BUILD CRYPTOGRAPHIC SYSTEM FROM MULTI-BIOMETRICS USING MEERKAT ALGORITHM

Duha Dawood Salman[1], Raghad Abdul-Aali Azeez[2], Abdul mohssen Jaber Abdul-hossen[3]
[1] Department of Computer Science /University of Technology , Baghdad, Iraq
111820@uotechnology.edu.iq
[2]College of Ibn Rushd /University of Baghdad , Baghdad, Iraq
Reghad.azeez@ircoedu.uobaghdad.edu.iq
[3] Department of Computer Science /University of Technology , Baghdad, Iraq
110116@votechnology.edu.iq

Abstract - Presenting uncouth proposal for the design of investigating ways to use extraction feature from biometric user, rather than memorable password or passphrase as an attempt to produce a new and randomly cipher keys. Human users find it difficult to remember long cipher keys. Therefore, the proposed work takes the eye and ear as a multi-biometrics feature extraction for generating the cryptography keys. Meerkat Clan Key Generation Algorithm (MCKGA) is used in this work for key generation, firstly we generate keys with 128-bits, then we enhance our method by generating 256-bits, and finally we mix the keys produced from (eye and ear) and get robust key with 512-bits length, these keys are tested by NIST statically test to generate random keys used in encryption process. Our approach generates unique keys used in cryptographic system by using Advanced Encryption Standard (AES) algorithm.

Keywords - Multi-Biometrics, Ear, Eye, Encryption, Decryption, Meerkat Algorithm.

I. INTRODUCTION

Encryption is one of the most important methods of data protection; therefore it is used in most systems of protection. Building a security system dependent on biometrics characteristics of human is a challengeable research field, since the biometrics subtleties differ from an individual to another, so imagining a scenario in which utilizes two of biometrics subtleties (eye and ear) of an individual, utilized together to maximize the level of security[1]. From previous studies, it was found that ear measurements come from its stability and the difficulty of rigging these organs in a person and their permanence during his life [2]. Through the perception of the conduct of some living creatures may show how they plan their characteristic practices to the algorithmic schedules. Those methodologies are meta-heuristics for worldwide enhancement and are mostly assembled by means of choosing the ideal structure and through a structure of randomization. The previous studies control the converging of the calculation to the optimality (for example the use), and the long ways ahead will maintain a strategic distance from both losing the assortment and keeping the calculation from getting bordered in the local optima (for example assessment). A productive security among examination and use could bring about worldwide accomplishment of optimality [3].The proposed system intended to utilize the unique biometric characteristics of the individual, so as to produce encrypted keys used in cryptographic system.

In this system, more than one level is applied to create the key that is utilized for encryption; the produced key is utilizing Meerkat algorithm which will utilize the unique features of the individual to extract features. Though, this research protrudes a unique key length and high security level, and this secures systems and non-penetration.

Meerkats are the creatures which socially live in states of 5 to 30 individuals. Because of the way that they are sociable animals, they trade both parental consideration and latrine obligations [4]. All of the crowds have a main alpha male just as a main alpha female. All of the crowds have their own property where they once in a while move for the situation where the nourishment isn't found or for the situation where they are obliged with a harder horde. In which case, the crowd which are more fragile will attempt at expanding in another manner or stay until they get harder and recoup the lost tunnel [5].

All of the hordes likewise have what is known as a "guard" that shows somebody guarding over the crowd when spotting hazard and advising others on account of risk. The guard either sees from climbing a tree or starting from the earliest stage in hedges. The guard is answerable for looking out for both the plan of the tunnel and for the situation where the other crowd individuals are looking for nourishment. The guard gives a sound like boisterous bark on account of watching a hazard, and a short time later the horde will quickly dart to the concealing openings [6].

II. RELATED WORK

Many researchers create cryptographic keys from a client verification that relies on multimedia biometrics. It may begin to gain recognition as a legitimate verification strategy and a practical choice for classic identity technologies in multiple areas of application.

"L. Wu et al", 2004 [11], have improved a modern face by using 128-dimensional principal component analysis vectors as biometric cryptosystem and uses "Error Correction Codes"(ECC) proposed by "Reed-Solomon algorithm". A biometric key during the decryption phase is generated by use checking all contains of the generated table at the cipher phase. The final key is got by using both the biometric key and ECC.

Yan and Yu-Jin Zhang, 2007 [8] demonstrated a class-reliance highlight investigation system based on Correlation Filter Bank (CFB) method for a productive multimodal biometrics fusion at the element level. In CFB, an unrestricted relationship channel prepared for a specific methodology is obtained by improving the entire unique relationship.

The authors, 2010 [10] proposed a unimodal biometric ; it is a framework for fixed cryptographic key production as traits that are instable in nature. The best result and conclusion are the advances to produce distinguishable and recognizable features resulting in a fix cryptographic key.

