are chaotic because they take different values at each step.

2.1.3. Bifurcation Diagram

One of the analysis methods to understand whether a system is chaotic is to examine bifurcation diagrams. By means of the method, which is plotted with the bifurcations that occur when the system parameters change, it is possible to analyze chaoticity of the system as well as at which points it is chaotic or not.

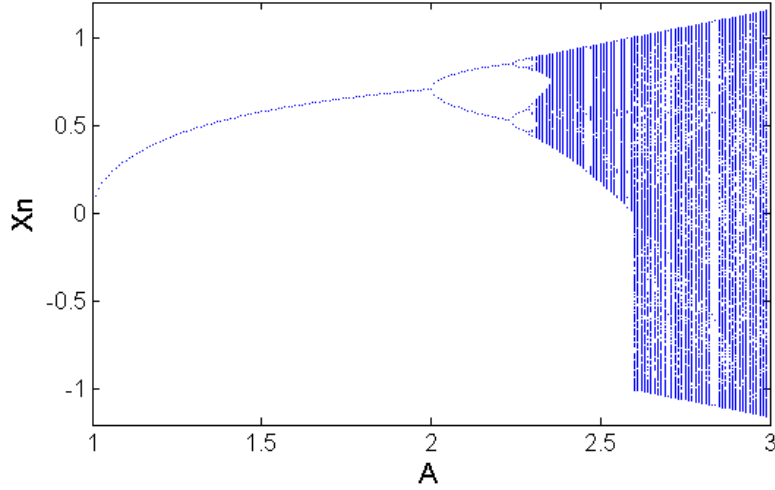


Figure 7: The bifurcation diagram of Cubic Map according to A parameter

The value of parameter A in the Cubic map system is given between 1 and 3, and the bifurcation diagram is shown in Figure 7. In the figure, it is seen that the chaotic range of the system is 2.3 – 3.

The value of parameter A in the Ricker's Population Model system is given between 10 and 25, and the bifurcation diagram is shown in Figure 8. In this figure, it is seen that the chaotic range of the system is $15 - 22.2$.

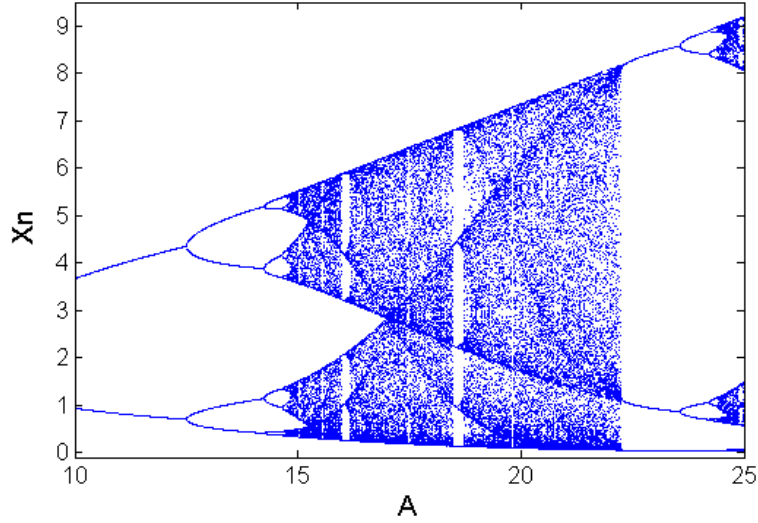


Figure 8: The bifurcation diagram of Ricker's Population Model according to A parameter

2.2. A novel security and authentication method

In this section, all steps performed in the article study are explained in summary form in a block diagram. When the block diagram given in Figure 9 is examined, firstly a raw image is taken from a IR camera module and the steps are started. In the registration step, Region of Interest (RoI) is extracted from the images and vein patterns in the region are processed. The resulting images are processed with Gray Level conversion, Gaussian Filter, and Contrast Limited Adaptive Histogram Equalization (CLAHE), respectively, to reveal the vein pattern.

After the image processing steps, numbers required for security operations are generated with a random number generator, and identity information received from people is encrypted with the numbers. Then, to increase security, the encrypted information is hidden in the LSB of pixels whose coordinates in the image are found by other generated random number generators in binary number format. In the last step, key-points are extracted from the processed images and saved to an identification database.

In the verification step, the real-time hand image taken by the camera is passed through the image processing steps again, and then key-point extraction is performed on the processed image. The key-points of images in the database are matched with the key-points of the image, and the verification processes are completed as "Accept" or "Reject".

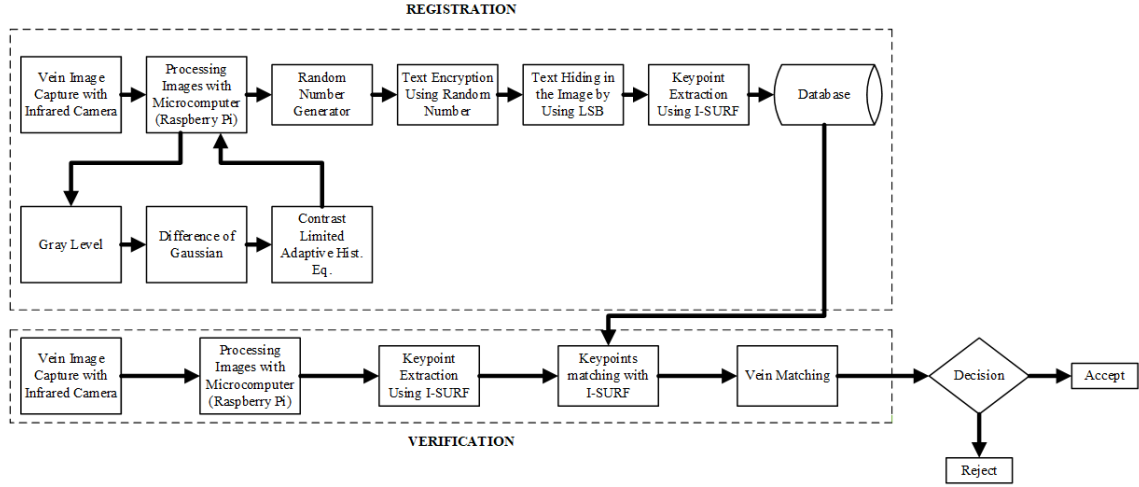


Figure 9: The general block diagram of security and authentication method

After the general explanation of the block diagram, details of the steps given in the diagram will be explained separately in the next sections. Also, after the detailed explanation of the diagram, an interface design will be introduced in Section 6 to show and facilitate the use of the work done.

