CHAPTER 1

INTRODUCTION

In the past few years the security and integrity of data is the main concern. In the present scenario almost all the data is transferred over computer networks due to which it is vulnerable to various kinds of attacks. To make the data secure from various attacks and for the integrity of data we must encrypt the data before it is transmitted or stored.

Cryptography is a method of storing and transmitting data in a form that only those it is intended for can read and process. It is a science of protecting information by encoding it into an unreadable format.

It is based on AES Key Expansion in which the encryption process is a bit wise exclusive or operation of a set of image pixels along with the a 128 bit key which changes for every set of pixels. The National Institute of Standards and Technology (NIST) has initiated a process to develop a Federal information Processing Standard (FIPS) for the Advanced Encryption Standard (AES), specifying an Advanced Encryption Algorithm to replace the Data Encryption standard (DES) the Expired in 1998.The Advanced Encryption Standard can be programmed in software or built with pure hardware thereby building an IoT based encrypted image data transfer system where we use raspberry pi to act as our image input system whose data shall be encrypted and transfer to our receiver pc.

CHAPTER 2

LITERATURE SURVEY

Sl.no	Name	Authors	Year	If any links
1.	Image Encryption Approach for Security Issues	J. V. Gorabal and Manjaiah D. H	2010	International Journal of Information Technology and Management Information Systems (IJITMIS), Volume 5, Issue 2, 2010, pp. 59 - 64, ISSN Print: 0976 – 6405, ISSN Online:0976 – 6413.
2.	Image Encryption Based on AES Key Expansion,	B.Subramanyan, Vivek.M.Chhabria, T.G.Sankar babu	2011	Second International Conference on Emerging Applications of Information Technology 978-0-7695-4329-1/11 $26.00 © 2011 IEEE DOI 10.1109/EAIT.2011.60 , 217_220
3.	Image Encryption and Compression Based on Compressive Sensing and Chaos	Prof. Maher K. Mahmood and Jinan N. Shehab	2014	International journal of Computer Engineering & Technology (IJCET), Volume 5, Issue 1, 2014, pp. 68 - 84, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.

4.	AES Encryption Engine For Many Core Processor Arrays For Enhanced Security	Dhanya Pushkaran and Neethu Bhaskar	2014	International journal of Electronics and Communication Engineering &Technology (IJECET), Volume 5, Issue 12, 2014, pp. 106 - 111, ISSN Print:0976- 6464, ISSN Online: 0976 – 6472.

CHAPTER 3

PROPOSED SYSTEM

The most common way to protect large multimedia files is by using conventional encryption techniques, Private key bulk encryption algorithms, such as Triple DES, are not so suitable for transmission of images. Due to complexity of their internal structure, they are not particularly fast in terms of execution speed and cannot be applied for images in the real time scenario Also traditional cryptographic techniques such as DES cannot be applied to images due to intrinsic properties of images such as bulk data capacity, redundancy and high correlation among pixels. Image encryption algorithms can become an integral part of the image delivery process if they aim towards efficiency and at same time preserve the security level.

CHAPTER 4

METHODOLOGY

4.1 HARDWARE SETUP:

- First we need to power up the Raspberry pi then we need to install its OS.
- To configure the camera module.
- To save and compress the captured image.

4.2 ENCRYPTION OF THE CAPTURED IMAGE:

Image encryption involves transforming an image into a format that is unreadable without the appropriate decryption key. Here's a basic outline of the image encryption process:

PREPROCESSING

Before encryption, you may choose to preprocess the image, such as converting it to grayscale or resizing it, depending on your requirements.

Choose Encryption Algorithm:

Select a suitable encryption algorithm. Common choices include symmetric encryption algorithms like AES or asymmetric encryption algorithms like RSA.

Key Generation:

Generate encryption keys. For symmetric encryption, you need a single secret key. For asymmetric encryption, you need both a public and a private key pair.

Encryption:

Apply the chosen encryption algorithm to the image data using the encryption key(s). The encryption process transforms the image data into ciphertext, making it unreadable without the decryption key(s).

The encryption may be applied to the entire image or specific regions of interest, depending on your requirements.

Storage or Transmission:

Store or transmit the encrypted image. If transmitting over a network, ensure secure transmission protocols like HTTPS or secure email are used to protect the encrypted image during transit.

Decryption (Optional):

If decryption is required, the recipient uses the appropriate decryption key(s) and algorithm to decrypt the ciphertext back into the original image data.

Decrypt the image data using the decryption key(s) and algorithm. This process transforms the ciphertext back into the original image, allowing it to be viewed.

Postprocessing (Optional):

After decryption, you may choose to apply any necessary postprocessing to the image, such as color adjustments or resizing.

It's important to note that encryption alone does not guarantee the security of the image. Proper key management, secure transmission/storage practices, and adherence to encryption best practices are essential for maintaining the confidentiality and integrity of the encrypted image. Additionally, the strength of the encryption depends on the chosen algorithm and key length.

4.3 SETTING UP OF RASPBERRY PI NETWORK

- There are several ways by which we can wirelessly connect a raspberry pi to a receiver device(such as phone or pc).
- Using any one of the method such as socketing technique or configuring a raspberry pi as dhcp server etc..,we can transmit encrypted image seamlessly and can be received at the receiver end.

