Journal of Theoretical and Applied Information Technology
29th February 2024. Vol.102. No 4
© Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NEW ENCRYPTION ALGORITHM BASED ON ARTIFICIAL INTELLIGENCE'S FACE RECOGNITION, SYMMETRIC ENCRYPTION, AND STEGANOGRAPHY

MIAD SALEM ALBALAWI [1] , NUHA KAMAL HUWAYKIM[2] , WAAD AHMED ALBRAIQI[3] , MOHAMMED ALWAKEEL[4]

[1,2,3]University of Tabuk, Department of Information Technology, Tabuk, Saudi Arabia

[4]University of Tabuk, Department of Computer Engineering, Tabuk, Saudi Arabia

E-mail: [1]441000382@stu.ut.edu.sa, [2] 431010400@stu.ut.edu.sa, [3]441009047@stu.ut.edu.sa, [4]alwakeel@ut.edu.sa

ABSTRACT

Encryption is critical to maintain the privacy, accuracy, and integrity of sensitive data. It protects the confidentiality of data by making it inaccessible to unauthorized parties. There are mainly two types of encryptions used widely, symmetric encryption and asymmetric encryption, in addition to encryption steganography is also used to protect data, where the data is hidden within another object to avoid being accessed by unauthorized user.

In this research we propose a new symmetric encryption technique, that uses artificial intelligence's face recognition, encryption, and steganography to encrypt and protect the data. Firstly, to generate a unique encryption key, the face attributes are extracted from a face photo using face recognition technique, and then the encryption algorithm process these attributes to generate the encryption key. Once the key is generated it is used by the encryption algorithm to encrypt the original data and generate the cyphered data, finally steganography technique is used to hide the cyphered data in a cover photo and hide the face photo that was used to generate the key in another cover photo, and then both cover photos will be sent to the receiver. At the receiver side, the face photo that was originally used to generate the key, and the cyphered data are retrieved from the two received cover images using steganography technique, then the artificial intelligence's face recognition technique is used to extract the face attributes from the retrieved face photo, finally the decryption algorithm process these attributes to generate the encryption key and used it to decrypt the cyphered data and retrieve the original data. In the final section of this research the encryption strength of the proposed technique is discussed.

Keywords: *Symmetric Encryption, Private Key Generation, Artificial Intelligence, Steganography, Face Attributes*

1. INTRODUCTION

In recent time due to the widespread use of information technology and its wide applications in many fields, it is necessary to have a highly efficient way to ensure the confidentiality and privacy of data and ensure the transmission of data in a secure way, hence the use of encryption in a system ensures that data can only be read and used by certain trusted authorized persons. There are mainly two types of encryptions, namely symmetric encryption, and asymmetric encryption. In symmetric encryption a single private key is used to encrypt and decrypt data [1-2], this type of encryption is fast due to the fact that relatively simple calculations are required to accomplish the encryption/decryption processes. Several symmetric encryption algorithms were developed such as Data Encryption Standard (DES), 3DES, Advanced Encryption System (AES) and Blowfish [3]. On the other hand, the asymmetric encryption uses a pair of keys, the first is a public key that is used for encryption and the second is a private key that is used for decryption [4]. In general the asymmetric encryption is more secure than symmetric encryption, however it is slower due to the fact that it uses more complex arithmetic operations and consumes a high amount of CPU processing ability to perform the encryption and decryption [5], Rivest–Shamir–Adleman (RSA) and Elliptic Curve Cryptography (ECC) [6] are examples of the asymmetric algorithms that were developed and used widely. In general, recently in both types of encryption the size of the keys usually ranges from 128 to 256 bits [7].

With technological progress and rapid globalization, information security is crucial and encryption is one of the most important guarantees of information security, and with the emerging of artificial intelligence (AI) in all aspects of life, the effect of AI on digital life and the internet necessitates an increased focus on cybersecurity [8].

Recently ai has been used to improve encryption techniques by creating secure keys, optimizing algorithms, breaking encryption, detecting intrusions, developing quantum-resistant algorithms, detecting threats, and ensuring secure communication, hence, ai is now widely used to encrypt and decrypt text, images, videos, signals, and data [9]. In addition to that, machine learning models and AI are used to analyze encrypted texts, identify hidden patterns, and achieving network encryption and security where the attacks and the threats are detected and analyzed using AI, and appropriate protection measures are taken into consideration based on the results of the analysis [10-11]. With that being said and as new serious risks emerge, AI has become an asset for companies and complex corporate environments to improve their performance in terms of security, therefore, developing strong encryption technologies and using tools that act responsibly to protect the privacy and security of sensitive data become critical issues and one of the most important aspects that every organization must ensure.

In addition to encryption techniques, steganography can be used as a method to secure the transmission of sensitive data between the transmitter and the receiver. Steganography can be defined as a method or a technique that is used to hide data within another object in order to avoid being accessed by unauthorized user, although this technology is centuries old it is still useful, and in parallel with technological advances steganography technology has evolved and adapted to the advent of computers and the internet. In the modern world steganography is mainly used to hide digital data into another digital medium in such a way that only the persons involved are aware of this and can retrieve the concealed data. There are several types of steganography including hiding text, image, video, or audio in another image, video, audio [12-13], there are also multiple techniques to do so, such as Least Significant Bit (LSB) steganography where each image pixel is represented by eight bits, and the last bit on the right side is used to hide the sensitive data and since the pixel value may be reduced by one, the human eye cannot notice that there is a difference between the two images [14]. Palette-based steganography is another technique that may be used where sensitive data is embedded in a palette-based picture such as GIF [15].

A third technique that may be used in steganography is secure cover selection, which is a quite complex technique of steganography that is used usually by cybercriminals, where they have to compare blocks of the original image to specific blocks of a malware. An exact match is required to carry the malware, the identical match is fitted into the original image which makes it extremely difficult to detect the malware with cybersecurity software or software applications [16]. Based on what was mentioned above Table 1 represents a brief comparison between steganography and cryptography.

Table 1 Steganography vs Cryptography

Criteria	Steganography	Cryptography
Definition	Used to conceal sensitive data	Used to make information Incomprehensible
Goal	Focus on keeping communication secure	Focus on keeping data protected
Key	Optional	Necessary
Carrier	All type digital media	Often depends on the text
Data Visibility	Never	Always
Failure	The discover of the hidden data makes it usable/understandable	When the decryption key is known, the original message is revealed
Data Structure	Does not altered the data's general structure	altering the overall data structure
Result	Stego file	Cipher text

2. LITERATURE REVIEW

In the past extensive amount of literature studies that include encryption, use of AI in encryption, and steganography have been published, and almost all aspects of these topics were investigated in these studies. Some of these studies that are related to our research are reviewed in the next sections.

2.1 Encryption Algorithms Literatures

[6], Presented a study to improve the security of cloud computing by combining symmetric and asymmetric encryption such that it improves encryption performance while the complexity is reduced, in addition to that the encryption time and throughput were improved. Since the speed factor is necessary in cloud computing, the author uses Elliptic-Curve based algorithm for key encryption and Blowfish algorithm for data encryption, and to ensure the integrity and confidentiality of the data he uses MD5-based digital signature technology, the evaluation of the proposed solution shows that it significantly superior to RSA, AES, 3DES and DES. The results were average in terms of execution time, throughput efficiency and memory utilization rate of the proposed solution compared to other solutions, but the AES outperformed it in terms of speed and throughput efficiency when the data volume was increased.

[1], developed an information security management system for records in colleges and universities to maintain their confidentiality and privacy based on symmetric encryption. The effectiveness of the developed system was evaluated in trial universities and the results show that the proposed safety management system is very acceptable and received a high performance and efficiency evaluation.

