An Effective Hybrid Encryption Model using Biometric Key for Ensuring Data Security

Saravanan Arumugam
Coimbatore Institute of Technology,
Anna University, India
a.saravanan21@gmail.com

Abstract: *Cybersecurity becomes a key concern in many applications as cybercrimes exploit system weaknesses. Cryptography helps protect sensitive data in everyday transactions and communications using passwords or tokens. However, the power of the encryption/decryption algorithms always depends on their ability to secure the data in any situation. This paper presents an effective hybrid encryption/decryption model that makes use of a biometric key along with an effective password to ensure data security. The biometric key utilised in the proposed model is generated from the fingerprint, which is a unique physical characteristic of an individual. The model initially encodes the data to be securely transmitted. The symmetric key encryption that makes use of a biometric key is applied over the encoded data. Another layer of defence is built by applying asymmetric key encryption to the encrypted data along with the details of the fingerprint. The Advanced Encryption Standard (AES) algorithm and Elgamal Encryption using Elliptical Curve Cryptography (E^3C^2) are used for symmetric and asymmetric encryptions. Experimental analysis is performed to analyse the model's computing speed and security and is compared with existing models and encoding/encryption techniques.*

Keywords: *Cybersecurity, encryption, decryption, fingerprint based biometric key, data security, symmetric key encryption, asymmetric key encryption.*

Received September 23, 2022; accepted March 6, 2023
https://doi.org/10.34028/iajit/20/5/12

1. Introduction

Security has become a major concern in this digital era due to the increase in the usage of the internet for many trusted communications. It not only increases the cybercrimes but also increases the risk of providing security [35]. Data security over the digital medium is becoming critical in our everyday lives [8, 19]. Most of the applications used for our digital activities employ the benefits of passwords to protect the system from unauthenticated users. These passwords recognise various entities, including bank accounts, mobile phones, laptops, and so on [33]. But the passwords must be chosen with care since many users utilise the same passwords for different systems, which often increases the vulnerability. Remembering different passwords is also difficult, as each user may have an account in at least a minimum of ten applications for which they need to remember their passwords individually [7].

Thus, most high-end defence systems use biometrics to distinguish a legitimate user from an attacker [46]. Biometric traits identify the unique characteristics of the user that replace the use of passwords in the authentication process [30]. The various biometric traits include fingerprint, iris, palm print, ear canal, face recognition, voice recognition, hand geometry, behavioural biometrics like typing speed, signature, and so on. The most common biometric systems utilise fingerprints, face recognition, and voice recognition [16]. However, the authentication process protects the entire system, which acts as a gatekeeper. Unfortunately, if the entire system gets compromised, the data becomes vulnerable to attacks. Thus, a system needs some form of security to secure the data stored in it or communicated for ensuring confidentiality, integrity, and authentication [38].

Cryptography is commonly used to secure data stored or communicated across a medium by applying codes known only to the sender and intended receiver [23]. In general, the encryption techniques can be divided into symmetric and asymmetric ones based on the keys used to encrypt and decrypt the message. Symmetric encryption utilises a single secret key shared by the sender and receiver to encrypt and decrypt the message. On the other hand, asymmetric encryption utilises two keys used by the sender and receiver, which are public (known to everyone) and private (known only to the receiver), in which the encryption is done by the public key of the receiver and the decryption process is done by the receiver through his private key [3]. Some systems utilise hybrid cryptography that makes use of both symmetric and asymmetric encryption models to increase the level of security [21].

Cryptography offers data security while transmitting it through an insecure medium with the use of keys. Here, keys play a precious role, as secure transmission relies on the lengthy key chosen for the cryptosystem [40]. Many systems utilise the keys generated by some functions or algorithms, like a random key generator. However, the keys generated by the functions will never be unique, and sharing the keys in some ways between the sender and receiver is an additional overhead. Despite a long set of procedures for creating the keys, the keys are often cracked using several possible attacks, like guessing or applying brute force attacks. Thus, biometric keys were used to generate a unique key due to cryptography's vulnerability in identifying keys [13]. This method secures data during transmission using cryptography and biometric authentication, increasing cryptosystem security. The most commonly used biometric trait in a cryptosystem is fingerprint identification [25]. And, using biometric verification has many challenges since the actual unique pattern cannot be extracted due to the noise, use of different sensors, and different orientation.

Moreover, several works were proposed based on biometric-based security systems, most of which focus on the authentication process [15] or secured communication [8, 26]. Only a few methods focus on generating keys using various biometric traits for securing data using encryption/decryption processes. Moreover, the computational and time complexity as well as the memory overhead are the major limitations of the existing data security models [11]. The primary advantage of using a biometric key is that it is hard to crack, helping to keep the data secure [25]. Though encryption and biometrics seem like an old concept, they are still in use due to their strong data protection and confidentiality. So, researchers are still looking into ways to use biometrics in cryptography to improve security and get better results.

Thus, to ensure data security, the proposed research work focuses on a hybrid cryptosystem that utilizes a biometric key for generating the encryption key. The general idea is to first encrypt the plain text using a symmetric key algorithm that makes use of secret keys generated using fingerprint based biometric keys, and then encrypt using asymmetric key encryption. This proposed model utilises the benefits of the Advanced Encryption Standard (AES) and Elgamal encryption (EE) using elliptic curve cryptography (ECC) (in short E^3C^2) for the symmetric and asymmetric key algorithms respectively.

Thus, the main contributions of this proposed work are:

a) Proposes a novel hybrid cryptographic model by generating biometric key to enhance data security.
b) Utilizes both biometric authentication and encryption techniques in single system to ensure data security.
c) The model that does not create extraneous difficulties or challenges.
d) The technique that reduces computational and time complexity.
e) It reduces the size of the encrypted file using lossless data compression.
f) It overcomes existing challenges and does not create new weaknesses to the security system.

The organization of the paper is as follows. Section 2 presents the literature survey carried out related to the proposed study. Section 3 describes the proposed hybrid cryptosystem for both encryption and decryption processes. Section 4 presents the detailed experimental and result analysis for the proposed model. Finally, the paper is concluded with the conclusion section along with the scope for future research.

2. Related Works

Several methods exist in the literature that focus on various vulnerabilities, network security, securing the data using cryptography, and biometric key generation from the biometric traits of the user. In today’s digital world, any user needs to exchange and share sensitive confidential data, which stimulates the intruders to initiate various attacks on the network for accessing the sensitive data [36]. To ensure the user's integrity and trustworthiness in the distributed environment, authorization policies have been framed to access sensitive data [34]. But this secures the entire system but not the data effectively and completely.

Several studies use the hybrid model to secure data due to cryptography and biometric advantages. In recent days, biometric keys for encryption have been most widely researched for protecting digital content [25, 26]. The biometric traits used for key generation include fingerprint [11], iris [1, 2], and facial features [5], among which most of the biometric key generation functions utilise fingerprint based biometric keys due to their cost-effective nature in extracting fingerprints from an individual [44]. The proposal put out a middleware framework in which users' private keys would be used to access the confidential information. A middleware framework was proposed in which users' private keys were used to access the confidential information [20].

