Encrypting Biometric File using Enhanced AES and Storing on the Cloud

[1]Singampalli Gowthami, [2]Kezia Rani Badhiti

[1]M.Tech, [2]Assistant Professor
University college of Engineering,
Adikavi Nannaya Univrsity, Rajamahendravaram, India

Abstract: **A biometric system is a technological system that uses information about a person (or other biological organism) to identify that person. Biometric systems rely on specific data about unique biological traits in order to work effectively. By using biometrics, it is possible to confirm or establish an individual's identity based on "who she is," rather than by "what she possess" (e.g., an ID card) or "what she remembers" (e.g., a password).**

This system is capable of capturing uni-modal biometric traits such as capturing finger-prints and sent them to the cloud by end-to-end encryption process. An enhanced encryption method with AES-256 algorithm is used for protecting the biometric traits on insecure communication path. The principal component of AES block cipher is the S-Box. To increase the difficulty of the system, AES is used in Round structure. The encrypted biometric information is stored on the cloud.

The encrypted biometric image is converted into binary code and stored into the local database. This binary code is decrypted into original image. This local database is connected with cloud SQL.

***IndexTerms* - Raspberry Pi, AES Algorithm, PN Sequence generator**

I. INTRODUCTION

Biometric authentication is that every person is unique and each individual can be identified by his or her intrinsic or behavior traits. Biometric technology is able to recognize a person on the basis of the unique features of their face, fingerprint, signature, DNA or iris pattern and then impart a secure and convenient method for authentication purposes. Biometrics is therefore the measurement and statistical analysis of a person’s physical and behavioral characteristics.

In the near future, we may no longer use passwords and PIN numbers to authenticate ourselves as they can be shared, which increases the likelihood of malicious or unaccountable use. These methods are proved to be insecure and unsafe time and time again. Technology has introduced us a much smarter solution called Biometrics. The word Biometrics derived from “bio” means life and “metric” that means measurement. Biometrics refers to the quantifiable data (or metrics) related to human characteristics and traits. Biometric identification (or biometric authentication) is used in computer science as a form of identification and access control. It is also used to identify individuals in groups that are under surveillance. Biometric identifiers are the distinctive, measurable characteristics used to label and describe individuals.

Biometric systems depend on “who is she” rather than “what she has”. The application of biometric systems in ATMs, banks, attendance management, aadhar card etc is increasing. In the proposed architecture, the physiology biometric traits like finger are captured using RPi compatible hardware devices like fingerprint scanner and are sent to cloud for better performance.

The fingerprint scanner is interfaced using LibScan API and libusb libraries on the RPi, is ready for enrolment process. As soon as the motion sensor detects motion, a desktop application pops up which initiates login process. After the image capturing process, the biometrics are encrypted on the RPi using the proposed AES-256 algorithm along with Round-structure and dynamic S-box generation based on pseudo noise sequence generator.

For the transmission of biometric traits to the cloud, these biometric data needs to pass through unsecure channel. There are chances that these important data may be captured or altered by intruders which will give them the right to access an important service etc. or can cause denial-of-service to the legitimate user.

II. LITERATURE SURVEY

The first modern use of fingerprints occurred in 1856 when Sir William Herschel, the Chief Magistrate of the Hooghly district in Jungipoor, India, had a local businessman, Rajyadhar Konai, impress his handprint on the back of a contract. Later, the right index and middle fingers were printed next to the signature on all contracts made with the locals. The purpose was to frighten the signer of repudiating the contract because the locals believed that personal contact with the document made it more binding. As his fingerprint collection grew, Sir Herschel began to realize that fingerprints could prove or disprove identity. Despite his lack of scientific knowledge in fingerprinting he was convinced that fingerprints are unique and permanent throughout life.

The French anthropologist, Alphonse Bertillon, devised the first widely accepted scientific method of biometric identification in 1870. The Bertillon system, Bertillonage, or anthropometry was not based on fingerprinting but relied on a systematic combination of physical measurements. These, among others, included measurements of the skull width, foot length, and the length of the left middle finger combined with hair colour, eye colour, as well as face and profile pictures. By grouping the data any single person could be placed into one of 243 distinct categories. For the next thirty years, Bertillonage was the primary method of biometric identification.