In this research [17], highlighted the mechanism of encryption used in the Internet of things, the researchers found that most of the encryption methods used in the Internet of things is often done with simple symmetric encryption because IoT devices are limited in battery and size, hence it is easy to know the encryption key, this makes IoT devices unsafe and vulnerable. The researchers proposed a new Genetic Algorithm-based symmetric key generation (GA) and used it to enhance DES performance they called it GADES algorithm. They used simulation to show that the developed method's is better than encryption with DES or GA algorithms, hence their method provides a low probability and more randomness to guess the encryption key.

An energy costs comparison between schemes that use asymmetric and symmetric key using two methods was presented by [2], In the first method they used energy cost of data usage (ECDU) to evaluate the energy costs of data usage on the internet globally, and for the other method they construct a small-scale network of wireless embedded devices as an experimental environment. A comparison of communication, computation costs, and an estimation of energy consumption for each solution were carried out. The research results showed that 58% of global energy costs can be saved when using symmetric key systems. The results also showed that when using the symmetric key encryption in wireless devices during the key agreement, it can achieve a decrease of up to 20 % in wireless device energy consumption.

New technique for encryption were presented by [3], the proposed technique consists of four layers, the layers combine anonymous text randomization, artificial intelligent encryption, steganography, and standard encryption algorithm to enhance and increase the security of any standard encryption algorithm. The results showed that the security level of the presented technique is higher than standard encryption, however the overhead is higher than the overhead of standard encryption or steganography. The presented work found that the implementation of the proposed technique is very simple and can be implemented easily with minimum cost and effort.

[5], reviewed several encryption algorithms and measured the performance of ten different algorithms using several metrics such as throughput, CPU utilization rate, level of security, and speed of encryption. They found out that RC4, RC6 and AES have the best encryption time and throughput, in addition they concluded that AES gave the best performance and level of security.

2.2 AI and Using AI in Encryption

[8] Presented an extensive literature review of researches on the use of AI in system security. They concluded that there will be huge demand on AI-based support system, also they stated that there is an urgent need to improve measures, standards, and strategies of cyber security, and finally they concluded that further developments of neural network platforms are required.

Using artificial intelligence in the area of electrical automation control systems was discussed by [18]. The authors concluded that despite the progress and development in the field of artificial intelligence applications in technology, science and social economics, however integration of artificial intelligence applications with electrical automation is facing many challenges, and there are many opportunities for improvements in this field.

[10], presented a new cyber security algorithm that uses artificial intelligence to detect attacks on networks, also he compared its performance with other practical algorithms. He concluded that the performance of the proposed technique was better and high accuracy indicators were achieved during the process of protecting users' social media profiles as the number of users increased.

[11] presented a survey on the impacts of using artificial intelligence in the field of cyber security. It has been concluded that artificial intelligence has a significant beneficial impact on cybersecurity according to the Authers survey, however there are also some limitations, and more research should be directed toward avoiding these limitations.

[19] in their research developed a new image encryption technique that may be used in AI, through integrating Rabinovich hyperchaos and compression sensing. The simulation's results of the proposed algorithm concluded that the proposed algorithm provides resistance to many types of attacks, including brute force, statistical and differential.

2.3 Steganography

In [12], reviewed hiding pattern taxonomy and introduced a new taxonomy tool that may be used in all steganography's domains, in addition to that they distinguish between hidden patterns' representation and the embedding process.

[20] presented a new steganography algorithm using LSB and Knight Tour Algorithm to improve stego image's security factor, then they used steganalysis the hidden message. The analysis shows that the level of security was improved, however there is a huge drawback in the proposed technique, where the size of the cover image must be dividable by 4 in order to use the Knight Tour Algorithm.

The work in [14] introduced an adapted multiway pixel-value differencing technique for image steganography utilizing general quantization ranges of pixel pairs' values. The results show that the proposed method is expected to produce better stego-images compared to the current multiway pixel-value differencing methods, specifically in terms of embedding rate and image quality.

[15] presented a new reversible steganography method to enhance payload security without affecting marked image's high quality. The results show the effectiveness of the proposed method with respect to image quality.

Several deep learning techniques that were used in steganography were discussed by [13], they also illustrate several evaluation metrics that can be used to evaluate the performance of different techniques. They concluded that LSB is the most common technique used in traditional steganography in addition to PVD, EMD, and DCT. Also, they concluded that the traditional methods' hiding capacity is limited, and exploiting additional pixels to hide the entire private message will produce a distortions in the cover image.

New method of steganography's cover image selection was presented by [16], the proposed method resists single object steganalysis and pooled steganalysis. They concluded that using the proposed method to select the cover image guaranteed steganography's security against pooled steganalysis and single object steganalysis.

3. PROBLEM STATEMENT AND RESEARCH QUESTIONS

As we have seen from the previous part, there are mainly two types of encryptions, symmetric encryption and asymmetric encryption, the standard symmetric encryption techniques require a private encryption key that is pre-agreed between the sender and the receiver, and the key should be known to both of them before sending and receiving the message. Hence, a prior connection between the sender and the receiver is required to exchange the encryption key, and the connection used in this stage must be completely secure, since what is needed to decrypt the message in standard symmetric encryption is only the private key [7-5]. With technology advancement many methods have been developed that can be used to break symmetric encryption fairly easily, and one of the simplest methods to do that is to attack using brute force, in addition to brute force there are many other adaptive ways to shorten the time needed to break the encryption, however they require more complex calculations and consume more processing power. To ensure the continuation of secure encryption using symmetric encryption methods at least two conditions must be fulfilled, the first is changing the encryption key periodically and at a relatively short intervals, and the second condition requires the use of a completely secure way to exchange the private encryption key between the sender and the receiver every time the key is changed, any lack of commitment with these two conditions results in a poor level of encryption security, which will ultimately lead to breaking the encryption and allow access to the data exchanged between the sender and the receiver by a third party who is not authorized to access and control the data. We also discussed in the previous section how the artificial intelligence may be used in encryption and decryption, and several methods to achieve this were mentioned. Another method to transfer data confidentially is steganography and several steganography techniques and the differences and the similarities between them were also presented, and a comparison between securing texts using the steganography technique and cryptography was presented. Our research aims to answer the following questions:

1. Could it happen a symmetric encryption be done without prior communication between the sender and the receiver to exchange the private encryption key.
2. Is there a secure way to send a private encryption key between the sender and the receiver together with the encrypted message such that it is changed for each encrypted message.
3. Can a mechanism be developed through which a private encryption key is created such that the mechanism can be used by the sender and the receiver to complete the encryption and decryption process without the need to send the same private encryption key directly between the sender and the receiver.
4. Is it possible to create a private encryption key using artificial intelligence technologies.
5. How to use additional security techniques such as steganography to increase the security level of messages sent between the parties.
6. How symmetric encryption, AI and steganography technologies can be combined together to create a new encryption mechanism that has better security level or better performance.
7. What is the security level of the proposed method compared to the security level of the standard encryption mechanisms.

4. THE CONTRIBUTIONS OF THE PROPOSED TECHNIQUE

In this research, we seek to introduce a new method to securely transfer data, and at the same time avoid some of the disadvantages of standard methods. This may be done by developing an encryption technique that allow sending encrypted data between the sender and the receiver; in a method similar to the symmetric encryption; however the two conditions mentioned in the previous section will be eliminated such that there will be no need for a secure prior communication between the two parties to exchange the private encryption key before sending the data, in addition to that; the developed technique will allow changing the encryption key every time a message is sent between the sender and the receiver easily without the need of exchanging the private encryption key prior sending the encrypted data. The developed technique will combine encryption technique, artificial intelligence, and steganography, where artificial intelligence will be used to generate a private key, then encryption will be used to encrypt the data using the generated key, and finally steganography will be used to hide the encrypted message.

