

# **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Deemed to be University)**

Submitted in partial fulfilment of the requirements for the award of  
Bachelor of Engineering Degree  
in

Computer Science and Engineering

By

**ADITYA HARI (Reg. No. 37110018)**

**ANISH J (Reg. No. 37110040)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF COMPUTING**

## **SATHYABAMA**

Institute of Science and Technology  
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B

Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 6000 119

**MARCH 2021**



# **SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY

**(DEEMED TO BE UNIVERSITY)**

**Accredited with “A” grade by NAAC I 12B Status by UGC I  
Approved by AICTE**

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

**[www.sathyabama.ac.in](http://www.sathyabama.ac.in)**

---

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bona fide work of Mr. \_\_\_\_\_  
**Aditya Hari (37110018) and Anish J (37110040)** who carried out the project entitled "**Video**  
**Surveillance using a smartphone**" under my supervision from November 2020 to March 2021.

Internal Guide  
DR.S.MURUGAN, M.E.,[Ph.D.,]

**Head of the Department**

---

Submitted for Viva voce Examination held on \_\_\_\_\_

**Internal Examiner**

**External Examiner**

## **DECLARATION**

We, **ADITYA HARI (Reg. No. 37110018)**, **ANISH J (Reg. No. 37110040)** hereby declare that the Project Report entitled “**Video Surveillance using a smartphone**” done by me under the guidance of **Dr. S. MURUGAN, M.E., [Ph.D.,]** at Sathyabama Institute of Science and Technology is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

**DATE:**

**PLACE:**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. SASIKALA. T, M.E., Ph.D.**, Dean, School of Computing and **Dr. L. Lakshmanan, M.E., Ph.D.**, and **S. VIGNESHWARI, M.E., Ph.D.**, Head of the Department, Dept. of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr. S. MURUGAN, M.E., [Ph.D.]** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

## **ABSTRACT**

Today's fast-growing world has adapted itself to the convenience of the smartphone and has enabled everyone to do virtually any task at the tip of their finger. This growth has influenced a change in the way day to day activities are undertaken. Security and surveillance have also been improved and enhanced over the years. This project aims to create an android application that provides a cheaper/simpler alternative to the traditional CCTV setup that is used today. This is achieved by using a smartphone as the surveillance device.

## TABLE OF CONTENTS

| Chapter No. | Title                                  | Page No. |
|-------------|----------------------------------------|----------|
|             | <b>ABSTRACT</b>                        | <b>i</b> |
|             | <b>LIST OF FIGURES</b>                 |          |
| <b>1</b>    | <b>INTRODUCTION</b>                    | <b>1</b> |
|             | 1.1 OUTLINE OF PROJECT                 | 1        |
|             | 1.2 LITERATURE REVIEW                  | 1        |
|             | 1.3 PROBLEM STATEMENT                  | 2        |
| <b>2</b>    | <b>AIM AND SCOPE</b>                   |          |
|             | 2.1 AIM OF PROJECT                     | 3        |
|             | 2.2 SCOPE OF PROJECT                   | 3        |
|             | 2.3 ADVANTAGES                         | 3        |
|             | 2.4 FLOW DIAGRAM                       | 4        |
| <b>3</b>    | <b>SYSTEM DESIGN &amp; METHODOLOGY</b> |          |
|             | 3.1 EXISTING SYSTEMS                   | 5        |
|             | 3.2 SYSTEM IMPLEMENTATION              | 5        |
|             | 3.3 SYSTEM FUNCTIONALITIES             | 8        |

|          |                                         |           |
|----------|-----------------------------------------|-----------|
|          | 3.4 SYSTEM REQUIREMENTS                 | 9         |
|          | 3.5 MODULES                             | 10        |
| <b>4</b> | <b>SOFTWARE DEVELOPMENT METHODOLOGY</b> |           |
|          | 4.1 ARCHITECTURAL DESIGN                | 11        |
|          | 4.2 DESCRIPTION OF DIAGRAM              | 12        |
|          | 4.3 ACTORS                              | 13        |
|          | 4.4 USE CASES                           | 13        |
| <b>5</b> | <b>RESULTS AND DISCUSSIONS</b>          |           |
|          | 5.1 RESULT                              | 14        |
|          | 5.2 OUTPUT & SCREENSHOTS                | 15        |
| <b>6</b> | <b>CONCLUSION AND FUTURE WORK</b>       |           |
|          | 6.1 CONCLUSION                          | 19        |
|          | 6.2 FUTURE WORK                         | 19        |
|          | <b>REFERENCES</b>                       | <b>20</b> |
|          | <b>APPENDIX</b>                         | <b>28</b> |

