Encryption on Multimodal Biometric using Hyper Chaotic Method and Inherent Binding Technique

Nalini M K[1]
IEEE member,
BMS College of Engineering,VTU,
Bangalore, India - 560 019

Dr. Radhika K R[2]
Senior IEEE Member,
Professor, BMS College of Engineering,VTU,
Bangalore, India - 560 019

***Abstract*—Chaotic maps are non-convergent and highly sensitive to initial values. The applications include secure digital identity in distributed systems. Face and fingerprint biometric templates are subjected to hyper-chaotic map leading to encrypted image. Encrypted image is fed as an input to deoxyribonucleic acid (DNA) sequence. Dimensionality of generated DNA sequence is reduced by hashing. The intra-variation for a subject is measured with inter-quartile range. Image set with minimal variation value is identified for selecting the consistent image of a subject. 256 bit Key is generated from the consistent image. Generated key is reduced to 128 bits by eliminating subject specific outliers and redundant values. User specific features are extracted for both the traits using ResNet 50 convolutional neural network and are fused by addition. Final key is bound to feature vector by permutation function and time taken towards key binding is estimated with benchmark database SDUMLA-HMT. Outcome reveals that time taken for key binding varies between 45ms and 58ms for an image of size 80 MB.**

***Keywords*—Chaotic systems; DNA sequences; cryptographic techniques; Convolutional Neural Networks (CNN); key binding**

I. Introduction

Securing biometric template by binding it with cryptographic key is known as key binding. Fuzzy vault and fuzzy commitment are two well-known approaches in key binding. Helper data is generated during enrollment phase and decrypted to retrieve key in authentication phase. Sutcu et.al (2007) proposed a new framework to bind cryptography keys with signature biometric using correlation filters. Results shows that smaller size images reduces the security and EER for recognition is 0.08%. Christian Rathgeb et al. (2011) developed a key binding system using fuzzy commitment scheme with iris biometric. Implementation states that for128 bits key FAR is less than 0.01%. Daniel et al. (2016) designed a method using function minimization for key binding with iris biometric and proved False acceptance rate is 0.0% for 256 bits key.

Biometric encryption is the technique of binding cyptographic key to a biometric template. The encryption procedure generates cryptographic key from biometric trait and finally template or key is retrieved for authentication [1], [2].The rapid growth of online transactions, multimedia communications, realtime applications, cloud technologies, etc. entails additional security meant for data protection. Safeguarding digital image against malicious threats is an enormous challenge in distributed networks [3]. Several approaches such as cryptographic algorithms, passcodes, etc. are currently available to secure digital image in real time applications [4]. However, the above mentioned techniques fail to differentiate between legitimate user and an attacker who has fraudulently acquired the passcode [5]. Several studies confirm that biometric encryption is resistant to multiple attacks and as such is highly recommended for verification and authentication [6].

Biometric authentication offers an innovative approach to key security which bestow direct connection between passcode and key [7]. The secured template must satisfy properties such as diversity, revocability, security and accuracy. Individual traits are incredibly imperative and eminent; therefore the templates should be protected using a highly secure technique [8]. In the past, several methods have been advanced using cryptographic techniques and cancelable biometrics to develop template security [9]. Revocable and non-invertible transformations are executed to obtain cancelable biometrics [10].

Biometric image have high resolution, high redundancy and strong correlations among adjacent pixels and occupies large storage space, as a result of which bio- metric image cannot compete with traditional encryption methods. We are looking to develop a system where the image is sensitive to even slight changes made to initial conditions, pseudo randomness, etc. Chaotic systems meet these obligations of the encryption system [11]. Strong parallel computing technique is realized by DNA sequences [12] and multimodal biometrics is brought to bear to increase the security level. The different forms of multimodal biometrics available are multi-sensor, multi-instance and multi-algorithmic, etc. Multimodal biometrics is used in several applications [8]. Accuracy is enhanced as various fusion techniques are used to pool multiple traits, is highly reliable and provide heightened security such that even if one trait fails, another trait can be used for authentication [13]. Multimodal biometrics is impenetrable to attacks and is used in several applications where high level of security is desired [14].

The key generated using chaotic maps and DNA sequences are extremely secure and effective [15]. Strength of the key is assessed using several security tests, such as histogram, key space, key sensitivity, correlation and information entropy analysis [16]. Implementation results results illustrate that the algorithm enhances security and is resistant to multiple attacks.

mds

July 1, 2021

II. Chaotic Map

Chaotic maps have a virtuous property of randomness and acute sensitivity to initial conditions. Chaos help to intersperse the original data prohibitively which is highly invertible and therefore is best suited for image encryption. Chaotic sequence is generated by scrambling pixels either by substitution or diffusion etc. Numerical methods such as phase diagrams method, Lyapunov index method, power spectral method, etc. are harnessed to magnify complexity in chaotic system [15]. Some of the widely used chaotic maps for encryption are Logistic chaotic map [17], sine map [18], cosine map, baker map, tent map, Lorenz map [19], Chens map [20]. Several chaotic maps are pooled together to form multidimensional chaotic maps [21]. 1D Logistic chaotic system is defined in equation (1) and 2D Logistic chaotic system is designated in equation (2).

$$\begin{array}{l} X_(n+1) = f(x) = \mu X_n(1-X_n)) \\ where \mu\epsilon(0,4), X_n\epsilon(0,1), n = 0,1,2,\ldots \end{array} \quad (1)$$

The results of the proposed system prove that when $3.56994 < \mu <= 4$ the system is chaotic.

$$\begin{array}{l} X_(i+1) = \mu_1 x_i(1-x_i) + \gamma_1 y_i^2 \\ y_(i+1) = \mu_2 y_i(1-y_i) + \gamma_2(x_i^2 + y_i x_i) \end{array} \quad (2)$$

when "$2.75 < \mu_1 <= 3.4, 2.75 < \mu_2 <= 3.45, 0.15 < \gamma_1 <= 0.21, 0.13 < \gamma_2 <= 0.1$",two sequences in the region [0, 1] are generated which is chaotic. The multidimensional chaotic system such as 3D or 4D chaotic system is designed by combining three or four 1D chaotic system. Equation (3) describes sine map chaotic system [22]:

$$X_(n+1) = f(x) = \mu 5_i sin(\pi X_n) \quad (3)$$

where $\mu 5_i\epsilon(0.87, 1)$ are parameters. Equation (4) describes cosine map chaotic system and Equation (5) describes baker's map chaotic system [23].

$$X_(n+1) = f(x) = \mu_6 cos(\pi|X_n - 0.5|) \quad (4)$$

where $\mu_6 = 0.98$.
$(x(n+1,)y(n+1)) =$

$$\{ \begin{array}{ll} (x_n/p, py_n) & 0 < x \leq p \\ ((x_n - p)/(1-p), (1-p)y_n + p) & p < x \leq 1 \end{array} \quad (5)$$

where $p = 0.5$ for standard bakers map.Equation (6) describes tent map,

$$x(i+1) = F_p(x_i) = \left\{ \begin{array}{ll} x_i/p & if x_i \leq p \\ (1-x_i)/(1-p) & if p < x_i \end{array} \right. \quad (6)$$

where $p\epsilon(0, 1)$ is a control parameter.Equation (7) describes Lorenz chaotic map [19],

$$\left\{ \begin{array}{l} x_i = \sigma(y_i - x_i) \\ y_i = \sigma x_i - y_i - x_i z_i \\ z_i = x_i y_i - \beta z_i \end{array} \right. \quad (7)$$

where x_i, y_i, z_i are first time derivatives,$\sigma = 10, \sigma = 28, \beta = 8/3$. Equation (8) describes Chen's chaotic map,

$$\left\{ \begin{array}{l} \dot{p} = a(q-p) \\ \dot{q} = -pr + dp + cq - s \\ \dot{r} = pq - br \\ \dot{s} = p + k \end{array} \right. \quad (8)$$