Dr. Henry Faulds, British Surgeon-Superintendent of the Tsukiji Hospital in Tokyo, took up the study of fingerprints in the 1870's after noticing finger imprints on prehistoric pottery. In 1880, in the October 28 issue of the British scientific periodical Nature, Dr. Faulds was the first to publish a scientific account of the use of fingerprint as a means of identification. In addition to recognizing the importance of fingerprints, for identification he devised a method of classification as well. Dr. Faulds is credited for the first fingerprint identification-based on a fingerprint left on an alcohol bottle. The method of classification proposed by Dr. Faulds is called Henry Classification system and is based on patterns such as loops and whorls, which is still used today to organize fingerprint card files.

Continuing the work of Dr. Faulds, Sir William Herschel and Sir Francis Galton established the individuality and permanence of fingerprints. "Fingerprints" from 1892, contains the first fingerprint lassification system containing three basic pattern types: loop, arch, and whorl. The system was based on the distribution of the pattern types on the ten fingers, e.g. LLAWL LWWLL. The system worked, but was yet to be improved with a classification that was easier to administer. Sir Galton identified the characteristics used for personal identification, the unique ridge characteristics known as minutiae, which are often referred to as "Galton's details"

During the 1890's, Sir Edward Richard Henry, a British official in Bengal believed that a fingerprinting system was the solution to his problem of verifying the identity of criminals. He studied the works of Sir Galton and Sir Henry and proved that they could be used to produce 1,024 primary classifications, which was instituted in Bengal in 1897. The system is described in his book, "Classification and Uses of Finger Prints". In June 1897, Bertillonage was replaced and the Henry Classification System became the official method of identifying criminals in British India.

In 1901, Sir Henry, established the first fingerprint files in London. Subsequently, within the next 25 years, the Henry Classification System was adopted as the universally accepted method of personal identification by law enforcement agencies throughout the world. It is still in use, though several variants of the Henry Classification System exist.

In 1903, the Henry classification system was used to differentiate two prisoners who were identical twins. The Bertillon system was not able to make out the difference between identical twins and thus Henry classification system was further strengthened. Juan Vucetich also worked on a classification system based on the findings of Sir Galton and years of experience in fingerprint forensics. His system was published in his book, "Dactiloscopía Comparada". His system, the Vucetich System, is still used in most Spanish-speaking countries.

.

III. Raspberry Pi

The Raspberry Pi is low-cost small single board mini-computer developed in UK by the Raspberry Pi Foundation Team with the aim of teaching young students programming skills. The RPi is capable of doing everything what a PC can do. Different models of RPi are A, A+, B, B+ and the RPi 2. In this research, Raspberry Pi Model B is used which costs US $35. It has 2 USB ports and HDMI port for connection with the display, SD/MMC/SDIO card slot as RPi model B for booting and data storage on RPi do not have on-board storage. Also it has 10/100 Mbit/s Ethernet (8P8C) USB adapter for internet connection. To make RPi portable in this project wireless USB Wi-Fi adapter is used. The OS used is Raspbian (Jessie). RPi model B needs power supply of 5V-700mA (3.5 W).

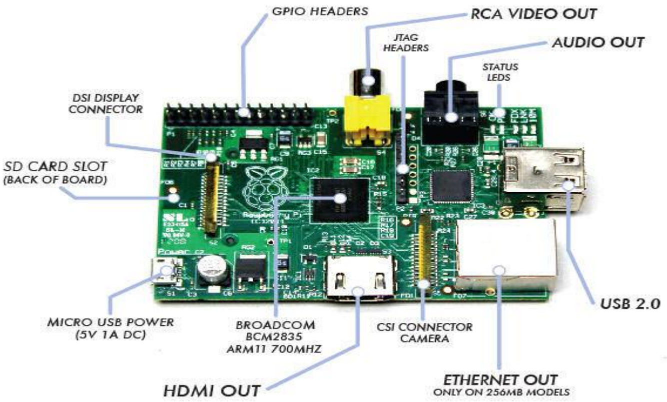

GPIO HEADERS
RCA VIDEO OUT
JTAG HEADERS
AUDIO OUT
STATUS LEDS
DSI DISPLAY CONNECTOR
SD CARD SLOT (BACK OF BOARD)
USB 2.0
MICRO USB POWER (5V 1A DC)
BROADCOM BCM2835 ARM11 700MHZ
CSI CONNECTOR CAMERA
HDMI OUT
ETHERNET OUT ONLY ON 256MB MODELS

Fig 1: Raspberry pi 3 Model B

IV. Encryption-AES 256

Security of image data is very important. In this project, a new Image Encryption technique is proposed. Here the face and the finger-print images are sent to the cloud for authentication purpose. Since they are transmitted via unsecure channel, it is important to have secure transmission. For this encryption is done using AES-256 Cipher Block Chaining mode in MATLAB. Substitute bytes: Uses an S-box to perform a byte-by-byte substitution of the block.
Shift-Rows: A simple permutation

Mix-Columns: A substitution that makes use of arithmetic over GF (2^8).
AddRoundKey: A simple bitwise XOR of the current block with a portion of the expanded key.