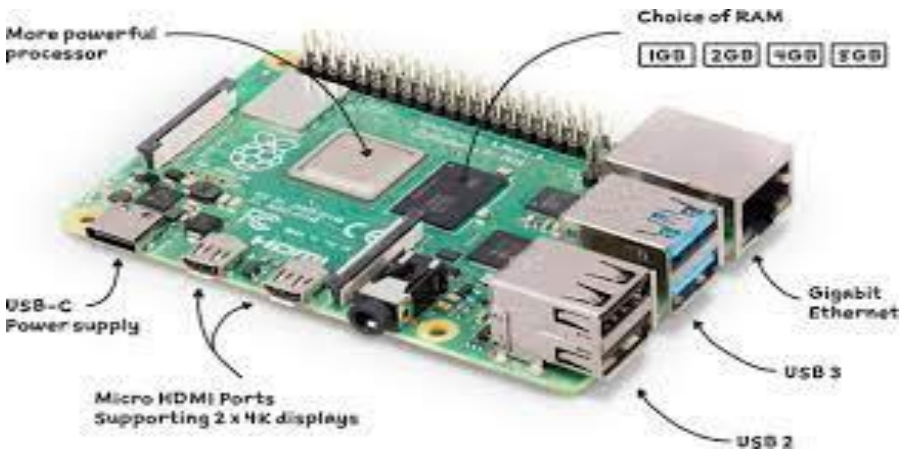


Figure 4.1-Raspberry pi 4

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 HARDWARE

5.1.1 Raspberry pi 4 Model B

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
- 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 × micro-HDMI® ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.1, Vulkan 1.0
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature: 0 – 50 degrees C ambient
- A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.

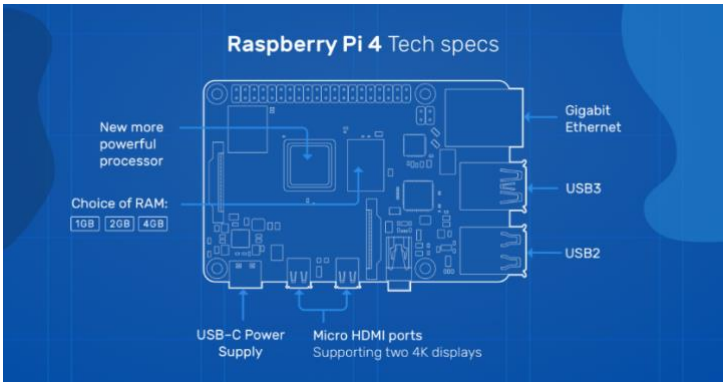


Figure 5.1-Overview of Raspberry pi

5.1.2 Raspberry pi 5MP Camera module

- Original Raspberry Pi Camera, supports all revisions of the Pi
- 5 megapixel OV5647 sensor in a fixed-focus module
- 2592 × 1944 still picture resolution
- Support 1080p30, 720p60 and 640x480p60/90 video record
- Dimension: 25mm x 24mm x 9mm

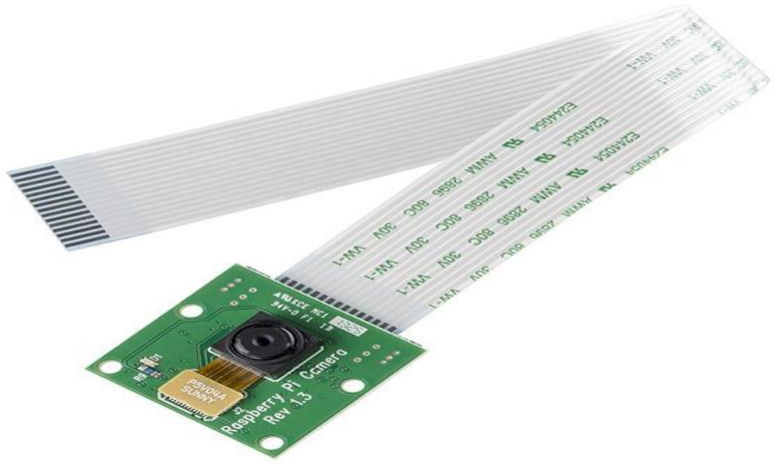


Figure 5.2-Camera Module

5.1.3 SanDisk 64 GB SD card

- 64GB Storage Capacity.
- UHS-I / V30 / U3 / Class 10.
- Max Read Speed: 200 MB/s.
- Max Write Speed: 90 MB/s.
- Min Write Speed: 30 MB/s.
- Records Full HD, 3D, and 4K Video.
- Built-In Write-Protect Switch.

Figure 5.3-SD card

5.2 SOFTWARE

For setting up of Raspberry Pi 4 we use,

5.2.1 PuTTY software

- **Developer**: Simon Tatham and contributors
- **License**: MIT License
- **Supported Platforms**: Windows, macOS, Linux, and Unix-like systems

➢ **Key Features**

- **Terminal Emulation**
  - **VT100 Emulation**: Provides VT100 terminal emulation, supporting a wide range of terminal operations.
  - **xterm, VT102, and ECMA-48 Emulation**: Extends support for additional terminal types and standards.

- **Network Protocols**
  - **SSH (Secure Shell)**: Supports SSH-1 and SSH-2 protocols for secure remote login.
  - **Telnet**: Provides support for the Telnet protocol.
  - **rlogin**: Allows connections using the rlogin protocol.
  - **Serial Communication**: Supports serial connections for managing hardware devices.
  - **SCP and SFTP**: Secure file transfer protocols for transferring files over SSH.

- **User Interface**
  - **Graphical User Interface (GUI)**: Offers a simple and intuitive GUI for configuring and managing connections.
  - **Command-Line Interface (CLI)**: Includes a command-line version (Plink) for automated tasks and scripts.
  - **Customizable Appearance**: Allows customization of the terminal appearance, including font, color, and window behavior.

- **Encryption**: Supports various encryption algorithms such as AES, DES, 3DES, and Blowfish for securing SSH connections.

  - **Public Key Authentication**: Allows the use of public key authentication for SSH, enhancing security.
  - **Host Key Verification**: Verifies the server's host key to prevent man-in-the-middle attacks.
  - **Session Logging**: Logs session activity for later review and auditing.