where a, b, c, d, k are system parameters. when $a = 36, b = 3, c = 28, d = 16$ and$-0.7 \leq$k≤ 0.7 the system behaves hyper chaotic by generating four chaotic sequences. Equation (9) describes 4D nonlinear hyper chaotic system [24],

$$\left\{ \begin{array}{l} (x_1) = a_1 x_1 + a_2 x_4 - x_2 x_3 \\ (x_2) = -a_3 x_1 + a_4 x_2 + b_1 x_1 x_3 \\ (x_3) = a_5 x_3 + b_2 x_1 x_2 + b_3 x_1 x_4 \\ (x_4) = a_6 x_2 + a_7 x_4 - b_4 x_1 x_3 \end{array} \right. \quad (9)$$

where $a_i < 0, i = 1, 2, 3\ldots 7, b_i > 0, x_j\epsilon R, j = 1, 2, 3, 4$. Equation (10) describes 5D hyper-chaotic maps [25],

$$\left\{ \begin{array}{l} x_1 = a(x_2 - x_1) + x_2 x_3 x_4 \\ x_2 = b(x_1 + x_2) + x_5 - x_1 x_3 x_4 \\ x_3 = -cx_2 - dx_3 - ex_4 + x_1 x_2 x_4 \\ x_4 = -fx_4 + x_1 x_2 x_3 \\ x_5 = -g(x_1 + x_2)) \end{array} \right. \quad (10)$$

where "a, b, c, d, e, f, g" are system control parameters.when "$a = 30, b = 10, c = 15.7, d = 5, e = 2.5, f = 4.45, g = 38.5$",the system generates five choatic sequences and behaves hyperchaotic [26].

Chaotic maps are classified based on properties and complexity of the system. Logistic chaotic map is less complex and highly efficient system with linear and non-linear dynamic function. Spatiotemporal is highly dynamic in nature, which improves complexity during cryptography process. Efficiency is also enhanced during encryption and decryption. Hyper chaotic is highly random which generates more Lyapunov components. Based on the number of Lyapunov components the complexity of the system is decided. The system which is non-linear and hyper chaotic, generates an highly chaotic image.

III. DNA Coding Sequences

DNA sequence has four nucleic acid bases such as "Adenine (A), Cytosine (C), Guanine (G) and Thymine (T)" in which A and T, G and C are complement to each other. Based on the similarity and uniqueness of complementary properties, DNA and binary values are related together [27]. In binary 0 and 1 are complementary, similarly 00 and 11, 01 and 10 are complementary pairs in DNA encoding. Pairing mode of DNA base and binary is given in Table I. About $c_4^4 c_2^1 = 8$types of pairing rules are available. Maximum length of DNA sequence is 4 for each pixel in an image. When C, A, T, G are denoted as 11, 00, 01, 10 and

TABLE I. DNA Encoding

	A	T	C	G
Rule 1	00	11	10	01
Rule 2	00	11	01	10
Rule 3	11	00	10	01
Rule 4	11	00	01	10
Rule 5	10	01	00	11
Rule 6	01	10	00	11
Rule 7	10	01	11	00
Rule 8	01	10	11	00

TABLE II. DNA ADDITION AND SUBTRACTION

+	A	C	T	G	-	A	C	T	G
A	C	A	G	T	A	C	G	A	T
C	A	C	T	G	C	A	C	T	G
T	G	T	C	A	T	G	T	C	A
G	T	G	A	C	G	T	A	G	C

TABLE III. DNA COMPLEMENTARY RULE

Original base	A	C	T	G
Complementary base	T	G	A	C

for 8 bit grey images, the DNA sequence length is 4 (e.g. 10101101 is encoded in to DNA sequence GGCT). DNA addition and DNA subtraction [Table II], DNA complementary [Table III], DNA XOR rule [28] and DNA XNOR rule [29] [Table IV] are shown below.

TABLE IV. DNA XOR AND DNA XNOR

XOR	A	G	C	T	XNOR	A	C	T	G
A	A	G	C	T	A	C	A	G	T
G	G	A	T	C	T	T	G	C	A
C	C	T	A	G	C	A	C	T	G
T	T	C	G	A	G	T	G	A	C

IV. ENCRYPTION TECHNIQUES

Biometric template protection is performed by feature transformation or biometric cryptosystem. There are few feature transformation techniques which are salting and non-invertible transform whereas biometric cryptosystem includes key binding and key generation [17], [30].

The properties of template protection are, there should be no cross matching across databases, able to reproduce new template based on same biometric data, computationally hard to obtain original template. The cryptosystem techniques used for template protection are Fuzzy vault, Fuzzy commitment, shielding function etc. Chaotic maps and DNA sequences are also used for image encryption which generates highly secured image. Few methods are discussed below.

Zhang et al, proposed a method with three round scrambling-diffusion structure. Hill matrix permutation is computed based on hyper-chaotic chen's chaotic values and scrambles the pixel position [27]. DNA coding rules are used as key for 'F' function of fiestal network and finally image encryption is obtained by XOR operation of previous pixel. This method effectively resists plaintext attack, differential attack and statistical attack. Chakraborty et al, proposed a system with josephus traversing and mixed chaotic map [31] . Traversing scrambles the plain image followed by diffusion which includes four 1D chaotic maps. Finally XOR operation is performed to obtain the encrypted image. This method is resistant to several security attacks. Fu et al, proposed a method using CML chaotic system for scrambling and DNA complementary rule for generating encrypted image. Analysis proves that this system is safe and effective [32].

Li et al., proposed a method where initially Bit level scrambling and pixel level scrambling is performed. Using 5D hyperchaotic system and DNA XOR coding rule is applied to generate image encryption [25]. The above method is proved to be highly secured and reliable. Ye Tian et al., proposed a method using CT cascade map, NC map and DNA encoding rule. Chaotic maps are used for scrambling S-box followed by DNA addition and XOR operation to generate encrypted image. Results of the proposed system handles several security analysis and are resistant to attacks.

The image encryption algorithm with DNA masking, SHA-2 and Lorenz system developed by Zhang et al. [27] have high key sensitivity, large key space, which enhances the security against statistical attacks and exhaustive attacks. The results show that the algorithm improves encoding efficiency. Secured image encryption algorithm was developed by combining logistic and spatiotemporal chaotic systems along with DNA encoding technique and the system was proposed by Wang et al. [22]. The experimental results show that the proposed system enhances security to various attacks such as brute-force attack, statistical attack and differential attacks. The algorithm based on Lorenz and Chen's chaotic systems with Dynamic S-boxes was proposed by Ahmad et al. [24]. The results show that the proposed scheme is resistant to several attacks and have very good efficiency. A new 1D chaotic system for image encryption was designed by Chai et al. [33]. To enhance the security level, several 1D chaotic maps are combined and the proposed system results show that the scheme is resistant to various attacks. The DNA encoding along with chaotic systems generates 192 bit key, which was implemented by Wu et al. [34]. The proposed system results show that the image is highly chaotic and resistant to several security analyses. The 4D hyper chaotic system with DNA sequence was proposed by Maddodi et al. [35]. The system provides hyper chaotic sequence, which is pseudo random in nature. The chaotic sequence is converted in to DNA sequence and the image blocks are diffused. The proposed system implementation states that the system is capable of handling known-plaintext and chosen-plaintext attacks.