For AES, each with a fixed block size of 128 bits but three different key lengths: 128, 192 and 256 bits.AES is based on a design principle known as substitution-permutation network, a combination of both substitution and permutation.AES operates on a 4X4 column-major order matrix of bytes, termed as state. Most AES calculations are done in a particular finite field. The key size used for an AES cipher specifies the number of repetitions of transformation rounds that converts the input called the plain-text into final output called the cipher-text. The number of cycles of repetitions is as follows i.e., 10 cycles for 128-bit, 12 cycles for 192-bit and 14 cycles for 256-bit keys.

Each round consists of several processing steps, each containing four similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform cipher-text back into the original plaintext using the same encryption key.

Key Expansions-round keys are derived from the cipher key.

Initial Round-

a.AddRoundKey-each byte of the state is combined with a block of the round key using bitwise xor.

Rounds

a. SubBytes- a non-linear substitution step where each byte is replaced with another according to the look-up table.

b. ShiftRows-a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

c. MixColumns- a mixing operation which operates on the columns of the state, combining the four bytes in each column.

d.AddRoundKey

Final Round consists of SubByes, ShiftRows and AddRoundKey.

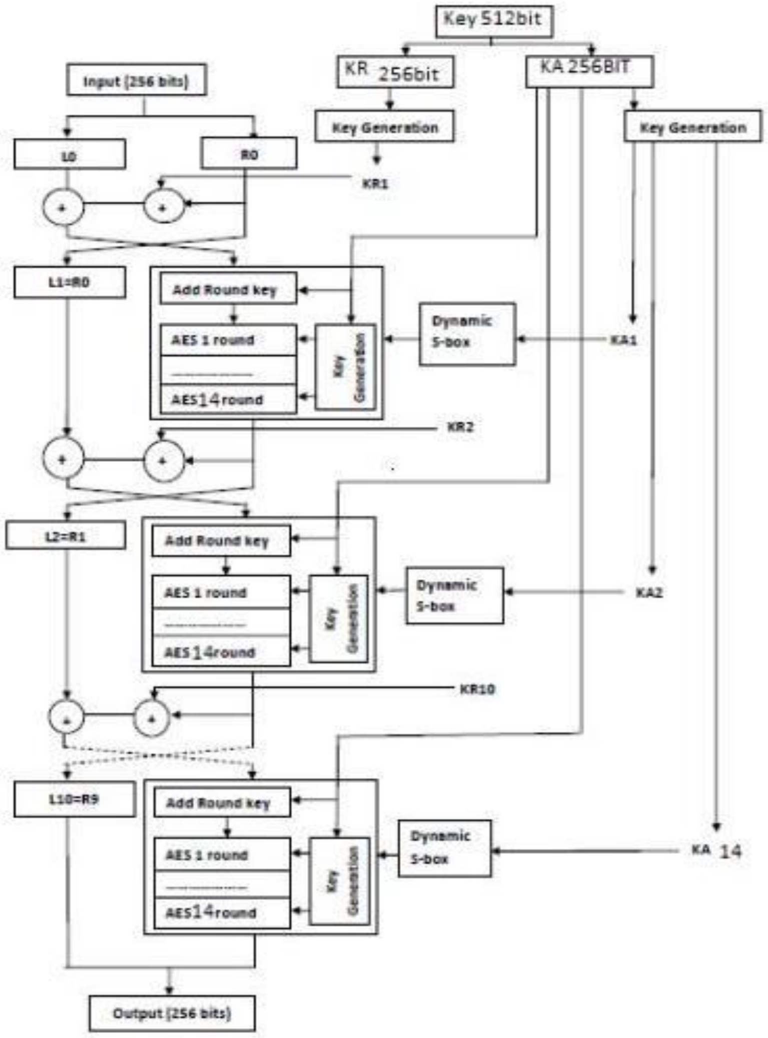


Fig2: Round AES with Dynamic S-Box

1. The Round structure of AES is used. Here the Input Data is split into two blocks of 128 bits each.

2. One Block is given as Input to the AES section of the System. The other Block is given as Input to the AES section of the System in the next round as per the Round structure.