## **LIST OF FIGURES**

Figure 2.1 Flow Diagram

Figure 4.1 Architecture Diagram

Figure 5.1 UI

Figure 5.2 Generation of Public & Private Keys

Figure 5.3 Generation of Public & Private Keys

Figure 5.4 Encryption Graph

Figure 5.5 Decryption Graph

# **CHAPTER 1**

## **INTRODUCTION**

Android is a software stack for mobile devices that an operating system, middleware and key applications. Android is a software platform and operating system for mobile devices based on the Linux operating system and developed by Google and the Open Handset Alliance. It allows developers to write managed code in java- like language that utilizes Google-developed java libraries but does not support programs developed in native code.

This app aims to provide a simple and cheaper alternative to establishing a secure surveillance system which can be easily set up by using a smartphone. A connection is set up between two phones

### **1.1 OUTLINE OF THE PROJECT**

The main goal of this application is to be able to provide a simple user – friendly means of setting up a home surveillance system. The application sets up a connection between two devices wherein one device is a sender and one is a receiver.

It uses:

- RSA Encryption Algorithm
- Agora API

### **1.2 LITERATURE REVIEW**

S. A. Ahirrao et al [1] discussed about how we can use a phone camera and send feed to a PC, through peer-to-peer connection. This enabled a surveillance system using the phone camera. This approach could be used for a peer-to-peer network, not for remote surveillance which restricts its use while the user is out of the network.

Aman Chada et al [2] speaks about the use of RSA to perform the task of encryption. Using RSA, we can secure and encrypt the video frames. This approach

enables an improved and enhanced security system. It proposes the use of pseudo noise while splitting of the video into R, G, B frames.

Aswin Vishal Kumar et al [3] tells about how conventional methods can require human intervention and restrict also easy and convenient access. This project stresses on the latter part of how the application can be easy and enable portability in turn improving the convenience.

The objective behind making this application was to provide a simple and easy way to set up a surveillance system in any place they choose to, be it a room in their house or an establishment they own. So, while surveying as to on which platform or rather operating system the project must be implemented, we selected android for the following reasons:

- a. Android is an open-source platform
- b. Supports multifunction
- c. Provides rich tools to make interactive application

### **1.3 PROBLEM STATEMENT**

Surveillance is a part of almost every aspect of our daily lives. Every retail store in the world possesses a security surveillance system to monitor the comings and goings of the store. The above tech is usually done with the help of separate cameras and also limits the use cases and portability. What if we need to have it for a specific time frame? What if the setup requires to be moved from one place? That could open up ideas and various ways of implementing which in turn makes our lives easier.

This app looks to use an extra smartphone to act as the camera used to perform the surveillance process. This will solve the above-mentioned issues like temporary need and portability. The application could be niche, but the use cases are unthinkable based on each and every users need.

## **CHAPTER 2**

### **AIM AND SCOPE**

#### **2.1 AIM OF PROJECT**

The aim of this project is to use an android smartphone as a means of setting up a simple, user-friendly surveillance system

#### **2.2 SCOPE OF THE PROJECT**

Surveillance is a part of almost every aspect of our daily lives. Every retail store in the world possesses a security surveillance system to monitor the comings and goings of the store. The above tech is usually done with the help of separate cameras and also limits the use cases and portability. What if we need to have it for a specific time frame? What if the setup requires to be moved from one place? That could open up ideas and various ways of implementing which in turn makes our lives easier.

This app looks to use an extra smartphone to act as the camera used to perform the surveillance process. This will solve the above-mentioned issues like temporary need and portability. The application could be niche, but the use cases are unthinkable based on each and every user need.

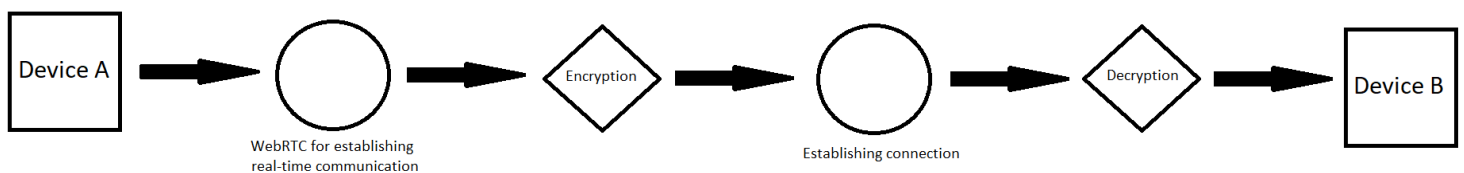
#### **2.3 ADVANTAGES**

- i. User Friendly
- ii. Cost Effective
- iii. Portable setup
- iv. Improved Security System
- v. Monitoring and surveillance from remote locations