The main contributions of this research are as follows:

1. A new encryption technique will be presented.
2. Eliminating the two conditions that were mentioned in the previous section which are required for standard symmetric encryption.
3. A new method to generate the private encryption key will be developed using AI's human faces' attributes extraction technique.
4. Encryption, AI and steganography techniques will be integrated together to securely transfer data between the sender and the receiver.

5. SUGGESTED SOLUTION

The proposed method to introduce a new symmetric encryption/decryption technique will include three stages for both encryption and decryption as shown below.

5.1 Suggested Encryption Stages

- Stage 1: AI face recognition technique will be used to extract the attributes of a human face (which are usually set of coordinates that may be

used to recognize the face) from a face photo stored in an image file, then the face attributes are used to generate a unique private encryption key using equations that manipulate the digits from the extracted face attributes.
- Stage 2: Encryption technique will use the private key generated in stage 1 to encrypt the data and generate the cypher data file.
- Stage 3: Steganography technique namely LSB will be used to hide two files into two different stego image files, the first stego image file will contain the hidden cypher data file and the second stego image file will contain the hidden face photo's image file e that was used in stage 1 to generate the private key.

These encryption stages are illustrated in figure 1.1 Below.

5.2 Suggested Decryption Stages

Two stego image files will be received at the receiver site, the first contains the face photo image file used to generate the private key, and the second stego file contains the cypher data file. The decryption process will have three stages as follow:

- Stage 1: The steganography technique will be used to extract the face photo image file and the cypher data file from the stego image files that were received from the sender.
- Stage 2: AI face recognition technique will be used to extract the attributes of the face from the face photo image file extracted in stage 1, then the face attributes are used (in a similar way that they were used at the sender site) to generate the unique private key that were used to encrypt the text.
- Stage 3: Decryption technique will use the private key generated in stage 2 at the receiver site to decrypt the cypher data and display the original data to the receiver.

These decryption stages are illustrated in figure 1.2 Below.

6. MPLEMENTATION AND ILLUSTRATIONS

6.1 Implementation of the Encryption

The proposed technique will encrypt the original message and decrypt the cyphered message using an application that we developed. The developed application consists of three stages, namely AI face recognition stage, encryption stage, and steganography stage. In the following subsections the three stages of the developed encryption technique will be illustrated.

6.1.1 AI Face Recognition Stage

The first stage is an AI face recognition that will be used to extract the attributes of a face from a face image file which is only known by the sender who knows also the original message, this is done by detecting the locations of 128 distinguished landmarks points on the face; as shown in figure 3.1 below, and store the locations of these landmarks in a text file (for the purpose of illustration we will call it "face_attributes.txt"). Table 3.1 shows a sample of the first 60 locations of the face landmarks.

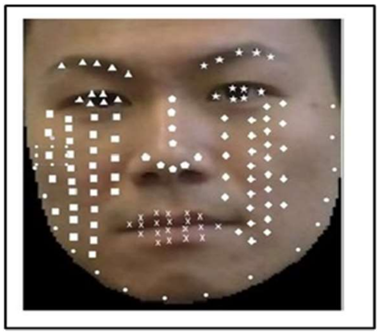


Figure 3.1 The location of 128 landmark points on the face

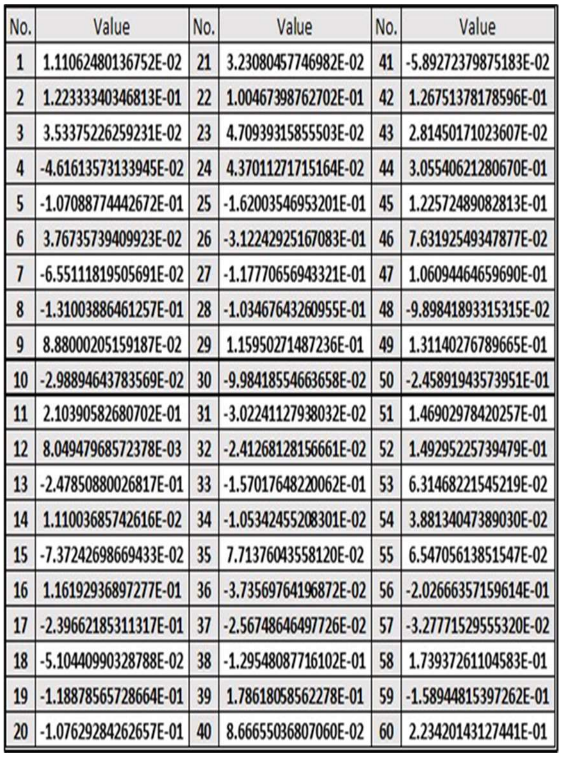
6.1.2 Encryption Stage

The message encryption stage will start once the first stage is completed, in this stage the locations of the face landmarks stored in the file "face_attributes.txt" will be used to generate the encryption key, and this key will be used to encrypt the original message. To generate the key we will start by extracting every decimal digit (i.e 0, 1,…,8,9) from the "face_attributes.txt" file and arrange them as a decimal digits string, figure 3.2 shows part of the decimal digits string that was extracted from the first five locations of the face landmarks listed in table 3.1. Several key generation techniques can then be used to select a subset from the full decimal digits string to generate the encryption key (which is represented by decimal digits at this point), then to represent the encryption key in binary digits each decimal digit of the encryption key will be represented by 8 bits which is the ASCII code of the decimal digit, for example if the length of the encryption key is 128 bits then the key generation technique will select 16 decimal digits from the decimal digits string and then find the ASCII codes of these digits and use it as an encryption key.

To simplify the illustration of the techniques that can be used to generate the encryption key, let us define the variables and the parameters that are used in these techniques:

- *DDS* : Denotes the decimal digits string as defined in section 3.1.2.
- *DDS$_i$* : Denotes the *ith* decimal digit in *DSS.*
- *KDDS* : Denotes the encryption key represented as decimal digits string.
- *KDDS$_i$* : Denotes the *ith* decimal digit in *KDSS.*
- *K* : Denotes the length of the key in bits (note that for encryption key *K* is always a multiple of 8.
- *N* : Denotes the length of the key in decimal digits where *N=K÷8*
- *BK* : Denotes the encryption key represented in binary digits.
- *ConvertToBinary*(*X*) : Denotes a function that accept a string of characters (*X*) as input, and convert it to Binary string by replacing every character in the string *X* with its ASCII code.
- *Concatenate*(*X,Y*) : Denotes a function that accept two string of characters (*X* and *Y*) and concatenate them to generate a single string of characters.
- *GRandom( )* : Denotes a function that generate a random integer between (1 and N*)*