Getting the features from the fingerprint and using them to generate the keys was one of several ideas. Obviously, the stable features extracted from the finger impression improve the accuracy of key generation [39]. Some models make use of scoring techniques to authenticate the user using a fingerprint [31] or unique key generator in various forms, such as a biometric key [8, 19] or Quick Response (QR) code [4]. Similarly, a novel approach of generating a fingerprint template that can reproduce the fingerprint from the stored template was suggested for an automatic fingerprint recognition system [43]. A hybrid verification technique with the use of biometric and encryption systems was proposed [15]. It uses a fingerprint based biometric technique along with the AES to implement a trustworthy cryptosystem specifically for cloud environments. A two-tier architecture for ensuring security in a multi-cloud platform was introduced, that uses two encryption techniques such as Paillier and Blowfish algorithms to secure the data storage and computation [37].

Multimodal symmetric hybrid cryptography that is simple to implement and suitable for small and medium-scale industries was proposed [22]. This multimodal cryptosystem makes use of an enhanced hill cipher algorithm with modulo 37 and a combination of simple positive and negative integers that offers high computational speed. Similarly, biometric keys can also be generated through handwritten biometrics [10] based on the given constraints. Here the idea is to split the feature space into cells in which each cell participates in generating the biometric key [14].

Fuzzy Fingerprint Biometric Based Key Security (FFBKS) was introduced to secure the data transmission through a wireless sensor network. It utilises feature extraction by producing a unique private key for the user [24]. Fuzzy Public-Key Encryption (FPKE), which is a variation of a public-key encryption scheme, was proposed that exploits fuzzy data (generated by adding noise to the biometric) as a unique key to decipher the cipher [9]. Similarly, to provide more security, a secure key with an optimised fuzzy extraction was suggested that even addresses unauthorised access [27].

Though there exist several models for securing the data through biometrics and cryptography, many of the methods suffer implementation and computational complexity. In other cases, many models consume higher costs in implementing the system which may not be suitable for all applications. Some approaches are quite simple and may not be suited for applications where data security is a critical concern.

3. Proposed Hybrid Encryption Model

The overall architecture of the proposed hybrid encryption model is shown in Figure 1. Here, the plain text is encoded into a particular format, which is then encrypted using symmetric key encryption.

The key is created by concatenating the shared passcode with the biometric key, which is generated from the fingerprint of the user who initiates the communication. The obtained ciphertext is concatenated with a few details of the fingerprint, which will be helpful for the receiver to generate the biometric key from the given detail. Again, the concatenated text undergoes asymmetric key encryption, which encrypts the text using the receiver's public key. The obtained final ciphertext can be communicated to the receiver. The model has the ability to encrypt any form of data, like text, images, and voice. Thus, any data is initially encoded into a specific format, which is then encrypted further to ensure data security. This model can be applied to various applications and uses encoding, symmetric and asymmetric key encryptions, biometric passcodes, to secure critical data.

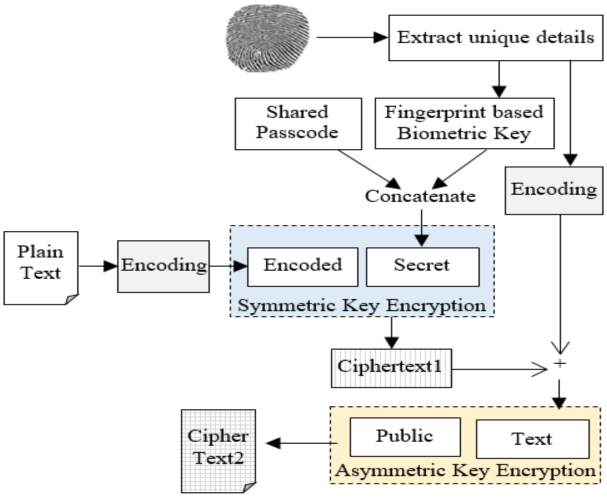


Figure 1. Proposed hybrid encryption using a biometric key.

The proposed model uses Arithmetic Coding (AC) [45] to encode the information that needs to be kept safe. This method encodes the data to provide additional security along with performing data compression. AC, an entropy-based model uses a lower number of bits to compress the data by computing the probability for each symbol at each interval. It encodes or compresses the entire message into a single floating-point number. The general idea to reduce the size of the data is that the most frequently occurring symbols will be assigned fewer bits than the least frequently occurring symbols. This step also helps to convert the given data, which can be in the form of text, image, video, or audio, into a particular form suitable for applying an encryption algorithm.

The encoded data is encrypted using symmetric key encryption, in which the cryptographic key used is a combination of a 64-bit shared passcode, a key shared between the sender and the receiver, and the 64-bit biometric key generated from the unique details of the user’s fingerprint. To generate the biometric key, the user's fingerprint is given as input, which is then pre-processed and the core point of the fingerprint is identified. The unique features, such as minutiae, are identified from the fingerprint image with the core point as the reference. From the well-recognized minutiae, the coordinates are extracted, from which the 64-bit biometric key is generated. The secret key for symmetric key encryption is the 128-bit string obtained by concatenating the shared passcode and the biometric code. The symmetric key encryption used in the proposed model is AES. Due to its speed and low memory requirement, the model is extensively used.

To decrypt the plaintext that was encrypted with the AES algorithm, the user must know the secret key that was used to encrypt it. However, the fingerprint of the sender is a unique biometric feature, and the receiver will be unaware of the unique details of the fingerprint, which are then communicated along with ciphertext. Thus, the extracted unique details are encoded using AC and concatenated with the ciphertext obtained from the symmetric key encryption process. This concatenated text is again encrypted using asymmetric key encryption. The asymmetric key encryption used is improved E^3C^2, which uses the receiver's public key to encrypt messages [41]. This Elgamal encryption variation is more secure than other public key methods. This encrypted text is then transmitted to the receiver.

3.1. Arithmetic Coding

AC is an entropy-based encoding specifically used for lossless data compression. The main focus of the algorithm is to reduce the size of the data to a rational number that lies between 0 and 1. Initially, the interval starts with [0, 1), and at each iteration the interval is subdivided into subintervals. Each symbol in the input file is processed one at a time and assigned a subinterval that lies between 0 and 1 according to its probability of occurrence. For larger input, the number of bits to represent the interval will be the minimum. On the other hand, the number of bits needed to represent the interval will be high for short inputs. The final interval derived after all the iterations is the codeword of the given input text. This method takes the data to be encoded as well as the frequency table of all possible symbols in the given text, which are obtained from the previous messages, as input for the algorithm. The frequency of symbols is then converted into probabilities. Then the message is encoded according to the probability of the symbol.

Thus, for each symbol *S*, the range is computed for all symbols in which it begins from the value *b* and ends at the value *e* which is computed using Equation (1) as:

$$e = C + (P(S) \times R) \tag{1}$$

Where *C* is the cumulative sum of all the probabilities of the given input text, *P(S)* is the probability of the symbol *S*, and *R* represents the range that starts by subtracting the beginning and end of the range. The average of the starting and ending values of the final range represents the encoded value, which will be a real number. The algorithm for AC is presented in Algorithm (1) [6].