3. Random Number Generator Design

Chaos-based random number generation method is given in Algorithm 1. According to the algorithm, the system parameters must be entered initially. Since two different systems are used, the values of 3 are entered when using Cubic Map and 20 when using Ricker's Population Model. Since a 1 Mb random number array is required for statistical tests, $1,000,000/16 = 62500$ steps are calculated, considering that 16-least significant bit is selected at each step. At each of the 62500 steps, float point number obtained from a chaotic system is converted to a 32-bit binary number. The 16-least significant bits of the binary number are then inserted into the random number array. Hereby, the 1-Mb random number array is created for both systems.

Random numbers obtained from the Cubic Map system are used in text encryption of a person data. Additionally, Random numbers obtained from the Ricker's Population Model system are used to determine coordinates of a dorsal hand vein image matrix overlaying the text data encrypted with the Cubic Map system. After the generation phase, statistical tests should be made for the performance of random numbers and their use in some applications. After the generation phase, statistical tests are performed to analyze the performance of random numbers and to use them in some applications.

Algorithm 1: Random Number Generator Algorithm Pseudo Code

Result: Ready to use 10000000 random number

Start

Entering system parameters \rightarrow (A=3 for Cubic Map, A=20 for Ricker's Population Model);

Entering initial condition \rightarrow ($x_0 = 0.1$)

Number of steps \rightarrow 62500

Constant \rightarrow 0

```
for  $i = 0$ :Number of steps do
    32bit_random=float to 32 bit binary (x(i))
    for  $j = 17 : 32$  do
        Constant = Constant+1
        random_number(Constant) = 32bit_random(j)
    end
end
End
```

3.1. Randomness Tests

NIST-800-22 and ENT tests are used for randomness performance of the numbers generated in this article, as they are internationally accepted and widely used. There are 16 different statistical tests in the NIST-800-22 test, which analyze the randomness of bit array. [32]. In order for the bit array subjected to the NIST-800-22 test to be successful, it must pass these tests successfully. In the NIST-800-22 test, the results are measured according to the P-value, which can be changed. For example, if the P-value of 0.001 is chosen as a condition, the P-value must be greater than 0.001 for a test to be successful.

Random numbers obtained from the two discrete-time chaotic systems used in the article successfully passed all NIST-800-22 tests and the results are given in Table 1. According to the table, the numbers that pass all tests are suitable for encryption and data hiding applications.

Table 1: NIST-800-22 Test Results

NIST-800-22 Tests	P-Value	
	Cubic Map	Ricker's Population Model
1) Frequency Monobit	0.6469	0.9664
2) Frequency Test within..	0.9229	0.2834
3) Run Test	0.1756	0.5511
4) Test for the longest	0.2538	0.0702
5) Binary Matrix Rank	0.9561	0.6217
6) Discrete Fourier Transform	0.5149	0.0706
7) Non overlapping	0.0032	0.0201
8) Overlapping Temp	0.0082	0.7046
9) Maurier's Universal	0.6904	0.4755
10) Linear Complexity	0.6322	0.5529
11) Serial Test	0.9292	0.1495
12) Approximate Entropy	0.0441	0.7468
13) Cumulative Sums(Forward) Test	0.1987	0.8852
14) Random Excursion Test	0.8418	0.9201
15) Random Excursion Variant Test	0.3074	0.8561

Another reliable test for randomness, the ENT test, is a test application developed by John Walker that applies various tests to byte arrays produced by pseudo-random number generator applications [33]. There are 5 different statistical tests in the ENT test that define randomness in a bit array. The mean values of the ENT test results of the random numbers obtained from the two systems are given in Table 2. In order for the test results to be successful, the arithmetic mean should be close to 128, the entropy should be close to 8, the optimum compression value should be close to 0, and the Monte Carlo π estimation should be close to 0. According to the table, it is seen that the random numbers passed all tests successfully.

Table 2: ENT Test Results

ENT Tests	Cubic Map	Ricker's Population Model
1) Arithmetic Mean	127.4999	127.2764
2) Entropy	7.9985	7.9986
3) Optimum Compression	0.00087466	-0.0015484
4) Chi-Square	263.5894	239.9391
5) Monte Carlo π Estimation	3.1652 (0.0075051)	3.1389 (0.000867)

3.2. Text Encryption Application

The pseudocode of chaos-based text encryption application is given in the Algorithm 2. In the algorithm, first all the characters of text to be encrypted are converted to double numbers according to their ASCII values. As a second step, random numbers with random_number variables obtained by using the Cubic system in the Algorithm 1 are obtained as 8-Bits. The 8-Bit array are converted from 8-Bit binary to decimal values. As the last step, the XOR operation is performed with the double values and the decimal values. The encryption process is completed when the values obtained after the XOR operation are converted into characters according to ASCII values.

Algorithm 2: Text encryption algorithm pseudo code

Result: Encrypted Text

Start

for $i = 1:length\ of\ text$ **do**

 double_text=char to double (text(i))

 decimal_random=Binary to decimal (random_number((i-1)*8+1: (i-1)*8+8))

 new_text (i) = bitxor(double_text, decimal_random)

end

encrypted_text = double to char (new_text)

End

In the interface application given as an example in Figure 10, after a user enters personal information, the information is received as a whole in text format. Encryption is performed with Algorithm 2. For example, the first character of text, the letter P, corresponds to the number 80 in ASCII characters code. The first 8-Bit of the random numbers obtained using the Cubic Map system in the Algorithm 1 corresponds to 11111110 and the decimal equivalent is 254. As seen in Algorithm 2, the XOR operation is performed with numbers of 254 and 80, and the result is 174. Since the ASCII equivalent of this value is the '@' character, the first character of the encrypted text becomes '@'.