3. This is done for all fourteen rounds respectively. These outputs are then combined together to form 256 bit block of encrypted data.

4. Dynamic S-box is applied to the Round structure of AES.

5. In the round structure, AES is applied n times to the block of data hence total n different S-boxes is created hence it is called dynamic S-box.

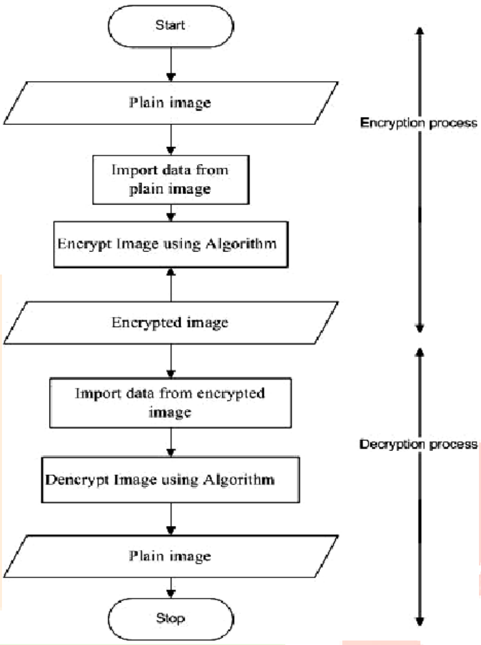

Start
Plain image
Import data from plain image
Encrypt Image using Algorithm
Encrypted image
Import data from encrypted image
Dencrypt Image using Algorithm
Plain image
Stop
Encryption process
Decryption process

Fig3: Flow chart of image using AES

V. PROPOSED SYSTEM

Biometrics is an automatic identification of a person by using certain physiological features associated with the person. Biometrics data is unique for every individual. So our project aims at storing Biometric data of user for the authentication process. For encryption process, Biometric data is encrypted using AES 256 algorithm into the form of a byte array and this byte array is converted to string, is stored into mysql database.
For decryption process, convert that retrieved string from database into byte array and decrypt into original image. Local storage ie., mysql database is connected to cloud SQL.

For connecting google cloud Sql with local host i.e., mysql. First an instance is created in cloud sql with local host ip and saved with instance name,an username and password is created for that instance.In local host, we use that instance ip address and access the databases,tables stored in cloud sql.

VI. THE ENHANCED AES 256 WITH ROUND STRUCTURE AND THE DYNAMIC S-BOX

The 256bits input and the 512bits key length is given to the enhanced AES system. The 512bits key length is divided into 256bits each. First 256bits are given to the round structure and the next 256bits are given to the AES algorithm. The AES secret key is modified and used as an initial state to the pseudo random sequence generator. Also XOR operation of all the bytes of the round key is taken to produce a new value. This new value along with the output of the pseudo random sequence generator is used to produce dynamic S-Box.

VII. THE PN SEQUENCE GENERATOR

The AES secret key is modified and given as input to the pseudo noise PN generator. The proposed generator consists of three maximal length linear feedback shift registers (LFSR) with thirty one, nineteen and fourteen taps. The feedback functions are chosen primitive to achieve a maximum period for each register. The output of the generator is two 16 hexadecimal values, *PNseq*.

PN sequence Generator The key length of this PN generator is (14+19+31) 64 so this generator needs 64 initial values. The AES-256 bits secret key is reshaped to four vector of 64 bit length and these four vectors are XORed with each other and the result is fed to PN generator as initial state. The Dynamic S-Box generation, the round key and the PNseqare used to produce a new number which is used as a shift value. The S-Box is rotated by that shift value. Thus the static S-Box is dynamically changed using the cipher key and the pseudo random sequence generator. The inverse S-Box is also revised after the S-Box to get the correct inverse values. In this paper, the enhanced AES can be used depending on the level of security required. In demand of moderate security Case 1 is used and when high security is required Case 2 is used. For the both the cases, the similar part is: Let modified AES secret key be the input to the pseudo random sequence generated where the output is 32 hexadecimal values, *PNseq*.

Case 1: Select the first byte of the *PNseq* and the first byte of the round key generated using the key expansion algorithm and XOR them. Use the resulting value as the shift value. Now the standard S-Box is rotated using this shift value.