Algorithm 1: Arithmetic coding

```
Input: Plaintext, frequency values
Output: Encoded text
Begin
  Define the initial interval between 0 and 1.
  For each symbol in the given input message
    1. Identify the subinterval from the previous interval one
       for each symbol in which the size of the subinterval
       corresponds to the probability of the next symbol.
    2. Select the identified subinterval to the next symbol and
       assign it as a new subinterval.
  Average of final range represents encoded and compressed
  text.
End Procedure
```

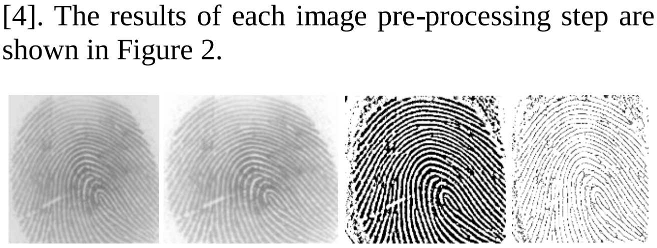
AC is superior to Huffman Coding (HC). HC transmits the compressed data in the form of the Huffman table. However, AC transmits the compressed data with respect to its length. Many of the steps in lossless compression end with generating headers or other predictable patterns that make the cryptanalysis easier. Thus, while choosing encoding techniques for cryptography, care must be taken that they do not produce any patterns that degrade the security of the system. Though several variations of AC have been developed, they suffer from higher computational complexity with additional headers [42]. Therefore, the proposed model makes use of simple AC for effective lossless compression based on entropy encoding.

3.2. Fingerprint based Biometric Key Generation

For symmetric key encryption, the user's fingerprint is used as an image input to make the secret key. Though there are several biometric traits, the fingerprint is considered to be the most promising yet simple to use, and the tools to capture the fingerprints are cost-effective. The fingerprint represents the unique pattern of friction ridges from the finger impression. Each fingerprint impression is uniquely identified based on the analysis of various extracted features or parts of the fingerprint. Thus, in the proposed model, the input image undergoes several processing steps such as pre-processing, identifying unique points, and generating the biometric key in accordance with the generation of a unique value from the thumb impression of the user. The processing steps are explained below.

3.2.1. Pre-Processing

The intensity values of the fingerprint images are normalized by applying the Contrast Level Adaptive Histogram Equalization method (CLAHE) [28, 29]. This method enhances the fingerprint image by normalising the grey levels and optimising the contrast level accordingly. Once the image is enhanced, the unique points of the image are accurately identified, and the image is binarized by marking each pixel either 1 or 0. Then, the ridgelines are thinned in order to remove the duplicate pixels in the ridgelines. Ridge thinning is responsible for thinning the ridgelines by a single pixel [4]. The results of each image pre-processing step are shown in Figure 2.

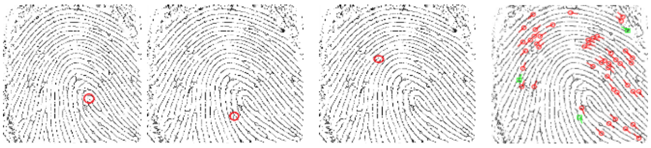


a) Input image. b) Image enhancement. b) Image binarization. d) Image thinning.

Figure 2. Fingerprint image pre-processing steps.

3.2.2. Unique Point Identification

Upon pre-processing the fingerprint image, the unique points of the fingerprint images are identified. The core point is one of the unique points, which can be defined as the top point in the innermost ridges. Identification of the core point is a significant step as it helps to generate a unique key even if the fingerprint images of the same individual are taken in different orientations. Several methods exist for detecting the core point of the fingerprint. This proposed model makes use of the Poincaré index method by transforming the given fingerprint image into an orientation image [18]. Once the core point is identified, the minutiae points are to be identified by taking the core point as a reference. Minutia is a discontinuity in the ridgeline that interferes with the smooth flow at an end, bridge, or bifurcation. However, ridge termination and bifurcation are the most widely used minutia in fingerprints. The core point of the fingerprint and sample minutia with ridge bifurcation and termination, along with the minutiae points detected, are shown in Figure 3.



a) Core point. b) Ridge bifurcation. c) Ridge termination. d) Minutia detection.

Figure 3. Core point and minutiae of a fingerprint image.

This set of minutiae is the features of the fingerprint that are used to identify the user uniquely. There are several methods for identifying the minutiae points in the fingerprint image. The proposed model utilises CNs method. Here, the intensity value of each pixel is compared with those of its adjacent pixels, and the identification of minutiae is carried out by computing the CN as half of the difference between the adjacent pixel intensity values. Crossing Number (CN) 1 represents the ridge termination, 2 represents the normal ridge, and 3 represents the ridge bifurcation [31]. Each type of minutia detection is presented in Figure 4 with a window size of 3x3.

1	0	0
0	1	0
0	0	0

a) Ridge termination (CN = 1).

1	0	0
0	1	0
0	0	1

b) Continuing ridge (CN=2).

0	1	0
0	1	0
1	0	1

c) Ridge bifurcation (CN=3).

Figure 4. Identification of minutiae points using CN of 3x3 window.

3.2.3. Biometric Key Generation

As the result contains a huge number of minutiae points, to identify the minutiae that are unique for generating a unique key, the Region Of Interest (ROI) with the core point as its centre is identified. ROI is a region that can be represented as a fixed square with a core point as its centre. Thus, only the minutiae that are inside the ROI are considered for generating the biometric key [4]. The coordinates representing the position of minutiae points concerning ridge termination and bifurcation along with its angle are extracted. Here, the location of the minutiae points in which the origin is shifted to the core point is the coordinates of the minutiae (x, y), and the angle between the horizontal axis and the corresponding ridgeline of the termination and bifurcation is the orientation angle of the minutiae (θ). Thus, this step results in the set of minutiae inside the ROI that is symbolised as (x, y, θ). Finally, the average values of the x, y, and θ are computed, and each mean value is converted to 64 bits, for which the XOR operation is performed [19]. This 64-bit value is the generated biometric key. For the given input fingerprint shown in Figure 2, the proposed model detects 43 minutiae points, for which the mean value for x and y coordinates concerning the core point as origin, along with the angle value, are computed and converted to a 64-bit value, which is then XORed to generate a biometric key. The complete workflow for generating the biometric key from the given fingerprint image is shown in Figure 5.

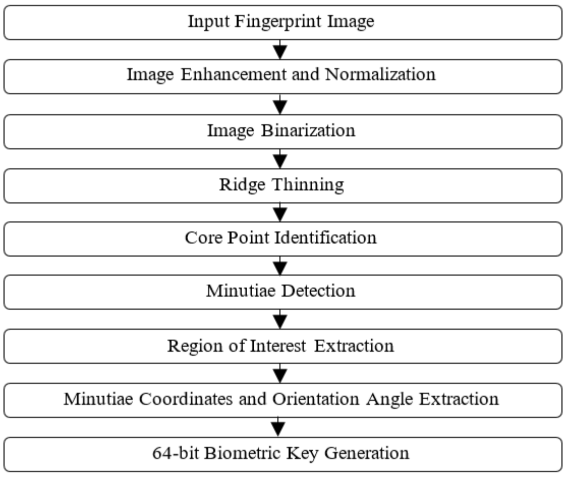

Input Fingerprint Image
Image Enhancement and Normalization
Image Binarization
Ridge Thinning
Core Point Identification
Minutiae Detection
Region of Interest Extraction
Minutiae Coordinates and Orientation Angle Extraction
64-bit Biometric Key Generation

Figure 5. Complete workflow of biometric key generation.

However, the receiver does not have the fingerprint of the sender, and so the mean value of the x and y coordinates along with the orientation angle are encoded and sent along with the ciphertext. The receiver on the other side extracts the mean values and performs an XOR operation to obtain the secret key.