V. EXISTING KEY GENERATION TECHNIQUES

There are several key generation techniques using chaotic systems and DNA sequences where few are discussed in this section. Fu et al [36] developed key stream sequence as shown in equation (11). The logistic chaotic map is used to generate permutation and substitution key stream sequences. Al et al [37] introduces the scheme that takes a master key of 320 bit and produces a group of sub keys with length 32 bit, 128 bit using tent chaotic map as shown in equation (12). Chai et al [38] and Al et al [39] proposed SHA256 key generation sequences for initial values of 2D logistic-adjusted-Sine-map and Lorenz chaotic system. The initial values are expressed as 8-bit blocks and four 52 bit sequences as shown in equation (13) and equation (14). Li et al [29] proposed four pairs 48 bit key sequences generation using chaotic logistic map and DNA substitution method as shown in equation (15). Abanda et al [21] generated the secret key made of two triplets of initial values and a coefficient. The secret key uses mixed chaotic maps of Colpitts and Duffing Oscillators.

$$pk_m = pos(pk_m) + (1 + mod(sig (abs(ps_m, \alpha), ((len(imgdata) - 1) - pos(pk_m)))) \quad (11)$$

where pk_m is the permutation key stream sequence.

$$\text{key(i)} = concat(\text{SK, Chao, compkey , Chao(i+1)}) \quad (12)$$

where SK is the SubKey,$compKey$ is the complement of SK, $Chao, Chao(i+1)$ is the chaotic keys from the Tent map.

$$K = K_1, K_2 ... K_{32} subjectto" K(i,0), K(i,1), K(i,7)" \quad (13)$$

where $K(i,j), i$ denotes the "character number" and j is the "bit number" in K_i.

$$x_h = digits(x_0 - floor(x_0), 52) \quad (14)$$

where x_0 is the initial secret key and x_h is the key value generated.

$$\begin{array}{l} b1b2b3b4 \ldots b12 \\ (b1b2b3b4)XOR(b5b6b7b8)XOR(b9b10b11b12) \\ r1r2r3r4 \end{array} \quad (15)$$

where $r1 \ldots r4$ are four pairs of 48 bit key sequences.

VI. Pre-trained Deep Networks

The convolutional neural network has several architectures which are pre-trained to solve complex problems [40]. The pre-trained models uses a benchmark dataset to achieve the solution [41],[42].

A. LeNet5

The LeNet5 architecture model [43] was developed for identification of handwritten character recognition. This mainly concentrates on automatic learning than hand-designed heuristics. The architecture comprises of 7 Layers with three convolutional layers (C1, C3, C5), two sub-sampling layer (S2, S4) and one output function (F6). The number of trainable parameters and connections are given in the Table V.

TABLE V. Entropy Analysis

Layers	Trainable Parameters	Connections
C1	156	304
S2	12	5880
C3	1516	151600
S4	32	2000
C5	10612	48120

F6 represents the Euclidean Radial Basis Function units (RBF) in the output layer. Each RBF output is calculated as given in equation (16),

$$y_i = \sum_j (x_j - w_{ij})^2 \quad (16)$$

where x_j is the state of unit j, $w_i j$ is the weight of the respective input function. The pretrained model used MNIST database with 60000 parameters which gives error rate 0.95% without deformation and dropped to 0.8% with distortion.

B. ImageNet

ImageNet architecture model [44] was developed for classification of images. This architecture comprises of eight layers with five convolutional layers (C1 –C5) and three fully connected layers (F1-F3). The input image size is $224 \times 224 \times 3$, F1- F3 have 4096 neurons each. The model uses stochastic gradient descent with batch size of 128 examples, momentum of 0.9 and weight of 0.0005 for training and weight is updated as in equation (17).

$$v_{i+1} = 0.9 \cdot v_i - 0.0005.\epsilon \cdot w_i - \epsilon \cdot \langle dL|dw|w_i \rangle D_i \quad (17)$$

Where "i is the iteration index, v is the momentum variable,ϵ is the learning rate, $\langle dL|dw|w_i \rangle D_i$ is the average over ith batch D_i of the derivative with respect to w, evaluated at w_i". The network is trained for 60 million parameters in the entire architecture. GPU trains the model for ILSVRC-2010 dataset with error rate 17.0%

C. GoogleNet

GoogleNet architecture model [45] was developed with deep convolutions for classification and detection with increased accuracy rate. The network comprises of 22 layers deep with accuracy 0.6%. GoogleNet used 5 million (V1) and 23 million (V3) parameters used in the network. ILSVRC 2014 database is used with error rate of 6.67%.

D. ResNet

Residual network was mainly developed for deeper networks [46]. Research concludes that adding more layers may degrade the final performance of the system. Residual blocks are implemented with residual function and are placed in the intermediate layers of a block. Residual function helps to adjust the input feature map for high quality features. When the distinct feature extraction is not required, the weight of the residual function is reduced to zero.

Residual network was enhanced by increasing the network's width (channel depth) and is considered as the most effective way of expanding the capacity of the entire network as shown in Fig. 1. This was implemented in ResNet34 architecture and ResNet50 architecture [47].

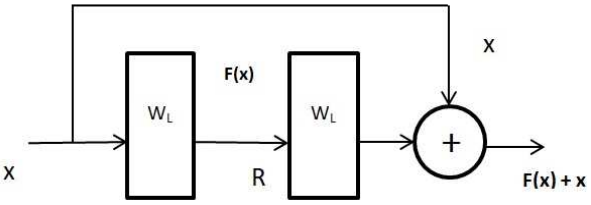


Fig. 1. Residual Model.

where W_L is the weight layer, R is the relu function, x is the input and F (x) is the residual function.

ResNet50 model replaces each two layer residual block with a three layer bottleneck block as shown in Fig. 1. The residual layer uses 1x1 convolutions in the implementation, which reduces and subsequently restores the channel depth

as shown in Fig. 2. when calculating the 3x3 convolution the computational load is significantly reduced. It uses "25 million parameters" with 80% accuracy rate.

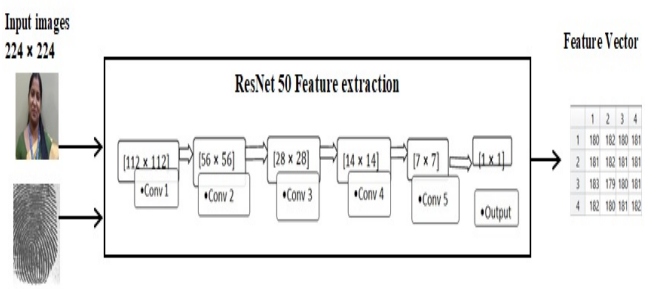


Fig. 2. ResNet50 Feature Extraction.

E. ResNeXt

The architecture of ResNeXt [48] is an extension of the deep residual network. In this network the standard residual block is replaced by one of the Inception models. The block's input is represented into a series of lower (channel) dimensional representations where convolution filters are applied before merging the results. The implementation is done using 8 GPUs with a mini batch size of 128. There are about 25 million parameters used for implementation.

F. DenseNet

Within the dense block, both the feature map and input are concatenated.

$$\text{Dense block} \implies \text{Feature map} \sim \text{input of each successive layer}$$

Features are reused within the network by concatenating feature-maps learned by different layers. When the features of different layers are concatenated the efficiency is improved. Variation in the input of subsequent layers increases. DenseNet [41] is capable to work with very small output channel depths by reducing the number of parameters. The parameters used for implementation is 40 million with 3.46% error rate.

VII. Proposed Key Generation Technique

Key generation procedure combines chaotic theory and DNA sequence as shown in Fig. 3. The procedure for key generation is stated below

Step 1: Input two Images and convert in to equal binary blocks. Consider $I1 and I2$ are input images and divide the images in to two blocks $B1, B2$.

Step 2: Beta chaotic map sequence is generated with initial values $[x1, x2, c1, c2, b1, b2, k] are chosen.$

Step 3: The blocks are converted in to scrambled images using chaotic sequence values $|p, q, z, r|$

Step 4: The scrambled images are encoded using DNA addition.

Step 5: The complement value is generated using DNA complement.

Step 6: The blocks are combined and SHA 256 is applied to generate the fixed size key for encryption.

Step 7: Outliers are eliminated in the key sequences and finally only one key is accepted.

Key generated is highly sensitive to initial parameters, more efficient and resistant to several attacks.