Case 2: XOR all the bytes of *PNseq*. Let the new number be called as *PNvalue.* Also XOR all the bytes of the round key call it as *round Value.* Get the shift value by XOR-ing the *PNvalue* and the *roundValue*. Finally the standard S-Box is shifted using the shift value.

One important thing to note here is roundkey and hence the *roundValue* will change in each round. The *PNseq* and hence the *PNvalue* will be fixed throughout the algorithm.

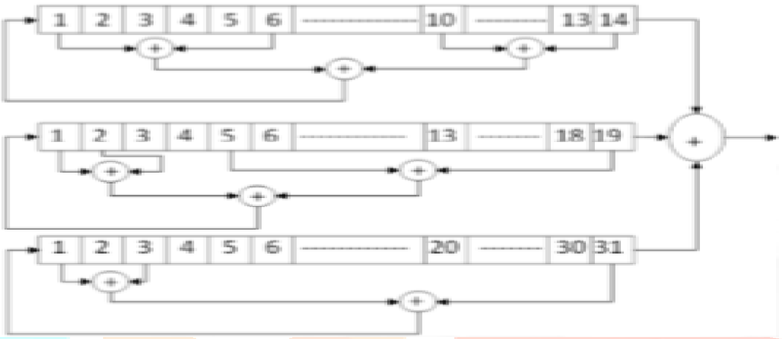


Fig 4: PN Sequence generator

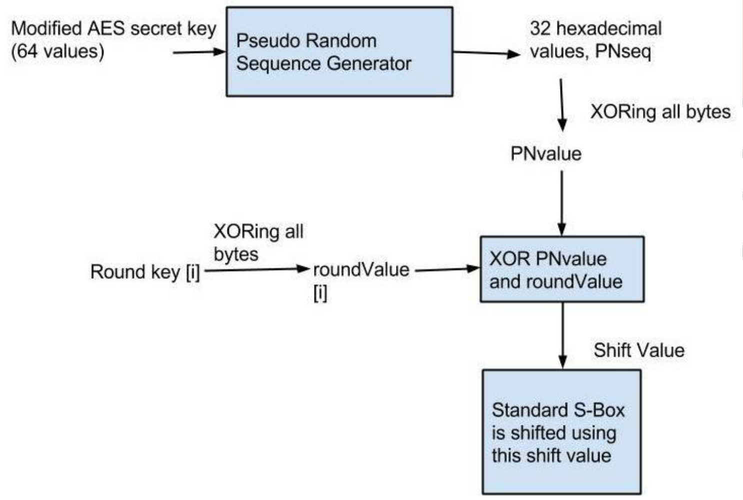
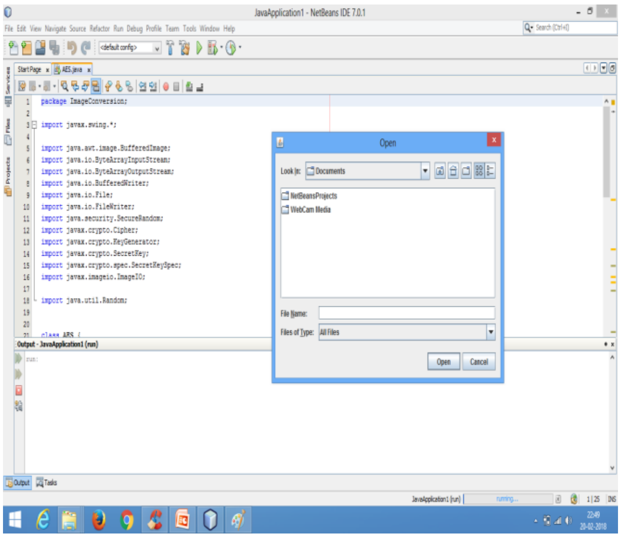

Modified AES secret key (64 values)
Pseudo Random Sequence Generator
32 hexadecimal values, PNseq
XORing all bytes
PNvalue
Round key [i]
XORing all bytes
roundValue [i]
XOR PNvalue and roundValue
Shift Value
Standard S-Box is shifted using this shift value