Table 3.1 Sample of the face's landmark Locations



No.	Value	No.	Value	No.	Value
1	1.11062480136752E-02	21	3.23080457746982E-02	41	-5.89272379875183E-02
2	1.22333340346813E-01	22	1.00467398762702E-01	42	1.26751378178596E-01
3	3.53375226259231E-02	23	4.70939315855503E-02	43	2.81450171023607E-02
4	-4.61613573133945E-02	24	4.37011271715164E-02	44	3.05540621280670E-01
5	-1.07088774442672E-01	25	-1.62003546953201E-01	45	1.22572489082813E-01
6	3.76735739409923E-02	26	-3.12242925167083E-01	46	7.63192549347877E-02
7	-6.55111819505691E-02	27	-1.17770656943321E-01	47	1.06094464659690E-01
8	-1.31003886461257E-01	28	-1.03467643260955E-01	48	-9.89841893315315E-02
9	8.88000205159187E-02	29	1.15950271487236E-01	49	1.31140276789665E-01
10	-2.98894643783569E-02	30	-9.98418554663658E-02	50	-2.45891943573951E-01
11	2.10390582680702E-01	31	-3.02241127938032E-02	51	1.46902978420257E-01
12	8.04947968572378E-03	32	-2.41268128156661E-02	52	1.49295225739479E-01
13	-2.47850880026817E-01	33	-1.57017648220062E-01	53	6.31468221545219E-02
14	1.11003685742616E-02	34	-1.05342455208301E-02	54	3.88134047389030E-02
15	-7.37242698669433E-02	35	7.71376043558120E-02	55	6.54705613851547E-02
16	1.16192936897277E-01	36	-3.73569764196872E-02	56	-2.02666357159614E-01
17	-2.39662185311317E-01	37	-2.56748646497726E-02	57	-3.27771529555320E-02
18	-5.10440990328788E-02	38	-1.29548087716102E-01	58	1.73937261104583E-01
19	-1.18878565728664E-01	39	1.78618058562278E-01	59	-1.58944815397262E-01
20	-1.07629284262657E-01	40	8.66655036807060E-02	60	2.23420143127441E-01

{1110624801367521286122333340346813201901353
375226259
23156740246161357313394546510210708877444267
2729501}

Figure 3.2 Subset of decimal digits string extracted from the first five values in table 3.1

In the next subsections using the above definitions, some of the techniques to generate the encryption key are illustrated.

6.1.2.1 Encryption Key Generation Technique A

Since the image of the face that is used to generate the key is unknown to anyone except the sender this technique is a straightforward, the key is generated by selecting the first *N* decimal digits from the decimal digits string as a subset, then convert this subset to binary set based on the ASCII code of each digit in the subset. The generated binary set is then used as an encryption key. For example, if *K=32* bits and the *DDS* is as shown in figure 3.2, then *N=32÷8=4* decimal digits, hence the first 4 digits will be selected from *DDS* as a subset (i.e. 1110) and then the equivalent ASCII code of this subset represent *BK* (i.e. the key will be "00110001 00110001 00110001 00110000"). This technique can be represented by the following simple algorithm:

```
K= input("Enter the length of the key :")    %# Length of the key in bits #%
KDDS = []                                    %# Defined KDDS as an empty character set #%
N=K/8
For c=1 To N
        {KDDS=concatenate(KDDS,DDS_c)}
BK=ConvertToBinary(KDDS)
```

6.1.2.2 Encryption Key Generation Technique B

In this technique to generate the encryption key, we will use some mathematical equations to locate the location of each digit in *DDS* that will be used as part of *KDDS*. The first digit in *KDDS* is selected such that $KDDS_1 = DDS_N$ and for the rest of *KDDS* digits they are selected such that $KDDS_i = DDS_{x_i}$, such that $x_1 = N$ and $x_i = x_{(i-1)} + 1 + \left(\left(DDS_{x\_((i-1))} + N\right) mod\ 9\right)$. For example, if *K=128* bits and the *DDS* is as shown in figure 3.2, then *N=128÷8=16* decimal digits and the following algorithm represent how *BK* can be generated:

```
K= input("Enter the length of the key :")    %# Length of the key in bits #%
KDDS = []                                    %# Defined KDDS as an empty character set #%
N=K/8
$w = N$
```

```
$KDDS_1 = DDS_N$
        For c=2 To N
        {       $x = w + 1 + \left(\left(DDS_{x\_((i-1))} + N\right) \bmod 9\right)$
        KDDS=concatenate(KDDS, $DDS_x$)}
$w = x$
BK=ConvertToBinary(KDDS)
```

Once the key is generated it can be used to encrypt the original message. The first step to encrypt the original message is to convert it to binary, then if the length of the message in binary is not a multiple of the key length, extra zeros are added to the end of the message such that the new length of the message is a multiple of the key length. The message is then divided into blocks with length equal to the key length, and to encrypt the message an XOR operation is carried out between each block and the encryption key, this process is illustrated in figure 3.3. After encrypting all the blocks, they will be concatenated together to form the encrypted message, which is stored in a binary file (for the purpose of illustration we will call it “Encrypted_Message.bin”). Figure 3.4 shows the developed application screen shot for the first and the second stage.

6.1.3 Steganography Stage

In this stage the encrypted message is hidden into a cover image and the face image that was used in the AI face recognition stage is hidden into another cover image, and the sender will send both cover image files to the receiver. In this stage several steganography techniques can be used, and for our proposed technique we used LSB technique, where each image pixel in the cover image is represented by eight bits and the last bit on the right side is used to hide the encrypted data, and since the pixel value may be changed by one, the human eye usually cannot notice the difference between the two images. Figure 3.5 shows a screen shot of the steganography stage, and figure 3.6 shows the original cover image and the stego image that include the face image that was used to generate the key hidden in it. From figure 3.5 and 3.6 we can notice that the human eye can’t distinguish between the original cover image and the stego image.

By the end of the three stages of the encryption process the sender will have two stego image files. The first file contains a cover image which includes the face image that was used in the first stage hidden within it. The second file contains another cover image which includes the encrypted message that was generated in the second stage hidden within it. For both files LSB steganography was used to hide the face image and the encrypted message as we mention in the above section. The sender will then send the two stego image files to the receiver as the encrypted message.

6.2 Implementation of the Decryption

At the receiver side the decryption process starts by receiving two image files from the sender. The decryption process also consists of three stages, namely steganography stage, AI face recognition, and finally the decryption stage. In the following subsections these three stages are illustrated.

6.2.1 Steganography Stage

In this stage we use the same steganography technique that was used during the third stage of the encryption process at the sender side to retrieve respectively the hidden face image from the first received file and the encrypted message from the second received file. The extracted face image and encrypted message are passed to the second stage of the decryption process.

6.2.2 AI Face Recognition Stage

In this stage the same AI face recognition technique that was used in the first stage of the encryption process is used to extract the attributes of the face from the face image that was retrieved in the previous stage. This is done by detecting the locations of the same 128 distinguished landmarks points on the face which were detected in the first stage of the encryption process at the sender side; and store the locations of these landmarks in a text file (for the purpose of illustration we will call it “face_attributes.txt”).

6.2.3 Decryption Stage

In this stage the locations of the face landmarks stored in the file face_attriute.txt will be used to generate the decryption key using the same method and algorithm that were illustrated in section 3.1.2, section 3.1.2.1, and section 3.1.2.2. Once the key is generated it can be used to decrypt the encrypted message that was retrieved in the first stage of the decryption process. The encrypted message which is in binary at this stage is then divided into blocks with length equal to the key length, and to decrypt the message an XOR operation is carried out between each block and the decryption key, this process is illustrated in figure 3.7. After decrypting all the blocks, they will be concatenated together to form the original message in a binary code, then the original message will be converted into characters (to enable the receiver to read it) by using the ASCII code to convert each 8 bits from the decrypted message into its equivalent character. Figure 3.8 shows the developed application screen shot that is used to decrypt the message.