In 2010 [12], a biometric-based cryptography key generation have been generated using fingerprint information of a person. With regard to the security views, this method is safe and further flexible in the process of the key generation. User's personal data could be encrypted by using this key. The fingerprints characteristic (Reference Points) have produced two-ways area, directional element and probability distribution. An efficient approach to secure cryptographic key generation from iris and face biometric traits has been presented in, 2015 [6]. The presented system generates 256-bits crypto key generated from iris and face biometric traits. The proposed system employs three modules: feature extraction module, face feature extraction module and key generation module.

"Chang et" al, 2015 [7], proposed a unimodal biometric system for generating constant encryption keys from unimodal biometric attributes. The main feature of their research is to deal with the production of achieving consistent cryptographic key and recognizable biometric features. Although the presentation of the proposed system is estimated using a "facial" database that includes "facial" manifestations and types of head movement, the scholars have expressed that the structure is closely related to other biometrics methods as well.

The authors in, 2016 [13] creates every time various keys based on the snapshot captured from the scanner. Identical biometric cryptography key can be obtained from the fingerprints captured with several qualities of the image from distinct scanners.

In another approach, a biometric-based key release is used to protect a randomly generated cryptographic key, 2017 [9]. In this scheme, an error in biometric data propagates the error in cryptographic key. This error is handled with the help of Error Correction Coding (ECC) technique(s).

III. The Proposed System

In this system, more than one level is applied to generate the key that is used for encryption; we use a key length of 128-bits. It's generated by using Meerkat which will use the unique features of the person to extract features that is used to generate the key. Figure (1) shows the proposed system architecture and algorithm (1) shows the main phases in the proposed key generation algorithm

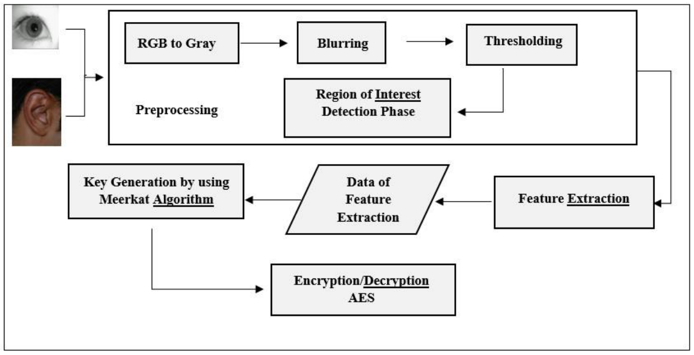

RGB to Gray
Blurring
Thresholding
Preprocessing
Region of Interest Detection Phase
Key Generation by using Meerkat Algorithm
Data of Feature Extraction
Feature Extraction
Encryption/Decryption AES

Figure (1) Structure of the proposed system

A. *Preprocessing algorithm*

For both the ear and the eye images, they are pre-treated as follows to provide the features extraction from them.

1) RGB to gray conversion: In this step, the entered colored image will be converted to gray scale to reduce the amount of information processed in the system and remove noise. Each colored pixel will be converted to gray scale pixel using the following equation (1):

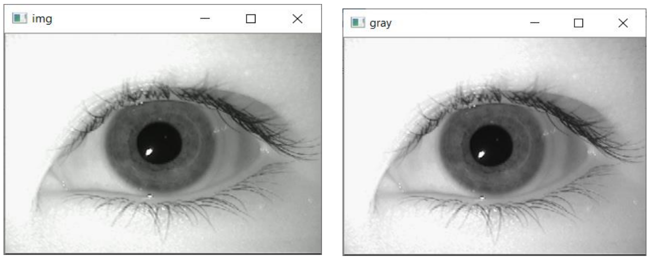
For each image pixel with red, green and blue values of (R, G, B):

$L=0.299R+0.587G+0.114B$ ………… (1)

Figures (2 and 3) show the application of the RGB conversion to gray scale

(a) **(b)**

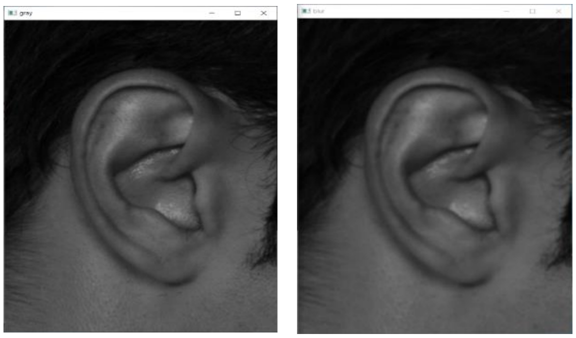
Figure (2) Application of ear RGB conversion, a: original image b: gray scale image



(a) **(b)**

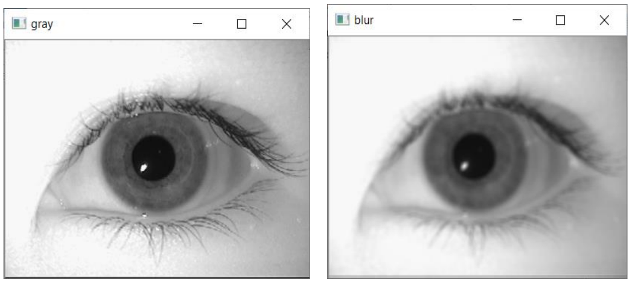
Figure (3) Application of eye RGB conversion, a: original image, b: gray scale image

2) Blurring: This is done via applying the 2D Gaussian filter to enhance the image and reduce the noise within that image show, and Figures (4 and 5) show blurring of ear and eye images.