A. Pre-processing Techniques

Zhang et al used global bit scrambling to generate binary sequence [49]. Ahmad et al applied the size of the resultant matrix obtained after chaotic map sequences to resize the image [24]. Girdhar et al converted the image to fixed size blocks for pre-processing and scrambling is performed using chaotic sequences [19]. The proposed system uses arnold chaotic map for scrambling input images.

B. Beta Chaotic Map

Ahmad et al used Beta chaotic map which is determined from beta function which has high chaotic behavior [24] . The mapping is polynomial and is given in equation (18). chaotic map generates four chaotic map sequences. The variables generated in the chaotic sequence have high randomness with two positive Lyapunov exponents.

$$x_(n+1) = k * \beta(x_n; x_2, p, q) \tag{18}$$

where $p = b_1 + c_1 * a$ and $q = b_2 + c_2 * a$ with b_1, c_1, b_2 and c_2 are constants. The parameter is used to control amplitude of the beta map and denote the bifurcation parameter. The beta function is given in equation (19).

$$Beta(x_n; x_2, p, q) = \begin{cases} ((x - x_1)/(x_c - x_1))^p, \\ ((x_2 - x)/(x_2 - x_c))^q if x \epsilon |x_1, x_2| \\ 0 \qquad\qquad otherwise \end{cases} \tag{19}$$

Fig. 3. Encrypted Images.

Algorithm 1: Key generation

1: The input from database folder of face and fingerprint is read.
2: im1 is the image from face database folder and im2 is the image from fingerprint database folder.
3: The input image is resized and stored it in B1 and B2.
4: The binarized values of the input images are stored in to bim1 and bim2.
5: The chaotic variables are initialized to $a = 0.8, x1 = -1, x2 = 1, k = 0.85, b1 = 5, c1 = 1, b2 = 3, c2 = -1$.
6: Beta chaotic map values are calculated **for** $i \leftarrow 0$ ***to*** n **do**
    $p \leftarrow b1 + c1 * a$;
    $q \leftarrow b2 + c2 * a$;
    $z \leftarrow ((p * x2) + (q * x1))/(p + q)$;
    $r \leftarrow ((B1(i) - x1)/(z - x1))^p * ((x2 - B1(i))/(x2 - z))^q$;
    $B1(1 + i) \leftarrow k * r$;
7: Images are scrambled using chaotic variable values **for** $inc \leftarrow 1$ ***to*** $num1$ **do**
    for $row \leftarrow 1$ ***to*** $rown$ **do**
        for $col1 \leftarrow 1$ ***to*** $coln$ **do**
            $nrowp \leftarrow row$;
            $ncolp \leftarrow col$;
            for $ite \leftarrow 1$ ***to*** inc **do**
                $newcord \leftarrow [11; 12] * [nrowpncolp]$;
                $nrowp \leftarrow newcord(1)$;
                $ncolp \leftarrow newcord(2)$;
            $newim1(row, col) \leftarrow im1((mod(nrowp, rown) + 1), (mod(ncolp, coln) + 1))$;
8: Repeat step 7 to get newim2 by scrambling
9: Image1 is encoded using DNA addition $bases \leftarrow' ATGC'$;
    $scbin1 \leftarrow im2bw(newim1)$;
    for $k \leftarrow 1$ ***to*** $length(scbin1)$ **do**
    $index \leftarrow 2 * scbin1(k) + scbin1(k + 1) + 1$;
    $result1((k + 1)/2) \leftarrow bases(index)$;
10: Repeat step 9 to encode image2 using DNA condition
11: Compute DNA complement for encoded image1 **for** $o \leftarrow 1$ ***to*** $length(result1)$ **do**
    if $result1(o) =' A'$ **then**
        $l1(o) =' T'$;
    if $result1(o) =' G'$ **then**
        $l1(o) =' C'$;
    if $result1(o) =' T'$ **then**
        $l1(o) =' A'$;
    else
        $l1(o) =' G'$;
12: Repeat step 11 for encoded image2
13: Call SHA256 for image1 and image2 $im1sha256 \leftarrow uint8(l1)$ $im2sha256 \leftarrow uint8(l2)$ $Key \leftarrow im1sha256 + im2sha256$
14: Intraclass variation of keys function is called
15: Outlier elimination function is called
16: Final key is accepted after eliminating outliers

C. Key Generation

Beta chaotic map is highly sensitive to its parameters, has strong chaotic behavior, more efficient and resistant to several attacks. The proposedkey generation procedure is shown in Fig. 4.

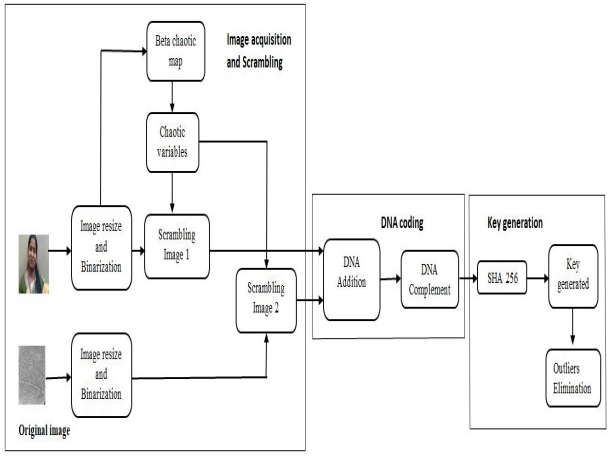


Fig. 4. Key Generation.

D. Scrambling Techniques

Liu et al. used permutation and substitution to obtain scrambled sequence. Input bits are transformed to output bits using S-boxes and P-boxes. The key introduced in each round is derived from beta chaotic map [15].

Zhan et al applied global bit scrambling technique to generate binary sequence [49]. Here 1D binary sequence b^0 contains binary digits where each digit represents intensity value of each pixel. k^x is the hyper-chaotic sequence in ascending order. b^0 is scrambled with k^x sequence as shown in equation (20). The author in [30] experimented with shuffling algorithm and substitution algorithm to rearrange the pixels. Assume m*n is the length of the input binary sequence, a_i is the shuffling sequence, $d_i$is the sequence after shuffling. Equation (21) explains shuffling sequence. Li et al experimented with pixel level scrambling [25] as shown in equation (22) and bit level scrambling as shown in equation (23).

$$b_i^1 = b_(k_i^x)^0 i\epsilon[1, 8_m * n] \tag{20}$$

$$d_(a_i) = c_i 1 \leq i \leq m * n \tag{21}$$

If F is the m×n substitution matrix, B is the resultant matrix after shuffling and then substitution is performed as shown in equation (24).

$$FoB = G_ij|G_ij = F_ij + B_ij(mod256) \tag{22}$$

$$q^{'}(i,j) = q(i^{'}, j^{'}), q(i^{'}, j^{'}) = q(i,j) \tag{23}$$

where "$q^{'}(i,j)$ is the scrambling image positioned at $(i,j), q(i^{'}, j^{'})$and $q(i,j)$are the original image positioned at $(i^{'}, j^{'})$ and $(i,j), i = 1, 2, \ldots m, j = 1, 2, \ldots n$.". Equation 24 shows the circular shift operation.

$$CS(r) = circshift[Q^{'}(r), LSB(k_3^{'}(r)), k_3^{'}(r)] \tag{24}$$

where "$circshift[u, q, v]$ means $v-$bit cycle shift on the binary sequences u. LSB(z) means the least bit of z. A right cycle shift or a left cycle shift will be decided by q =1 or q =0."

