

**Short thesis for the degree of doctor of philosophy  
(PhD)**

**Developing a Biometric Based  
Authentication to Secure Transferring Data  
via Cloud Environment**

**by Kaabi Jasim Hussein Lafta**

**Supervisor: Dr. Tornai Róbert**



**UNIVERSITY OF DEBRECEN  
Doctoral School of Informatics**

**Debrecen, 2025**



# Contents

|                                                                      |          |
|----------------------------------------------------------------------|----------|
| <b>Introduction</b>                                                  | <b>1</b> |
| <b>1 Theses</b>                                                      | <b>4</b> |
| 1.1 Authentication and Biometric Contributions . . . . .             | 4        |
| 1.2 Cryptographic Contributions . . . . .                            | 5        |
| 1.2.1 Biometric-Based AES Key Generation . . . . .                   | 5        |
| 1.2.2 RSA Public Key Generation and Modified<br>RSA (MRSA) . . . . . | 5        |
| 1.3 Performance Overview . . . . .                                   | 6        |
| <b>2 Results</b>                                                     | <b>7</b> |
| 2.1 Case study 1 . . . . .                                           | 7        |
| 2.1.1 Preprocessing the Eyes . . . . .                               | 7        |
| 2.1.2 Extract the Features . . . . .                                 | 7        |
| 2.1.3 Authentication Stage . . . . .                                 | 10       |
| 2.1.4 Confidentiality Stage . . . . .                                | 11       |
| First step . . . . .                                                 | 11       |
| Second step . . . . .                                                | 13       |
| 2.1.5 Evaluate and test the AES key encryption . .                   | 16       |
| Frequency Test . . . . .                                             | 16       |
| Frequency Mono Bit Test . . . . .                                    | 17       |
| Run Test . . . . .                                                   | 18       |
| Entropy . . . . .                                                    | 19       |
| 2.2 Case study 2 . . . . .                                           | 19       |
| 2.2.1 Preprocessing the Eyes . . . . .                               | 19       |
| 2.2.2 Extract the Features . . . . .                                 | 20       |
| 2.2.3 Authentication Stage . . . . .                                 | 20       |
| 2.2.4 Confidentiality Stage . . . . .                                | 21       |
| 2.2.5 Evaluate and test the AES key encryption . .                   | 22       |
| Frequency Test . . . . .                                             | 22       |

|                                   |           |
|-----------------------------------|-----------|
| Frequency Mono Bit Test . . . . . | 23        |
| Run Test . . . . .                | 23        |
| Entropy . . . . .                 | 24        |
| <b>Publications</b>               | <b>25</b> |

# Introduction

Cloud internet technology has emerged to provide various electronic computing services. According to researchers, there are base risks anytime users log onto a computer and access the internet, but these risks are more significant when computing in the cloud. Security algorithms for advanced encryption and decryption can be developed to enhance security over cloud networks. Some cryptography techniques mainly depend on prime numbers generated using weak algorithms such as an RNG (Random Number Generator) and SRNG (Pseudorandom Number Generator). These algorithms first generate random numbers, then primes are selected from these numbers after some tests.

The main goal of this dissertation is to enhance the two stages of information security in cloud computing: confidentiality and authentication. Confidentiality can be achieved through two processes. First, an AES encryption key is generated from the iris biometric to encrypt the data using a B-spline curve. Second, Bézier curve interpolation is used to reduce periodicity to develop the most significant prime number as a powerful key used in the modified RSA algorithm called the Mersenne RSA (MRSA) algorithm to encrypt the key of the AES algorithm. From an authentication point of view, we can do that by features of the iris that will be considered a control point of a Bézier-spline as unique user identification.

The proposed approach extracts the features from the iris by applying Zernike moments as a good feature representation. These features are used as control points to draw the curve of a B-spline and to generate a 128-bit AES encryption key, which is used to encrypt the data before sending or storing it in the cloud. Also, these control points are used in the B-spline interpolation to obtain prime numbers that will be expanded by using Mersenne

prime numbers. The most enormous prime numbers are used to create the public and private keys in the modified RSA algorithm (MRSA). The MRSA algorithm is used to protect the AES private key.

In addition, these features are used as an authentication tool. Authentication is an additional step to enhance the security in this dissertation. Thus, this dissertation proposes two encryption techniques, asymmetric and symmetric keys with RSA and AES algorithms, to improve data security in cloud computing.

The proposed system can obtain the most significant prime number, 29923 digits (2988 digits), then convert each digit of the prime number produced from Mersenne prime numbers to a 4-bit binary number. The final key will be 11952 bits, a prime number used to generate the encryption and decryption keys in the modified RSA (MRSA) algorithm.

The proposed approach is robust against iris rotation and noise. The iris datasets used to examine its performance are from the Biometric Recognition MMUD and CASIA Iris datasets.

## Research Goals

The main objective of this study is to enhance the security of cloud computing by:

1. Using biometrics to generate a symmetric key for the AES algorithm by using the Zernike moment to extract the features from biometrics. The AES algorithm is one of the algorithms used in cloud computing.
2. Using biometric and Bézier curves to generate a large prime number through applying Mersenne prime numbers to use in the RSA algorithm, which is used in cloud computing.
3. Using control points of B-spline to authenticate the user in cloud environment through extracting the eight features from biometric.