(a) (b)

Figure (4) Application of ear blurring, a: gray scale image, b: blurred image



(a) (b)

Figure (5) Application of eye blurring, a: gray scale image, b: blurred image

3) Thresholding: It will convert the image to binary which will be entered to the region of interest extraction by using contour which needs binary inputs, since it senses the black spots.

B. *Region of Interest (ROI) detection phase*

The detected part needs to be bounded since it is the only part that will need to be extracted by using feature extraction algorithm, and the detection of region of interest is done by using contour algorithm which will bound the ear or eye and discard other information on the image.

C. *Feature extraction*

This phase aims to extract features from bounded region of interest founded by contour; and this is done via algorithm LDA.

D. *Key Generation Proposal with Meerkat Algorithm*

The features extracted from eyes and ears are used for key generation by MCA Algorithm. From prior clarifying concerning Meerkat creature enlivened MCA, the following algorithm illustrates Meerkat process.

Algorithm 1: Meerkat Clan pseudocode

```
Input
    a   clan size (number of keys)
    R   care size     a- z -1
    z   foraging size     where z < 80% of a
    Rr  worst care rate
    Gr  worst foraging rate
    N   neighbor solution
Output
 Generated strong Key (Sentry)
Process
  Produce random clan of Keys from extracted feature clan(a)
  Calculate the fitness for each Key in clan using
          $F= a + (€ / m)$
  Sentry = optimal clan Key
  Divide the clan to 2 sets (which are foraging and care)
  While not end of generations
    For i=1 to z
      Call neighbor_generated (N, Sentry, foraging(i), strong_key)
      foraging(i)= strong_key from N neighbor
    End For
    Swap the worst for Gr Keys in foraging set with the optimal ones' Keys in care group;
    Drop worst Rr Keys from care set and produce ones' Key from extracted features;
    Choose optimal Key of foraging call it strong _key
    If strong  key <= Sentry then
      Swap the Sentry with strong_key
    EndIf
End  while
End
```

E. *Key Encryption*

Encryption has come up as a solution, and plays an important role in information security system. This security mechanism uses some algorithms to scramble data into unreadable text which can be only decoded or decrypted by parties that possess the associated key. In this work we generate a strong key, but we also encrypt this key with AES algorithm.

AES algorithm is one of the most common and widely symmetric block cipher algorithm and uses symmetric key for encryption and decryption processes. The algorithm has three features, first, it has simple design, second, it is easy to understand, and lastly, it provides more security. It encrypts all block bits in one iteration, and composes a series of clan linked operations by inputting specific outputs as replacements called substitutions and rearranging bits around permutations as others including [14].

Advanced Encryption Standard algorithm in Steps [14]
" Step 1:- Input a plaintext of 128 or 256 bits of block cipher, which will be negotiated as 16 bytes.
Step 2: - Add Round Key
- Every byte is integrated with a block of key generation utilizing bitwise XOR.
Step 3:- Byte Substitution
The 16 input bytes are substituted by looking up a fixed table (S-box). The result will be a 4x4 matrix.
Step 4:- Shift Row
Every one of the four rows of the matrix is shifted to the left. Entry, which will be left, placed on the right side of the row. Shift is done as follows:
1st row is not shifted.
2nd row is shifted one byte position to the left.
3rd row is shifted two positions to the left.
4th row is shifted three positions to the left.
The result is a new matrix consisting of the same 16 bytes, yet shifted concerning one another.
Step 5:- Mix Columns:
Each column of 4 bytes will be altered by applying a Galois Field. This function takes as inputting 4 bytes of one column and outputting four totally new bytes, which replaces the original column. The result is another new matrix comprising of 16 new bytes.
Step 6:- Add Round Key
The 16 bytes of matrix will be pondered as 128 bits and will be XOR to 128 bits of the key generator.
Step 7:- This 128 bits will be taken as 16 bytes and comparable rounds will be performed.
Step 8:- At the 10th round which will be the last round, a ciphered text will be the output ".