## 2.4 FLOW DIAGRAM

Flow chart is a diagrammatic representation of an algorithm and essential part of planning the system. Flow charts are widely used in technical analysis and programming for easy writing programs and explaining them to others. So, one of the most popular type of flow charts is Technical Flow Chart. Technical Flow Chart can be drawn by pencil on the paper, but it will be easier to use for designing a special software. A Flowchart is a graphical representation of process, algorithm, workflow or step-by-step solution of the problem. It shows the steps as boxes of various kinds and connects them by arrows in a defined order depicting a flow.

The main reason of using Process Flowchart or PFD is to show relations between major parts of the system. Process Flowcharts for single unit or multiple units differ in their structure and implementations are shown in **Fig. 2.1**



**Fig. 2.1 Flow Diagram**

## **CHAPTER 3**

### **SYSTEM DESIGN & METHODOLOGY**

#### **3.1 EXISTING SYSTEMS**

The old technique of monitoring or surveillance uses IP camera which are bulky and fixed to one spot. These devices are permanent or intended to be for long term use. The proposed system enables the users to have the device portable to any location without having to allocate a particular space, unlike the IP camera.

The few other existing systems of this portable nature does not allow users to connect to the devices that are outside of the network. This is solely due to the fact that the existing system uses P2P connections where in the devices must be connected to the same network. The connection is established based on the network. This makes it difficult for the user to move away more than the distance allowed by the wireless local network. There are few other existing systems, like one being through wired medium which again limits the distance and increases hassle to use the device.

The proposed system enables the user to connect the device remotely through the internet as this is enabled by the Real time communication API. This will it less worrying to operate, enhance the portability and at the same time allows us to use it for our own personal needs which in turn increases the number of use cases. An illustration that satisfies a use case with the proposed system is for toddler or infant parents who can monitor their children who will be sleeping away from the parent who can monitor their movements and act accordingly without the need for a permanent CCTV in the room.

The above-mentioned portable systems also lack in the security aspects. In the system of P2P networks, anyone in the network can access the feed, because it is unprotected and not encrypted. To overcome the issue of security we have implemented the RSA 2048 encryption

The RSA 2048 is a strong asymmetric encryption algorithm. It works on the principle of using two different keys in order to perform the action of encryption and decryption. The public key as in the name, is public and is used to encrypt the data being sent. The private is unique and varies from person to person/ device to device. It is used for decrypting the data that is received.

#### **An example of asymmetric cryptography:**

- A client (for example browser) sends its public key to the server and requests for some data.
- The server encrypts the data using client's public key and sends the encrypted data.
- Client receives this data and decrypts it.

The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers and private key is also derived from the same two prime numbers. So, if somebody can factorize the large number, the private key is compromised. Therefore, encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024-bit keys could be broken in the near future.

### **3.2 SYSTEM IMPLEMENTATION**

Agora is a platform-as-a-service (PaaS) company. They provide API for real time communication on various modes like text, audio, video, live streaming services etc. The API used in the project is one-on-one video chat API for android.

#### **How to implement the Agora System:**

- Set up the Application ID and add the necessary SDK's
- UI consists of simple/basic elements such as Mute/Unmute, Switch Camera and disconnect
- Ensure permissions for the following are enabled in order to establish connections:

- i. Phone State
- ii. Internet
- iii. Camera
- iv. Audio
- v. Modify Audio Settings
- vi. Bluetooth
- vii. Access WIFI State

- Check if the user has given permission for the above mentioned using the following functions: onRequestPermissionsResult() and onCreate()
- Initialize the Agora engine and connect the video from the local feed to the receiver feed in order to establish connection.
- Set up the event handler using: RtcEngineEventHandler
- The channel is now set up and can be joined with another device.

### Key generation in RSA:

#### A. Public Key:

- i. Select two prime no's. Suppose **P = 53 and Q = 59**. Now First part of the Public key: **n = P\*Q = 3127**.
- ii. We also need a small exponent say **e**:
  - But e Must be an integer and not be a factor of n.
- iii. **1 < e <  $\Phi(n)$**
- iv. Let us now consider it to be equal to 3.
- v. Our Public Key is made of n and e

## **B. Private Key:**

i. We need to calculate  $\Phi(n)$  :

Such that  $\Phi(n) = (P-1)(Q-1)$  so,  $\Phi(n) = 3016$

ii. Now calculate Private Key,  $d$ :

$d = (k \cdot \Phi(n) + 1) / e$  for some integer  $k$

For  $k = 2$ , value of  $d$  is 2011.