5.2.2 VNC viewer

VNC (Virtual Network Computing) Viewer is a remote desktop application that allows users to connect to and control a remote computer's desktop environment over a network. Developed by RealVNC, it supports multiple operating systems and provides various features to facilitate remote access and management.

- **Developer**: RealVNC Limited
- **License**: Varies (Free and Commercial licenses available)
- **Supported Platforms**: Windows, macOS, Linux, Unix-like systems, iOS, Android

- **Key Features**

  - **Remote Desktop Access**
    - **Full Desktop Control**: Allows users to control the remote desktop as if they were physically present at the machine.
    - **View Only Mode**: Users can view the remote desktop without being able to control it, which is useful for demonstrations and monitoring.

  - **Cross-Platform Compatibility**
    - **Multi-Platform Support**: Connects to remote desktops running on different operating systems, including Windows, macOS, and Linux.
    - **Mobile Support**: Provides mobile applications for iOS and Android, enabling remote access from smartphones and tablets.

- **Security Features**
  - **Encryption**: Supports end-to-end encryption to protect data transmitted over the network.
  - **Authentication**: Requires authentication before establishing a connection, supporting both password-based and multi-factor authentication.
  - **Access Control**: Allows configuration of access control lists to restrict which users can connect to the VNC server.

5.2.3 Raspberry pi Imager

Raspberry Pi Imager is a tool developed by the Raspberry Pi Foundation to simplify the process of installing operating systems onto SD cards for use with Raspberry Pi devices.

- **Developer**: Raspberry Pi Foundation
- **License**: Open Source (GNU GPL)
- **Supported Platforms**: Windows, macOS, Linux

➢ **Key Features**

- **Operating System Installation**
  - **Wide OS Selection**: Supports a variety of operating systems specifically designed for Raspberry Pi, including Raspberry Pi OS, Ubuntu, and others.
  - **Official and Third-Party OSes**: Provides access to both official Raspberry Pi Foundation OSes and third-party operating systems.

- **User Interface**
  - **Simple and Intuitive GUI**: Easy-to-use graphical interface suitable for beginners and experienced users.
  - **Step-by-Step Process**: Guides users through selecting the OS, specifying the storage device, and writing the image to the SD card.

- **Image Download and Writing**
  - **Direct Download**: Automatically downloads selected operating system images from the internet.
  - **Local Image Writing**: Allows users to write images from local storage, supporting custom or previously downloaded OS images.
  - **Data Verification**: Verifies written data to ensure the integrity and correctness of the installation.

- For this project we use VS code software and later adapt to Thonny Rasp python IDE.

5.2.4 Thonny

Thonny is a free and open-source integrated development environment for Python that is designed for beginners. It was created by Aivar Annamaa, an Estonian programmer. It supports different ways of stepping through code, step-by-step expression evaluation, detailed visualization of the call stack and a mode for explaining the concepts of references and heap.

- Line numbers.
- Statement stepping without breakpoints.
- Live variables during debugging.
- Stepping through evaluation of the expressions (expressions get replaced by their values).
- Separate windows for executing function calls (for explaining local variables and call stack).
- Variables and memory can be explained either by using simplified model (name → value) or by using more realistic model (name → address/id → value).
- Simple pip GUI.
- Support for CPython and MicroPython.
- Support for running and managing files on a remote machine via SSH.
- Possibility to log user actions for replaying or analyzing the programming process.

CHAPTER 6

IMPLEMENTATION

6.1 SETTING UP OF RASPBERRY PI

Setting up a Raspberry Pi 4 from a laptop using VNC Viewer involves several steps, including installing the Raspberry Pi OS, enabling VNC, and connecting remotely. The steps are given below,

- Step 1: Install Raspberry Pi OS on the SD Card:
  - Download Raspberry Pi Imager: Download and install the Raspberry Pi Imager from the official website.

- Step 2: Prepare the SD Card:
  - Insert the SD card into your laptop.
  - Open the Raspberry Pi Imager, select the Raspberry Pi OS (32-bit) or another version you prefer, choose your SD card, and click "Write".

- Step 3: Headless Setup: Enable SSH and VNC in the SD Card:
  - After writing the OS to the SD card, open the boot partition on your laptop.
  - Create an empty file named ssh (no extension) to enable SSH.
  - Create a file named “wpa_supplicant.conf” with the following contents to configure Wi-Fi (replace SSID and PASSWORD with your Wi-Fi credentials):

```
country=INDIA

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev

update_config=1

network={

   ssid="SSID"

   psk="PASSWORD"

   key_mgmt=WPA-PSK

}
```

- Step 3: Boot the Raspberry Pi:
  - Insert the SD card into the Raspberry Pi and power it up.
  - The Pi will connect to your Wi-Fi network.

- Step 4: Find the Raspberry Pi's IP Address:
  - Use your router's interface to find the Pi's IP address or use a network scanning tool like "nmap".

- Step 5: Connect via SSH:
  - Open a terminal on your laptop and type ssh pi@<IP_ADDRESS> (replace <IP_ADDRESS> with the actual IP.
  - Enter the default password (raspberry) when prompted.

- Step 6: Enable VNC via SSH:
  - Once connected via SSH, enable VNC by running: sudo raspi-config
  - Navigate to "Interfacing Options" > "VNC" > "Yes" to enable VNC.

- Step 7: Connect via VNC Viewer:
  - Connect to the Raspberry Pi via VNC.

- Step 8: Find the IP Address of the Raspberry Pi:
  - Open a terminal on the Raspberry Pi and type hostname -I to get the IP address.

- Step 9: Open VNC Viewer on Your Laptop:
  - Open VNC Viewer.
  - Enter the Raspberry Pi's IP address in the VNC Server field.
  - Click "Connect."