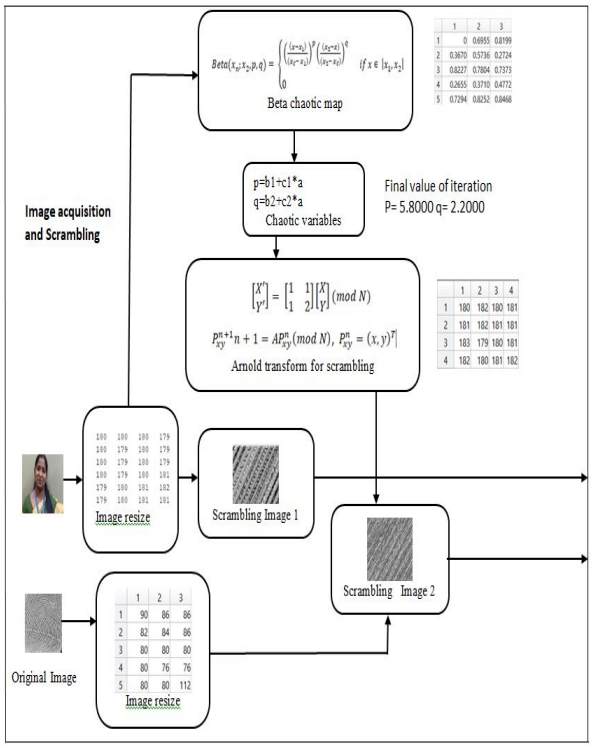


Fig. 5. Scrambling.

Arnold transform [28] is used in our proposed method for scrambling as shown in Fig. 5, because it is a reversible map without any attractor. Therefore, this transform is best used for scrambling and generalized 2D transform is shown in equation (25).

$$(x_(n+1)y_(n+1)) = ((ab)(cd))(x_n y_n) mod N \tag{25}$$

Where "a,b,c,d are positive integers and gcd(ad-bc,N)=1".

E. DNA Coding

In the proposed system, DNA addition and DNA complement is used for encoding as shown in Fig. 6. For example, if the pixel of an image is 82 and 174, the equivalent binary sequence is 10100100 and 10101110. According to DNA encoding rule-1, we obtain CCGA and CCTC. Additionally, after applying DNA addition we get CCAA and applying complement rule the final encoded value is GGTT. DNA addition operation is implemented by Li et al [26] and Kumar et al. [50] proposed a system to obtain the encrypted sequence. Zhan et al used DNA addition, DNA complement rule and DNA XOR between the hyperchaotic sequence and DNA sequence [49]. The sequence generated in the proposed system gives a robust encryption performance. Li et al. used DNA operations rules to diffuse gray images [29]. The DNA operations (XOR, XNOR, +, -) are executed randomly as shown in equation (26) to obtain transitional images.

$$\begin{aligned} op &= [x3] + 1 \\ pr^{'} &= propKI_e \\ pg^{'} &= pgopKI_e \\ pb^{'} &= pbopKI_e \end{aligned} \tag{26}$$

where op is the selected DNA operation, pr, pg, pb are plain image, $pr^{'}$, $pg^{'}$, $pb^{'}$ are transitional images. Zheng et al. implemented DNA addition and XOR rule to obtain encrypted image [51].

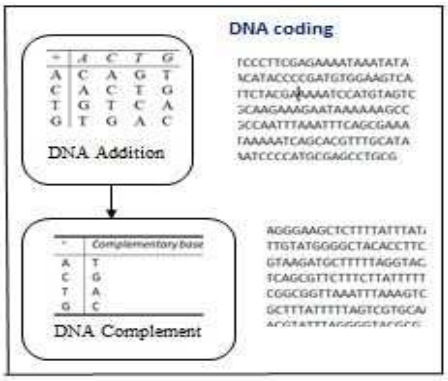


Fig. 6. DNA Coding.

F. SHA 256

Hashing is one of the widely used common cryptographic techniques due to its high security, uniqueness and integrity. It generates unique 256 bit (32 byte) signature for text which can be used as a key. In our proposed system, DNA encoded blocks of two images is combined and SHA-256 algorithm is implemented to obtain a key for encryption.

G. Intra-class Variation of Keys

After the generation of keys for all the traits, intra-class variation between the samples is calculated which is depicted in Fig. 7. The steps followed for intra-class variation of keys are as follows:

Step 1: Each column values are subtracted with the adjacent column to check the similarities between the samples of same class.

Step 2: All the column values are sorted and outliers are calculated based on zeros in the column values.

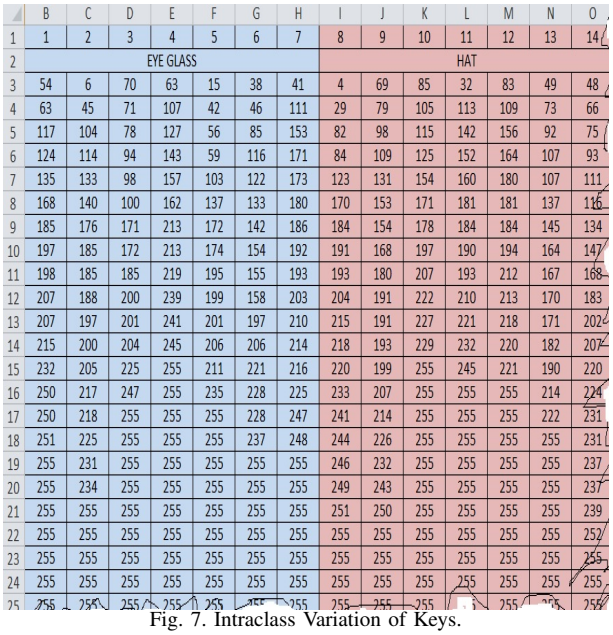
Step 3: The sample which has more zeros are considered as a class with more similarity therefore the sample is given low priority for selecting the final key.

Step 4: The above step is repeated for all the samples and the sample with minimum zero count is given high priority and the sample is used for selecting final key.

Step 5: Outliers are removed for the selected sample and final key is generated.

Algorithm 2: Intraclass variation

```
Each trait values are arranged column wise
for i = 1 to 12 do
    for j = 1 to 7 do
        A_j = A_j − A_{j+1}
    if i==7 then
        A_j = A_j − A_1
for i = 1 to 84 do
    Sort(A)
```



	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	EYE GLASS							HAT						
3	54	6	70	63	15	38	41	4	69	85	32	83	49	48
4	63	45	71	107	42	46	111	29	79	105	113	109	73	66
5	117	104	78	127	56	85	153	82	98	115	142	156	92	75
6	124	114	94	143	59	116	171	84	109	125	152	164	107	93
7	135	133	98	157	103	122	173	123	131	154	160	180	107	111
8	168	140	100	162	137	133	180	170	153	171	181	181	137	116
9	185	176	171	213	172	142	186	184	154	178	184	184	145	134
10	197	185	172	213	174	154	192	191	168	197	190	194	164	147
11	198	185	185	219	195	155	193	193	180	207	193	212	167	168
12	207	188	200	239	199	158	203	204	191	222	210	213	170	183
13	207	197	201	241	201	197	210	215	191	227	221	218	171	202
14	215	200	204	245	206	206	214	218	193	229	232	220	182	207
15	232	205	225	255	211	221	216	220	199	255	245	221	190	220
16	250	217	247	255	235	228	225	233	207	255	255	255	214	224
17	250	218	255	255	255	228	247	241	214	255	255	255	222	231
18	251	225	255	255	255	237	248	244	226	255	255	255	255	231
19	255	231	255	255	255	255	255	246	232	255	255	255	255	237
20	255	234	255	255	255	255	255	249	243	255	255	255	255	237
21	255	255	255	255	255	255	255	251	250	255	255	255	255	239
22	255	255	255	255	255	255	255	255	255	255	255	255	255	252
23	255	255	255	255	255	255	255	255	255	255	255	255	255	255
24	255	255	255	255	255	255	255	255	255	255	255	255	255	255
25	255	255	255	255	255	255	255	255	255	255	255	255	255	255

Fig. 7. Intraclass Variation of Keys.

H. Outlier Elimination

Outliers are patterns in the data which is not defined with the actual range of data values. Outliers are classified as point Outliers and contextual outliers based on the distribution of data points. The individual data, which is identified as distinct compared to rest of data, the instance is known as point outlier. If the data is distinct to a specific context then the data is a conditional outlier (or contextual outlier). The specific context is referred as either contextual attributes or behavioural attributes.