Fig 5: Proposed PN Sequence generator

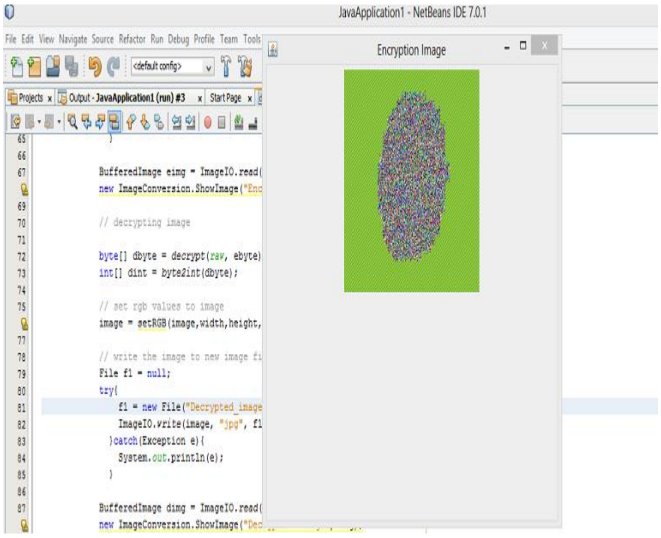
VIII. RESULTS



```
package ImageConversion;

import javax.swing.*;

import java.awt.image.BufferedImage;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.security.SecureRandom;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import javax.imageio.ImageIO;

import java.util.Random;


class AES {
```

JavaApplication1 - NetBeans IDE 7.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Start Page
AES.java
Output - JavaApplication1 (run)
run:
Open
Look In: Documents
NetBeansProjects
WebCam Media
File Name:
Files of Type: All Files
Open
Cancel

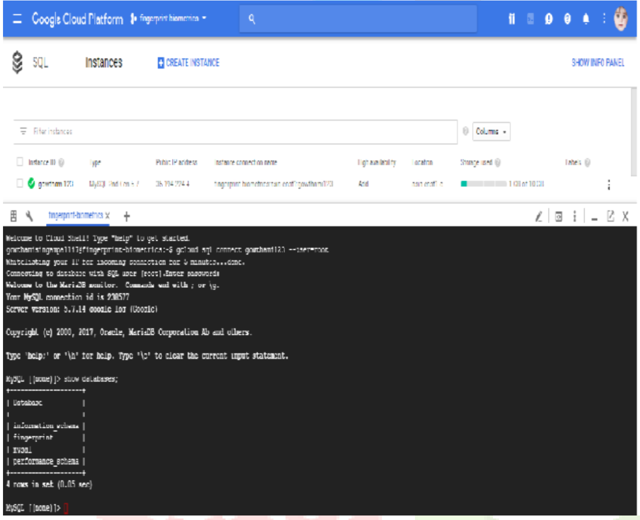
Choosing biometric files and encrypting them



```
        }

        BufferedImage eimg = ImageIO.read(
        new ImageConversion.ShowImage("Enc

        // decrypting image

        byte[] dbyte = decrypt(raw, ebyte)
        int[] dint = byte2int(dbyte);

        // set rgb values to image
        image = setRGB(image,width,height,

        // write the image to new image fi
        File f1 = null;
        try{
            f1 = new File("Decrypted_image
            ImageIO.write(image, "jpg", f1
        }catch(Exception e){
            System.out.println(e);
        }

        BufferedImage dimg = ImageIO.read(
        new ImageConversion.ShowImage("Dec
```

JavaApplication1 - NetBeans IDE 7.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools
<default config>
Projects
Output - JavaApplication1 (run) #3
Start Page
Encryption Image

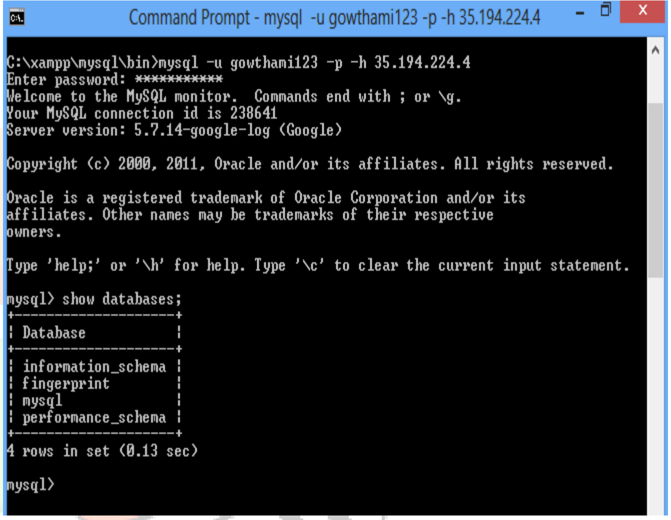
Encrypted Image of fingerprint



Instance ID	Type	Public IP address	Instance connection name	High availability	Location	Storage used
gowtham123	MySQL 2nd Gen 5.7	35.194.224.4	fingerprint-biometrics:asia-east1:gowtham123	Add	[illegible]	[illegible]