Table 1: Comparison: Proposed System With The Related Works

| Extract Features from Iris using Zernike Moments or CNN                                                                                                                                                                      |                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Reorder These Features to Produce Control Points of B-spline and Bézier                                                                                                                                                      |                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Generate Encryption Key                                                                                                                                                                                                      |                                                                                                                                                                                                                                         | Authentication                                                                                                                                                                                                                                                                                                                                                                                                                       |
| AES                                                                                                                                                                                                                          | RSA                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Using the control points of the B-spline to obtain 128 bits. This approach increases the randomness of the bit sequence for AES encryption keys. This key is then used to encrypt data before sending it to cloud computing. | Using the Bézier curve points to obtain the first prime number $p$ , then applying Mersenne prime numbers to get the second prime number $M_q$ . This approach produces a large prime number and generates a highly secure private key. | <ul style="list-style-type: none"> <li>- Reordering eight features into the control points of B-spline and Bézier curves produces highly congruent iris accuracy.</li> <li>- CNN is applied for individual eye recognition, which is used in security systems for person identification. The method has been tested on different images and provides the highest accuracy for image recognition used in security systems.</li> </ul> |

# 1 Theses

This chapter presents the core contributions of this dissertation, which introduces a hybrid security framework for cloud computing that combines biometric authentication and advanced cryptographic techniques. The system utilizes iris recognition for feature extraction via Convolutional Neural Networks (CNNs) and Zernike moments and applies B-spline and Bézier curve interpolation to generate control points. These are then used to create secure encryption keys for symmetric (AES-128) and asymmetric (RSA) encryption.

## 1.1 Authentication and Biometric Contributions

1. Developed a novel cloud authentication framework based on iris biometrics, leveraging CNNs or compact control points derived from Zernike moments and B-spline interpolation [P2, C1]. This reduced computational overhead by storing only four control points per user (Section 3.2.4 of my theses).
2. Enhanced iris feature extraction by employing rotation-invariant Zernike moments, improving robustness and accuracy in authentication scenarios [P4].
3. Applied B-spline and Bézier curve techniques to generate high-entropy, non-periodic control points, improving both the randomness and compactness of biometric representations [P6, C1].

## 1.2 Cryptographic Contributions

The techniques and findings presented in this section are thoroughly discussed in Section 3.2.5 of my theses.

### 1.2.1 Biometric-Based AES Key Generation

1. Proposed a secure method for AES-128 key derivation from biometric data using B-spline and Bézier curve control points, ensuring key invariance under iris rotation due to Zernike moment properties [P1, P4].
2. Strengthened AES key security by validating the generated keys through statistical randomness and entropy analysis, confirming resistance to brute-force attacks [P6].
3. Secured AES key transmission by integrating RSA encryption, mitigating key exchange vulnerabilities [C1].

### 1.2.2 RSA Public Key Generation and Modified RSA (MRSA)

1. Designed a novel RSA key generation approach where the first prime ( $p$ ) is derived from B-spline-based iris features, and the second ( $q$ ) from a Mersenne prime formula, enabling strong and biometric-tied public keys (Section 3.2.5 of my theses).
2. Introduced a Modified RSA (MRSA) scheme capable of handling extremely large primes (up to 2988 digits), enhancing asymmetric cryptographic strength (Section 3.2.5 of my theses).
3. Demonstrated a comprehensive hybrid encryption model combining symmetric (AES-128) and asymmetric (RSA) cryptographic methods for secure cloud-based data transmission [C1].

## 1.3 Performance Overview

The proposed system demonstrates the ability to handle very large prime numbers, supporting key lengths of up to 2988 digits for secure encryption. A noted limitation is the increased execution time required to encrypt AES-128 keys using the RSA algorithm, primarily due to the high computational cost of large prime generation and exponentiation. Despite this, the trade-off between security and performance is justified by the significant increase in key strength and randomness achieved through the combined use of B-spline, Bézier curves, and Mersenne primes.

## 2 Results

### 2.1 Case study 1

The first case in which the proposed system will be applied is an iris image from the MMU iris database.

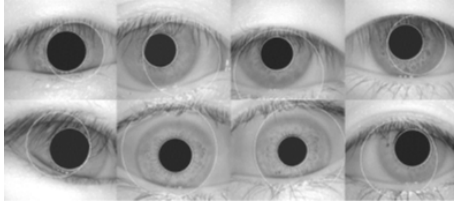
- Iris size.
- Type of file.
- AES Encryption Key, size (128 bits) to encrypt the data.
- RSA Encryption Key, size (variable) to encrypt the AES Encryption Key.

#### 2.1.1 Preprocessing the Eyes

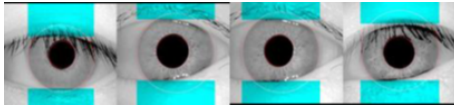
The first step after entering the iris image into the proposed system is to preprocess the iris by determining the locations of the iris and pupil, then to localize the partial iris pattern. After this, the iris is divided into  $3 \times 3$  subregions where the pupil is centered. Figure 2.1a illustrates the iris and pupil localization. Figure 2.1b illustrates the partial iris pattern determination process. The final step is dividing the iris into  $3 \times 3$  subregions where the pupil is centered; Figure 2.1c illustrates this step.