- Step 10: Authentication:
  - Enter the username and password for your Raspberry Pi when prompted.
  - Click "OK."

6.2 SETTING UP OF CAMERA MODULE

- Step 1: Connect the Camera Module:
  - Ensure your Raspberry Pi is powered off.
  - Attach the camera module to the CSI (Camera Serial Interface) port on the Raspberry Pi. Ensure the connector is properly seated.

- Step 2:Enable the Camera Interface:
  - Boot up your Raspberry Pi.
  - Open a terminal and run the configuration tool:

    sudo raspi-config

  - Navigate to "Interfacing Options" > "Camera" and select "Enable".
  - Reboot the Raspberry Pi if prompted.

- Step 3: Install Required Libraries:
- Step 4: Update Your System:
  - Open a terminal and run:

    sudo apt-get update
    sudo apt-get upgrade

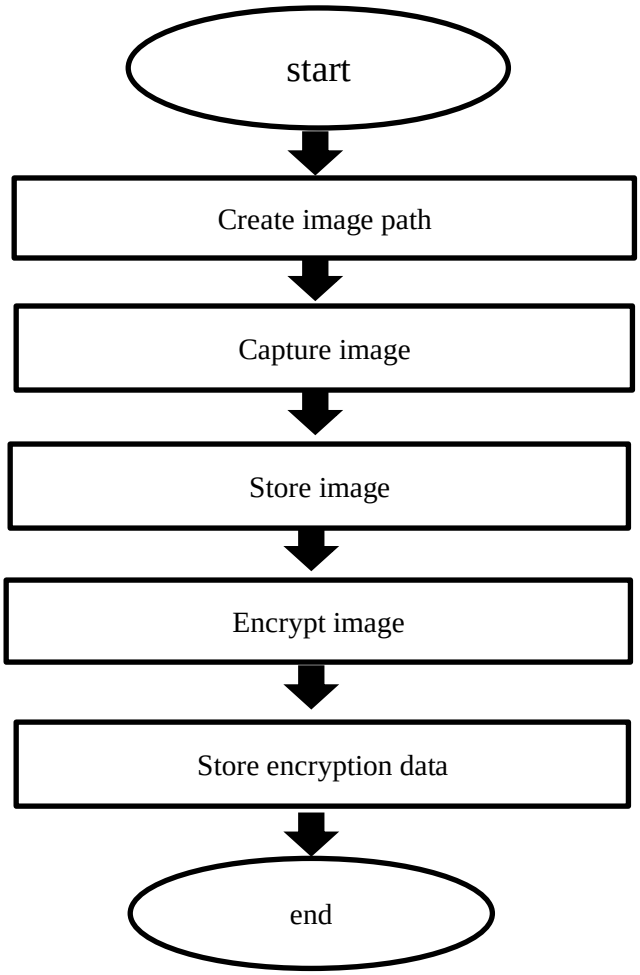
- step 5: Install picamera Library:
  - The picamera library provides a Python interface to the Raspberry Pi camera module.
  - Install it using pip:

    sudo apt-get install python3-picamera

- Step 6: Write Python Code to Control the Camera

Here is a basic example of how to capture an image and record a video using the picamera library.

FLOW CHART



//we create a folder called example in which image and key and IV and encrypted file

6.3 ENCRYPTION

Image encryption involves transforming an image into a format that is unreadable without the appropriate decryption key. Here's a basic outline of the image encryption process:

Preprocessing:

Before encryption, you may choose to preprocess the image, such as converting it to grayscale or resizing it, depending on your requirements.

Choose Encryption Algorithm:

Select a suitable encryption algorithm. Common choices include symmetric encryption algorithms like AES or asymmetric encryption algorithms like RSA.

Key Generation:

Generate encryption keys. For symmetric encryption, you need a single secret key. For asymmetric encryption, you need both a public and a private key pair.

Encryption:

Apply the chosen encryption algorithm to the image data using the encryption key(s). The encryption process transforms the image data into ciphertext, making it unreadable without the decryption key(s).

The encryption may be applied to the entire image or specific regions of interest, depending on your requirements.

Storage or Transmission:

Store or transmit the encrypted image. If transmitting over a network, ensure secure transmission protocols like HTTPS or secure email are used to protect the encrypted image during transit.

Decryption (Optional):

If decryption is required, the recipient uses the appropriate decryption key(s) and algorithm to decrypt the ciphertext back into the original image data allowing it to be viewed.

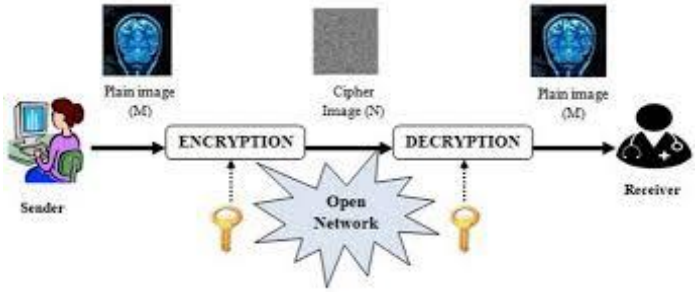


Figure 6.1-Encryption

6.4 SETTING UP OF RASPBERRY PI NETWORK

- There are several ways by which we can wirelessly connect a raspberry pi to a receiver device(such as phone or pc).
- Using any one of the method such as socketing technique or configuring a raspberry pi as dhcp server etc..,we can transmit encrypted image seamlessly and can be received at the receiver end.