6.3 Encryption Strength

In general, there are two ways to break the encryption, the first is brute force where the hacker tries all possible combinations of the key until he successfully decrypted the message, and in order to do so the hacker need to know the length of the key, however the brute force method may require an extensive amount of time. On the other hand, adaptive encryption breaking techniques (which usually required less time than brute force) may be used to decrypt the encrypted message, however these techniques usually require more processing power and more complicated algorithms than brute force. The strength of the encryption may be evaluated using several parameters, one simple parameter to evaluate the strength of the encryption is by assuming that the encryption strength (ES) is evaluated as ($ES = 2^{KB}$) where KB is the length of the key as we defined it earlier. For the proposed technique if brute force is used to break the encryption, the encryption strength is equivalent to the strength of the standard techniques like AES, DES… etc. since it depends only on the length of the encryption key. For the adaptive encryption breaking techniques it is obvious that the proposed technique will be much stronger than standard techniques, since the hacker is required to know several additional parameters than the parameters required to break the standard encryption, these parameters include knowing the exact details of the proposed technique's stages, knowing the steganography algorithm that was used during the encryption, knowing the techniques that was used to extract the face attributes and the locations of the landmark on the image face during the first stage of the proposed encryption technique, and finally knowing the technique that were used to generate the encryption key from the landmarks locations as explains in sections 3.1.1. and 3.1.2.

7. CONCLUSION

In this research we proposed a new symmetric encryption technique, the proposed technique uses encryption method, artificial intelligence, and steganography to enhance and increase the security level of the encrypted message. The proposed technique at the sender side uses three stages to encrypt the message, where artificial intelligence's face recognition technique is used to extract the face attributes from a face image that is selected by the originator of the message, then the encryption algorithm processes these attributes to generate the encryption key that is used to encrypt the message, and finally steganography technique namely LSB is used to hide the encrypted message within a cover image, also the same steganography technique is used to hide the face image that was used to generate the encryption key within another cover image, and both cover images are sent to the receiver. At the receiver the two cover images from the sender are received, and steganography technique is used to recover the encrypted message and the face image, then the artificial intelligence's face recognition technique is used to extract the face attributes from the face image, finally the decryption algorithm process these attributes to generate the encryption key and use it to decrypt the encrypted message and retrieve the original message.

It is expected that the proposed technique will outperform the standard encryption algorithms with respect to security level whenever any adaptive method is used by a nonauthorized entity to decrypt the message, and it will have at least similar performance to the standard encryption technique whenever brute force method is used to decrypt the message. The proposed technique also eliminates the conditions that must be fulfilled when using standard encryption.

Advance performance analysis and more sophisticated A.I. technique may be used to enhance the work done in this research. Hence, future research directions of this work may include advance sophisticated performance analysis of the proposed technique, based on several performance measures such as throughput, overhead, security level, processing time... etc., and compare the results with the performance of other relevant works and standard encryption and steganography methods. The proposed work also can be expanded to incorporate an advance A.I. technique to replace the face recognition technique that was used to generate the encryption key.

REFERENCES

[1] Wu, K., & Li, C. (2022). Application of Symmetric Encryption Algorithm Sensor in the Research of College Student Security Management System. Jo*urnal of Sensors*, 1230-1242. https://doi.org/10.1155/2022/3323547

[2] Halak, B., Yilmaz, Y., & Shiu, D. (2022). Comparative Analysis of Energy Costs of Asymmetric vs Symmetric Encryption-Based

Security Applications. *IEEE Access*, 76707-76719. https://doi.org/10.1109/ACCESS.2022.3192970

[3] Alfadhli, S., Sak, K., & Alwakeel, M. (2023). Enhancing Standard Encryption Algorithms Using Multilayers Encryption Technique. *Journal of Theoretical and Applied Information Technology*, 101(3),1230-1242.

[4] Luo, Q., & Zhang, Z. (2023). The Secure Data Transmission Method of a Cellular Communication Network Based on the Asymmetric Encryption Algorithm. *Journal of Communications*, 82-88.

[5] Alenezi, M. N., Alabdulrazzaq, H., & Mohammad, N. Q. (2020). Symmetric encryption algorithms: Review and evaluation study. *International Journal of Communication Networks and Information Security*, 12(2), 256-272

[6] Abroshan, H. (2021). A Hybrid Encryption Solution to Improve Cloud Computing Security using Symmetric and Asymmetric Cryptography Algorithms. *International Journal of Advanced Computer Science and Applications*, 12(6), 31-37. https://dx.doi.org/10.14569/IJACSA.2021.0120604

[7] Kilic, M. B. (2021). Encryption Methods and Comparison of Popular Chat Applications. *Advances in Artificial Intelligence Research*, 52-59.

[8] Raimundo, R., & Rosário, A. (2021). The impact of artificial intelligence on Data System Security: A literature review. *Sensors*, *21*(21), 1-19.

[9] Shawkat, S., & Al-Barazanchi, I. (2022). A proposed model for text and image encryption using different techniques. *TELKOMNIKA (Telecommunication Computing Electronics and Control*), *20(4)*, 858-866.

[10] Logeshwaran, J. (2021). AICSA-an artificial intelligence cyber security algorithm for cooperative P2P file sharing in social networks. *Data Science and Machine Learning*, 3(1), 252-253.

[11] Ansari, M., Dash, B., & Sharma, P. (2022). The Impact and Limitations of Artificial Intelligence in Cybersecurity. *Advanced Research in Computer and Communication Engineering*, 81-90. https://doi.org/10.17148/IJARCCE.2022.11912

[12] Wendzel, S., Caviglione, L., Mazurczyk, W., Mileva, A., Dittmann, J., Krätzer, C., Lamshöft, K., Vielhauer, C., Hartmann, L., Keller, J., & Neubert, T. (2021). A Revised Taxonomy of Steganography Embedding Patterns. *Association for Computing Machinery*, 1-12. https://doi.org/10.1145/3465481.3470069

[13] Subramanian, N., Elharrouss, O., Almaadeed, S., & Bouridane, A. (2021). Image Steganography: A Review of the Recent Advances. *IEEE Access*, 9, 23409-23423. https://doi.org/10.1109/ACCESS.2021.3053998

[14] Wu, D. C., Shih, Z. N., & Wu, J. H. (2019). Modified Multiway Pixel- Value Differencing Methods Based on General Quantization Ranges for Image Steganography. *IEEE Access*, 10,8824-8839. https://doi.org/10.1109/ACCESS.2021.3138895

[15] Nakaya, D., & Imaizumi, S. (2023). Security Enhancement for Reversible Data Hiding for Palette-Based Images. *Bulletin of the Society of Photography and Imaging of Japan*, *33*, 1:1–6. https://doi.org/10.11454/ephotogrst.33.1_1

[16] Wang, Z., & Zhang, X. (2019). Secure Cover Selection for Steganography. *IEEE Access*, 7, 57857-57867. https://doi.org/10.1109/ACCESS.2019.2914226

[17] Tsai, M. Y., & Cho, H. H. (2021). A High Security Symmetric Key Generation by Using Genetic Algorithm Based on a Novel Similarity Mode. *Mobile Networks and Applications*, 1386–1396. https://doi.org/10.1007/s11036-021-01753-1

[18] Panda, S., Mutallib, M., & Dash, B. (2022). Significance of AI in Electrical Control Systems and Automation. *International Journal of Advanced Research in Computer and Communication Engineering*, *11*, 25-30. https://doi.org/10.17148/IJARCCE.2022.111104

[19] Xu, D., Li, G., Xu, W., & Wei, C. (2023). Design of artificial intelligence image encryption algorithm based on hyperchaos. *Ain Shams Engineering Journal*, *14*(3), 1-8. https://doi.org/10.1016/j.asej.2022.101891

[20] Nie, S. A., Sulong, G., Ali, R., & Abel, A. (2019). The use of least significant bit (LSB) and knight tour algorithm for image steganography of cover image. *International Journal of Electrical and Computer Engineering (IJECE)*, 5218-5226. https://doi.org/10.11591/ijece.v9i6.pp5218-5226.