#### 2.1.2 Extract the Features

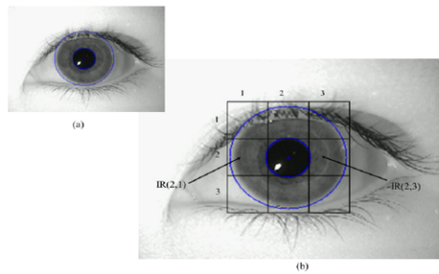
After preprocessing the iris, the system can extract the features from this enhanced image by applying the Zernike moments. Zernike moments can detect the fine details of objects in the image. Table 2.1 illustrates this set of polynomials.



**(a) Iris And Pupil Localization**



**(b) The Partial Iris Pattern**



**(c) The Iris Region is Divided into Nine Subregions**

Figure 2.1: Preprocessing Of Iris.

Table 2.1: List Of The 8 First Orders Zernike Moments

| n | m | Zernike moments |
|---|---|-----------------|
| 1 | 1 | $A_{1,1}$       |
| 2 | 0 | $A_{2,0}$       |
| 2 | 2 | $A_{2,2}$       |
| 3 | 1 | $A_{3,1}$       |
| 3 | 3 | $A_{3,3}$       |
| 4 | 0 | $A_{4,0}$       |
| 4 | 2 | $A_{4,2}$       |
| 4 | 4 | $A_{4,4}$       |

Table 2.2: Eight Features And Control Points

| Features | Value of Features | Control Points       |
|----------|-------------------|----------------------|
| $Z_1$    | 1.017072          | (1.017072, 1.653543) |
| $Z_2$    | 1.653412          | (1.653412, 2.393891) |
| $Z_3$    | 1.744543          | (1.744543, 3.255656) |
| $Z_4$    | 1.847913          | (1.847913, 8.708148) |
| $Z_5$    | 1.653543          | -                    |
| $Z_6$    | 2.393891          | -                    |
| $Z_7$    | 3.255656          | -                    |
| $Z_8$    | 8.708148          | -                    |

Moreover, Zernike moments are susceptible to noise. However, they are also a very efficient method for image feature extraction and are unaffected by image rotation. The order ( $p$ ) and repetition ( $q$ ) must be determined to generate the set of polynomials to obtain eight features using Zernike moments. Experience shows that  $p = 3$  and  $q = 5$  yield the best features for the proposed system.

This set of polynomials provides the eight features, which are then reordered to yield four points that will serve as the B-spline's control points. Table 2.2 illustrates these features and control points.

Table 2.3: Congruence Control Points With Database

|          |          |          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|----------|----------|
| 1.017072 | 1.653412 | 1.744543 | 1.847913 | 1.653543 | 2.393891 | 3.255656 | 8.708148 |
| 12.59776 | 18.73098 | 19.56652 | 32.7656  | 32.76805 | 26.37178 | 65.11989 | 20.04399 |
| 11.8573  | 18.8453  | 20.6162  | 21.98433 | 20.61446 | 14.4648  | 27.03797 | 10.26192 |
| 14.7314  | 44.70006 | 55.3483  | 62.42363 | 14.73702 | 93.8137  | 34.9924  | 17.2721  |
| 31.31452 | 34.64823 | 48.8476  | 76.49455 | 34.64321 | 10.44282 | 37.3523  | 14.0025  |
| ⋮        | ⋮        | ⋮        | ⋮        | ⋮        | ⋮        | ⋮        | ⋮        |

### 2.1.3 Authentication Stage

The proposed system will use the control points for authentication and confidentiality. The control points are checked for congruence with the database features to process authentication. Table 2.3 illustrates this congruence.

The authentication process begins by taking the first control point of the iris features and matching it with the first control points of all iris feature records in the database. This process is repeated for the remaining control points to verify the user's identity. If the features match any row in the database, the system authorizes the user to access cloud computing services. This approach represents the initial method in this dissertation to enhance the level of security in cloud computing.

Table 2.3 illustrates that the above process is done on the same features of the iris, and it will extract four values for the left side and four others for the right side of the same iris using ZM.

Using Zernike Moments (ZM) produces fractional numbers, which represent features for this part of the iris with defined parameters. Each execution yields these features, and we repeat this process eight times.

|          |          |          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|----------|----------|
| $Z_1$    | $Z_2$    | $Z_3$    | $Z_4$    | $Z_5$    | $Z_6$    | $Z_7$    | $Z_8$    |
| 1.653412 | 1.744543 | 1.017072 | 1.847913 | 1.653543 | 2.393891 | 3.255656 | 8.708148 |

We multiply these fractional numbers by 100,000 to increase their magnitudes, converting them to integer values by removing the decimal portion. Each resulting value is then restricted to four digits.

|       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|
| $Z_1$ | $Z_2$ | $Z_3$ | $Z_4$ | $Z_5$ | $Z_6$ | $Z_7$ | $Z_8$ |
| 1653  | 1744  | 1017  | 1847  | 1653  | 2393  | 3255  | 8708  |

Next, we arrange the first four numbers in ascending order to represent the x-axis, while the remaining four numbers are used as the y-axis coordinates.

| $Z_1$ | $Z_2$ | $Z_3$ | $Z_4$ | $Z_5$ | $Z_6$ | $Z_7$ | $Z_8$ |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 1017  | 1653  | 1744  | 1847  | 1653  | 2393  | 3255  | 8708  |

These four points, consisting of an ordered pair of numbers in each of them, are used as control points to draw the curve of the B-spline and Bézier-spline interpolator methods. Each pair of these values will represent data for the control point  $K(x_i, y_i) \mid i = 1 : 4$  shown in Table 4.4, on the curve to be approximated by the interpolators used with B-spline  $[(Z_1, Z_5), (Z_2, Z_6), (Z_3, Z_7), (Z_4, Z_8)]$ .