Outlier Detection using Indegree Number (ODIN) is the one of the outlier detection scheme proposed by Hautamaki et al. [52] that works based on local density. ODIN method is generally used for cluster thinning by removing vectors that are overlapping with other regions. Overlapping between clusters happen with the assumption that the regions are of low density. All the higher density regions grouped near the cluster centroid. The outlyingess of x_i is given in equation (27).

$$OL_i = \frac{1}{ind\,(x_i) + 1} \tag{27}$$

where "$ind(x_i)$ is the indegree of the vertexx_i, i.e. the number of edges pointing to x_i".

The outlyingness factor for each vector by finding the minimum distance to centroid dmax as shown in the equation (28).

$$d_{\max} = \max_i \{\|\boldsymbol{x}_i - \boldsymbol{c}_{p_i}\|\}, \quad i = 1, \ldots, N \tag{28}$$

The outlyingess of x_i is given in equation (29),

$$o_i = \frac{\|\boldsymbol{x}_i - \boldsymbol{c}_{p_i}\|}{d_{\max}} \tag{29}$$

where x_i is the vector, c_p is the centroid and d_max is the distance. The factors that are responsible for an outlier, in the dataset are normalized to the scale [0, 1]. The vector with greater value is considered as an outlier. The proposed system uses Interquartile Range technique as shown in Fig. 8. The outliers in the key generated are calculated by considering, that the outliers lie either on the first quarter or on the last quarter of the given data after sorting the key values. The final key generated is shown in Fig. 9. The steps followed are

Step 1: The point furthest from the mean of data is calculated.

Step 2: IQR is calculated by finding the difference between the mean of the upper quartile(q2) and mean of the lower quartile(q1).

Step 3: The fence is calculated using the formula, F=1.5 * IQR

Step 4: Lower bound and upper bound are calculated.

LB=q1 – F, UB=q2+F

Step 5: The values between the lower bound and upper bound are accepted for the final key.

Step 6: The accepted key size should not exceed 128 bits. If the size of the key is greater than 128 bits, steps 3 to step 5 are repeated.

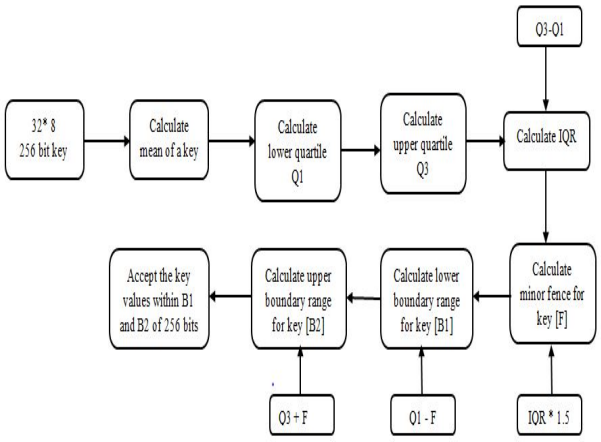


Fig. 8. Outlier Elimination.

Algorithm 3: Outlier elimination

1: Upper quartile(Q2) and lower quartile(Q1) are calculated using the given formula,

$$Q1 = \frac{1}{16}\Sigma_{1=1}^{16} k_i$$

$$Q2 = \frac{1}{16}\Sigma_{i=17}^{32} k_i$$

2: Inter Quartile Range is calculated using Q1 and Q2 $IQR = Q2 - Q1$

3: Upperbound(UB) and lower bound(LB) are calculated,

$$\text{LB} = \text{Q1} - \text{F}, \text{UB} = \text{Q2} + \text{F}$$

4: The values between the upperbound and lowerbound are considered for Final key is LB <= Key <= UB

5: **if** *Final key* > 128 *bits* **then**

Repeat Step 2 to 4 till the key is 128 bits

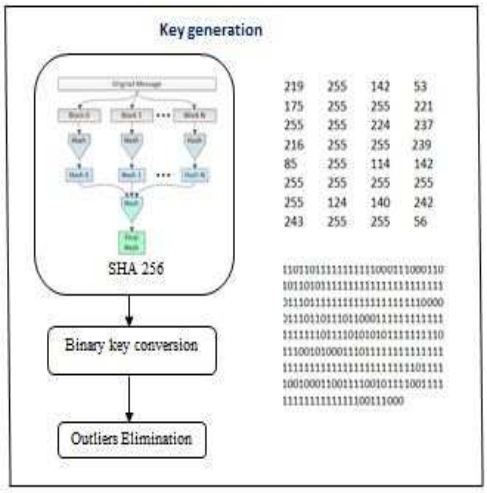


Fig. 9. Final Key.

I. Feature Extraction using CNN

ResNet increases the seed of training the deep networks, reduces the number of parameters by increasing the depth of the networks instead of widening the network, helps to vanish Gradient problem and gives high accuracy in network performance.

During the training period some neuron can die and cause information loss which is known as Gradient problem. The above problem is rectified by using residual network, by mapping non-linear function H(x) to F(x) which is defined as H(x) – x. The output of the second layer adds F(x) to x and carries important information to the next layer (Relu). Therefore, information loss is reduced and no gradient problem in ResNet. Increasing the depth of the network may result in bias problem but ResNet avoids negative outcomes, increases accuracy, fast training of the network while the network depth increases. ResNet is classified in to three models, 50, 101, 152 based on the depth of networks. The feature extraction analysis using CNN for Resnet50 network is shown in Fig. 12.

Algorithm 4: CNN Feature Extraction

1: Multimodal sample images are read

2: The sample images are divided for testing and training

3: The pre-trained ResNet-50 network is loaded

4: Features are extracted from the deeper layer of a pre-trained network

5: The training features are used to train the classifier

6: The accuracy level is calculated

Fig. 10. Resenet 50 Analysis.

J. Key Binding Technique

The feature extraction is performed using Resnet 50 CNN as shown in Fig. 10. The training features are extracted for both the traits and fused using additive operation to get the original vector. The generated key of the respective trait is embedded with the original vector using random permutation function. The permutation function used is randi which returns the array of integers from discrete uniform distribution on the given interval. $X = \text{randi}\left([\text{imin}_\alpha\, 1\, \text{max}], \text{sz}\, 1, \text{szn}\right)$ Random positions are generated with respect to the key size in the original template vector. Now the random position values are replaced with the key value. Final image is the key embedded within the template as shown in Fig. 11. The steps followed for key binding are,

Step 1: Original fused vector of face and fingerprint is loaded.

Step 2: Random positions are generated using permutation function.

Step 3: The key values are replaced to the random positions of the original vector.

Step 4: The key is now embedded with the biometric template.

Algorithm 5: Key binding

1: V - Original vector, imin, imax - min and max position of V, SZ1,SZN - min and max position of Key, K - Key
2: index = randi([imin,imax], SZ1,SZN)
3: V(index) = K
4: New vector (NV) = embedded image

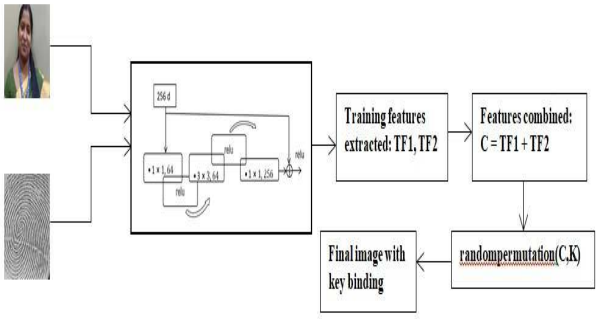


Fig. 11. Key Binding.