3.2.4. Secret Key Generation

However, to perform symmetric key encryption, the secret key (K_s) is to be generated. Here, both the sender and receiver share the same Passcode (PC) which is then converted to 64-bit binary values. If the passcode exceeds 64 bits, then the extra bits are truncated or if it has minimum bits, then 0's are padded at the beginning to make the length 64 bits. Finally, the secret key can be generated by concatenating the 64-bit passcode with the 64-bit generated biometric key as given in Equation (2).

$$K_s = PC + (\bar{x} \oplus \bar{y} \oplus \bar{\theta}) \tag{2}$$

The algorithm for fingerprint based secret key generation is given in Algorithm (2).

Algorithm 2: Fingerprint based Symmetric Key Generation

```
Input: Fingerprint Image, Shared Passcode PC
Output: Secret key
Begin secret_key_generation()
  sum_x = 0, sum_y = 0, sum_θ = 0, count=0
  //Preprocesing the input image
  Apply contrast level adaptive histogram equalization on input file
  Binarize the image into 0's and 1's.
  Perform ridge thinning on the image with one-pixel width.
  //Unique Point Identification
  Identify core point of fingerprint using Poincaré Index method
  Compute CN for each pixel to detect the minutia points
      If CN is 1 or 3 then Mark the pixel as a minutiae point
  Shift the origin to the core point of the fingerprint
  Extract ROI of the processed image according to the core point
  For each minutiae point inside the ROI
    Extract location coordinates (x, y) respect to the core point
    Compute θ between x-axis & minutiae ridgeline
    sum_x=sum_x + x; sum_y=sum_y + y; sum_θ=sum_θ + θ;
    count = count + 1
  End For
  mean_x = sum_x/count; mean_y = sum_y/count;
  mean_θ = mean_θ/count;
  biometric_key = b(mean_x) ⊕ b(mean_y) ⊕ b(mean_θ);
  secret_key = b(PC)+biometric_key;
End Procedure
```

3.3. Symmetric Key Encryption

The plain text encoded using AC is encrypted using the generated symmetric secret key. The proposed model utilises the AES algorithm, a block cipher that is a type of symmetrical key algorithm that uses the same key for both encryption and decryption [32]. The model is most widely used because it is faster than many other encryption algorithms, especially triple Data Encryption Standard (DES). It operates on bytes using an iterative procedure with substitution (replacing the bytes) and permutation (replacing the bytes) at each iteration. It takes a 128-bit block of input (16 bytes) in the form of a matrix of size 4X4 and produces a 128-bit block of ciphertext. It works on various key sizes, and the number of iterations or rounds depends on the size of the key. The key sizes 128, 192, and 256 have 10, 12, and 14 rounds, respectively, as the larger key size takes more rounds to complete. At each round, it uses operations such as substituting bytes, shifting rows, mixing columns, and adding the round keys. The proposed model exploits the key size of 128 bits, so the number of rounds to be iterated is 10.

The ciphertext obtained from the AES algorithm is then encrypted using an asymmetric key algorithm. As the receiver must know some details about the fingerprint, the mean values of the coordinate points and the orientation angle are then encoded using AC and are then concatenated along with the ciphertext obtained from the symmetric key algorithm, and this will become the input for the asymmetric key algorithm.

3.4. Asymmetric Key Encryption

In general, the most widely used symmetric key algorithm is Rivest, Shamir, Adleman commonly known as (RSA). The increase in the security of the RSA algorithm depends on the increase in the key length. Unfortunately, increasing the key length degrades the performance as it slows down the system. Thus, Elliptic Curve Cryptography (ECC) has become a significant alternative for RSA in a rapidly growing digital era. The main operations of the ECC are the addition and multiplication of points on the elliptic curves. The proposed model makes use of E^3C^2 [41]. Elgamal cryptosystem uses elliptic curve discrete alogarithm problem which needs to find a value k where $P \circ k=Q$ as well as P, Q fits the similar points G on an elliptic curve.

E^3C^2 encrypts the message m using α^k and β^k in which α is the primitive root of a prime p, k is a random integer, and β is computed as α^a where a is the secret key of the receiver. In this model, the set of values (α, β, p) is public and known by both the sender and the receiver. The encryption for the message m is given in Equation (3).

$$Encrypt\ (m) = (\alpha^k, \beta^k m) \tag{3}$$

The receiver decrypts the message m using the secret value a and the received ciphertext $(\alpha^k, \beta^k m)$ as in Equation (4) which can be rewritten as in Equation (5).

$$Decrypt\ (\beta^k m) = (\alpha^k)^{-a} \times (\beta^k m) \tag{4}$$

$$(\alpha^{-a})^k \times (\beta^k m) = (\beta^k m) \times (\beta^k m) \Rightarrow m \tag{5}$$

Where α and β are the points on the elliptic curve with the addition and multiplication of points.

Consider the elliptic curve C as $y^2=x^3+bx+c$, $mod\ p$, and a secret a. Consider a point $\alpha(x_1, y_1)$ that lies on the curve and the point $\beta(x_2, y_2)$ computed as $\beta=a \circ \alpha$. With the public values p, α and β and an elliptic curve C, the sender encrypts the message m by performing integer modulo with prime p. The sender utilises the random number k to compute two additional points as $r(x_3, y_3) =(x_3, k \circ \alpha)$ and $t(x_4, y_4=(x_4,m+k \circ \beta)$ where $\circ$ represents the multiplication and + represents the addition. The user sends the y values (y_3, y_4) to the receiver. With this, the receiver decrypts the message m as y_4-$(a \circ y_3)$. Substituting the values of y coordinates results in the message m and is given in Equation (6).

$$(m + k \circ \beta) - a \circ (k \circ \alpha) = m + (a \circ k \circ \alpha) - (a \circ k \circ \alpha) \tag{6}$$

3.5. Hybrid Decryption using Biometric Key

The decryption procedure for the proposed hybrid model is the inverse process of the encryption steps. The received ciphertext is decrypted using E^3C^2, an asymmetric key algorithm, and the receiver's private key. The decrypted text contains the encoded details of the fingerprint values as well as the ciphertext obtained from symmetric key encryption. The details of the fingerprint values are decoded and processed to obtain the biometric key. Then the shared passcode is concatenated with that of the biometric key, which is applied as a key to the AES algorithm to decrypt the ciphertext extracted from asymmetric key decryption. This results in encoded text, which is then decoded using AC to obtain the plain text. The process diagram for the proposed hybrid decryption model is shown in Figure 6.

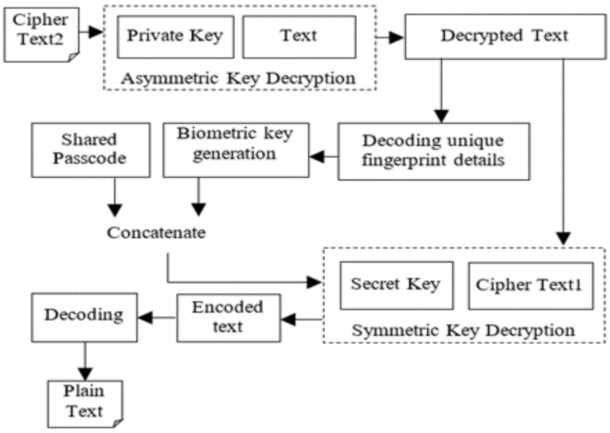


Figure 6. Proposed hybrid decryption using a biometric key.