Each pair of these values represents one of four points that will be interpolated by the B-spline as follows:  $(1017, 1653), (1653, 2393), (1744, 3255), (1847, 8708)$ .

The result of applying the B-spline interpolator for  $(F_1, F_2, F_3, F_4)$  is illustrated in Figure 2.2. Through the drawing, we can find the final values used to extract the keys.

Therefore, after completing the authentication stage, the proposed system will use the original features to initiate the confidentiality stage.

### 2.1.4 Confidentiality Stage

The next stage is the confidentiality process, which consists of two steps. The first step is generating the encrypted key, which encrypts the data using an AES algorithm before storing it in the cloud computing data center. The second step is protecting the encrypted key that encrypts the data. This process can be done by encrypting the encryption key using the RSA algorithm.

#### First step

This step utilizes the control points extracted from the iris using Zernike moments to draw the B-spline curve. Next, two intersection points are selected. The first eight digits in their actual form

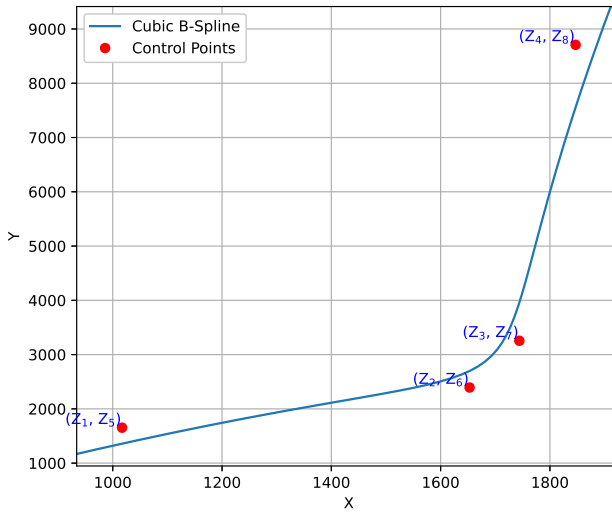


Figure 2.2: B-Spline Curve

are taken from each coordinate of these points. These eight digits are then converted into binary format by grouping every two digits. The last eight digits are used to ensure the extraction of 128 bits, which will be used in the AES algorithm for data encryption. Table 2.4 illustrates obtaining 128 bits from the intersection points.

Table 2.4: Select The Last Eight Digits To Obtain 128 Bits

| Last eight digit of x | Binary Form                         |
|-----------------------|-------------------------------------|
| 1017.19153781         | 00010011 00001111 00100101 01010001 |
| 1520.61634755         | 00111101 00111111 00101111 00110111 |
| Last eight digit of y | Binary Form                         |
| 1653.66624762         | 01000010 00111110 00101111 00111110 |
| 1922.78266287         | 01001110 00011010 00111110 01010111 |

From the table above, take only the first 128 bits and ignore the reminder bits, and the result is:

|          |          |          |          |
|----------|----------|----------|----------|
| 00010011 | 00001111 | 00100101 | 01010001 |
| 01000010 | 00111110 | 00101111 | 00111110 |
| 00111101 | 00111111 | 00101111 | 00110111 |
| 01001110 | 00011010 | 00111110 | 01010111 |

After this process, the bits are grouped into consecutive sets of eight and converted to hexadecimal to obtain the symmetric key for the AES algorithm, which will be used to encrypt the data. This process is illustrated in Table 2.5 before sending the data to cloud storage.

Table 2.5: 128-Bit Hexadecimal Encrypted Key

| Hexadecimal Values |    |    |    |
|--------------------|----|----|----|
| 13                 | F  | 25 | 51 |
| 22                 | 3E | 2F | 3E |
| 3D                 | 3F | 2F | 37 |
| 4E                 | 1A | 3E | 57 |

After generating the encrypted key from the iris, this key represents a unique encryption key for each individual. The key generated by this method will encrypt data before it is sent to cloud storage. This process represents the second approach in this dissertation to enhance the level of security in cloud computing. Table 2.6 and Figure 2.3 illustrate the encryption time for files of different sizes.

**Second step**

This step will protect the AES-encrypted key by utilizing the points extracted by Zernike moments from the iris as control points. In addition, this key is encrypted by the MRSA algorithm, which requires two prime numbers. The first is obtained from the Bézier-spline, and the second is obtained from Mersenne prime numbers.