### **3.3 SYSTEM FUNCTIONALITIES**

The proposed system is devised using java, XML and Agora API. The java connects the API and the front-end basic UI elements of the XML to the app. The API key will be used to link the application to the API. This can be tested using a temporary app token which enables to test the project.

The proposed system has the following functions:

- Can connect to any remote devices
- Can mute/unmute the audio
- Can switch the camera from front to rear.

The user can connect to the other device using the internet through various channels. After the user has joined the channel, he/she can monitor on what is happening on the other side either using the front or rear camera. Keeping in mind the security of this application we have implemented RSA encryption

The RSA algorithm that is used in the system provides an improved, updated and enhanced security. The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers and private key is also derived from the same two prime numbers.

The proposed system can do a lot of cost cutting and at the same time be beneficial to the internet by cutting down on e-waste. Reduction of e-waste can benefit in the betterment of the environment.

### 3.4 REQUIREMENT SPECIFICATION

Specifications required to run the system:

| HARDWARE                  | SPECIFICATIONS          |
|---------------------------|-------------------------|
| OPERATING SYSTEM          | WINDOWS 7/8/10          |
| RAM(RANDOM ACCESS MEMORY) | 4.00 GB(3.88 GB USABLE) |
| HARD DISK SPACE           | 2 GB                    |
| TOOLS REQUIRED            | ANDROID STUDIO & AVD    |
| DEBUGGING                 | USB CABLES              |
| PHONE                     | ANDROID PHONE           |

| SOFTWARE         | SPECIFICATIONS                                                                                      |
|------------------|-----------------------------------------------------------------------------------------------------|
| SOFTWARE         | JAVA DEVELOPMENT KIT (JDK)<br>SOFTWARE DEVELOPMENT KIT (SDK)<br>REAL TIME COMMUNICATION API (AGORA) |
| ANDROID SOFTWARE | ANDROID STUDIO                                                                                      |

### 3.5 MODULES

This project contains two modules; i.e. The User Module and Security Module. The functionality of both modules are explained below

1) User Module:

- Consists of a device (Sender/Receiver Phone)
- Encryption on Sender Phone
- Video footage is transmitted
- Decryption on Receiver Phone
- Video is then streamed

2) Security Module:

The security module consists of two sections; Encryption and Decryption, which are explained below

i) Encryption:

In this section of the security module, the encryption takes place.

Encryption takes place on the phone that acts as a sender phone

This process of encryption is done using the public key. The public key is generated through a series of arithmetic calculations which start by using the dimensions of the video frame in the generation of the key value