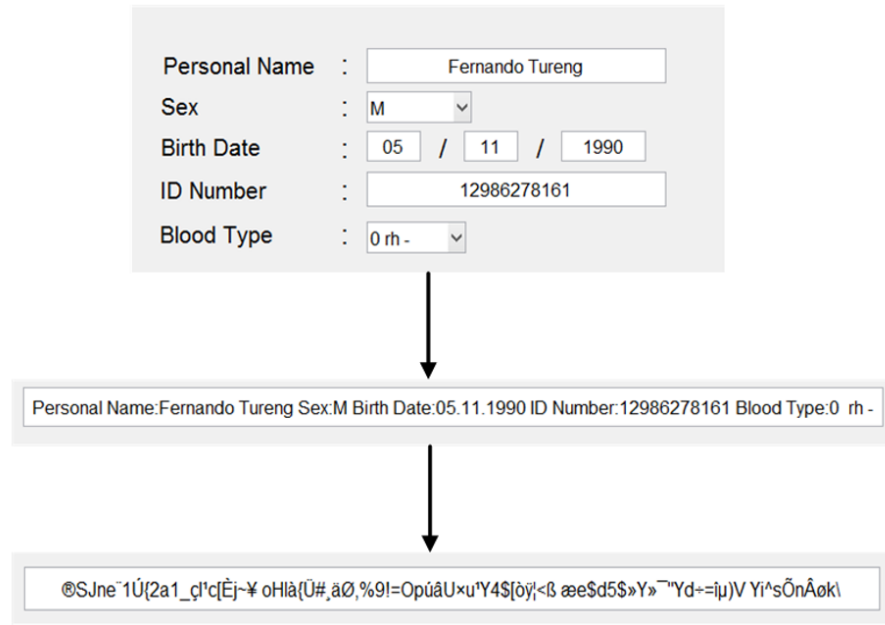


Figure 10: Text Encryption Interface Screen

4. Data Hiding Application

For two-stage security, the data encrypted in the article study is also subjected to data hiding. Dorsal vein images are used for data hiding. In order to hide the encrypted text data in a dorsal vein image, first, the image coordinates need to be determined. For this, random numbers obtained by using Ricker's Population Model system in Algorithm 1 are used to calculate rows and columns where the data will be hidden, as shown in Algorithm 3.

Algorithm 3: Coordinate determination algorithm pseudo code

Result: Ready to use line up the length of text coordinates of rows and columns

Start

[row column]=size(image)

series = 1: row* column

for $i = 1: \text{length of text} * 8$ **do**

 value=fix (mod (x (i) , (0.0065536-0.0000001* (i - 1))) * 1⁷)

 row_series (i) = ceil (series (value) / column)

 column_series (i) = mod(series (value) ,column)

 series(i) = delete

end

End

According to the Algorithm 3, firstly rows and columns of a person's dorsal vein image are obtained. An array with variable name *series* is obtained as numbers of elements of the image matrix. Values of the Ricker's Population Model system are subject to mod operation starting with the first step 0.0065535. When the result of the mode operation is divided by the number of the column and then the value obtained is rounded up, the row value for which the data will be hidden in the image is obtained. In addition, the remainder obtained after dividing the result of the mode operation by the number of the column is the column value for which the data will be hidden in the image. For example, the initial value of the Ricker's Population Model system is 0.1. When the value of 0.1 is substituted for $x(i)$ in the 5th line of Algorithm 3, the result becomes 16960. Since it is the first step, element 16960 corresponds to 16960 in the array. In a 256x256 dorsal vein image, this value is 67 using the 6th row of Algorithm 3. The value of the column is 64 using the 7th row. After these results, the first bit of the first value in the encrypted text is put into the least significant bit (LSB) of the pixel in the 67th row and 64th column of the dorsal vein image. Finally, in order to prevent repetitive numbers and to avoid losses in data extraction, the value 16960 is deleted from the array so that it is not selected again.

Algorithm 4: Data Hidden algorithm pseudo code

Result: Data is hidden
Start
[*row column*]=size(image)
for $i = 1:length\ of\ text * 8$ **do**
 value = image(*row_series*(i) , *column_series*(i))
 8bit = decimal to 8 bit binary (value)
 8bit(8)= encrypted_text(i)
end
End

Algorithm 4 was applied to hide the encrypted data after the Algorithm 3 stage was finished. In the Algorithm 4, the data is converted to 8-bit binary arrays. After the first bit of the array is placed in the LSB bit of the pixel in the 67th row and 64th column of the image, the next coordinate is calculated to place the second bit of the array. This process continues for 8 times the text length. Thus, the data encrypted with the random numbers obtained from the Cubic Map chaotic system is hidden in the image with the random numbers obtained from the Ricker's Population Model chaotic system.

4.1. Hiding Duplicated Data into Dorsal Vein Image

Data hidden inside a dorsal vein image may be lost due to data loss, noise, rotation, size reduction, etc. For this, it may be necessary to hide the chaotic encrypted data into the the image multiple times. Thus, although the data hiding speed decreases, data that cannot be obtained because of data loss and noise can be recovered. Accordingly, the loop in the Algorithm 3 is realized not by the number of bits of the encrypted data, but by multiplying this number of bits by a specified

Table 3: Impact of Data Duplication on Data Hiding

Data Size	SSIM	MSE	PSNR	NC	TIME (sn)
1	0.9999	0.0057	70.5552	0.9999	0.014
10	0.9999	0.0591	60.4174	0.9998	0.071
20	0.9998	0.1183	57.4132	0.9997	0.097
40	0.9997	0.2369	54.3856	0.9995	0.121
80	0.9995	0.4732	51.3807	0.9992	0.192

number. For example, in this study, the encrypted data is duplicated 10, 20, 40 and 80 times and then hidden inside the image. The number of characters of the encrypted data is 97 in the example given in Figure 10. If the encrypted data is hidden once, 776 different coordinates are required since each character consists of 8 bits. However, if the data is to be written 10 times, 7760 different coordinates must be found. From another point of view, the data can be hidden in different coordinates up to 84 times in a 256x256 image. Thus, the specified amount of data can be hidden in the image.