4. Experimental Analysis

An experimental analysis has been made to analyse the performance of the proposed model. For evaluation, the following hardware characteristics are utilized: a system with 8.00 GB of RAM and an Intel (R) Core (TM) i3-4005U Central Processing Unit (CPU) at 1.70 GHz. The software utilised for the model implementation is Java. Various analyses are made concerning the execution time of the proposed model and individual modules used in the model by varying the size and type of the input file. The obtained results are analysed and compared with the existing algorithms. The various parameters used in the analysis are biometric key generation time, encryption time, and decryption time. Biometric key generation time is the time taken by the system to generate a biometric key from the given input fingerprint image by the sender. On the receiver side, biometric key generation time will be the time taken to generate the key from the given values that represent the fingerprint. Encryption time is the time taken by the sender to encrypt the content with a key known only to the intended receiver. Decryption time represents the time taken to decipher the received text using the key [37].

4.1. Result Analysis

Initially, an analysis on generating fingerprint based biometric keys is made, and the results obtained are assessed. The execution time for generating the biometric key by the sender and receiver is evaluated with the fingerprints obtained from five users, which undergo scanning, preprocessing, identification of the core point, extraction of the location of minutiae with reference to the core point, and computing the key based on the location of minutiae points. The input image for evaluating the biometric key generation is listed in Figure 7. The computing time for each of the five fingerprint images is presented in Table 1.

a) Image 1. b) Image 2. c) Image 3. d) Image 4. e) Image 5.

Figure 7. Sample fingerprint images used.

From the analysis, the average computation time for generating the biometric key obtained from the scanned fingerprint images by the sender is 0.1752 seconds, whereas that of generating the binary key from the details of the fingerprint image by the receiver is 0.0256 seconds. Because the computation complexity of generating the biometric key is low, the result appears to be quite impressive.

Table 1. Biometric key generation.

Image name	**Key generation time (seconds)**	
	Sender	**Receiver**
Image 1	0.131	0.014
Image 2	0.125	0.019
Image 3	0.223	0.021
Image 4	0.184	0.043
Image 5	0.213	0.031

To evaluate the performance of the AC, various types of files have been encoded, for which the ratio of compressed data along with the time to encode the data are analysed. Three types of files, such as byte files, text files, and image files, are utilised in the analysis. The files, namely file1 and file2, are byte files with low and high redundancy. The files specified as files 3 and 4 are text files with English characters, whereas files 5 and 6 are image files of simple and difficult images [17]. The results of AC are compared with those of existing HC and its variations such as Adaptive Huffman Coding (AHC) and Canonical Huffman Coding (CHC). The size of the encoded files, the ratio of compression, and the time taken to encode the file for various algorithms are presented in Table 2.

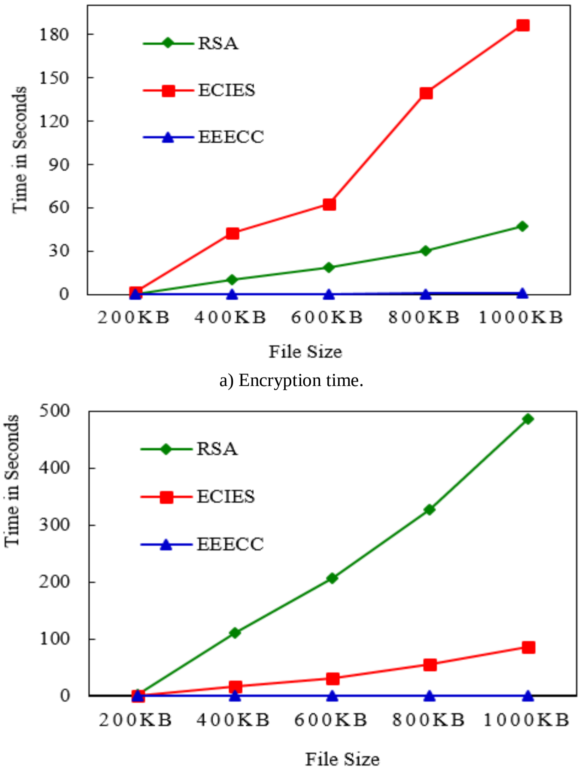
Table 2. Comparison of various encoding techniques.

File name		**File 1**	**File 2**	**File 3**	**File 4**	**File 5**	**File 6**
File size		102358	214589	124856	156789	132589	159634
HC	**Encoded size**	83256	67896	89632	81457	96512	81236
	Ratio	0.1866	0.6836	0.2821	0.4804	0.2721	0.4911
	Time (secs)	**1.28**	**2.45**	1.63	1.82	1.69	2.32
AHC	**Encoded size**	**82147**	62478	87112	80147	92348	82369
	Ratio	**0.1974**	0.7088	0.3023	0.4888	0.3035	0.4840
	Time (secs)	1.59	2.87	**1.55**	1.96	1.18	2.48
CHC	**Encoded size**	82478	61478	83692	**79584**	**90489**	80478
	Ratio	0.1942	0.7135	0.3296	**0.4924**	**0.3175**	0.4958
	Time (secs)	1.54	2.96	1.71	**1.76**	1.49	2.37
AC	**Encoded size**	82345	**60147**	**82369**	79699	90896	**80079**
	Ratio	0.1955	**0.7197**	**0.3402**	0.4916	0.3144	**0.4983**
	Time in (secs)	1.47	2.57	1.59	1.91	**1.06**	**2.21**

From the obtained results, it is clear that the AC has a better compression rate than most of the algorithms under comparison. Also, when focusing on the execution times of various encoding techniques used for lossless compression, AC has a shorter execution time than HC, AHC, and CHC in most of the cases. Also, the average time taken for HC, AHC, and CHC is 1.87, 1.94, and 1.97 seconds, respectively, and for AC, the average execution time is 1.8 seconds. Thus, when compared with other models, AC offers better compression within an optimised timeframe in many cases. Consequently, the model utilises the power of AC to secure the data in a better way.

The asymmetrical algorithm used in the model is E^3C^2, which has been compared with various other algorithms such as RSA and the Elliptic Curve Integrated Encryption Scheme (ECIES). The analysis has been made by varying the input files of different sizes: 200KB, 400KB, 600KB, 800KB, and 1000KB. The results obtained are compared as a graph in Figure 8.

From the analysis, the encryption time of RSA is less than the decryption time, whereas the encryption time of the ECIES algorithm is higher than the decryption time. The average time taken for encryption using RSA and ECIES is 21.16 seconds and 86.78 seconds, respectively, and the decryption time for RSA and ECIES is 226.39 seconds and 37.99 seconds, respectively. The E^3C^2 algorithm is computationally strong, as it takes the minimum time for both encryption and decryption at 0.398 and 0.294 seconds, respectively



a) Encryption time.