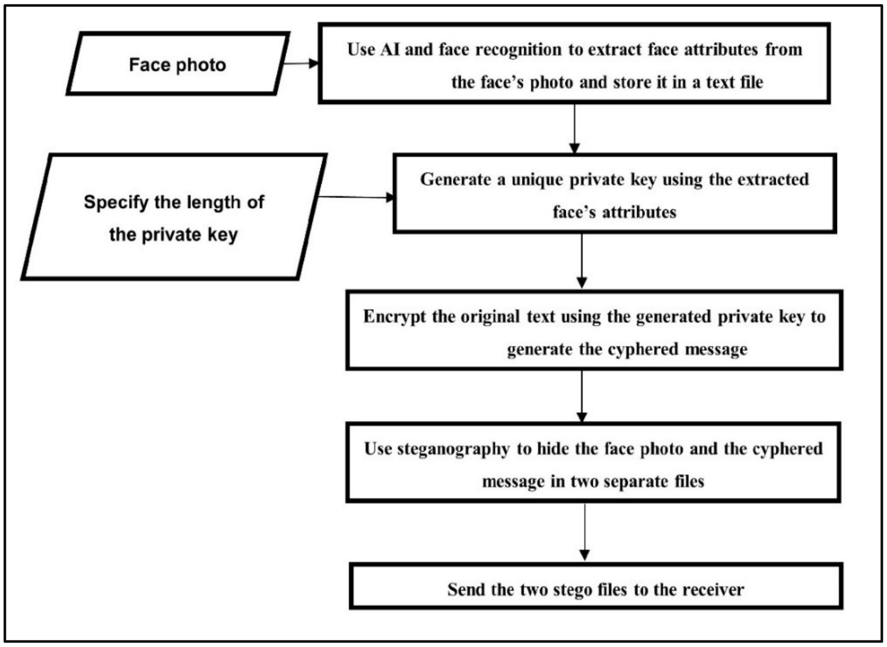

Face photo
Use AI and face recognition to extract face attributes from the face's photo and store it in a text file
Specify the length of the private key
Generate a unique private key using the extracted face's attributes
Encrypt the original text using the generated private key to generate the cyphered message
Use steganography to hide the face photo and the cyphered message in two separate files
Send the two stego files to the receiver

Figure 1.1 Encryption process flowchart

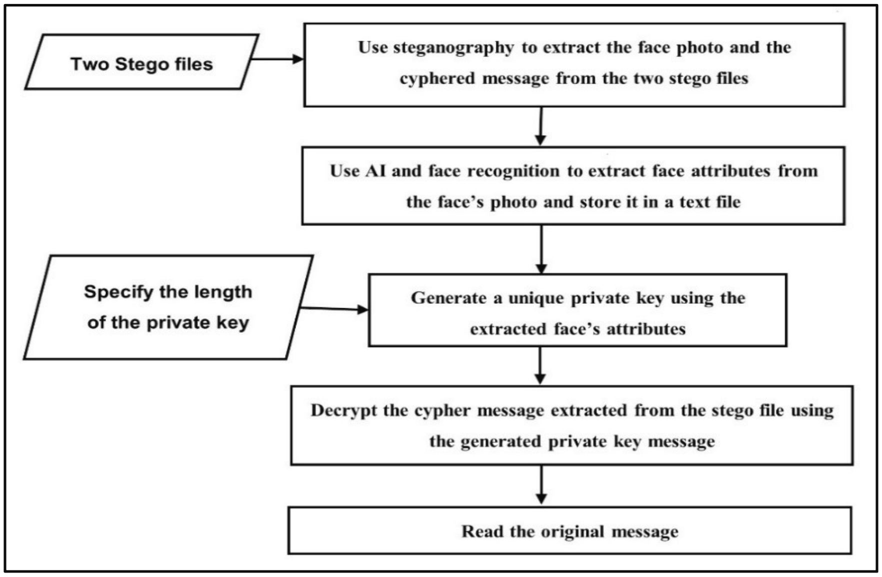

Two Stego files
Use steganography to extract the face photo and the cyphered message from the two stego files
Use AI and face recognition to extract face attributes from the face's photo and store it in a text file
Specify the length of the private key
Generate a unique private key using the extracted face's attributes
Decrypt the cypher message extracted from the stego file using the generated private key message
Read the original message

Figure 1.2 Decryption process flowchart

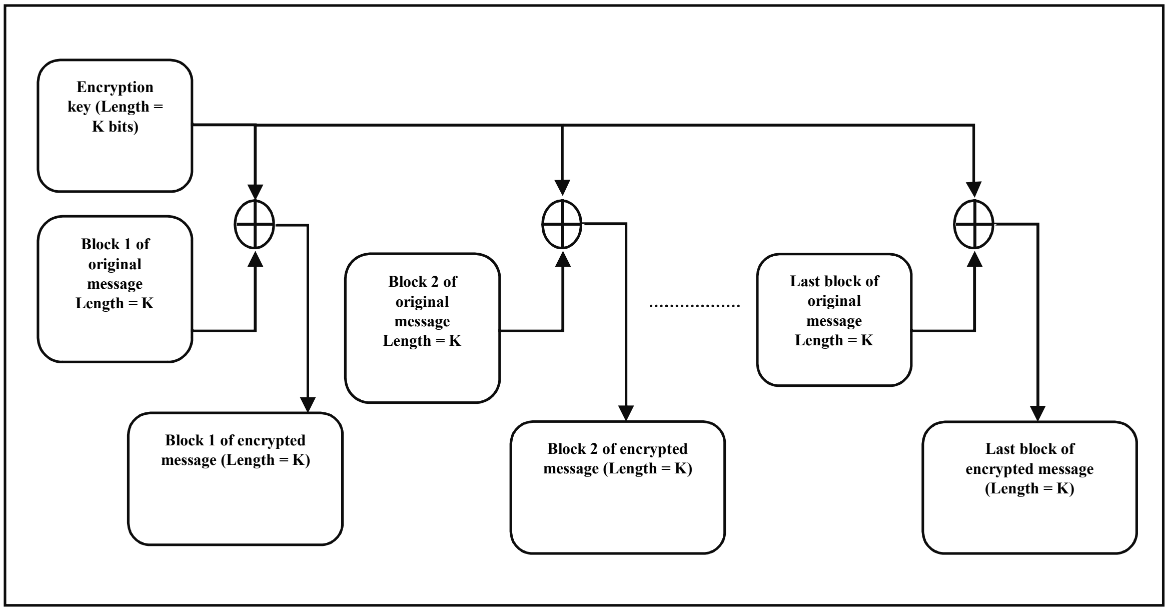
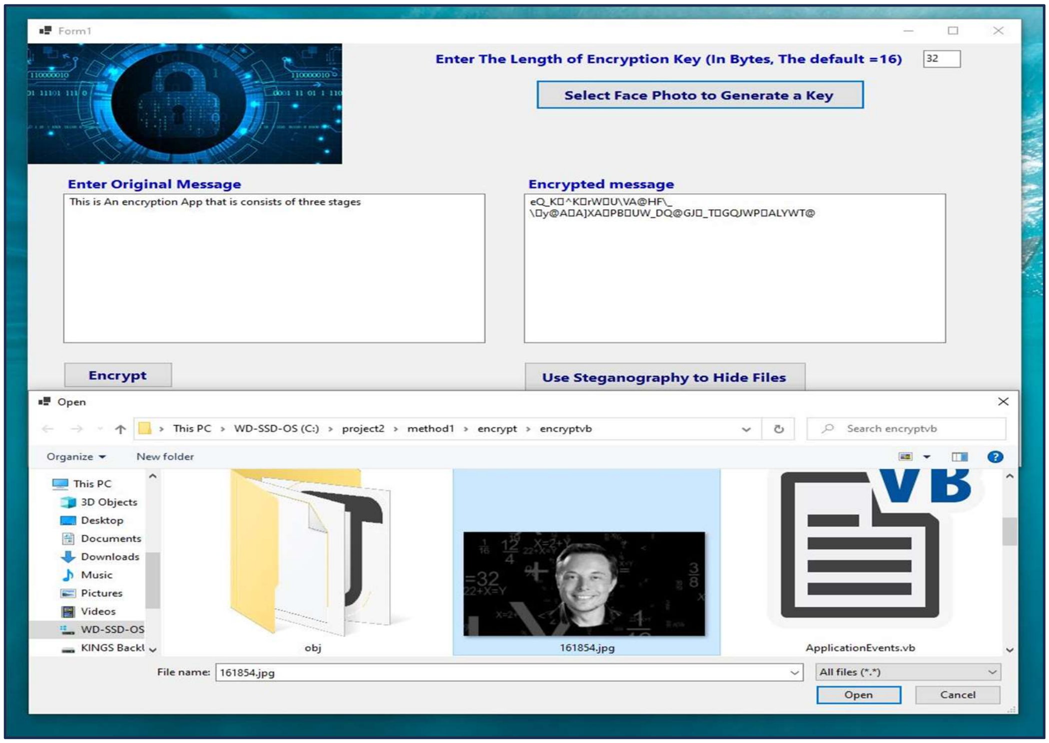