Because of duplicating and hiding the data, according to analysis results shown in Table 3, the Structural Similarity Index Measure (SSIM) value close to 1 shows that the two images are structurally similar. As a result of this analysis, it has been shown that there is not much change in the images after the data is hidden. In Mean Squared Error (MSE) analysis, if the MSE value is close to 0, it shows how many pixels have changed between the original and the modified image. As a result of the analysis, it has also been observed that almost half of the pixels of the image in which the data is hidden 80 times have changed.

$$MSE(I, I_0) = \frac{1}{M \times N} \times \sum_{y=1}^M \sum_{x=1}^N [I - I_0]^2 \quad (3)$$

Since the Peak Signal-to-Noise Ratio (PSNR) and the MSE are inversely proportional, the large differences between the obtained values show that the two images are more similar to each other. In the analysis results of Table 3, the difference between the two analysis values decreases as the amount of duplicated data increases, showing that the similarity between the two images decreases. In addition, when examining in terms of time, we observe that the processing time increases as the amount of duplicated data increases.

$$PSNR = 20 \times \log_{10} \left(\frac{255}{\sqrt{MSE(I, I_0)}} \right) \quad (4)$$

4.2. Extraction of Hidden Data From Deformed and Noisy Image

Encrypted data hidden in dorsal vein images may not be obtained, even though it is encrypted in the deformation and noise in the image. The reason for this is that there is a deformation in pixels where the data is hidden and, as a result, the data is lost. The lost data is difficult to recover because the deformed coordinates of the image are unknown. Therefore, by hiding the encrypted data in over 1 different coordinates, the probability of obtaining the data is increased. For example, a dorsal vein image in which encrypted data is hidden and 10% is deformed is shown in Figure 11. The data extracted from the image in this figure is shown in Figure 12

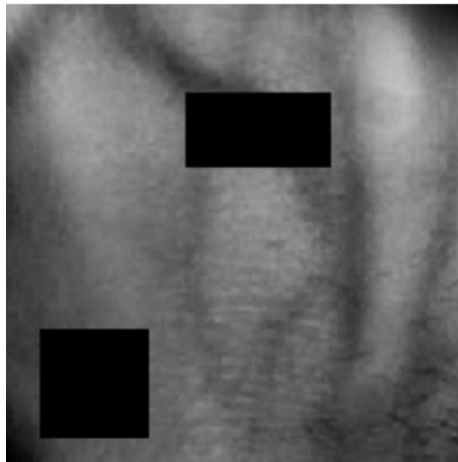


Figure 11: A Deformed Image

```

Dārso^al8Name:FESnant Twreng āx:l Bip h PAVEz05&11.1'94 ID ĩumj-r*!29862?81 9 Bloid @qpe:8 rh -
Perso.ai Ncmm°Fevnindo UubenÇ Sex*ĭ Viruh Da4E:05.91;3990 ID OUmby2°1298v27<369!Nlkĭd(Type: ĭrh(/
Rursonil&ĖamE:Fupnane (Turālg sex:M^Bistj Date: 5&11+19=0 IL ^wmb&3z3;;862780:1 Jloot Type:8`rh -
Đersonad NmMg:FEōnan0o Turung {eo>M Birt``Date:'5@11,19q2 ID!ĭ)mR`r 12{862781> Jjowt TYpu:0$rh -
Pev(onAm Nam!:Æernando T)ōeng Qep:MĭRir|h DĀte:05. 9.199° ID Number:52=86"78069 @lood,ōype*0 vi -
Perso`l Name: urnanā/0TeōEng Sex:M Birth'Da\m285.8!/1y93 I Numper:q2982278±6'`Blood(\{Pe:0 rh /
Personal NaoEzFernandk Tubejg Sex: iv|h F`0e:0p.01.1990` D Nuegar:1"986*68171 B|oo`"Ty0e: vh -
pErso&Ax Āmf:FEōNan`/2tUōeng Se|:E Birth Date:07.19.1990!ID Num`er:12986 ·8±61`BLkod Type:1 r-
Persooal Fkoe*fez*a~ln āureng Wul:M B)rqh Diōc:45.11.199$ AD Lum`er 12986278161 KLood0\ypm:0 rh`%
Pārsola|(Name8Feō&indo tuseng ex:I @kōth Daōe:27.11@19 "ID Nuober -298626:161 Ālgie(TI0D:1 rh -

```



```

Personal Name:Fernando Tureng Sex:M Birth Date:05.11.1990 ID Number:12986278161 Blood Type:0 rh -

```

Figure 12: Data Extracted From Deformed Image

In Figure 12, it is seen that the hidden data is not completely correct because 10% of the image is deformed. We have developed a different method to solve the challenges in the data obtained with some characters incorrectly. In this method, first, the length of the data is found, and it is determined how many times the encrypted data is hidden in the image. After this determination, the first 10 data in Figure 11 were obtained. In order to find the correct data, the characters of the 10 data obtained

are compared on an index basis. Here, the most identical character from the characters in each index is selected as the correct character. For example, since the P character is found 6 times in the first indexes of 10 different obtained data, the first character in the result data is chosen as the P character. After applying the same method for 97 characters, the encrypted data is successfully extracted.



Figure 13: 10% Salt-and-pepper Noisy Dorsal Vein Image

In Figure 13, there is a dorsal vein image where the encrypted data is hidden and exposed to 10% salt-and-pepper noise. The noise are made by randomly choosing coordinates of the image so that the hidden data is exposed to the noise. However, since the data is hidden multiple times, the data is still unaffected in some coordinates of the image. Figure 14 shows the data extracted from the image.

```
@erron`n Náoe FErêando TureHg Sex:M bir|h Date:15,11.1'90#ID0nulb%r:q09862781&l Blood *(Te80 rh(-
QezwOoal N#m *Fer.andO vufmnc$Sex:M"Rarth Datg: 5&!1.1992 ID NumbEr:1298 26<161 B,ood!T pe:2 rh -
PesSonAl Naau:Nurnando Tu2mng Sex: hrté!Datm;0u*19.q991 ID RumBeb(1290>278161 Rlocd yp%:4`^h m
Pessljal Name:Ernando(Puzehg mx:I BiRTh D!Te:"6fl!+990$H@!N5mbeR 52;86278!61 Blo d`T)pM:0 rh -
Pdrqoncl Name:Fernando Öuvejg Sex:M Rirth`Äatm:15. 1.1 9p ID0numr%R:1298627,#61`Bl od ype:0 bh -
PebqofAlName8Fernando TeR%ng Sep2M Birth`Dapm:04.11.1990 KD$NumbAr:±2;:62',16s$Blood Type:0brh`-
Pe2son`l!N!mgzFernand/$\Ureng Sex:I Birth(dátd:0%.11.1990 ID Num`eR:12'86r7<161 Bnugd0Pyrâ:0 r`!-
PErs~N`l Naim:Fgzfalfo"Ö}reng0Sax:U"Biôtj DAre:"µ. 3/1992 IL Nueber:12986²·(161 B,oYd Tyqm: rh -
PursOnal$na)e:Fe2~indo$üu"eno Sax:M Birth$Date:p5.11.1990 ID`Number: r98vrw8±61(Blood TyXe:0 rh$%
Pevconal N!mm FDsi!nd L5reng Sgø:M BirTh Fáte8 5.19.1=q0!IG0Fumb'p:1298625816! BlY/d T'0e 0`rh
```



```
Personal Name:Fernando Tureng Sex:M Birth Date:05.11.1990 ID Number:12986278161 Blood Type:0 rh -
```