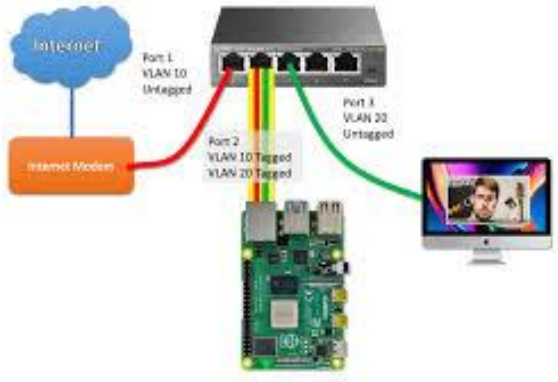


Figure 6.2-Raspberry pi network

6.5 DECRYPTION

Image decryption generally involves reversing the encryption process to restore the original image from its encrypted form. The process typically requires knowledge of the encryption algorithm used and the correct decryption key.

There are various methods to encrypt and decrypt images, such as:

1. AES (Advanced Encryption Standard):
   Encrypt: The image is treated as a byte array, and AES encryption is applied.
   Decrypt: AES decryption is applied to the byte array to retrieve the original image.

2. Steganography:
   Encrypt: The image is hidden within another image or media file.
   Decrypt: The hidden image is extracted from the carrier file.

3. LSB (Least Significant Bit) Modification:
   Encrypt: The least significant bits of the image pixels are modified to hide information.
   Decrypt: The hidden information is retrieved by examining the least significant bits.

4. Chaos-based Encryption:
   Encrypt: The image pixels are shuffled or altered using a chaotic system.
   Decrypt: The original order of pixels is restored using the inverse chaotic system.

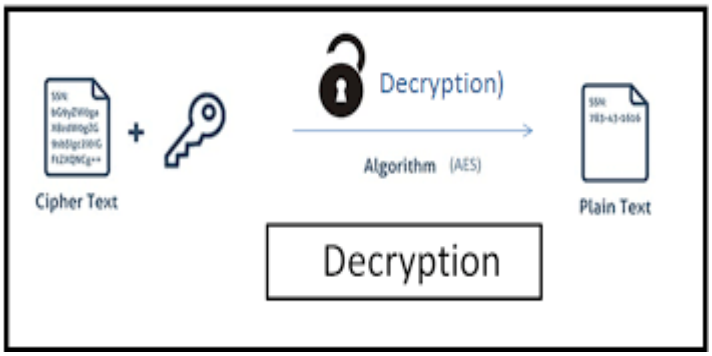
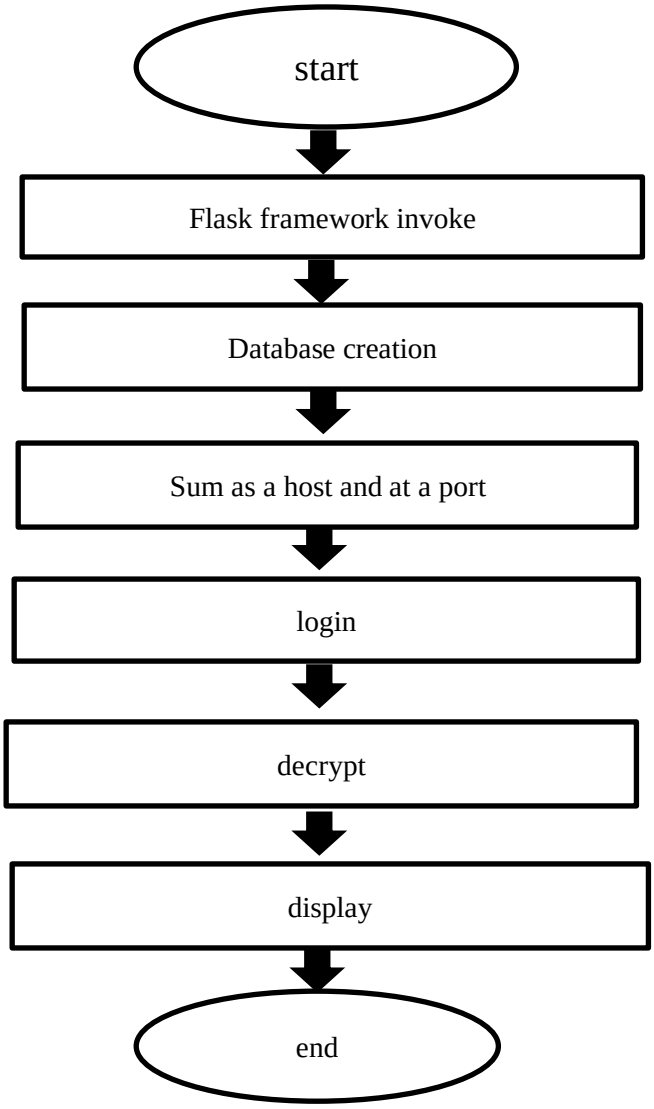
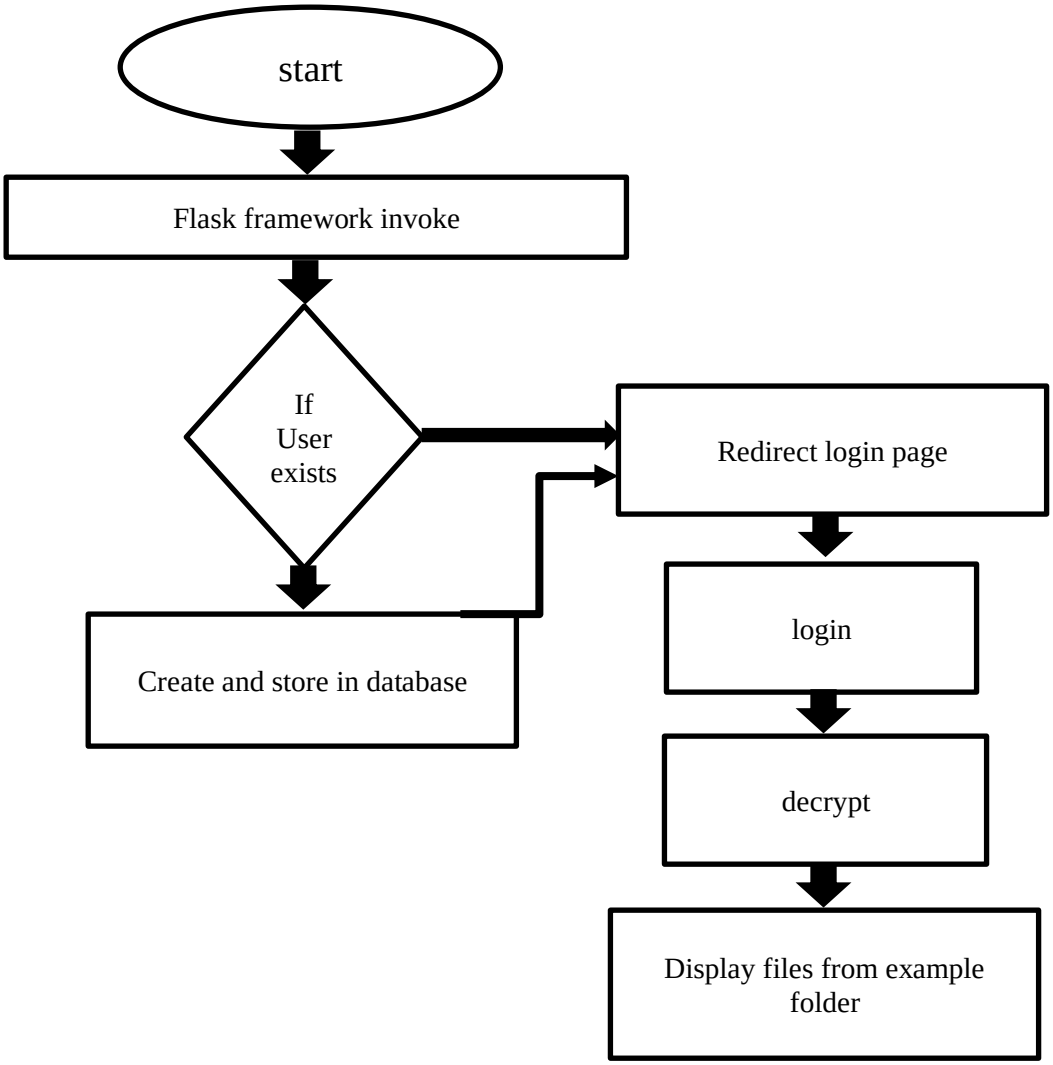