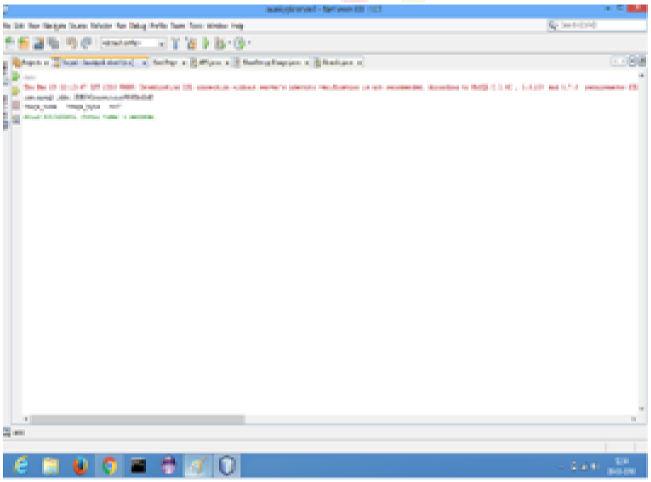
```
Welcome to Cloud Shell! Type "help" to get started.
gowthamisingampalli17@fingerprint-biometrics:~$ gcloud sql connect gowtham123 --user=root
Whitelisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 238577
Server version: 5.7.14-google-log (Google)

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| fingerprint        |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.05 sec)

MySQL [(none)]>
```

Creating a database in google cloud sql mysql

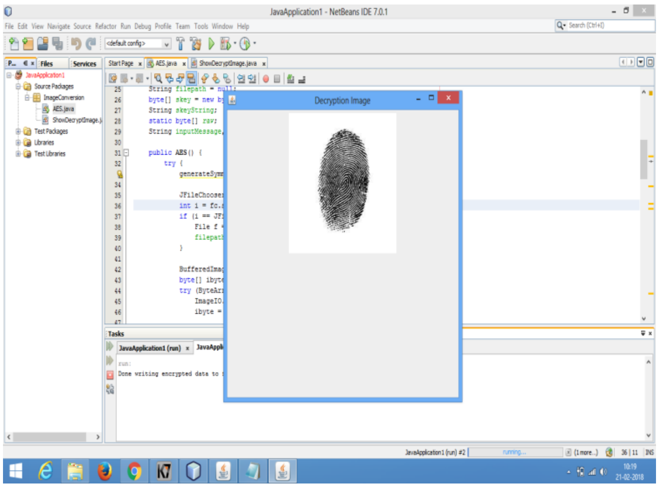


```
C:\xampp\mysql\bin>mysql -u gowthami123 -p -h 35.194.224.4
Enter password: ***********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 238641
Server version: 5.7.14-google-log (Google)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| fingerprint        |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.13 sec)

mysql>
```

Accessingdatabase in cloud sql from local host



Displaying the cloud sql data on a netbeans IDE



```
    String filepath = null;
    byte[] skey = new by
    String skeyString;
    static byte[] raw;
    String inputMessage,

    public AES() {
        try {
            generateSymm

            JFileChooser
            int i = fc.s
            if (i == JFi
                File f =
                filepath
            }

            BufferedImag
            byte[] ibyte
            try (ByteArr
                ImageIO.
                ibyte =
```

```
run:
Done writing encrypted data to f
```

Decrypted Image of fingerprint

IX CONCLUSION AND FUTURE WORK

The AES is implemented and the biometric traits are encrypted using Round Structure and the dynamic S-box. Using AES in Round structure with more no of rounds, runtime is increased but complexity of network is also increased. Increasing complexity will make the system attack resistant and secure data from attackers. So this system can be used in the application where time is not the constraint. The time taken by the AES in Round structure with one round is nearly same as traditional AES hence it can be used in the applications where speed is required with complexity.

The encrypted images will be sent to the cloud and decrypted there and feature extractions will be done. Performance measures of the enhanced AES will be calculated based on encryption and decryption time. Proposed system has application in all the biometric access control systems.