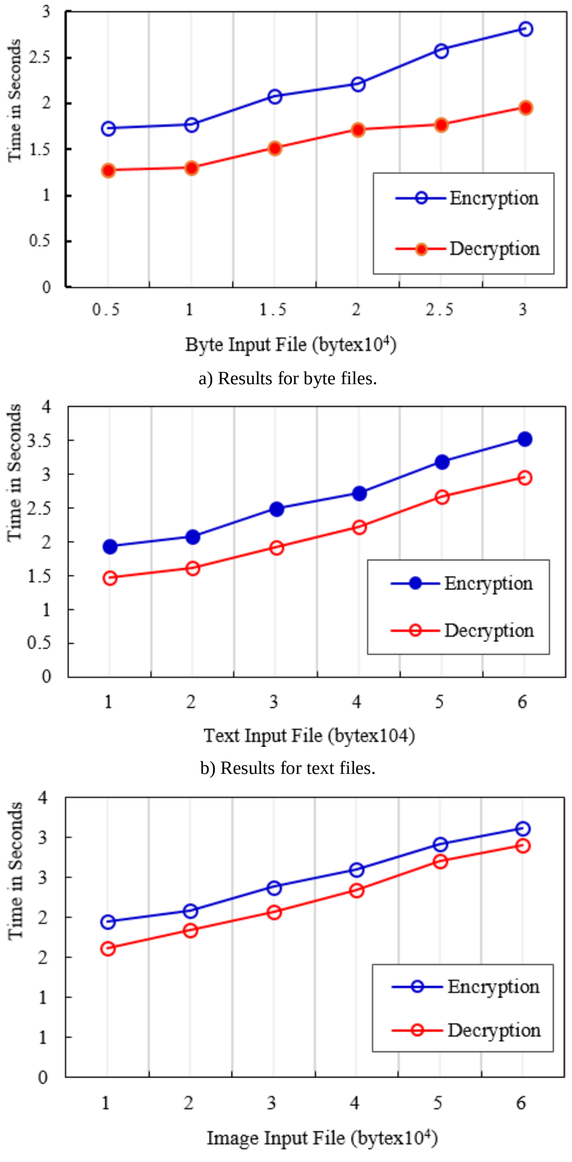
b) Decryption time.

Figure 8. Encryption and decryption time of various asymmetric key algorithms.

4.2. Performance Analysis

The performance of the entire model has been analysed by executing it and computing the time taken for encryption and decryption. This has been carried out by varying the types of input files of different sizes. The analysis utilises various types of files as input, including byte files, text files, and image files. Each type of file is analysed by varying the input size as follows: 5000 bytes, 10000 bytes, 15000 bytes, 20000 bytes, 25000 bytes, and 30000 bytes, respectively, as in [15].

The average time taken to encrypt the byte file is 2.199 seconds and decrypt the byte file is 1.593 seconds. Similarly, the time taken to encrypt and decrypt the text files is 2.66 and 2.145 seconds, respectively, and for the image files, the encryption time and decryption time are 2.508 and 2.248 seconds, respectively. Thus, the proposed model has a minimum time to process byte and text files. The results obtained for byte files, text files, and image files are presented in Figure 9.



a) Results for byte files.

b) Results for text files.

c) Results for image files.

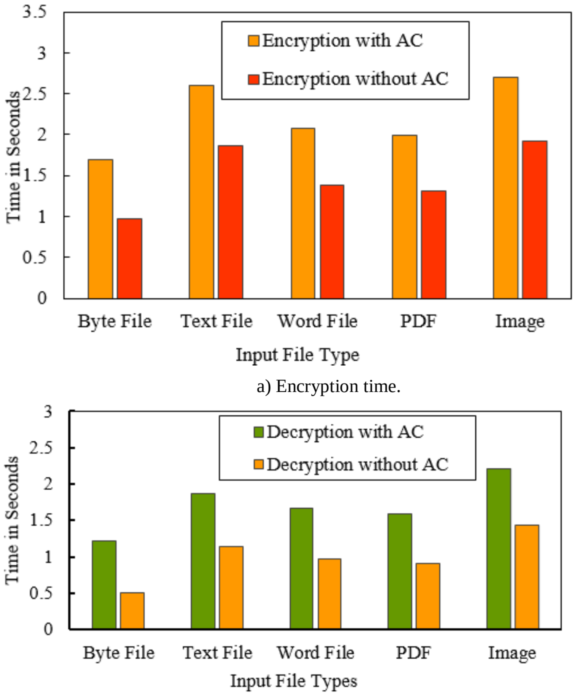
Figure 9. Encryption time and decryption time for different file types.

An analysis has been made with different types of files, such as byte file, text file, word file, pdf file, and image file. The size of the encrypted file, encryption time, and decryption time after applying the proposed hybrid model with and without the AC are evaluated, and the obtained results are presented in Table 3.

Table 3. Comparison of execution time for various file types.

File type		Byte	Text	Word	PDF	Image
File size (Bytes)		8145	15432	10896	9874	20895
Encrypted file size	With AC	6789	11304	8604	8109	14885
	Without AC	8337	15624	10908	10066	20907
Encryption time	With AC	1.689	2.598	2.081	1.987	2.697
	Without AC	0.977	1.867	1.38	1.309	1.915
Decryption time	With AC	1.214	1.872	1.673	1.589	2.214
	Without AC	0.502	1.141	0.972	0.911	1.432

Here, the size of the encoded file is minimal when compared with the model implemented without AC. The decrease in the size of the file after applying the AC is 22.8%, 38.2%, 26.8%, 24%, and 40.5% for byte, text, word, pdf, and image files, respectively. The time taken to encrypt and decrypt various types of files using the proposed model with and without AC is shown in Figure 10. Though the time taken for encoding the files directly influences the execution time of the proposed model, it can be negligible when considering the security provided to the input file by the proposed hybrid model.



a) Encryption time.

b) Decryption time.

Figure 10. Analysis of encryption and decryption time for different file types.

4.3. Comparison with the Existing Models

The performance of the proposed biometric key based encryption technique is also compared with that of other similar existing works. A similar model was proposed by Goyal and Kant [12]. Though the method uses several strong cryptographic algorithms, such as AES, SHA-1, and ECC, it does not use a biometric key for encryption. Thus, the method is vulnerable to various attacks. However, the proposed method makes use of biometric key based encryption, which reduces the attack vulnerability. Additionally, existing methods such as a method for authenticating the cloud storage using biometric based encryption technique takes 1.03 seconds [15], the encryption and decryption processes in fingerprint biometric-based cryptographic key generation for cloud data take 1.87 seconds [11], and a fuzzy fingerprint biometric key takes 13.1 milli seconds [24]. However, most of the methods do not consider reducing the memory size while ensuring data security. Thus, the average time taken by the proposed model is 1.78 seconds, which indicates that the computational cost of the proposed model is little high than few existing models. This is because it applies lossless data compression using arithmetic encoding which also reduces the memory size by 40%.

Moreover, performance of the proposed system is also analysed based on the various performance metrics of the authentication system such as accuracy, True Acceptance Rate (TAR) and False Rejection Rates (FRR). The average accuracy of the proposed model is 99.55% which is greater than or on par with all the existing model under comparison. The TAR is greater for the proposed model at 99.4%, whereas the FRR is 0.7%, which is greater than other existing models except few models [24, 26]. The results of the various existing models and the proposed model are presented in Table 4 with self-explanatory values. Thus, the proposed results provide favourable results with respect to accuracy, TAR, and TRR.

Table 4. Performance comparison with various existing models.