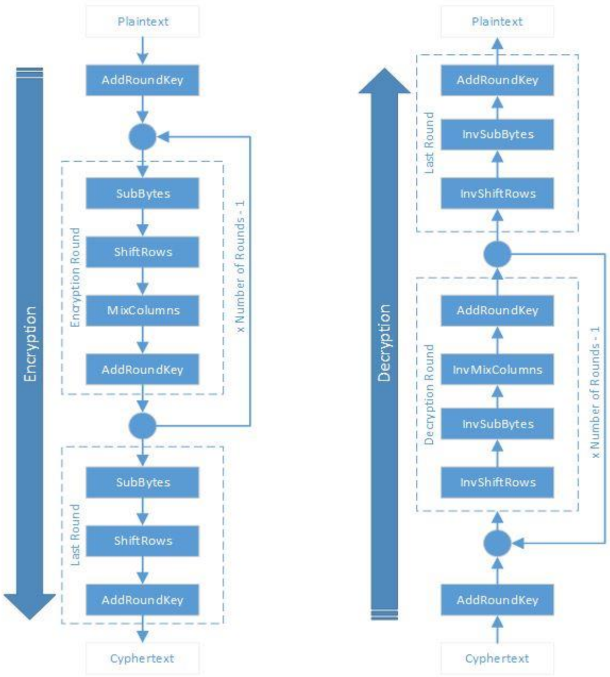


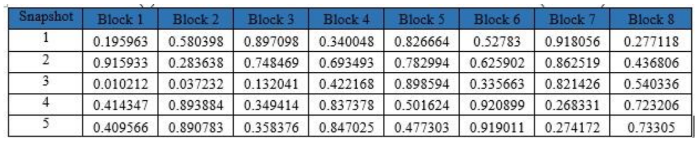
Figure (6) AES algorithm structure encryption/decryption [14]

Figure (6) shows the steps of the encryption process which results a cipher text, and reversing these steps from step8 to step1will gain the plain text again.

IV. Results

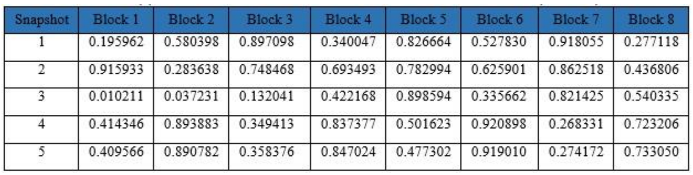
In this section there are two phases:
The first phase includes key generation of two lengths (128 and 256) for ear and eye images by using Meerkat algorithm as shown in tables (1, 2, 3, 4).
The second phase choses a plain text which is used for encryption / decryption by using the Advanced Encryption Standard (AES), and the keys used for encryption or decryption process are the keys generated using Meerkat algorithm as shown in figure(7) and tested via NIST(National Institute of Standards and Technology) packages.

TABLE (1): KEYS GENERATED BY MKGA FOR EAR WITH KEY SIZE 128 BITS (PERSON 1)



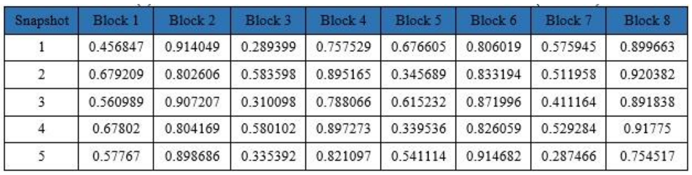
Snapshot	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	Block 8
1	0.195963	0.580398	0.897098	0.340048	0.826664	0.52783	0.918056	0.277118
2	0.915933	0.283638	0.748469	0.693493	0.782994	0.625902	0.862519	0.436806
3	0.010212	0.037232	0.132041	0.422168	0.898594	0.335663	0.821426	0.540336
4	0.414347	0.893884	0.349414	0.837378	0.501624	0.920899	0.268331	0.723206
5	0.409566	0.890783	0.358376	0.847025	0.477303	0.919011	0.274172	0.73305

TABLE (2): KEYS GENERATED BY MKGA FOR EAR WITH KEY SIZE 256 BITS (PERSON1)



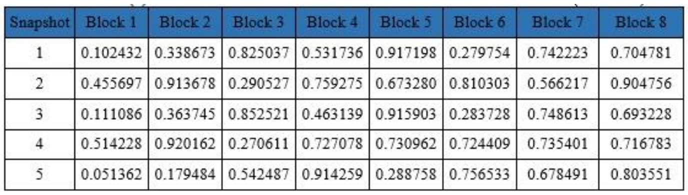
Snapshot	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	Block 8
1	0.195962	0.580398	0.897098	0.340047	0.826664	0.527830	0.918055	0.277118
2	0.915933	0.283638	0.748468	0.693493	0.782994	0.625901	0.862518	0.436806
3	0.010211	0.037231	0.132041	0.422168	0.898594	0.335662	0.821425	0.540335
4	0.414346	0.893883	0.349413	0.837377	0.501623	0.920898	0.268331	0.723206
5	0.409566	0.890782	0.358376	0.847024	0.477302	0.919010	0.274172	0.733050

TABLE (3): KEYS GENERATED BY MKGA FOR EYE WITH KEY SIZE 128 BITS (PERSON 1)



Snapshot	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	Block 8
1	0.456847	0.914049	0.289399	0.757529	0.676605	0.806019	0.575945	0.899663
2	0.679209	0.802606	0.583598	0.895165	0.345689	0.833194	0.511958	0.920382
3	0.560989	0.907207	0.310098	0.788066	0.615232	0.871996	0.411164	0.891838
4	0.67802	0.804169	0.580102	0.897273	0.339536	0.826059	0.529284	0.91775
5	0.57767	0.898686	0.335392	0.821097	0.541114	0.914682	0.287466	0.754517

TABLE (4): KEYS GENERATED BY MKGA FOR EYE WITH KEY SIZE 256 BITS (PERSON 1)