Figure 14: Data Extracted From Salt-and-pepper Noisy Image

In order to extract the hidden data, we can also use the method used to extract the data from the deformed dorsal vein image in the noisy dorsal vein image. First, the length of the data is found, and it is determined how many times the encrypted data is hidden in the image. After this determination, the first 10 data in Figure 14 were obtained. In order to find the correct data, the characters of the 10

Table 4: Accurate Results of Data Extracted From Salt-and-Pepper Noisy and Deformed Images

Data Hiding Number	Salt-and-Pepper Noise			Deformed			
	%10	%30	%50	%5	%15	%30	%50
1 time	X	X	X	X	X	X	X
10 times	✓	X	X	✓	✓	X	X
20 times	✓	X	X	✓	✓	X	X
40 times	✓	✓	X	✓	✓	✓	X
80 times	✓	✓	✓	✓	✓	✓	✓

data obtained are compared on an index basis. Here, the most identical character from the characters in each index is selected as the correct character. For example, since the T character is found 3 times in the 24th indexes of 10 different obtained data, the 24th character in the result data is chosen as the T character. This process is also applied for 97 characters of the hidden data.

In Table 4, the correct acquisition results of the data extracted from the salt-and-pepper noisy and deformed vein image are given. When the table is analyzed, if the encrypted data is hidden in the image only once, correct data is not extracted from both the salt-and-pepper noisy image and the deformed image. However, when the data is placed in the image 80 times, correct data is extracted from all the images. In hiding 10, 20 and 40 times, correct data is obtained from some data extracted from the images, while false data is obtained from some of them. Of these three different data hiding numbers, the most successful one is 40 times, and with this data hiding number, correct data is obtained in all the images, except for 50% salt-and-pepper noisy images and 50% deformed images. When the data hiding numbers in the table are taken into consideration, it is observed that the resistance against attacks increases if this number increases.

5. Matching and Identification

For identification, it is very important to obtain correct results by matching input images with pictures previously registered in an identification database. There are many methods for matching images. In this study, the Improved Speeded Up Robust Features (I-SURF) method, which is an improved version of the Speeded Up Robust Features (SURF) method, is used. Figure 15 shows a matching of features extracted from two dorsal vein images by the method. The image on the right of this figure is a dorsal vein image hidden in a person's identity data and then saved in an identification database. The image on the left is a dorsal vein image of the same person, got at a different time with an imaging device for authentication. These two different images of the same person is matched at 24 points.

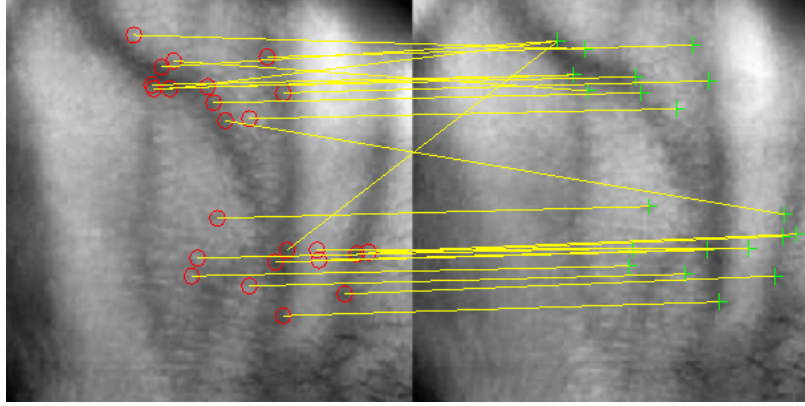


Figure 15: Example of Matching Two Dorsal Vein Images Using The SURF Method

5.1. Vein Region Matching Using SURF Method

When matching between two dorsal vein images, it can be better to prefer only at features extracted with SURF in vein patterns in order to obtain more accurate results and to gain speed before improvement. Algorithm for the SURF matching of two dorsal vein images is given in Algorithm 5.

Algorithm 5: Data Hidden algorithm pseudo code

Result: The vein region is obtained with SURF

Start

points1 = Detect SURF Features (image1) & points2 = Detect SURF Features (image2)

f1 = Extract SURF Features (image1 , points1) & f2 = Extract SURF Features (image2 , points2)

indexPairs = Match SURF Features (f1 , f2)

for $i = 1:length(indexPairs)$ **do**

if $indexPairs(i,1) > 50 \text{ — } indexPairs(i,2) > 50$ **then**

$indexPairs(i,:) = \text{delete}$

end

end

End

According to Algorithm 5, first, a dorsal vein image is obtained from a personal with an imaging device and the SURF features of this image are detected. Then, SURF features of dorsal vein images, in which the identity information of people previously registered in an identification database are encrypted, are detected. After the features are detected, the features are extracted according to the ‘f’ parameters. The extracted features are matched between the two images and only the features on the vein pattern are obtained from those matches.

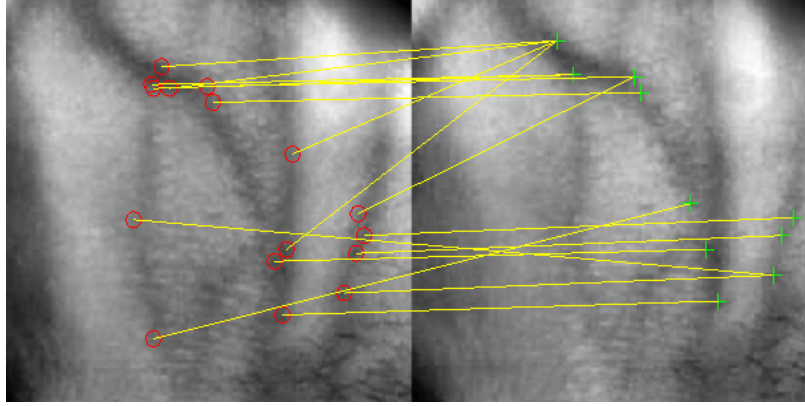


Figure 16: SURF Detection and Matching Process on Vein Pattern

In Figure 16, there are only SURF matches on the vein pattern after the SURF matching. While the SURF matching of all regions of the two images obtain 24 features, this number decreases to 16 in SURF matching within vascular regions. Thus, accuracy rate in matching between two images is increased.