Methods	Biometric type	Accuracy (%)	TAR (%)	FRR (%)
Ambadiyil *et al.* [4]	Fingerprint	86.60	86.3	13.1
Adamovic *et al.* [1]	Iris	98.13	100	3.8
Jayapal [19]	Fingerprint	89.03	85.4	7.3
Panchal and Samanta [25]	Fingerprint	99.57	99.3	0.1
Al-Saggaf [2]	Iris	85.27	98.8	28.3
Nivedetha and Vennila [24]	Fingerprint	88.13	-	-
Panchal *et al.* [26]	Fingerprint	97.56	95.1	0.0
Dwivedi *et al.* [11]	Fingerprint	95.22	96.7	6.3
Proposed	Fingerprint	99.55	99.4	0.6

5. Conclusions

This paper introduces an effective hybrid encryption/decryption model to ensure data security with the use of a fingerprint based biometric key in addition to symmetric and asymmetric keys. The fingerprint of the user is processed at different stages to generate the biometric key by extracting unique core and minutiae points. The model then applies symmetric key encryption performed using a combination of a shared key and a biometric key. Finally, the decoded details of the fingerprint along with the ciphertext from symmetric key encryption are then encrypted using asymmetric key encryption. The proposed model utilises AES and DES for symmetric and asymmetric encryption. The computational complexity of the proposed model is analysed with various input files, in which the proposed model offers better results with an average of 2.46 seconds for encryption and 2 seconds for decryption, approximately. The size of the encrypted file is reduced by 40% with the use of AC. However, the research work lacks a security analysis of the proposed model with different attacks. Moreover, though the proposed model offers better security, the time taken by the process is high and must be reduced further. Thus, the future research work focuses on providing a complete analysis of the security of the model along with utilising the model in real-time applications.

References

[1] Adamovic S., Milosavljevic M., Veinovic M., Sarac M., and Jevremovic A., “Fuzzy Commitment Scheme for Generation of Cryptographic Keys Based on Iris Biometrics,” *IET Biometrics*, vol. 6, no. 2, pp. 89-96, 2017. https://doi.org/10.1049/iet-bmt.2016.0061

[2] Al-Saggaf A, “Secure Method for Combining Cryptography with Iris Biometrics,” *Journal of Universal Computer Science*, vol. 24, no. 4, pp. 341-356, 2018.

[3] Al-Shabi M.A., “A Survey on Symmetric and Asymmetric Cryptography Algorithms in Information Security,” *International Journal of Scientific and Research Publications*, vol. 9, no. 3, pp. 576-589, 2019. DOI: 10.29322/IJSRP.X.X.2018.pXXXX

[4] Ambadiyil S., Soorej K., and Pillai V., “Biometric Based Unique ID Generation and One to One Verification for Security Documents,” *Procedia Computer Science*, vol. 46, pp. 507-516, 2015. https://doi.org/10.1016/j.procs.2015.02.075

[5] Anees A. and Chen Y., “Discriminative Binary Feature Learning and Quantization in Biometric Key Generation,” *Pattern Recognition*, vol. 77, pp. 289-305, 2018. https://doi.org/10.1016/j.patcog.2017.11.018Get rights and content

[6] Brinda T. and Dharma D., “Enhancing the Compression Performance in Medical Images Using a Novel Hex-directional Chain Code (Hex DCC) Representation,” *Soft Computing*, vol. 25, no. 7, pp. 5807-5830, 2021. DOI:10.1007/s00500-021-05645-0

[7] Cavoukian A. and Stoianov A., *Biometric Encryption: A Positive-Sum Technology that Achieves Strong Authentication, Security AND Privacy*, Information and Privacy Commissioner of Ontario, 2007. https://www.genkey.com/wp-content/uploads/2016/11/bio-encryp.pdf

[8] Chandra S., Paul S., Saha B., and Mitra S., “Generate an Encryption Key by using Biometric Cryptosystems to Secure Transferring of Data over a Network,” *IOSR Journal of Computer Engineering*, vol. 12, no. 1, pp. 16-22, 2013. DOI:10.9790/0661-1211622

[9] Cui H., Au M., Qin B., Deng R., and Yi X., “Fuzzy Public-Key Encryption Based on Biometric Data,” *in Proceedings of the 11th International Conference on Provable Security*, Xi'an, pp. 400-409, 2017. https://doi.org/10.1007/978-3-319-68637-0_24

[10] Diaz M., Ferrer M., Impedovo D., Malik M., Pirlo G., and Plamondon R., “A Perspective Analysis of Handwritten Signature Technology,” *ACM Computing Surveys*, vol. 51, no. 6, pp. 1-39, 2019. https://doi.org/10.1145/3274658

[11] Dwivedi R., Dey S., Sharma M., and Goel A., “A Fingerprint Based Crypto-biometric System for Secure Communication,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1495-1509, 2020. http://dspace.iiti.ac.in:8080/jspui/handle/123456789/4867

[12] Goyal V. and Kant C., “An Effective Hybrid Encryption Algorithm for Ensuring Cloud Data Security,” *in Proceedings of the CSI Big Data Analytics*, Singapore, pp. 195-210, 2018. https://doi.org/10.1007/978-981-10-6620-7_20

[13] Hoque S., Fairhurst M., Howells G., and Deravi F., “Feasibility of Generating Biometric Encryption Keys,” *Electronics Letters*, vol. 41, no. 6, pp. 309-311, 2005. DOI:10.1049/el:20057524

[14] Hoque S., Fairhurst M., and Howells G., “Evaluating Biometric Encryption Key Generation Using Handwritten Signatures,” *in Proceedings of the Bio-inspired, Learning and Intelligent Systems for Security*, Edinburgh, pp. 17-22, 2008. DOI: 10.1109/BLISS.2008.8

[15] Hossain M. and Al Hasan M., “Improving Cloud Data Security Through Hybrid Verification Technique Based on Biometrics and Encryption System,” *International Journal of Computers and Applications*, vol. 44, no. 5, pp. 455-464, 2022. https://doi.org/10.1080/1206212X.2020.1809177

[16] Hossain E. and Chetty G., “Human Identity Verification by Using Physiological and Behavioural Biometric Traits,” *International Journal of Bioscience, Biochemistry and Bioinformatics*, vol. 1, no. 3, pp. 199-205, 2011. DOI: 10.7763/IJBBB.2011.V1.36

[17] Iombo C., Predictive Data Compression Using Adaptive Arithmetic Coding, LSU Master's Theses, Louisiana State University, 2007.

https://digitalcommons.lsu.edu/gradschool_theses/2717.