Snapshot	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	Block 8
1	0.102432	0.338673	0.825037	0.531736	0.917198	0.279754	0.742223	0.704781
2	0.455697	0.913678	0.290527	0.759275	0.673280	0.810303	0.566217	0.904756
3	0.111086	0.363745	0.852521	0.463139	0.915903	0.283728	0.748613	0.693228
4	0.514228	0.920162	0.270611	0.727078	0.730962	0.724409	0.735401	0.716783
5	0.051362	0.179484	0.542487	0.914259	0.288758	0.756533	0.678491	0.803551

TABLE (5): RUN TIME OF ENCRYPTION KEY

Bits Length of key	**Time Required in ms**
128	658 ms
256	1,052 ms

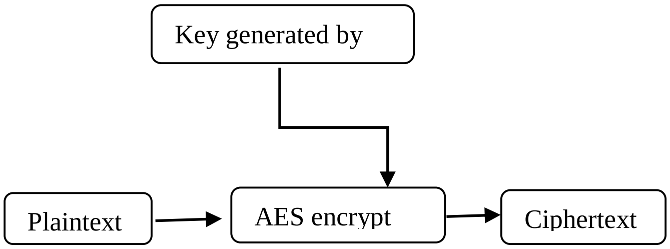


Figure (7) Encryption and decryption

P-value is calculated based on the degrees of freedom (K) data as the first parameter and the χ 2 data as the second parameter. There are ten tests generated by NIST used in order to verify the randomness of the keys before using it in the Encrypt/ Decrypt algorithms. NIST has adopted broadly two approaches, the P-value is calculated on application (ON LIN) Random number's name and p- value must be ranged between zero and one. Samples tests for keys generation (eye, ear) about length (128 and 256 bits) are shown in Tables (6…… 13), and vary as key 256 bits in most tests do not fail, but in key 128 bits some fail.

TABLE (6) EYE-128 BITS- RANDOM TEST FOR FREQUENCY TO BINARY MATRIX RANK

NO. OF SHOTS	Frequency (Monobit) Test	Frequency Test within a Block	Runs Test	Longest Run of Ones in a Block	Binary Matrix Rank Test
	0.282297	0.955550	0.323947	0.674916	0.039105
	0.612882	0.560367	0.993539	0.144935	0.039105
	0.849515	0.824371	0.848618	0.364511	0.039105
	0.657969	0.111375	0.756492	0.394240	0.039105
	0.375921	0.353732	0.819359	0.152368	0.039105
	0.486616	0.087537	0.911426	0.648665	0.039105
	0.164104	0.996416	0.311603	0.839146	0.039105
	0.949571	0.153548	0.657876	0.720812	0.039105
	0.486616	0.326832	0.383998	0.876990	0.039105
	0.751830	0.959877	0.701962	0.928763	0.039105

TABLE (7) EYE 128 BITS –RANDOM TEST FOR DISCRETE FOURIER TO CUMULATIVE

NO. OF KEYS	Discrete Fourier Transform (Spectral)	Non Overlapping Template	Serial		Approximate	Cumulative Sums
	0.146793	0.001334	0.720255	0.751940	1.000000	0.282297
	0.146793	0.078957	0.751574	0.384785	1.000000	0.823133
	0.245739	0.051659	0.999772	0.973110	1.000000	0.618347
	0.146793	0.031921	0.934859	0.844195	1.000000	0.765607
	1.000000	0.026696	0.986738	0.986215	1.000000	0.200192
	0.146793	0.077897	0.871263	0.557825	1.000000	0.328147
	0.383988	0.018879	0.010458	fail	1.000000	0.200192
	0.383988	0.937575	0.997629	0.921674	1.000000	0.823133
	0.081659	0.744239	0.884780	0.740173	1.000000	0.589898
	0.081659	0.064500	fail	0.067167	1.000000	0.022824

Table (8) ear 256 bits ear Random test for frequency to binary matrix

NO. OF KEYS	Frequency (Monobit) Test	Frequency Test within a Block	Runs Test	Longest Run of Ones in a Block	Binary Matrix Rank Test
1	0.205903	0.980789	0.131642	0.512625	0.039105
2	0.375921	0.964293	0.595299	0.884161	0.039105
3	0.849515	0.933757	0.848618	0.648665	0.039105
4	0.704336	0.121942	0.902935	0.436015	0.039105
5	0.311572	0.983366	0.505719	0.973813	0.039105
6	0.066636	0.959877	0.833233	0.479766	0.039105
7	0.410968	0.070909	0.038723	0.421398	0.039105
8	0.899343	0.308558	0.752210	0.149276	0.039105
9	0.527089	0.176710	0.518692	0.839146	0.039105
10	0.849515	0.985884	0.183735	0.969343	0.039105

Table (9) Ear 256 bits Random test for Discrete Fourier transform to Cumulative sums