5.2. Improved SURF

After features of vein regions on vein images are extracted with SURF, the features are matched. However, some of their matches are incorrect. In order to eliminate these mismatches, improvements are required during the matching process of the SURF method. Therefore, Improved SURF (I-SURF) algorithm is given in Algorithm 6 to improve the matching process of the SURF method.

Algorithm 6: Improving Matching Process

Result: Matching of the features obtained from the vein regions is done by improving

Start

```
for  $i = 1:\text{length}(\text{indexPairs})$  do
    [row1 column1] =  $\text{indexPairs}(i,1)$  Location & [row2 column2] =  $\text{indexPairs}(i,2)$ 
    Location
    Length of distance =  $\text{sqrt}((\text{row2} - \text{row1})^2 + (\text{column2} - \text{column1})^2)$ 
    Inclination angle =  $\text{arctan}((\text{column2} - \text{column1}) / (\text{row2} - \text{row1}))$ 
    if Length of distance < median(Length of distance) - 30 | Length of distance >
        median(Length of distance) + 30 then
        |  $\text{indexPairs}(i,:) = \text{delete}$ 
    end
    if Inclination angle < median(Inclination angle) - 3 | Inclination angle > median
        (Inclination angle) + 3 then
        |  $\text{indexPairs}(i,:) = \text{delete}$ 
    end
end
End
```

In the Algorithm 6, the coordinates of the indexPairs from Algorithm 5 are calculated. Then, match lengths and gradients are calculated according to the coordinates of the matches. Here, the match lengths and the gradients are averaged first, and then matches that are far from these averages are eliminated.

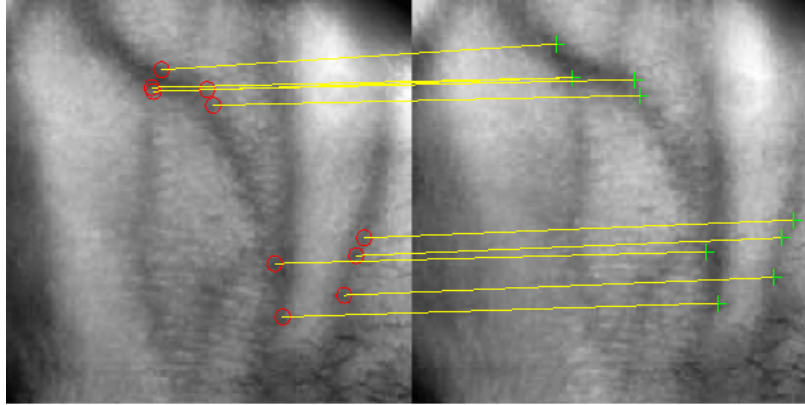


Figure 17: Matching after I-SURF

In Figure 17, the matching of dorsal vein images from features only in the vein regions after I-SURF algorithm is shown. While there are 16 different matches only from the vein regions of the images, when the I-SURF algorithm is applied here, 6 of these matches are found to be incorrect. Thus, after incorrect matches are eliminated, the most matching image is detected and identity of the person is determined.

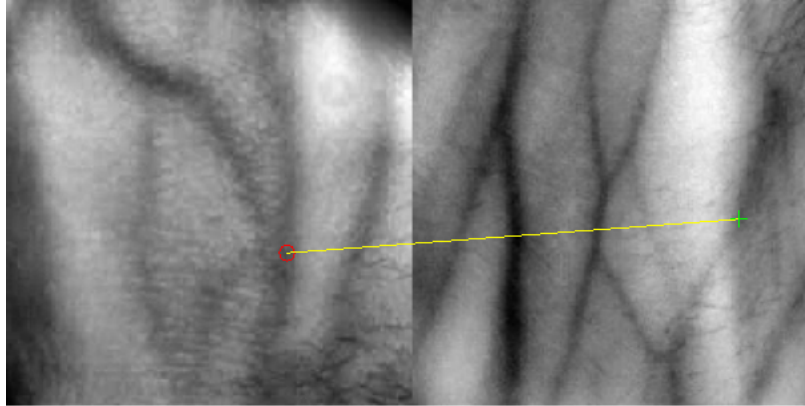


Figure 18: Matching Two Dorsal Vein Images Taken from Different People With I-SURF Algorithm

In Figure 18, the matching of two dorsal vein images of different people with I-SURF is given. These two images have 14 matches with classic SURF. If the match is limited to vein regions only, the number of matches will be 3. In addition, if the I-SURF algorithm is applied, the number of matches will be only 1. As can be seen, the number of matches for two images belonging to the same person in Figure 17 is 10, while the number of matches for two images belonging to a different person in Figure 17 is 1. Thus, more efficient results are obtained when the I-SURF algorithm is used in the matching phase of the features extracted from the images.


6. Interface Study

In this section, the interface realized in the article study is introduced. The interface is shown in Figure 19. In the first stage of using this interface, a person's vein image is taken by clicking the 'Take Dorsal Hand Vein Image' button. Then, the person enters their personal information in the upper middle part of the interface. By clicking the 'Get Text' button under the part, the information is displayed as a single line of text just below the button. When 'Encrypt Text' button is clicked, the text data is encrypted. In order to specify how many times the encrypted data will be written into the image, a positive integer number is entered into the 'Number of Data' text field in the upper right part of the interface. Then 'Hide Data' button just below the text field is clicked, and the encrypted data is hidden in the image by the number specified in the 'Number of Data' field. Finally, name of the image in which the data is hidden is requested from the person and the image is saved as a jpg extension.