[18] Iwasokun G. and Akinyokun O., "Fingerprint Singular Point Detection Based on Modified Poincare Index Method," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 7, no. 5, pp. 259-272, 2014. http://dx.doi.org/10.14257/ijsip.2014.7.5.23

[19] Jayapal R., Biometric Encryption System for Increased Security, *UNF Graduate Theses and Dissertations*, University of North Florida, 2017. https://digitalcommons.unf.edu/etd/746

[20] Kasım Ö., "An Efficient Ensemble Architecture for Privacy and Security of Electronic Medical Records," *The International Arab Journal of Information Technology*, vol. 19, no. 2, pp. 272-280, 2022. https://doi.org/10.34028/iajit/19/2/14

[21] Kuppuswamy P. and Al-Khalidi S., "Hybrid Encryption/Decryption Technique Using New Public Key and Symmetric Key Algorithm," *International Journal of Information and Computer Security*, vol. 6, no. 4, pp. 372-382, 2014. DOI:10.1504/IJICS.2014.068103

[22] Kuppuswamy P. and Al-Khalidi S., "A Novel Symmetric Hybrid Cryptography Technique Using Linear Block Cipher (LBC) and Simple Symmetric Key," *Journal of Theoretical and Applied Information Technology*, vol. 99, no. 10, pp. 2216-2226, 2021. http://www.jatit.org/volumes/Vol99No10/4Vol99No10.pdf

[23] Laskar S. and Hemachandran K., "Secure Data Transmission Using Steganography and Encryption," *International Journal on Cryptography and Information Security*, vol. 2, no. 3, pp. 161-172, 2012. https://wireilla.com/papers/ijcis/V2N3/2312ijcis14.pdf

[24] Nivedetha B. and Vennila I., "FFBKS: Fuzzy Fingerprint Biometric Key Based Security Schema for Wireless Sensor Networks," *Computer Communications*, vol. 150, pp. 94-102, 2020. https://doi.org/10.1016/j.comcom.2019.11.007

[25] Panchal G. and Samanta D., "A Novel Approach to Fingerprint Biometric-based Cryptographic Key Generation and its Applications to Storage Security," *Computers and Electrical Engineering*, vol. 69, pp. 461-478, 2018. https://doi.org/10.1016/j.compeleceng.2018.01.028

[26] Panchal G., Samanta D., and Barman S., "Biometric-based Cryptography for Digital Content Protection without any Key Storage," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 26979-27000, 2019. https://doi.org/10.1007/s11042-017-4528-x

[27] Patil K., Mandal I., and Rangaswamy C., "Hybrid and Adaptive Cryptographic-based Secure Authentication Approach in IoT Based Applications Using Hybrid Encryption," *Pervasive and Mobile Computing*, vol. 82, pp.101552, 2022. https://doi.org/10.1016/j.pmcj.2022.101552

[28] Pisano E., Zong S., Hemminger B., DeLuca M., Johnston R., Muller K., Braeuning M., and Pizer S., "Contrast Limited Adaptive Histogram Equalization Image Processing to Improve the Detection of Simulated Speculations in Dense Mammograms," *Journal of Digital Imaging*, vol. 11, no. 4, pp. 193-200, 1998. doi: 10.1007/BF03178082.

[29] Prabha P., Sheetlani J., and Pardeshi R., "Fingerprint Based Automatic Human Gender Identification," *International Journal of Computer Applications*, vol. 170, no. 7, pp. 1-4, 2017. DOI:10.5120/ijca2017914910

[30] Ratha N., Connell J., and Bolle R., "Enhancing Security and Privacy in Biometrics-based Authentication Systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614-634, 2001. DOI: 10.1147/sj.403.0614

[31] Ravi J., Raja K., and Venugopal K., "Fingerprint Recognition Using Minutia Score Matching," *International Journal of Engineering Science and Technology*, vol. 1, no. 2, pp. 35-42, 2009. https://arxiv.org/ftp/arxiv/papers/1001/1001.4186.pdf

[32] Rijmen V. and Daemen J., "Advanced Encryption Standard," *in Proceedings of the Federal Information Processing Standards Publications, National Institute of Standards and Technology*, Gaithersburg, pp. 19-22, 2001. https://doi.org/10.6028/NIST.FIPS.197-upd1

[33] Rui Z. and Yan Z., "A Survey on Biometric Authentication: Toward Secure and Privacy-Preserving Identification," *IEEE Access*, vol. 7, pp. 5994-6009, 2019. doi: 10.1109/ACCESS.2018.2889996.

[34] Saravanan A., Irfan Ahmed M., and Sathya Bama S., "Automated Policy Based Remote Attestation in Trusted Computing," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 7, pp. 4485-4491, 2016. http://www.arpnjournals.org/jeas/research_papers/rp_2016/jeas_0416_3982.pdf

[35] Saravanan A., and Sathya Bama S., "A Review on Cyber Security and the Fifth Generation Cyberattacks," *Oriental Journal of Computer Science and Technology*, vol. 12, no. 2, pp. 50-56, 2019. DOI : http://dx.doi.org/10.13005/ojcst12.02.04

[36] Sathya Bama S., Irfan Ahmed M., and Saravanan A., "Network Intrusion Detection Using Clustering: A Data Mining Approach,"

International Journal of Computer Applications, vol. 30, no. 4, pp. 14-17, 2011.

[37] Seth B., Dalal S., Jaglan V., Le D., Mohan S., and Srivastava G., "Integrating Encryption Techniques for Secure Data Storage in the Cloud," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, pp. e4108, 2022. https://doi.org/10.1002/ett.4108

[38] Sharma M., Elmiligi H., and Gebali F., *Handbook of Big Data Privacy*, Springer, 2020. https://doi.org/10.1007/978-3-030-38557-6_9

[39] Shibata Y., Mimura M., Takahashi K., and Nishigaki M., "A Study on Biometric Key Generation from Fingerprints: Fingerprint-key Generation from Stable Feature Value," *in Proceedings of the International Conference on Security and Management*, Las Vegas, pp. 45-51, 2007. https://dblp.org/rec/conf/csreaSAM/ShibataMTN07.bib

[40] Sudeepa K., Aithal G., Rajinikanth V., and Satapathy S., "Genetic Algorithm Based Key Sequence Generation for Cipher System," *Pattern Recognition Letters*, vol. 133, pp. 341-348, 2020. https://doi.org/10.1016/j.patrec.2020.03.015

[41] Sunuwar R. and Samal S., "Elgamal Encryption Using Elliptic Curve Cryptography," Cryptography and Computer Security, University of Nebraska, Lincoln, 2015. https://medium.com/asecuritysite-when-bob-met-alice/elgamal-and-elliptic-curve-cryptography-ecc-8b72c3c3555e

[42] Triantafyllidis G. and Strintzis M., "A Context Based Adaptive Arithmetic Coding Technique for Lossless Image Compression," *IEEE Signal Processing Letters*, vol. 6, no. 7, pp. 168-170, 1999. doi: 10.1109/97.769360.

[43] Trotter I., "Mapping Fingerprints to Unique Numbers," *University of Oslo, Master's Thesis*, 2007. http://urn.nb.no/URN:NBN:no-26155

[44] Verma G., Liao M., Lu D., He W., Peng X., and Sinha A., "An Optical Asymmetric Encryption Scheme with Biometric Keys," *Optics and Lasers in Engineering*, vol. 116, pp. 32-40, 2019. https://doi.org/10.1016/j.optlaseng.2018.12.010

[45] Witten I., Neal R., and Cleary J., "Arithmetic Coding for Data Compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520-540, 1987. https://web.stanford.edu/class/ee398a/handouts/papers/WittenACM87ArithmCoding.pdf

[46] Yang W., Wang S., Hu J., Zheng G., and Valli C., "Security and Accuracy of Fingerprint-Based Biometrics: A Review," *Symmetry*, vol. 11, no. 2, pp. 141, 2019. https://doi.org/10.3390/sym11020141

Saravanan Arumugam completed his Ph.D. in Computer Applications at Anna University, Chennai, Tamil Nadu, India. He is an Associate Professor at Coimbatore Institute of Technology, Tamil Nadu, India. He has an experience of 25 years in teaching, including 12 years of research. He has a good number of publications in refereed journals. His areas of interest include web security, network security, web mining, software engineering, and databases.