NO. OF KEYS	Discrete Fourier Transform (Spectral)	Non Overlapping Template	Serial		Approximate	Cumulative Sums
	0.042221	0.042221	0.291810	0.258252	1.000000	0.291508
	1.000000	0.001339	0.765243	0.509566	1.000000	0.618347
	0.561658	0.001644	0.986343	0.882034	1.000000	0.850473
	0.245739	0.074730	0.955698	0.862054	1.000000	0.989720
	0.561658	0.001272	0.730770	0.782710	1.000000	0.291508
	0.042221	0.000841	0.390009	1.000000	1.000000	0.073758
	1.000000	0.054418	0.092340	1.000000	1.000000	0.227688
	0.020259	0.084852	0.999886	0.993351	1.000000	0.765607
	0.771671	0.092880	0.912121	0.883339	1.000000	0.483072
	0.383988	0.001163	0.591851	0.468814	1.000000	0.647327

Table (10) Eye Random test for frequency to binary matrix test

NO. OF KEYS	Frequency (Monobit) Test	Frequency Test within a Block	Runs Test	Test for the Longest Run of Ones in a Block	Binary Matrix Rank Test
	0.113846	0.146200	0.515762	0.688351	0.039105
	0.657969	0.949154	0.482656	0.727632	0.039105
	0.704336	0.121942	0.483705	0.810619	0.039105
	0.657969	0.733712	0.756492	0.789799	0.039105
	0.447884	0.888650	0.458597	0.497085	0.039105
	0.486616	0.102529	0.911426	0.555331	0.039105
	0.311572	0.783900	0.924946	0.738212	0.039105
	0.949571	0.355453	0.657876	0.886893	0.039105
	0.949571	0.267555	0.527170	0.696398	0.039105
	0.751830	0.819918	0.409124	0.891617	0.039105

Table (11) Eye Random test for Discrete Fourier transform to Cumulative sums

B NO. OF KEYS	Discrete Fourier Transform (Spectral)	Non Overlapping Template	Serial		Approximate	Cumulative Sums
	0.383988	0.066494	0.108857	0.123736	0.889631	0.113846
	0.146793	0.001057	0.985113	0.943729	0.898760	0.850473
	0.771671	0.075643	0.961917	0.970719	0.946084	0.458362
	0.146793	0.031921	0.934859	0.934859	0.879957	0.765607
	0.042221	0.001721	0.979505	0.958517	0.944730	0.794732
	0.146793	0.077897	0.871263	0.557825	0.823967	0.850473
	0.383988	0.001921	0.772008	0.636775	0.505470	0.483072
	0.383988	0.089515	0.997629	0.921674	0.922587	0.876384
	0.383988	0.054268	0.939758	0.855239	0.895621	0.989720
	0.383988	0.001381	0.634688	0.262024	0.988241	0.971736

Table (12) Ear -128- Random Test for frequency to binary matrix rank

NO. OF KEYS	Frequency (Monobit) Test	Frequency Test within a Block	Runs Test	Test for the Longest Run of Ones in a Block	Binary Matrix Rank Test
1	0.899343	0.079343	0.949973	0.312667	0.039105
2	0.447884	0.178914	0.458597	0.720812	0.039105
3	0.342782	0.170972	0.391031	0.541472	0.039105
4	0.569214	0.982306	0.126355	0.914509	0.039105
5	0.311572	0.992456	0.505719	0.973813	0.039105
6	0.410968	0.187092	0.720090	0.212739	0.039105
7	0.751830	0.228327	0.701962	0.112114	fail
8	0.899343	0.399105	0.752210	0.149276	fail
9	0.949571	0.932330	0.949470	0.257132	fail
10	0.849515	0.956748	0.183735	0.969343	fail

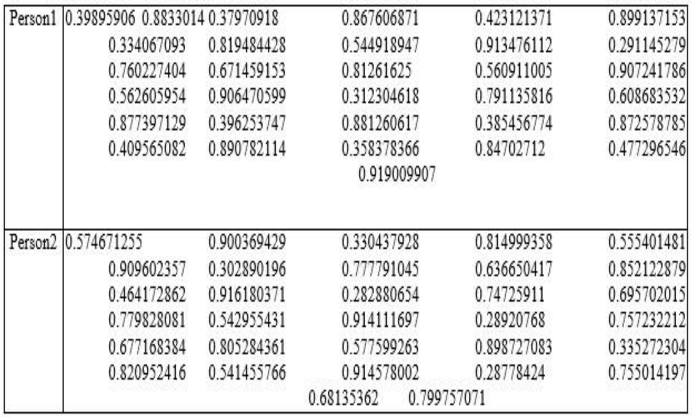
Table (13) Ear -128- Random test for discrete Fourier to cumulative sums

NO. OF KEYS	Discrete Fourier Transform (Spectral)	Non Overlapping Template	Serial		Approximate	Cumulative Sums
	0.561658	0.373738	0.768632	0.346838	1.000000	0.618347

	0.561658	0.038775	0.887405	0.853867	1.000000	0.765607
	0.383988	0.041843	0.560107	0.461648	0.999999	0.291508
	0.561658	0.001060	0.638271	0.508299	1.000000	0.941731
	0.561658	0.001272	0.730770	0.782710	1.000000	0.291508
	0.561658	0.095090	0.867084	0.652532	1.000000	0.618347
	0.245739	0.067539	0.966231	0.930787	1.000000	0.958243
	0.383988	fail	0.982911	0.980465	0.925712	0.922381
	0.383988	fail	0.591851	0.468814	0.585713	0.647327
	0.146793	fail	0.798533	0.535330	0.885780	0.850473