Hiding_Data

Take Dorsal Hand Vein Image

Received Dorsal Hand Vein Image



Personal Name : Fernando Tureng

Sex : M

Birth Date : 05 / 11 / 1990

ID Number : 12986278161

Blood Type : 0 rh -

Get Text

Personal Name:Fernando Tureng Sex:M Birth Date:05.11.1990 ID Number:12986278161 Blood Type:0

Encrypt Text

©SJne 1Ú[2a1_cl'c]Êj~¥ qHlâ[Ü#_aØ,%9I=OpúâU×u"Y4S[oy](<ß æe\$ð5\$»Y» ~Yd--=ijj)V YI^sÔnÅok\

Encryption time = 0.004 sn


Data hiding time = 0.188 sn

Total Time = 0.192 sn

Number of Data = 80

Hide Data

Data Hidden Image



Name of the Image to be Saved

1

Save Image

Figure 19: Data Hiding GUI

The identity matching interface is shown in Figure 20. By clicking ‘Take Dorsal Hand Vein Image’ button in the upper left part of the interface, the person’s current dorsal vein image is taken. The image is extracted with SURF to match the database images. The image is then matched with the images in the database with the proposed I-SURF. Finally, the matching image in the database is displayed, and the identity information of the person is obtained after decrypting the encrypted data hidden in the image.

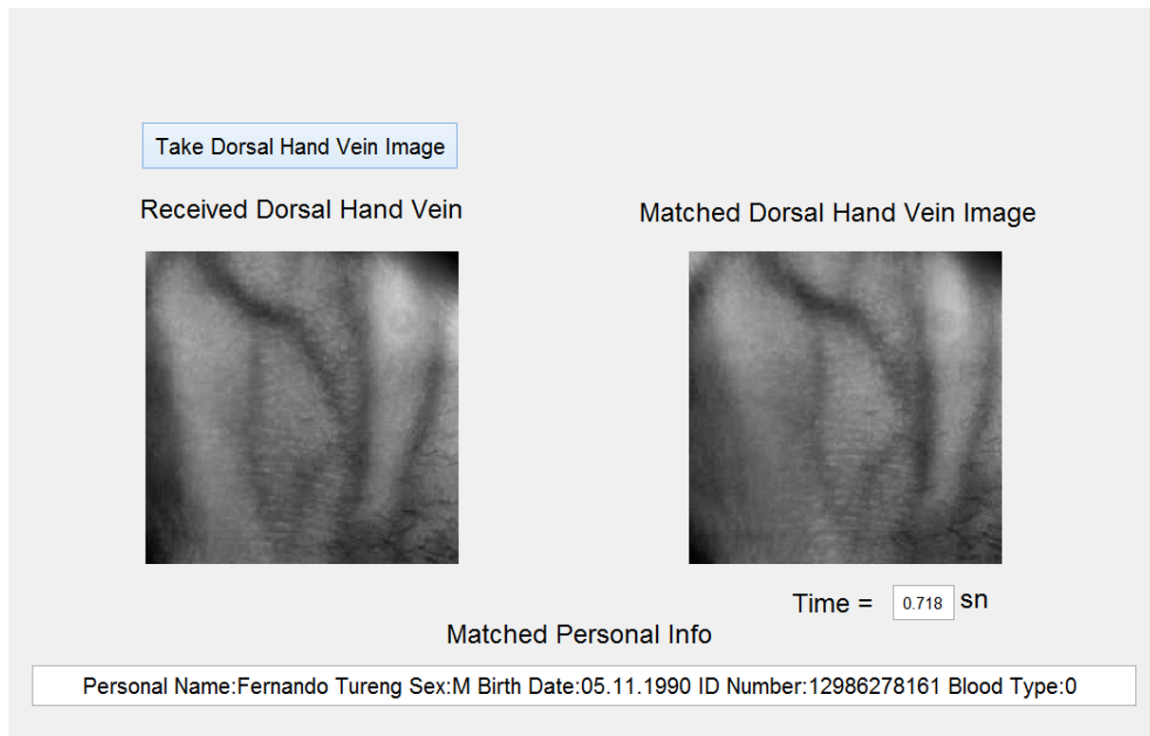


Figure 20: Identity Matching GUI

7. Conclusion and Future Works

In this article study, a new dorsal vein image identification application is introduced that provides personal data security by using two different discrete time chaotic systems and matches features extracted from images with higher performance using the I-SURF method. In the first phase of the study, registration process is explained. In the process, first, the identity information of people whose dorsal vein images are taken is encrypted using Cubic Map chaotic system. Then, the encrypted data of the people are hidden as a binary value in the LSB of pixels selected from their dorsal vein images with the Ricker's Population Model chaotic system, and finally the image is saved in an identification database with its extracted features by SURF method. Thus, personal data is protected as a double layer during registration to the database. Afterwards, the identification and matching process is explained in the article. At the beginning of this process, a vein image taken from a person for identification is extracted with SURF method. The extracted features are matched with features of pictures in the database with the proposed I-SURF method, and thus the registered picture of the same person in the database is found.

In the study, some experiments are carried out to test the performance of the application. As a first performance test in the article, Cubic Map and Ricker's Population Model chaotic systems are subjected to NIST-800-22 and ENT tests. Thus, it is determined that chaotic systems are reliable. Then, differences of two same dorsal vein images, one of which has data encrypted to its pixels and the other one being original, are compared according to some criteria and it is determined that there is not a big difference between the two images. Finally, correct extraction of encrypted data from dorsal vein images that have corrupted and encrypted data in their pixels is tested. In this test, it is observed that as the number of hiding data in dorsal vein image pixels increases, the rate of extracting correctly hidden the data increases.

In future studies, it is planned to design and use different chaotic systems or methods to increase the security of personal data. In addition, different techniques can be considered for data hiding operations. Also, to further increase the performance of dorsal vein image matching, more effective results can be obtained by supporting it with artificial intelligence.

Acknowledgments

Dr. Akif Akgul acknowledges financial support by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant No. 120E318.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Availability of data and material

500 healthy adults (260 males, 240 females, age range: 18–55) participated in the study (P1-P500). First subjects were asked to hand placement their right and left hands at fist position under the infrared camera on a white surface during approximately 3s. The procedures posed no harm for the participants. All participants gave written consents and the experiments were approved by the Institutional Review Board for Research with Human Subjects of Sakarya University (no.71522473/050.01.04/86).