Figure 6.3-Decryption

BACKEND FLOWCHART



FRONTEND FLOWCHART



CHAPTER 7

RESULTS AND DISCUSSIONS

RESULTS

Cryptography and its principles have been studied carefully. I read and learned about Cryptography from various materials available on the internet. Encryption including why data encryption is necessary and types of encryption algorithms were studied.

Features and principles of Symmetric key algorithm were studied from various materials available. Image encryption and decryption techniques using Advanced Encryption Standard (AES) algorithm is proposed.

The usage of 256 bit cipher key to achieve the high security, because 256 bit cipher key is difficult to break. As a result of this secure transmission of image can be possible.

The goal of this research is to study the application of Advanced Encryption Standard algorithm (AES) for secure and efficient image encryption. The importance of image encryption by AES algorithm processes have been studied. It is also expected that AES algorithm study will have an effective role in strategic applications, because the encryption algorithm applied on hardware is also take a place in strategic communications equipment, it is safe and possible to develop this algorithm in terms of height and speed of time that have approved mainly on logistical aspects not on the technical aspects.

It is possible to encrypt and decrypt by AES encryption used in many highly sensitive applications like Image encryption. We have reason to believe that use this method to encrypt the image will have a very good prospect in the future.

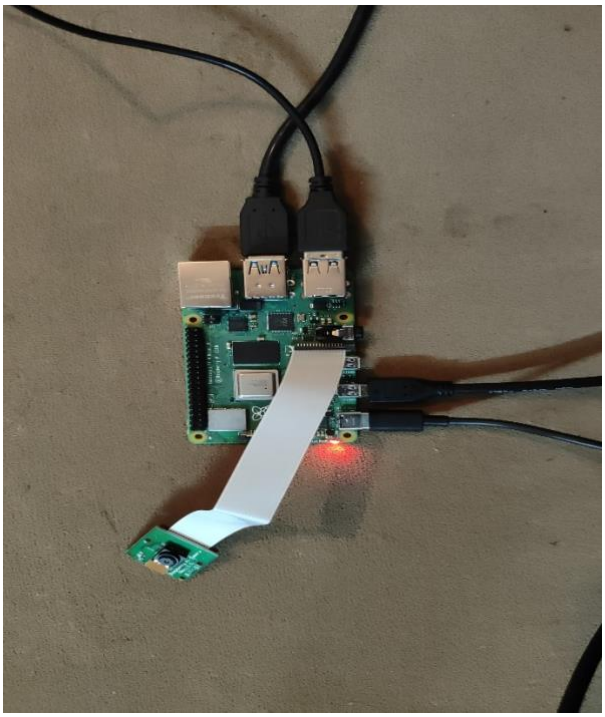


Figure 7.1-Raspberry pi with camera module

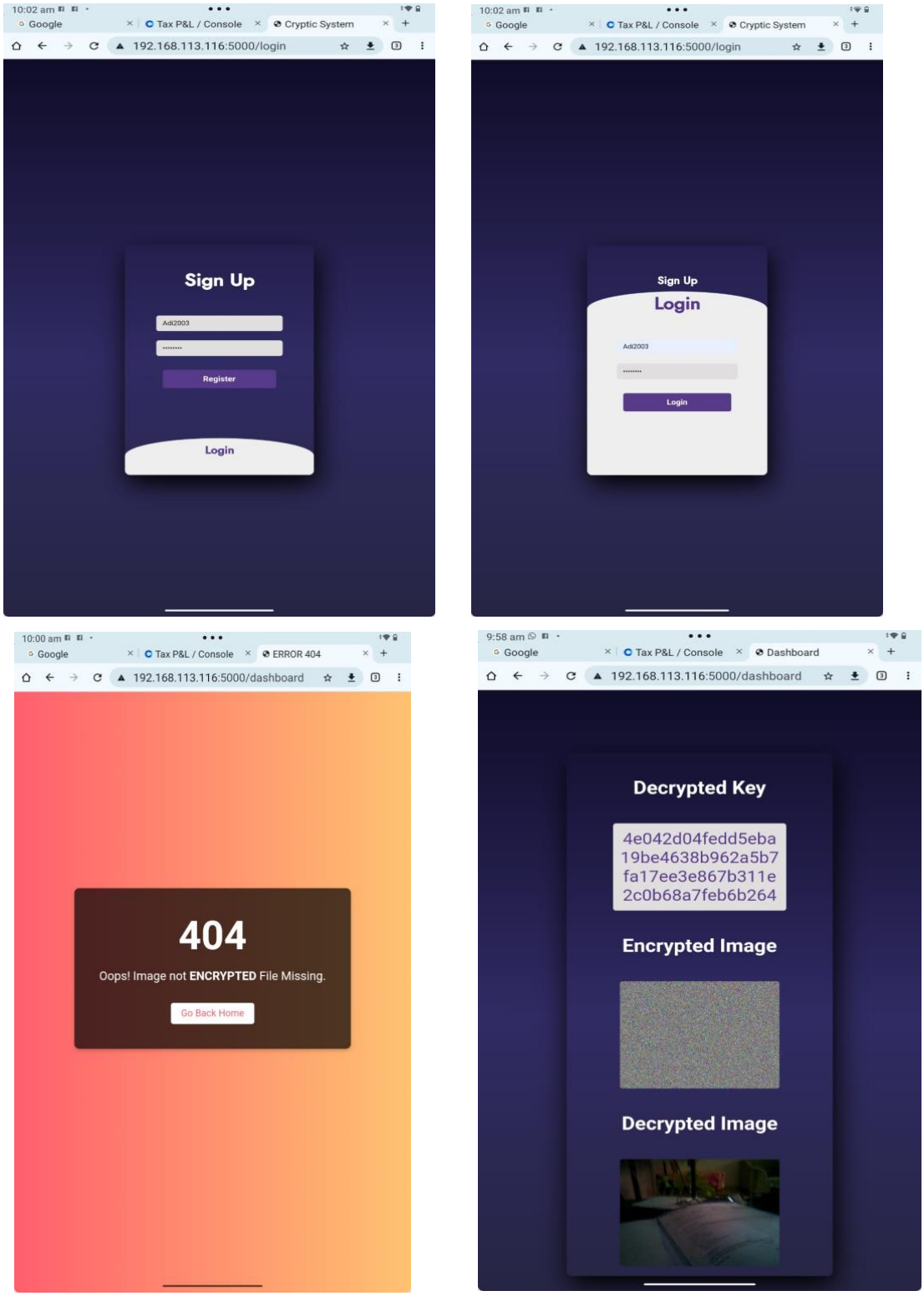


Figure 7.4-Final Output

CHAPTER 8

ADVANTAGES AND DISADVANTAGES

8.1 ADVANTAGES

- **Security**: Encryption ensures that the images are transmitted securely, minimizing the risk of unauthorized access or interception. This is crucial, especially when dealing with sensitive or confidential information.

- **Privacy**: Encrypted image transfer protects the privacy of individuals whose images are being transmitted. This is particularly important in applications such as healthcare, where patient confidentiality is paramount.