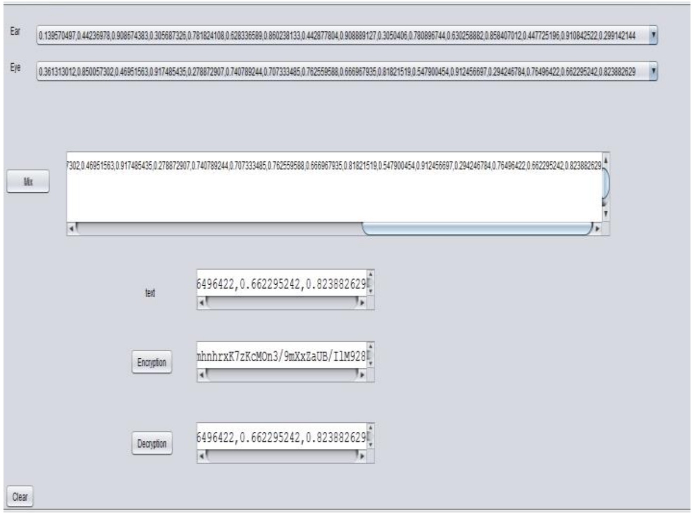
After the test, the better key is the less p- value. Each four persons by mathematic operation statistical action and mixed in continuation function, one key ear and one key eye become new key length 512, as shown in table 14 below:

TABLE (14) MIXED BEST KEYS (EAR 256 BITS, EYE 256 BITS)



Person1	0.39895906 0.8833014 0.37970918 0.867606871 0.423121371 0.899137153
	0.334067093 0.819484428 0.544918947 0.913476112 0.291145279
	0.760227404 0.671459153 0.81261625 0.560911005 0.907241786
	0.562605954 0.906470599 0.312304618 0.791135816 0.608683532
	0.877397129 0.396253747 0.881260617 0.385456774 0.872578785
	0.409565082 0.890782114 0.358378366 0.84702712 0.477296546
	0.919009907

Person2	0.574671255 0.900369429 0.330437928 0.814999358 0.555401481
	0.909602357 0.302890196 0.777791045 0.636650417 0.852122879
	0.464172862 0.916180371 0.282880654 0.74725911 0.695702015
	0.779828081 0.542955431 0.914111697 0.28920768 0.757232212
	0.677168384 0.805284361 0.577599263 0.898727083 0.335272304
	0.820952416 0.541455766 0.914578002 0.28778424 0.755014197
	0.68135362 0.799757071



Ear 0.139570497,0.44236978,0.908674383,0.305687326,0.781824108,0.628336589,0.860238133,0.442877804,0.908889127,0.3050406,0.780896744,0.630258882,0.858407012,0.447725196,0.910842522,0.299142144

Eye 0.361313012,0.850057302,0.46951563,0.917485435,0.278872907,0.740789244,0.707333485,0.762559588,0.666967935,0.81821519,0.547900454,0.912456697,0.294246784,0.76496422,0.662295242,0.823882629

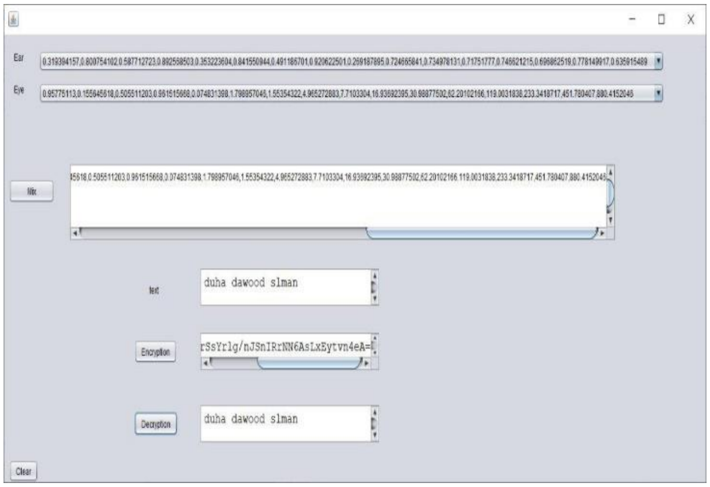
7302,0.46951563,0.917485435,0.278872907,0.740789244,0.707333485,0.762559588,0.666967935,0.81821519,0.547900454,0.912456697,0.294246784,0.76496422,0.662295242,0.823882629

text 6496422,0.662295242,0.823882629

Encryption mhnhrxK7zKcMOn3/9mXxZaUB/IlM928

Decryption 6496422,0.662295242,0.823882629

Figure (7) key Encryption and decryption processes



Ear 0.319394157,0.800754102,0.587712723,0.892558503,0.353223604,0.841550944,0.491186701,0.920622501,0.259187895,0.724665841,0.734978131,0.71751777,0.746621215,0.696862519,0.778149917,0.635915489