Table 2.6: Implementation Of AES Algorithm

| File Size (Megabit) | Time of AES Execution (ms) |
|---------------------|----------------------------|
| 0.512               | 17.5                       |
| 1                   | 18.2                       |
| 1.5                 | 19.08                      |
| 2                   | 19.5                       |
| 2.5                 | 20.3                       |

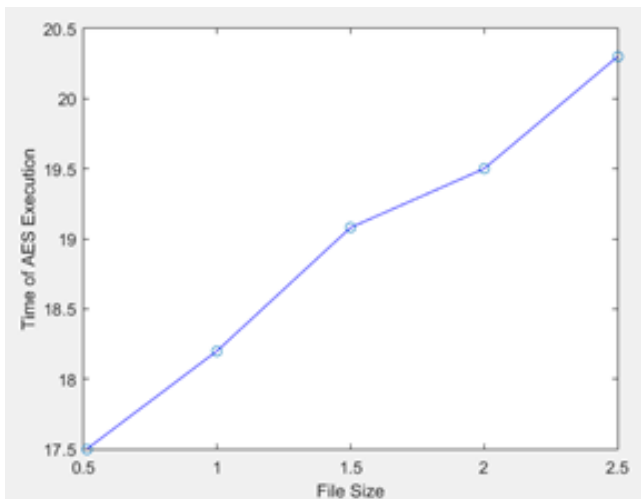


Figure 2.3: Period When Implemented The AES Algorithm

Table 2.7: Expand The Value Of Control Points

| X Coordinate of Control Points | Y Coordinate of Control Points |
|--------------------------------|--------------------------------|
| 1017                           | 1653                           |
| 1653                           | 2393                           |
| 1744                           | 3255                           |
| 1847                           | 8708                           |

Before using the control points in the Bézier curve, the proposed system multiplies and rounds each coordinate (x and y) by a constant value, such as 10.000, to expand the value of each point. Table 2.7 illustrates this process.

The mechanism for selecting the first prime number  $p$  involves comparing the primes of the x-values and y-values, selecting the two largest nearest primes, and then choosing the larger one to represent the prime number  $p$ . The second prime number is selected from the Mersenne prime numbers by comparing the  $p$  value with the nearest larger Mersenne prime, representing the second prime number  $M_q$ . This prime number  $M_q$  is used in the RSA algorithm to encrypt the AES encryption key.

In this case, the prime number  $p$  selected from the points on the Bézier curve is 1847, and the nearest larger Mersenne prime is 8707. This number produces the second prime number by applying the Mersenne prime formula:

$$M_q = 2^{8707} - 1 = 1.169963321862486603522673270657 \times 10^{2621} - 1.$$

The length of the private key is 2622 digits.

These prime numbers  $p$  and  $M_q$  are used in the RSA algorithm to generate the public and private keys, which are then used to protect the AES encryption key. Figure 2.4 illustrates the relationship between the length of the private key and the time required to generate it.

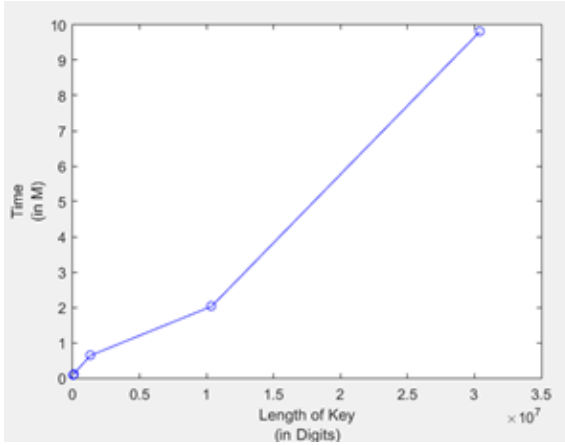


Figure 2.4: Relation Between The Length Of Private Key And Time

## 2.1.5 Evaluate and test the AES key encryption

### Frequency Test

The test statistic is given by:

$$X^2 = \frac{(n_0 - n_1)^2}{n}$$

Where the good sequence must satisfy:

$$0 < X^2 < 3.84$$

Where:

- $n$ : total digits of the key sequence
- $n_0$ : number of zeros in the key sequence
- $n_1$ : number of ones in the key sequence

Given:

$$n = 128, \quad n_0 = 60, \quad n_1 = 68$$

We compute:

$$X^2 = \frac{(60 - 68)^2}{128} = \frac{64}{128} = 0.4923077$$

The key succeeds this test since  $X^2 = 0.4923077 < 3.84$ .

### Frequency Mono Bit Test

The frequency mono-bit test is used to ensure the randomness of a sequence based on the proportion of 0's and 1's in the entire sequence. The number of zeros and ones expected in a random sequence should be the same. To succeed in this test, the P-value must be greater than 0.01. The steps of this test are as follows:

1. Convert the zeros into  $(-1)$  and the ones into  $(+1)$  for the key sequence, and sum them to produce:

$$S_n = x_1 + x_2 + \cdots + x_n$$

where  $S_n = 8$ .

2. Compute:

$$S_{\text{obs}} = \frac{|S_n|}{\sqrt{n}} = \frac{8}{\sqrt{128}} = \frac{8}{11.31} = 0.70733$$

3. Compute the P-value:

$$P\text{-value} = \text{erfc} \left( \frac{S_{\text{obs}}}{\sqrt{S_n}} \right) = \text{erfc} \left( \frac{0.70733}{\sqrt{8}} \right) = 0.7257210 > 0.01$$

Since the P-value is more significant than 0.01, this key successfully passes the frequency monobit test.