- **Data Integrity**: Encryption can also include mechanisms to ensure the integrity of the transmitted images. This prevents tampering or alteration of the images during transit, maintaining their reliability and trustworthiness.

- **Efficiency**: IoT-based systems can optimize the transfer process, ensuring that images are transmitted efficiently and reliably. This can be particularly beneficial in applications where real-time or near-real-time image transfer is required.

- **Cost-Effectiveness**: While there may be initial investment costs associated with implementing an IoT-based encrypted image transfer system, in the long run, it can be cost-effective. It can streamline processes, reduce manual intervention, and prevent data breaches or loss, leading to potential cost savings over time.

8.2 DISADVANTAGES

- **Compatibility Issues**: Ensuring compatibility between different IoT devices, protocols, and encryption standards can be challenging. Incompatibilities may arise when integrating devices from different manufacturers or when attempting to interface with legacy systems, leading to interoperability issues and potential data loss or corruption.

- **Performance Limitations**: Encrypting and decrypting images can introduce latency and overhead, impacting the performance and responsiveness of the system, especially in real-time applications. Balancing security requirements with performance considerations is crucial to maintaining an acceptable user experience.

- **Dependency on Internet Connectivity**: IoT-based systems rely heavily on internet connectivity for data transmission. Any disruptions or outages in internet connectivity can affect the system's reliability and availability, potentially leading to delays or data loss.

CHAPTER 9

APPLICATIONS

- **Smart Home Security Systems**

  - **Encrypted Video Feeds**: Cameras installed in smart homes capture and transmit video feeds. Encrypting these feeds ensures that only authorized users can view them, preventing unauthorized access and potential breaches.
  - **Secure Cloud Storage**: Images and videos from smart cameras are often stored in the cloud. Encrypted storage protects this data from being accessed by unauthorized parties.