References

- [1] Cihui Xie and Ajay Kumar. Finger vein identification using convolutional neural network and supervised discrete hashing. In *Deep Learning for Biometrics*, pages 109–132. Springer, 2017.
- [2] Toshiyuki Tanaka and Naohiko Kubo. Biometric authentication by hand vein patterns. In *SICE 2004 Annual Conference*, volume 1, pages 249–253. IEEE, 2004.
- [3] Jun Wang, Guoqing Wang, Ming Li, and Wenkai Du. Hand vein recognition based on pcet. *Optik*, 127(19):7663–7669, 2016.
- [4] ZH Liu, Jun Yin, and Zhong Jin. An adaptive feature and weight selection method based on gabor image for face recognition. *Acta Photonica Sinica*, 40(4):636–641, 2011.
- [5] Ömer Faruk Boyraz, Muhammed Ali Pala, Murat Erhan Çimen, Ali Fuat Boz, and Mustafa Zahid Yıldız. Mikrobilgisayar tabanlı el-bilek damar örüntüleri kullanılarak biyometrik kimlik doğrulama işleminin yapılması. *Academic Perspective Procedia*, 2(3):593–600, 2019.
- [6] Wonseok Song, Taejeong Kim, Hee Chan Kim, Joon Hwan Choi, Hyoun-Joong Kong, and Seung-Rae Lee. A finger-vein verification system using mean curvature. *Pattern Recognition Letters*, 32(11):1541–1547, 2011.
- [7] WenXiong Kang, HuaSong Li, and FeiQi Deng. Direct gray-scale extraction of topographic features for vein recognition. *Science China Information Sciences*, 53(10):2062–2074, 2010.
- [8] Wei-Yu Han and Jen-Chun Lee. Palm vein recognition using adaptive gabor filter. *Expert Systems with Applications*, 39(18):13225–13234, 2012.
- [9] Yakun Zhang, Weijun Li, Liping Zhang, Xin Ning, Linjun Sun, and Yaxuan Lu. Adaptive learning gabor filter for finger-vein recognition. *IEEE Access*, 7:159821–159830, 2019.
- [10] Jing Zhang and Jinfeng Yang. Finger-vein image enhancement based on combination of gray-level grouping and circular gabor filter. In *2009 International Conference on Information Engineering and Computer Science*, pages 1–4. IEEE, 2009.

- [11] Jinfeng Yang and Jinli Yang. Multi-channel gabor filter design for finger-vein image enhancement. In *2009 Fifth International Conference on Image and Graphics*, pages 87–91. IEEE, 2009.
- [12] Wenxiong Kang. Vein pattern extraction based on vectorgrams of maximal intra-neighbor difference. *Pattern Recognition Letters*, 33(14):1916–1923, 2012.
- [13] Eui Chul Lee, Hyeon Chang Lee, and Kang Ryoung Park. Finger vein recognition using minutia-based alignment and local binary pattern-based feature extraction. *International Journal of Imaging Systems and Technology*, 19(3):179–186, 2009.
- [14] Leila Mirmohamadsadeghi and Andrzej Drygajlo. Palm vein recognition with local binary patterns and local derivative patterns. In *2011 International Joint Conference on Biometrics (IJCB)*, pages 1–6. IEEE, 2011.
- [15] Bakhtiar Affendi Rosdi, Chai Wuh Shing, and Shahrel Azmin Suandi. Finger vein recognition using local line binary pattern. *Sensors*, 11(12):11357–11371, 2011.
- [16] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *International workshop on analysis and modeling of faces and gestures*, pages 168–182. Springer, 2007.
- [17] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.
- [18] Herbert Bay. *From wide-baseline point and line correspondences to 3D*. PhD thesis, ETH Zurich, 2006.
- [19] Francisco J Escribano, Alexandre Wagemakers, and Miguel AF Sanjuán. Chaos-based turbo systems in fading channels. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(2):530–541, 2013.
- [20] Yicong Zhou, Long Bao, and CL Philip Chen. A new 1d chaotic system for image encryption. *Signal processing*, 97:172–182, 2014.
- [21] Dhivya Ravichandran, Padmapriya Praveenkumar, John Bosco Balaguru Rayappan, and Rengarajan Amirtharajan. Dna chaos blend to secure medical privacy. *IEEE transactions on nanobioscience*, 16(8):850–858, 2017.
- [22] K Shankar, Mohamed Elhoseny, E Dhiravida Chelvi, SK Lakshmanaprabu, and Wanqing Wu. An efficient optimal key based chaos function for medical image security. *IEEE Access*, 6:77145–77154, 2018.

- [23] Xian Hui Mai, Bo Zhang, Xiao Shu Luo, et al. Controlling chaos in complex motor networks by environment. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 62(6):603–607, 2015.
- [24] Hui Xu, Xiaojun Tong, and Xianwen Meng. An efficient chaos pseudo-random number generator applied to video encryption. *Optik*, 127(20):9305–9319, 2016.
- [25] Sergio Callegari, Riccardo Rovatti, and Gianluca Setti. Embeddable adc-based true random number generator for cryptographic applications exploiting nonlinear signal processing and chaos. *IEEE transactions on signal processing*, 53(2):793–805, 2005.
- [26] Toni Stojanovski, Johnny Pihl, and Ljupco Kocarev. Chaos-based random number generators. part ii: practical realization. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(3):382–385, 2001.
- [27] Peyman Ayubi, Saeed Setayeshi, and Amir Masoud Rahmani. Deterministic chaos game: a new fractal based pseudo-random number generator and its cryptographic application. *Journal of Information Security and Applications*, 52:102472, 2020.
- [28] Octaviana Datcu, Corina Macovei, and Radu Hobincu. Chaos based cryptographic pseudo-random number generator template with dynamic state change. *Applied Sciences*, 10(2):451, 2020.
- [29] Abdullah Qayyum, Jawad Ahmad, Wadii Boulila, Saeed Rubaiee, Fawad Masood, Fawad Khan, William J Buchanan, et al. Chaos-based confusion and diffusion of image pixels using dynamic substitution. *IEEE Access*, 8:140876–140895, 2020.
- [30] Teh-Lu Liao, Pei-Yen Wan, and Jun-Juh Yan. Design of synchronized large-scale chaos random number generators and its application to secure communication. *Applied Sciences*, 9(1):185, 2019.
- [31] Murat Tuna and CAN FIDAN. A study on the importance of chaotic oscillators based on fpga for true random number generating (trng) and chaotic systems. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 33(2):473–492, 2018.
- [32] Ismail Koyuncu. *Kriptolojik uygulamalar için FPGA tabanlı yeni kaotik osilatörlerin ve gerçek rasgele sayı üreteçlerinin tasarımı ve gerçekleştirilmesi*. PhD thesis, Sakarya University, Fen Bilimleri Enstitüsü, 2014.
- [33] Pseudorandom number sequence test program, Jan 2008. <https://www.fourmilab.ch/random/>, visited 2022-01-14.