### Run Test

This test determines whether the number of 0s and 1s in the sequence is approximately the same as expected in a truly random sequence. When the P-value is greater than 0.01, the test is satisfied. The steps of this test are as follows:

1. Compute the pretest proportion:

$$\pi = \frac{\sum_j \zeta_j}{n} = \frac{60}{128} = 0.4687$$

2. Compute  $\tau$ :

$$\tau = \frac{2}{\sqrt{n}} = \frac{2}{\sqrt{128}} = 0.1768$$

3. Check the condition:

$$\left| \pi - \frac{1}{2} \right| \leq \tau \quad \Rightarrow \quad -0.0313 \leq 0.1768$$

4. Compute  $v_n(\text{obs})$ :

$$v_n(\text{obs}) = \sum_{k=1}^{n-1} r(k) + 1,$$

where  $r(k) = 0$  if  $\zeta_k = \zeta_{k+1}$ ,  $r(k) = 1$  otherwise

Thus,  $v_n(\text{obs}) = 60$ .

5. Compute the P-value:

$$P\text{-value} = \text{erfc} \left( \frac{|v_n(\text{obs}) - 2n\pi(1 - \pi)|}{2\sqrt{2n\pi(1 - \pi)}} \right)$$

$$P\text{-value} = \text{erfc} \left( \frac{|60 - 2 \times 128 \times 0.4687 \times (1 - 0.4687)|}{2\sqrt{2 \times 128 \times 0.4687 \times (1 - 0.4687)}} \right)$$

$$P\text{-value} = 0.449 > 0.01$$

This key successfully passes the run test since the P-value is more significant than 0.01.

### Entropy

Entropy is an important parameter to test the strength of the encryption key. When measuring the entropy of this key using a MATLAB function, the value of entropy is:

$$\text{Entropy} = 9.971804 \times 10^{-1}$$

## 2.2 Case study 2

The second case applies the proposed system on the iris image from the CASIA iris database.

- Iris size (Kbytes).
- Type of file.
- AES Encryption Key, size (128 bits) to encrypt the data.
- RSA Encryption Key, size (variable) to encrypt the AES Encryption Key.

### 2.2.1 Preprocessing the Eyes

The first step in the proposed system for preprocessing is to make the iris image a square image.

$$x = x_1(1:N, 1:N)$$

where:

- $x_1$ : the iris image,
- $N$ : the minimum dimension of the iris image.

Now, the proposed system will preprocess the iris; Figure 2.5 illustrates the preprocessing of the iris.

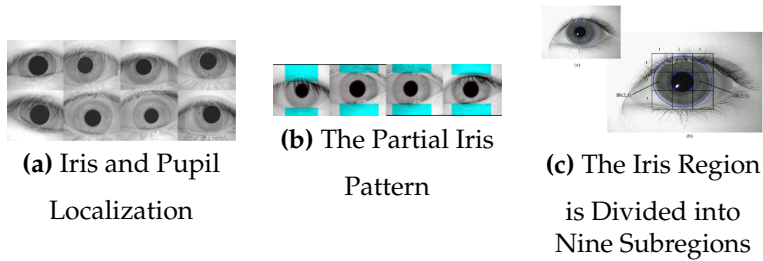


Figure 2.5: Preprocessing Of Iris.

### 2.2.2 Extract the Features

The proposed system will extract eight features from the iris by Zernike moments and reorder them to get the four points that will represent the control points to draw the curve of a B-spline. Table 2.8 illustrates these features and the control points.

Table 2.8: Eight Features And Control Points For Case Study 2

| Feature | Value of Feature | Control Points       |
|---------|------------------|----------------------|
| $Z_1$   | 1113.336         | (1113.336, 1113.957) |
| $Z_2$   | 1590.908         | (1590.908, 4653.255) |
| $Z_3$   | 2162.809         | (2162.809, 1326.555) |
| $Z_4$   | 9923.468         | (9923.468, 2335.987) |
| $Z_5$   | 1113.957         | -                    |
| $Z_6$   | 4653.255         | -                    |
| $Z_7$   | 1326.555         | -                    |
| $Z_8$   | 2335.987         | -                    |

### 2.2.3 Authentication Stage

The control points are used for authentication. This is achieved by using these control points to congruence with the features in the database; Table 2.9 illustrates this congruence.

Table 2.9: Congruence Of Control Points With Database

|          |          |          |          |           |          |          |          |
|----------|----------|----------|----------|-----------|----------|----------|----------|
| 1798.324 | 1803.471 | 2783.828 | 9560.785 | 2783.6892 | 2598.767 | 1277.892 | 2082.352 |
| 2703.776 | 5639.068 | 7221.632 | 7757.561 | 5639.852  | 1082.078 | 1996.989 | 4622.349 |
| 2776.762 | 2493.503 | 6756.662 | 6332.435 | 6217.465  | 2493.473 | 2492.797 | 1759.122 |
| 1597.843 | 2338.062 | 3203.731 | 4581.633 | 4581.772  | 6399.361 | 5704.294 | 1736.133 |
| 1113.336 | 1590.908 | 2162.809 | 9923.468 | 1113.957  | 4653.255 | 1326.555 | 2335.987 |
| ...      | ...      | ...      | ...      | ...       | ...      | ...      | ...      |

Table 2.10: Select The Last Eight Digits To Obtain 128 Bits For Case Study 2

| Last eight digit of x | Binary Form                         |
|-----------------------|-------------------------------------|
| 9892,76853264         | 01001100 01010101 00100000 01000000 |
| 9920,67191566         | 01000011 00010011 00001111 01000010 |
| Last eight digit of y | Binary Form                         |
| 2331,08685890         | 00001000 01000100 00111010 01011010 |
| 2334,69743006         | 01000101 01001010 00011110 00000110 |

## 2.2.4 Confidentiality Stage

To generate the encryption key used in the AES algorithm to encrypt the data before storing it in cloud computing, the last eight digits are taken from each control point and converted into binary form at every two digits. Then, only 128 bits are selected as an encryption key. Table 2.10 illustrates the result of this process.

```

01001100 01010101 00100000 01000000
01000011 00010011 00001111 01000010
00001000 01000100 00111010 01011010
01000101 01001010 00011110 00000110

```

The final step to obtain the AES-128 encryption key is to convert these 128 bits to hexadecimal, which is used as an encryption key for the AES algorithm to encrypt the data. This process is illustrated in Table 2.11 before sending the data to cloud storage.