Eye 0.95775113,0.155645618,0.505511203,0.961515668,0.074831398,1.798957046,1.55354322,4.965272883,7.7103304,16.93692395,30.98877502,62.20102166,119.0031838,233.3418717,451.780407,880.4152046

45618,0.505511203,0.961515668,0.074831398,1.798957046,1.55354322,4.965272883,7.7103304,16.93692395,30.98877502,62.20102166,119.0031838,233.3418717,451.780407,880.4152046

text duha dawood slman

Encryption rSsYrlg/nJSnIRrNN6AsLxEytvn4eA=

Decryption duha dawood slman

Figure (8) Encryption and decryption processes sample

Encryption process:

In this study, Meerkat algorithm is used to generate a key for encryption process as shown in Fig. 7, and to obtain a robust key we mix keys produced from (eye and ear) to get key length 512-bits, these keys are encrypted with AES algorithm, then the input plaintext is encrypted with AES algorithm by using the keys indicated above as shown in Fig. 8.

Decryption process:

It is the process of switching unreadable cipher text to readable information. As shown in Fig. 7, the cipher text is decrypted with AES algorithm by also using the keys generated by Meerkat algorithm.

V. CONCLUSIONS

This paper displays an efficient way to deal with cryptographic key generation from multiple biometric modalities. The proposed system creates keys produced from eye and ear biometric qualities. Our system utilizes two modules: eye and ear Multimodal biometric as features extrication module to generate random keys from Meerkat algorithm. Meerkat Clan Key Generation Algorithm (MCKGA) is used in this work for keys generation with two keys length such that 128-bits and 256-bits. Even more, for security reasons and also to get more secure keys we mix the keys produced from eye and ear and get robust keys with 512-bits length. All keys are tested statically by NIST test to generate random keys used in encryption process, and our approach generates a unique key used in cryptographic system by using Advanced Encryption Standard AES algorithm. As per the outcomes got from explorers, they show that the AES algorithm can significantly give more security in contrast with different algorithms. The exploratory results have shown that the security of the proposed way to deal with producing client specific cryptographic key is improved.

REFERENCES

[1] Belguechi, R., Alimi, V., Cherrier, E., Lacharme, P., & Rosenberger, C. (2011). An overview on privacy preserving biometrics. Recent Application in Biometrics, 65-84

[2] A. S. Rahma and Raghad A.Azeez, (2006), "A Proposed Passive Identification System
Using Ear Biometric Images", Informatics Institute for Postgraduate Studies of
the Iraqi Commission for Computers and Informatics , PHD thesis.

[3] A. Jagadeesan, T. Thillaikkarasi, and D. K. Duraiswamy, "Cryptographic Key Generation from Multiple Biometric Modalities: Fusing Minutiae with Iris Feature," Int. J. Comput. Appl., vol. 2, no. 6, pp. 16–26, 2010.

[4] Ahmed T. Sadiq Al-Obaidi, Hasanen S. Abdullah, Zied O. Ahmed: Meerkat Clan Algorithm: a New Swarm Intelligence Algorithm. In Indonesian Journal of Electrical Engineering and Computer Science, 2018

[5] R. Jayapal and P. Govindan, "Biometric encryption system for increased security," IMCIC 2018 - 9th Int. Multi-Conference Complexity, Informatics Cybern. Proc., vol. 2, no. Imcic, pp. 6–11, 2018.

[6] Abuguba, Saad, Milan M. Milosavljevic, and Nemanja Macek. "An efficient approach to generating cryptographic keys from face and iris biometrics fused at the feature level." International Journal of Computer Science and Network Security (IJCSNS) 15.6 (2015): 6.

[7] N. Maček, B. Đorđević, J. Gavrilović, and K. Lalović, "An approach to robust biometric key generation system design," Acta Polytech. Hungarica, vol. 12, no. 8, pp. 43–60, 2015.

[8] J. G. Jo, J. W. Seo, and H. W. Lee, "Biometrie digital signature key generation and cryptography communication based on fingerprint," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 4613 LNCS, pp. 38–49, 2007.

[9] Panchal, Gaurang, and Debasis Samanta. "Directional area based minutiae selection and cryptographic key generation using biometric fingerprint." Proceedings of the First International Conference on Computational Intelligence and Informatics. Springer, Singapore, 2017

[10] Upmanyu M, Namboodiri A, Srinathan K, Jawahar C (2010) IEEE Trans Inf Forensics Secur 5:255.

[11] Chang, Yao-Jen, Wende Zhang, and Tsuhan Chen. "Biometrics-based cryptographic key generation." 2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No. 04TH8763). Vol. 3. IEEE, 2004.

[12] Wu, Lifang, et al. "A novel key generation cryptosystem based on face features." IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS. IEEE, 2010.

[13] Panchal, Gaurangkumar, and Debasis Samanta. "Comparable features and same cryptography key generation using biometric fingerprint image." 2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB). IEEE, 2016.

[14]M. A. Ako, "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data," Cryptogr. Netw. Secur., no. June, 2017.