Encryption key (Length = K bits)
Block 1 of original message Length = K
Block 2 of original message Length = K
....
Last block of original message Length = K
Block 1 of encrypted message (Length = K)
Block 2 of encrypted message (Length = K)
Last block of encrypted message (Length = K)

Figure 3.3 Encryption of the original message blocks



Enter The Length of Encryption Key (In Bytes, The default =16) 32

Enter Original Message
This is An encryption App that is consists of three stages

Encrypted message
eQ_K□^K□rW□U\VA@HF_
\□y@A□A]XA□PB□UW_DQ@GJ□_T□GQJWP□ALYWT@

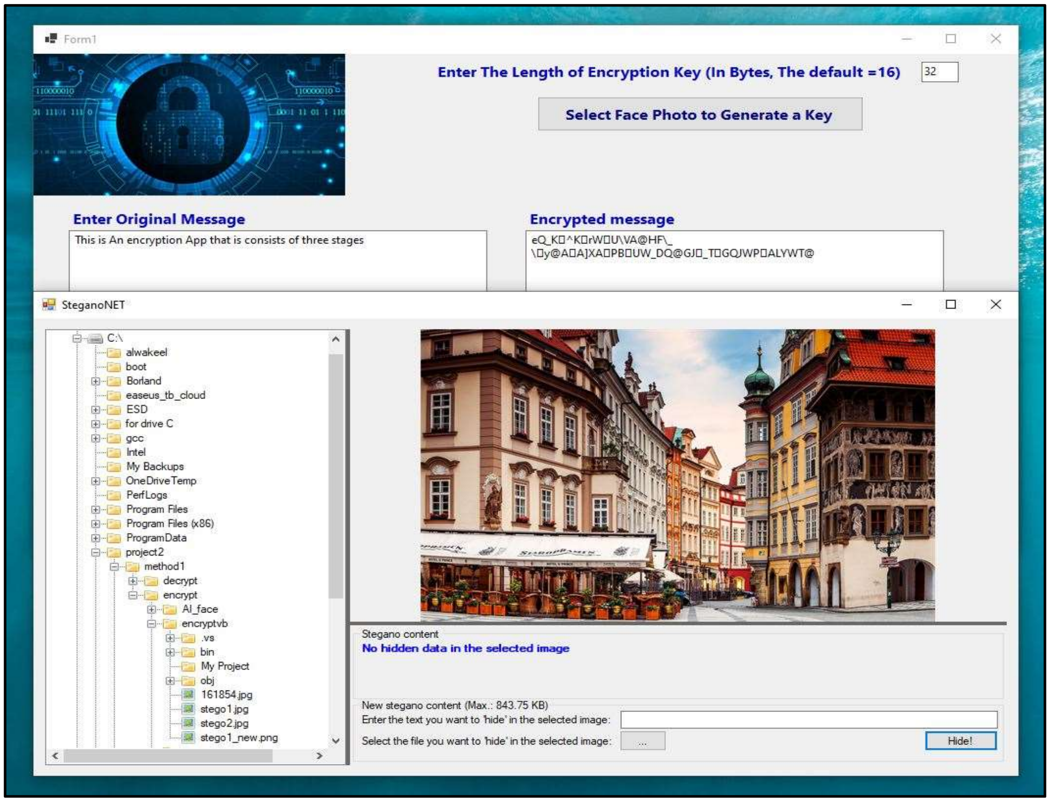
This PC > WD-SSD-OS (C:) > project2 > method1 > encrypt > encryptvb

Search encryptvb

File name: 161854.jpg

All files (*.*)

Figure 3.4 Screen shot of the encryption stage



Enter The Length of Encryption Key (In Bytes, The default =16) 32

Enter Original Message
This is An encryption App that is consists of three stages

Encrypted message
eQ_K□^K□rW□U\VA@HF_
\□y@A□A]XA□PB□UW_DQ@GJ□_T□GQJWP□ALYWT@

Stegano content
No hidden data in the selected image

New stegano content (Max.: 843.75 KB)
Enter the text you want to 'hide' in the selected image:
Select the file you want to 'hide' in the selected image:

Figure 3.5 Screen shot of the steganography stage

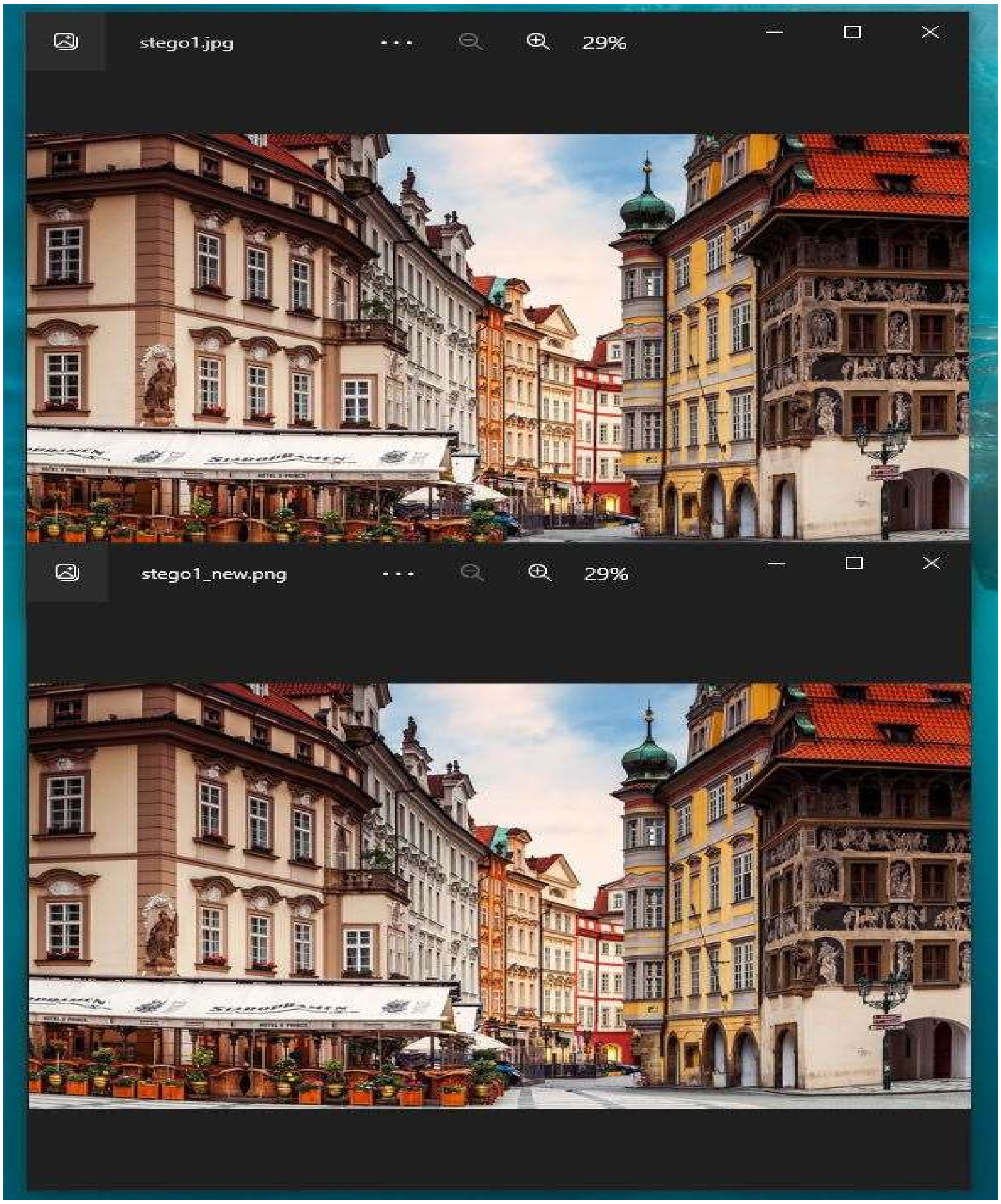


Figure 3.6 Screen shot of the cover image and stego image

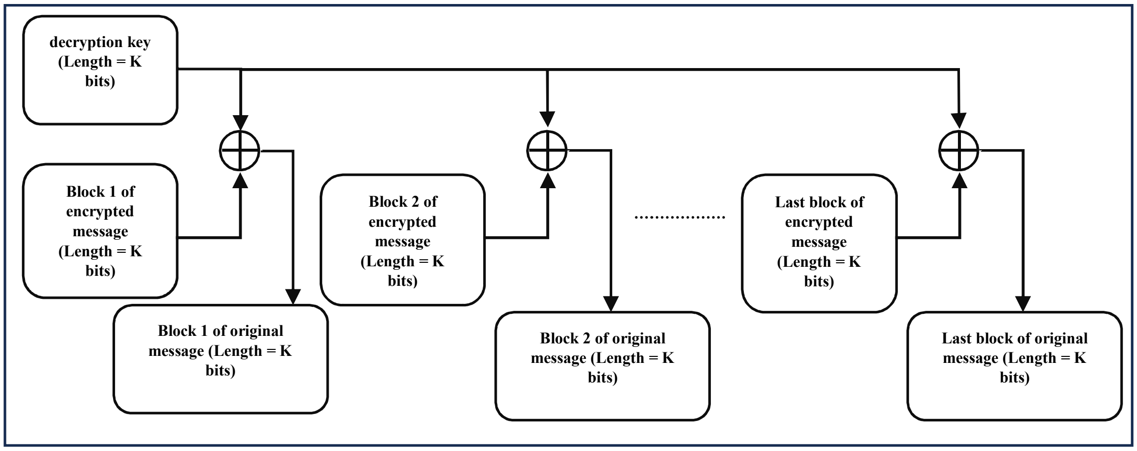
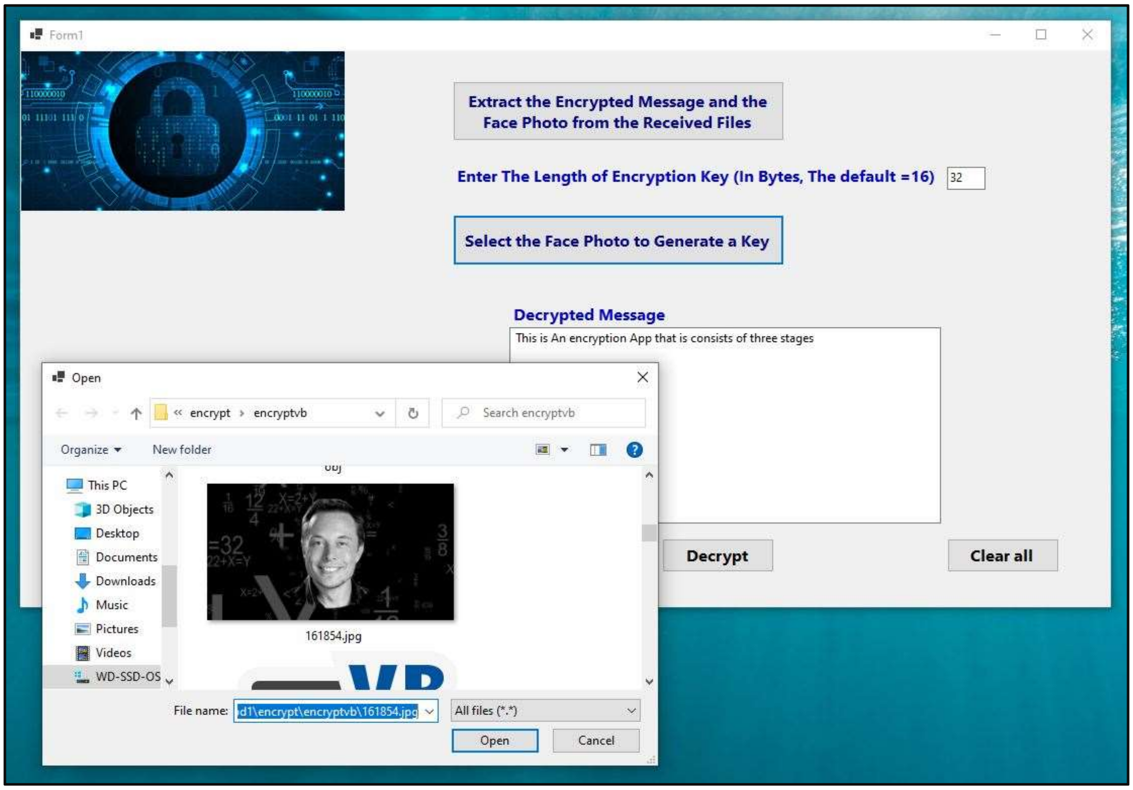

decryption key (Length = K bits)
Block 1 of encrypted message (Length = K bits)
Block 2 of encrypted message (Length = K bits)
....
Last block of encrypted message (Length = K bits)
Block 1 of original message (Length = K bits)
Block 2 of original message (Length = K bits)
Last block of original message (Length = K bits)

Figure 3.7 Decryption of the encrypted message blocks



Enter The Length of Encryption Key (In Bytes, The default =16) 32

Decrypted Message

This is An encryption App that is consists of three stages

File name: d1\encrypt\encryptvb\161854.jpg

All files (*.*)

Figure 3.8 Screen shot of the Decryption Application