The second step in the proposed system's confidentiality is to protect the encryption key by encrypting it using the MRSA algorithm. Therefore, in order to apply the MRSA algorithm, two prime numbers must be obtained in the beginning. The first

Table 2.11: 128-Bit Hexadecimal Encrypted Key

| Hexadecimal Values |    |    |    |
|--------------------|----|----|----|
| 4C                 | 55 | 20 | 40 |
| 43                 | 13 | F0 | 42 |
| 08                 | 44 | 3A | 5A |
| 45                 | 4A | 1E | 06 |

prime number could be obtained from the Bézier curve by utilizing the control points to draw the curve of a Bézier-spline.

The first prime number can be obtained from the curve's points of the Bézier-spline as  $p = 9923$ . The second prime number is derived by finding the nearest Mersenne prime  $q = 4673$ , and then applying the Mersenne prime formula:

$$M_q = 2^{9923} - 1 = 5.2548491497447161918815538985055 \times 10^{9907} - 1.$$

The length of the resulting private key is 2988 digits.

## 2.2.5 Evaluate and test the AES key encryption

### Frequency Test

We can calculate:

$$n = 128, \quad n_0 = 84, \quad n_1 = 44.$$

The test statistic is:

$$X^2 = \frac{(84 - 44)^2}{128}$$

$$X^2 = 12.5 > 3.84$$

This key fails the test since  $X^2 = 12.5 > 3.84$ .

### Frequency Mono Bit Test

To perform the frequency mono-bit test:

1. Convert the zeros into  $(-1)$  and the ones into  $(+1)$  for the key sequence, and sum them to produce:

$$S_n = x_1 + x_2 + \cdots + x_n$$

where  $S_n = -40$ .

2. Compute:

$$S_{\text{obs}} = \frac{|S_n|}{\sqrt{n}} = \frac{40}{\sqrt{128}} = 3.535$$

3. Compute the P-value:

$$P\text{-value} = \text{erfc} \left( \frac{S_{\text{obs}}}{\sqrt{S_n}} \right) = \text{erfc} \left( \frac{3.535}{\sqrt{-40}} \right)$$

This key fails because the test statistic is invalid due to the negative number inside the square root.

### Run Test

This test aims to determine whether the number of 0s and 1s in the sequence is approximately the same as expected in a truly random sequence. The steps are as follows:

1. Compute the pre-test proportion:

$$\pi = \frac{\sum_j \zeta_j}{n} = \frac{84}{128} = 0.6562$$

2. Compute  $\tau$ :

$$\tau = \frac{2}{\sqrt{n}} = \frac{2}{\sqrt{128}} = 0.1768$$

3. Check the condition:

$$\left| \pi - \frac{1}{2} \right| \leq \tau \quad \Rightarrow \quad 0.1562 \leq 0.1768$$

4. Compute  $v_n(\text{obs})$ :

$$v_n(\text{obs}) = \sum_{k=1}^{n-1} r(k) + 1,$$

where  $r(k) = 0$  if  $\zeta_k = \zeta_{k+1}$ ,  $r(k) = 1$  otherwise

Thus,  $v_n(\text{obs}) = 63$ .

5. Compute the P-value:

$$P\text{-value} = \text{erfc} \left( \frac{|v_n(\text{obs}) - 2n\pi(1 - \pi)|}{2\sqrt{2n\pi(1 - \pi)}} \right)$$

$$P\text{-value} = \text{erfc} \left( \frac{|63 - 2 \times 128 \times 0.6562 \times (1 - 0.6562)|}{2\sqrt{2 \times 128 \times 0.6562 \times (1 - 0.6562)}} \right)$$

$$P\text{-value} = 46.9 > 0.01$$

This key successfully passes the run test since the P-value is more significant than 0.01.

## Entropy

Entropy is an important parameter to test the strength of the encryption key. When measuring the entropy of this key, the value of entropy is:

$$\text{Entropy} = 92.8$$

# List of Publications

## Conferences

- C1** Hussein Lafta Hussein, Ahmed Abdulrudah Abbass, **Jasim H. Lafta**, Robert Tornai: *Novel Method to Generate symmetric key from Iris Using Polynomial Interpolation and Zernike Moment*. The 12<sup>th</sup> International Conference on Applied Informatics (ICAI 2023), Eger, Hungary, March 2-4, 2023.