- **Healthcare and Medical Imaging**

  - **Patient Privacy**: Medical IoT devices, such as connected imaging systems (e.g., MRI, X-ray machines), generate sensitive patient data. Encrypting these images ensures patient confidentiality and compliance with regulations like HIPAA.
  - **Telemedicine**: In telemedicine, encrypted transmission of medical images ensures that patient data remains secure during remote consultations and diagnoses.

- **Smart Cities and Surveillance**

  - **Public Safety**: Surveillance cameras deployed in smart cities collect vast amounts of video data. Encrypting these images helps protect citizen privacy and secures data from tampering.
  - **Traffic Management**: Cameras monitoring traffic conditions can use encrypted images to transmit data securely to traffic management systems, preventing data manipulation.

1. **Industrial IoT (IIoT)**

   - **Quality Control**: IoT devices in manufacturing use cameras for quality inspection. Encrypting these images ensures that sensitive manufacturing processes and proprietary information remain confidential.
   - **Equipment Monitoring**: Cameras monitoring industrial equipment can transmit encrypted images to central control systems, ensuring secure monitoring and analysis.

- **Agriculture**

  - **Precision Farming**: Drones and IoT sensors capture images of crops for analysis. Encrypting these images protects data integrity and ensures that competitive agricultural data is not exposed to unauthorized parties.
  - **Wildlife Monitoring**: Encrypted transmission of images from wildlife cameras prevents poaching activities by protecting the location and movement data of endangered species.

- **Retail and E-commerce**

  - **In-store Security**: Cameras in retail stores capture images to prevent theft and ensure security. Encrypting these images ensures customer and store data privacy.
  - **Virtual Try-ons**: IoT devices enabling virtual try-ons for clothes or accessories transmit customer images. Encryption ensures these personal images are securely handled.

- **Transportation and Logistics**

  - **Vehicle Monitoring**: Dashcams and onboard cameras in fleet management systems capture and transmit images. Encrypting these images helps in securely monitoring driver behavior and vehicle conditions.
  - **Cargo Security**: IoT-enabled cargo monitoring systems use cameras to check cargo integrity. Encrypted images ensure that sensitive shipment data remains secure.

- **Military and Defense**

  - **Surveillance Drones**: Military drones capture reconnaissance images. Encrypting these images ensures that sensitive information about troop movements and positions is secure.
  - **Border Security**: Cameras used in border security can transmit encrypted images to ensure that data regarding border crossings and incidents remains confidential.

- **Smart Infrastructure**

  - **Building Management**: Cameras monitoring smart buildings for security and operational efficiency transmit encrypted images, ensuring that building layouts and security measures are not compromised.

- **Energy Sector**: Cameras monitoring critical infrastructure like power plants can use encryption to securely transmit images, protecting against cyber threats.

➢ **Consumer Electronics**

- **Smartphones and Wearables**: Devices with cameras, like smartphones and smart glasses, can encrypt captured images to ensure user privacy and protect sensitive data.
- **Gaming and VR**: Virtual reality systems with cameras can encrypt images to protect user data and gaming environments from unauthorized access.

CHAPTER 10

CONCLUSION AND FUTURE SCOPE

CONCLUSION

This system provides security from intrusion attacks and the usage of AES technique allows the encryption and decryption process to be more secure and faster.

Thus this system provides security in storage and transmission of digital images. The cryptographic methodology proposed in this paper will further be tested on different types of input images with change in size of the image and keys of AES encryption algorithm.

The report shows the study in which a system could be used for effective image data encryption and key generation in diversified application areas, where sensitive and confidential data needs to be transmitted along with the image. The next step in this direction will be system implementation, and then analyzing it for its efficiency, accuracy and reliability. As a future work, I am going to continue this research in order generating more secure key to get the maximum encryption speed in limited implementation area. I will implement a novel mechanism in which AES algorithm will be apply to encrypt and decrypt images securely for further applications in image communication system.

FUTURE SCOPE

Future scope is, it can be used in various applications like mall remote controlled iot security,

Hospitals image transfer systems, Military communication, Forensics, Intelligent systems etc.

Definitely our present project has its flaws as mentioned such as the compatibility issues across various iot cameras, encryption standards, integration compatibility issues, personnel issues, performance limitations dues to the low ram and the other devices integrated may also be slow due to which the cost of a good system may be a bit high and the dependency on internet which limits itself to the internet connectivity and the use of the more refined and efficient encryption standards.

Once these drawbacks are solved the system will provide a very dependent solution to the easy access of highly secured files within a small business or for a whole system to operate on publicly.

REFERENCES

- P.Karthigaikumar, Soumiya Rasheed, Simulation of Image Encryption using AES Algorithm, IJCA Special Issue on Computational Science - New Dimensions & Perspectives NCCSE, 2011, 166-172
  .
- Shraddha Soni, Himani Agrawal, Dr. (Mrs.) Monisha Sharma, Analysis and Comparison between AES and DES Cryptographic Algorithm, International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 6, December 2012, 362-365.

- Irfan AbdulGani Landge, Implementation of AES Encryption and Decryption using VHDL, International J. of Engg. Research & Indu. Appls. (IJERIA). ISSN 0974-1518, Vol. 4, No. III (August 2011), 395-406.

- Priyanka Chauhan and Girish Chandra Thakur, “Efficient Way of Image Encryption Using Generalized Weighted Fractional Fourier Transform with Double Random Phase Encoding” International Journal of Advanced Research in Engineering & Technology (IJARET), Volume 5, Issue 6, 2014, pp. 45 - 52, ISSN 0976-6480, ISSN Online: 0976-6499.