In cloud computing, the users are first enrolled in the biometric system supplied by a service provider or the cloud platform. During the enrolment process, individuals must submit multiple biometric fingerprints. These are then stored on the cloud provider's side as templates. The user is prompted to provide a scan of his fingerprint whenever they want to access a cloud service. The fingerprint is compared to the stored template. Users are authenticated and gain access once there is a positive match between the fingerprint scan and the stored template. The images from the user and the fingerprint templates are both encrypted for additional security. The biometric system is put into action every time a user wants to access a cloud service. The user will be directed to the cloud service platform that they have clearance to access once they complete successful authentication.|

REFERENCES

[1]https://www.codeproject.com/Articles/839230/Introduction-to-Raspberry-Pi-with-Raspbian-OS#Introduction

[2]https://www.google.co.in/search?q=how+to+create+cloud+for+raspberry+pi&oq=how+to+create+cloud+for+raspberry+pi&aqs=chrome..69i57j0l5.12463j0j7&sourceid=chrome&ie=UTF-8#kpvalbx=1

[3] Raspberry Pi, Available: http://en.Wikipedia.org/wiki/Raspberry_Pi

[4] Available: http://www.siongboon.com/projects/2013-07-08_raspberry_pi/images/raspberry_pi_circuit.jpg

[5] Available: http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[6] Camera Module,Available: http://www.raspberrypi.org/ documentation/ usage/camera/README.md.

[7] RPi Verified Peripherals, Available:http://elinux.org/RPi_VerifiedPeripherals#Fingerprint_Scanners

[8]https://www.elprocus.com/different-types-biometric-sensors

[9]https://www.tutorialspoint.com/biometrics/multimodal-biometric-systems.htm

[10]https://www.youtube.com/watch?v=zB8EqP2jltA

[11]S.Anandakumar "Image Cryptography Using RSA Algorithm in Network Security", IJCSET, September 2015, Volume5, Issue 9,326-330

[12] Seyed Hossein Kamali, Reza Shakerian, Maysam Hedayati, Mohsen Rahmani, "A New Modified Version of Advanced Encryption Standard (AES) Based Algorithm for Image Encryption" (2010)

[13] https://www.pi-supply.com/make/build-your-own-cloud-storage-file-server-with-a-raspberry-pi/

[14] https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-raspberry-pi-kit-python-get-started

[15] https://learn.adafruit.com/how-to-use-matlab-and-simulink-with-raspberry-pi/set-up-matlab-and-simulink-support-package-for-raspberry-pi

[16]https://blog.adafruit.com/2017/09/15/turn-your-raspberry-pi-3-into-a-personal-cloud-storage-piday-raspberrypi-raspberry_pi/

[17]https://normally.online/2016/04/29/owncloud-9-0-1-on-raspberry-pi-3-step-by-step/

[18] https:// github.com/leoxiong/image-Steganography/blob/master/src/Steganography.java

[19]www.biometrics-solutions.com/fingerprint-recognition.html

[20]blog.weston-fl.com/configure-netbeans-to-test-and-deploy-raspberry-pi-project

[21]https://youtu.be/hrShPd3Me5g

[22]https://youtu.be/xbazHnZR8AA

[23]https://cloud.google.com/sql/docs/mysql/connect-external-app#java

[24] J. C. Yang, N. X. Xiong, A. V. Vasilakos and Zh. J. Fang, ―A fingerprint recognition scheme based on assembling invariant moments for cloud computing communications,‖ IEEE Systems Journal, vol. 5, no. 4, Dec. 2011.

[25] K. S. Zhang, J. She, M. X. Gao, and W. B. Ma, ―Study on the embedded fingerprint image recognition system,‖ in Proc. Conference of Information Science and Management Engineering, IEEE, 2010, pp. 169-172.

[26] B. Y. Hiew, A. B. J. Teoh, and Y. H. Pang, ―Digital camera based fingerprint recognition,‖ in Proc. the 2007 IEEE International Conference on Telecommunications and Malaysia Conference on Communications, Penang, May 14-17, 2007

[27] D. R. Wan and J. Zhou, ―Fingerprint recognition using model-based density map,‖ presented at IEEE Transactions On Image Processing, vol. 15, no. 6, June 2006.

[28]www.circuitbasics.com/how-to-connect-to-a-raspberry-pi-directly-with-an-ethernet-cable/

[29] https://www.youtube.com/watch?v=XAowXcmQ-kA

[30]www.youtube.com/watch?v=meGhTnlS9k4