VIII. Security Analysis for Key

A. Key-Space Analysis

The iterative procedure of checking all possible keys to find the decryption key is known as Key space analysis. The proposed system ensures that the key generated is 256-bit hash value and reduce to 192 bits, which handles Brute force attack effectively.

B. Key Sensitivity Analysis

The perfect image encryption key should be sensitive so that the key will be highly secured and inaccessible by hackers easily. The minor change in initial parameter of the chaotic system will alter the entire key because of which the generated key in our proposed system is highly sensitive. A small change in the parameter results in a large variation in the encryption process. The variation measures the sensitivity of the generated key.

C. Statistical Analysis

Statistical analysis measures the performance of image encryption system. The encrypted image histogram is different from the original image histogram so that the hacker will not be able to get the original image through encrypted image histogram.

D. Pixel Correlation Analysis

The adjacent pixels correlation in both original and the encrypted image either horizontally or vertically or diagonally is known as correlation co-efficient analysis as shown in equation (30). When the similarity between the original and encrypted image is less, then the value of correlation coefficient is low. The values obtained by calculating Correlation coefficient shows that the system is capable of handling statistical attacks. Table VI gives the correlation values of the proposed system with the existing techniques.

$$r = \frac{\sum_m \sum_n \left(A_{mn} - \overline{A}\right)\left(B_{mn} - \overline{B}\right)}{\sqrt{\left(\sum_m \sum_n \left(A_{mn} - \overline{A}\right)^2\right)\left(\sum_m \sum_n \left(B_{mn} - \overline{B}\right)^2\right)}} \tag{30}$$

where $\overline{A} = \text{mean}\,2(A),$ and $\overline{B} = \text{mean}\,2(B)$

TABLE VI. Correlation Analysis

Image	Reference	Correlation
Pepper	Ref.1	0.2650
Lena	Ref.2	0.0214
Lena	Ref.3	0.0015
Lena	Ref.5	-0.0012
Lena	Proposed system	-0.0044

1) Differential Analysis: The sensitivity of the encrypted image is determined by Differential analysis. Minor variation in the original image gives more impact on the encrypted image. "Number of pixels changed between the original and the encrypted image is calculated using NPCR [number of pixels change rate]". It gives the ratio of the two encrypted images when there is a slight variation in the input image. NPCR is calculated as given in equation (31). Table VII gives the NPCR analysis of the proposed system with the existing techniques.

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M * N} * 100 \tag{31}$$

Where, "D is the auxiliary matrix which is created when $C1(i,j) = C2(i,j), then D(i,j) = 0$, otherwise D(i,j) = 1".

TABLE VII. NPCR Analysis

Image	Reference	NPCR
Pepper	Ref.1	99.62
Lena	Ref.2	99.60
Lena	Ref.7	99.48
Lena	Ref.5	99.50
Lena	Proposed system	99.51

The calculation of the average intensity between the same cipher images is known as "unified average change intensity [UACI]". When the values are high, the parameters shows that the minor variation in original image gives more impact in encrypted image. UACI is calculated as given in equation (32). Table VIII gives the UACI analysis of the proposed system with the existing techniques.

$$UACI = 1/(MxN)[(i=0)^{(}M-1)(i=0)^{(}N-1)|C_1(i,j) - C_2(i,j)|/255]100 \tag{32}$$

TABLE VIII. UACI Analysis

Image	Reference	UACI
Pepper	Ref.1	34.26
Lena	Ref.2	33.44
Lena	Ref.7	28.45
Lena	Ref.5	35.65
Lena	Proposed system	33.16

2) MSE: The mean square error value between the decrypted image(I_2) and plain image (I_1). If there is more data loss the value is high or the value is low. The formula is given in equation (33).

$$MSE = 1/(MxN)(i=1)^M(J=1)^N I_1(i,j) - I_2(i,j)^2 \quad (33)$$

3) PSNR: "PSNR [Peak Signal to Noise Ratio] is used to test the quality of the attacked encrypted image". The formula is calculated as given in equation (34).

$$PSNR = 10xlog_10((255x255)/MSE)(dB) \quad (34)$$

E. Information Entropy Analysis

The robustness of encrypted image is verified by information entropy analysis. For an image, 8 bit length random system is generated and entropy of both images are calculated as given in equation (35). When the "Entropy value of cipher image is approximately 8 then the system is proved to be resistant to entropy analysis".

$$H(x) = -(i=1)^n(x_(i)logp(x_(i)) \quad (35)$$

Where, "x is a set of symbols, n is the total number of symbols, $x_(i)\epsilon x, p(x_(i))$is the probability of $x(i)$in x". Table IX gives the entropy value of the proposed system with the existing techniques.

TABLE IX. ENTROPY ANALYSIS

Image	Reference	Entropy
Pepper	Ref.1	7.9993
Lena	Ref.2	7.9993
Lena	Ref.3	7.9994
Lena	Ref.5	7.9982
Lena	Proposed system	7.7795

F. Time Analysis of Key Binding Technique

Biometric template of 80MB size is considered for the experiment. Result shows that the time taken for key binding of several template varies from 40ms to 58ms as shown in Fig. 12.

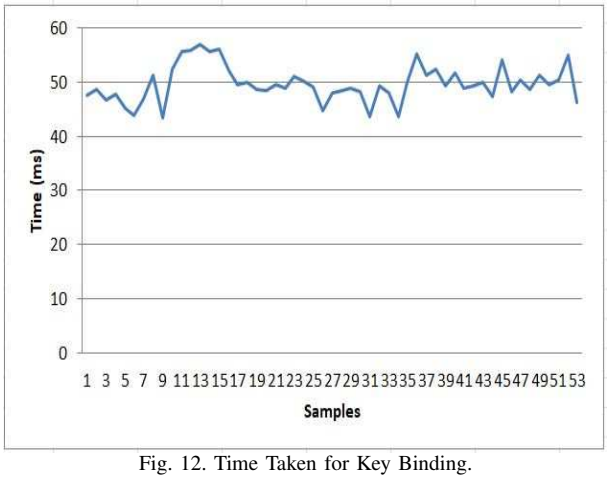


Fig. 12. Time Taken for Key Binding.

IX. CONCLUSION AND FUTURE WORK

An innovative algorithm has been proposed for key generation and key binding. Results prove that the key generated is highly secured as intraclass variations are applied to remove outliers and takes less time for key binding. The proposed system uses chaotic, DNA and CNN concepts and implemented using Matlab R2019b. The database used is SDUMLA-HMT which is a multimodal biometric of face and fingerprint with 106 traits of 96 samples each. The above work can be enhanced for touchless real-time biometric samples and measure the performance of the system.

ACKNOWLEDGMENT

I like to thank School of computer science and technology, Shandong university for providing multimodal biometric database SDUMLA-HMT for my experiment.

REFERENCES

[1] C.-C. Lee, C.-T. Chen, P.-H. Wu, and T.-Y. Chen, "Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices," *IET Computers & Digital Techniques*, vol. 7, no. 1, pp. 48–55, 2013.

[2] D. G. Martínez, F. J. G. Castano, E. A. Rúa, J. L. A. Castro, and D. A. R. Silva, "Secure crypto-biometric system for cloud computing," in *2011 1st International Workshop on Securing Services on the Cloud (IWSSC)*. IEEE, 2011, pp. 38–45.

[3] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. Jawahar, "Blind authentication: a secure crypto-biometric verification protocol," *IEEE transactions on information forensics and security*, vol. 5, no. 2, pp. 255–268, 2010.

[4] V. Matyáš and Z. Říha, "Security of biometric authentication systems," in *2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*. IEEE, 2010, pp. 19–28.

[5] M. Rahman and P. Bhattacharya, "Remote access and networked appliance control using biometrics features," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 2, pp. 348–353, 2003.

[6] A. K. Jain and K. Nandakumar, "Biometric authentication: System security and user privacy." *IEEE Computer*, vol. 45, no. 11, pp. 87–92, 2012.