## Publications

- P1** Abbass, A. A., Al, A. S., Hussein, H. L., **Kaabi, J. H. L.**, Tornai, R.: *American standard code for information interchange mapping technique for text hiding in the RGB and gray images*. *Int. J. Electr. Comp. Eng.*, **12**(3), 2812–2817, 2022. ISSN: 2088-8708.  
DOI: <http://dx.doi.org/10.11591/ijece.v12i3.pp2812-2817>
- P2** Abbass, A. A., Hussein, H. L., Shukur, W. A., **Kaabi, J. H. L.**, Tornai, R.: *Efficient Eye Recognition for Secure Systems Using Convolutional Neural Network*. *Webology*, **19**(1), 4967–4978, 2022. ISSN: 1735-188X.  
DOI: <http://dx.doi.org/10.14704/WEB/V19I1/WEB19333>
- P3** Naji, S. A., Hussein, H. L., Ihsan, A. M., **Kaabi, J. H. L.**: *Enhancing LSB Algorithm against Brute-Force Attack using ASCII Mapping Technique*. *J Eng Appl Sci*, **14**(16), 5854–5863, 2019. ISSN: 1816-949X.

## Conference proceedings

- P4** Naji, S. A., Tornai, R., **Kaabi, J. H. L.**, Hussein, H. L.: *Iris Recognition Using Localized Zernike Features with Partial Iris Pattern*.  
In: *New Trends in Information and Communications Technology Applications*. Eds.: Abbas M. Al-Bakry, Safaa O. Al-Mamory, Mouayad A. Sahib, Haitham S. Hasan, George S. Oreku, Thaker M. Nayl, Jaafar A. Al-Dhaibani. Springer, Baghdad, Iraq, pp. 219–232, 2020. (Proceedings, ISSN: 1865-0929). ISBN: 9783030553395
- P5** Hussein, H. L., Abbass, A. A., Naji, S. A., Al, A. S., **Kaabi, J. H. L.**: *Hiding text in gray image using mapping technique*. *J. Phys. Conf. Ser.*, **1003**, 1–7, 2018. ISSN: 1742-6588.  
DOI: <http://dx.doi.org/10.1088/1742-6596/1003/1/012032>
- P6** Hussein, H. L., Abbass, A. A., Naji, S. A., **Lafta, J. H.**, & Tornai, R. (2025). *Extract features using Zernike Moment to generate symmetric encryption key with polynomial interpolations*. In *AIP Conference Proceedings* (Vol. 3264, p. 030028). 3<sup>rd</sup> International Conference on Mathematics, AI, Information and Communication Technologies: ICMAICT2023. AIP Publishing. <https://doi.org/10.1063/5.0258437>



Registry number:

DEENK//2025.PL

Subject:

PhD Publication List

Candidate: Jasim Hussein Lafta Kaabi

Doctoral School: Doctoral School of Informatics

MTMT ID: 10071757

### List of publications related to the dissertation

#### Foreign language scientific articles in international journals (3)

1. Abbass, A. A., Al, a. S., Hussein, H. L., **Kaabi, J. H. L.**, Tornai, R.: American standard code for information interchange mapping technique for text hiding in the RGB and gray images.  
*Int. J. Electr. Comp. Eng.* 12 (3), 2812-2817, 2022. ISSN: 2088-8708.  
DOI: <http://dx.doi.org/10.11591/ijece.v12i3.pp2812-2817>
2. Abbass, A. A., Hussein, H. L., Shukur, W. A., **Kaabi, J. H. L.**, Tornai, R.: Efficient Eye Recognition for Secure Systems Using Convolutional Neural Network.  
*Webology.* 19 (1), 4967-4978, 2022. ISSN: 1735-188X.  
DOI: <http://dx.doi.org/10.14704/WEB/V19I1/WEB19333>
3. Naji, S. A., Hussein, H. L., Ihsan, A. M., **Kaabi, J. H. L.**: Enhancing LSB Algorithm against Brute-Force Attack using ASCII Mapping Technique.  
*J Eng Appl Sci.* 14 (16), 5854-5863, 2019. ISSN: 1816-949X.

#### Foreign language conference proceedings (3)

4. Hussein, H. L., Abbass, A. A., Naji, S. A., **Kaabi, J. H. L.**, Tornai, R.: Extract features using Zernike Moment to generate symmetric encryption key with polynomial interpolations.  
In: 3rd International Conference on Mathematics, AI, Information and Communication Technologies: ICMAICT2023, AIP Publishing, Melville, NY, 030028-1-030028-9, 2025, (AIP Conference Proceedings, ISSN 0094-243X 1551-7616 ; 3264)
5. Naji, S. A., Tornai, R., **Kaabi, J. H. L.**, Hussein, H. L.: Iris Recognition Using Localized Zernike Features with Partial Iris Pattern.  
In: New Trends in Information and Communications Technology Applications. Ed.: Abbas M. Al-Bakry, Safaa O. Al-Mamory, Mouayad A. Sahib, Haitham S. Hasan, George S. Oreku, Thaker M. Nayl, Jaafar A. Al-Dhaibani, Springer, Baghdad, Iraq, 219-232, 2020, (Proceedings, ISSN 1865-0929) ISBN: 9783030553395



6. Hussein, H. L., Abbass, A. A., Naji, S. A., Al, a. S., **Kaabi, J. H. L.**: Hiding text in gray image using mapping technique.

*J. Phys. Conf. Ser.* 1003, 1-7, 2018. ISSN: 1742-6588.

DOI: <http://dx.doi.org/10.1088/1742-6596/1003/1/012032>

The Candidate's publication data submitted to the Tudóstér have been validated by DEENK on the basis of the Journal Citation Report (Impact Factor) database.

09 April, 2025