[7] Y. J. Lee, K. R. Park, S. J. Lee, K. Bae, and J. Kim, "A new method for generating an invariant iris private key based on the fuzzy vault system," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1302–1313, 2008.

[8] A. K. Jain, L. Hong, and Y. Kulkarni, "A multimodal biometric system using fingerprint, face and speech," in *2nd Int'l Conf. AVBPA*, vol. 10, 1999.

[9] R. Ang, R. Safavi-Naini, and L. McAven, "Cancelable key-based fingerprint templates," in *Australasian conference on information security and privacy*. Springer, 2005, pp. 242–252.

[10] C. Rathgeb and A. Uhl, "A survey on biometric cryptosystems and cancelable biometrics," *EURASIP*

Journal on Information Security, vol. 2011, no. 1, p. 3, 2011.

[11] R. Guesmi, M. A. B. Farah, A. Kachouri, and M. Samet, "A novel chaos-based image encryption using dna sequence operation and secure hash algorithm sha-2," *Nonlinear Dynamics*, vol. 83, no. 3, pp. 1123–1136, 2016.

[12] P. Zhen, G. Zhao, L. Min, and X. Jin, "Chaos-based image encryption scheme combining dna coding and entropy," *Multimedia Tools and Applications*, vol. 75, no. 11, pp. 6303–6319, 2016.

[13] K.-A. Toh, W.-Y. Yau, and X. Jiang, "A reduced multivariate polynomial model for multimodal biometrics and classifiers fusion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 2, pp. 224–233, 2004.

[14] T. B. Long, T. Hanh *et al.*, "Multimodal biometric person authentication using fingerprint, face features," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2012, pp. 613–624.

[15] L. Liu, S. Miao, H. Hu, and M. Cheng, "N-phase logistic chaotic sequence and its application for image encryption," *IET Signal Processing*, vol. 10, no. 9, pp. 1096–1104, 2016.

[16] K. Radhika and M. Nalini, "Biometric image encryption using dna sequences and chaotic systems," in *2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*. IEEE, 2017, pp. 164–168.

[17] H. Ren, J. Wang, and Q.-H. Wang, "An image encryption scheme of logistic modulation using computer-generated hologram and chaotic map," *Journal of Electrical and Computer Engineering*, vol. 2018, 2018.

[18] B. Awdun and G. Li, "The color image encryption technology based on dna encoding & sine chaos," in *2016 International Conference on Smart City and Systems Engineering (ICSCSE)*. IEEE, 2016, pp. 539–544.

[19] A. Girdhar and V. Kumar, "A rgb image encryption technique using lorenz and rossler chaotic system on dna sequences," *Multimedia Tools and Applications*, vol. 77, no. 20, pp. 27 017–27 039, 2018.

[20] J. Zhang, D. Hou, and H. Ren, "Image encryption algorithm based on dynamic dna coding and chens hyperchaotic system," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[21] Y. Abanda and A. Tiedeu, "Image encryption by chaos mixing," *IET Image Processing*, vol. 10, no. 10, pp. 742–750, 2016.

[22] X. Wang, X. Zhu, and Y. Zhang, "An image encryption algorithm based on josephus traversing and mixed chaotic map," *IEEE Access*, vol. 6, pp. 23 733–23 746, 2018.

[23] L. Liu, S. Hao, J. Lin, Z. Wang, X. Hu, and S. Miao, "Image block encryption algorithm based on chaotic maps," *IET Signal Processing*, vol. 12, no. 1, pp. 22–30, 2017.

[24] M. Ahmad, M. Doja, and M. S. Beg, "Security analysis and enhancements of an image cryptosystem based on hyperchaotic system," *Journal of King Saud University-Computer and Information Sciences*, 2018.

[25] Y. Li, C. Wang, and H. Chen, "A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation," *Optics and Lasers in Engineering*, vol. 90, pp. 238–246, 2017.

[26] T. Li, M. Yang, J. Wu, and X. Jing, "A novel image encryption algorithm based on a fractional-order hyperchaotic system and dna computing," *Complexity*, vol. 2017, 2017.

[27] X. Zhang, Z. Zhou, and Y. Niu, "An image encryption method based on the feistel network and dynamic dna encoding," *IEEE Photonics Journal*, vol. 10, no. 4, pp. 1–14, 2018.

[28] Y. Tian and Z. Lu, "Novel permutation-diffusion image encryption algorithm with chaotic dynamic s-box and dna sequence operation," *AIP Advances*, vol. 7, no. 8, p. 085008, 2017.

[29] X. Li, C. Zhou, and N. Xu, "A secure and efficient image encryption algorithm based on dna coding and spatiotemporal chaos." *IJ Network Security*, vol. 20, no. 1, pp. 110–120, 2018.

[30] Y. Zhang, "Test and verification of aes used for image encryption," *3D Research*, vol. 9, no. 1, p. 3, 2018.

[31] S. Chakraborty, A. Seal, M. Roy, and K. Mali, "A novel lossless image encryption method using dna substitution and chaotic logistic map," *International Journal of Security and Its Applications*, vol. 10, no. 2, pp. 205–216, 2016.

[32] X.-Q. Fu, B.-C. Liu, Y.-Y. Xie, W. Li, and Y. Liu, "Image encryption-then-transmission using dna encryption algorithm and the double chaos," *IEEE Photonics Journal*, vol. 10, no. 3, pp. 1–15, 2018.

[33] X. Chai, X. Zheng, Z. Gan, D. Han, and Y. Chen, "An image encryption algorithm based on chaotic system and compressive sensing," *Signal Processing*, vol. 148, pp. 124–144, 2018.

[34] X. Wu, K. Wang, X. Wang, H. Kan, and J. Kurths, "Color image dna encryption using nca map-based cml and one-time keys," *Signal Processing*, vol. 148, pp. 272–287, 2018.

[35] G. Maddodi, A. Awad, D. Awad, M. Awad, and B. Lee, "A new image encryption algorithm based on heterogeneous chaotic neural network generator and dna encoding," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 24 701–24 725, 2018.

[36] C. Fu, G.-y. Zhang, M. Zhu, Z. Chen, and W.-m. Lei, "A new chaos-based color image encryption scheme with an efficient substitution keystream generation strategy," *Security and Communication Networks*, vol. 2018, 2018.

[37] H. M. Al-Mashhadi and I. Q. Abduljaleel, "Color image encryption using chaotic maps, triangular scrambling, with dna sequences," in *2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT)*. IEEE, 2017, pp. 93–98.

[38] X. Chai, Z. Gan, K. Yuan, Y. Chen, and X. Liu, "A novel image encryption scheme based on dna sequence operations and chaotic systems," *Neural Computing and Applications*, vol. 31, no. 1, pp. 219–237, 2019.

[39] O. M. Al-Hazaimeh, M. F. Al-Jamal, N. Alhindawi, and A. Omari, "Image encryption algorithm based on lorenz chaotic map with dynamic secret keys," *Neural Computing and Applications*, pp. 1–11, 2017.

[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp.

2818–2826.

[41] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[43] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[46] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[48] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[49] K. Zhan, D. Wei, J. Shi, and J. Yu, "Cross-utilizing hyperchaotic and dna sequences for image encryption," *Journal of Electronic Imaging*, vol. 26, no. 1, p. 013021, 2017.

[50] M. Kumar, A. Iqbal, and P. Kumar, "A new rgb image encryption algorithm based on dna encoding and elliptic curve diffie–hellman cryptography," *Signal Processing*, vol. 125, pp. 187–202, 2016.

[51] W. Zheng, F.-Y. Wang, and K. Wang, "An acp-based approach to color image encryption using dna sequence operation and hyper-chaotic system," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 461–466.

[52] V. Hautamäki, S. Cherednichenko, I. Kärkkäinen, T. Kinnunen, and P. Fränti, "Improving k-means by outlier removal," in *Scandinavian Conference on Image Analysis*. Springer, 2005, pp. 978–987.