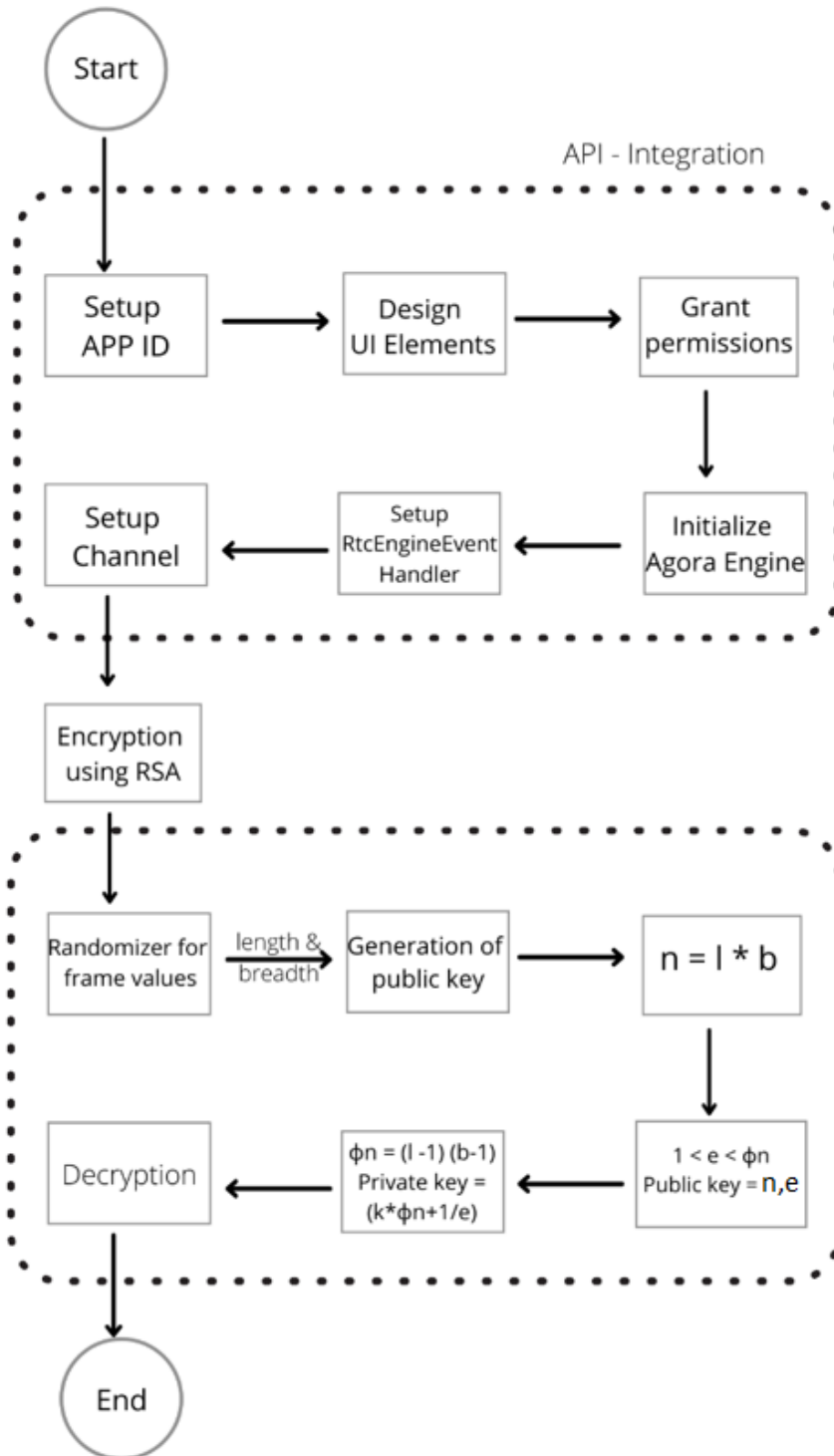
ii) Decryption:

This is the final step in the security module. Decryption takes place on the phone that acts as the receiver phone. The process of decryption is done using the private key. The variables involved in the generation of the public key are also used for generation the private key. The private key is unique for each user

## CHAPTER 4

### SOFTWARE DEVELOPMENT METHODOLOGY

#### 4.1 ARCHITECTURAL DESIGN



**Fig 4.1 Architecture Diagram**

## 4.2 DESCRIPTION OF DIAGRAM

Fig 4.1 shows the functioning of the system explaining every aspect of the application. The first block depicts the working of the Agora API. Agora API is used to establish a real time communication between the sender device and the receiver device. This will make it less difficult to operate, enhance the portability and at the same time allows us to use it for our own personal needs which in turn increases the number of use cases. An illustration that satisfies a use case with the proposed system is for toddler or infant parents who can monitor their children who will be sleeping away from the parent who can monitor their movements and act accordingly without the need for a permanent CCTV in the room.

The first step is to set up agora and link the application ID and the API. Next step involves the design of the UI. This will involve having an android application with basic functionalities such as call (to establish connection), swap between primary and secondary camera. This is achieved with interactable buttons on the bottom of the screen (**Fig 5.1**). The necessary permissions such as microphone, camera, etc. must be given in order for the footage to be streamed

The Agora engine is then initialized and the following handler is setup “RtcEngineEvent”. This engine sets up the Real Time Communication (RTC) between the sender and the receiver devices. The channel is then completely set up and the encryption is done.

Encryption is achieved by using RSA Encryption Algorithm. RSA is an asymmetric encryption algorithm. This means it consists of two different keys; A Public Key and A Private Key. The public key is used for encryption of the file and the private key is used for the decryption of the file. The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers and private key is also derived from the same two prime numbers. So, if somebody can factorize the large number, the private key is compromised. Therefore, encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially

### **4.3 ACTORS**

The actors of this designed system are people who are looking to set up a simple surveillance system. The user on either side can connect to the network for the live feed and surveillance video.

### **4.4 USE CASES**

This project shows a wide set of test cases. These include, Surveillance System. The designed system can work efficiently as a cost effective, user- friendly surveillance system.

This project can also be used as a video calling application or even be attached to a computer system/Laptop as an improved camera to the native system camera.

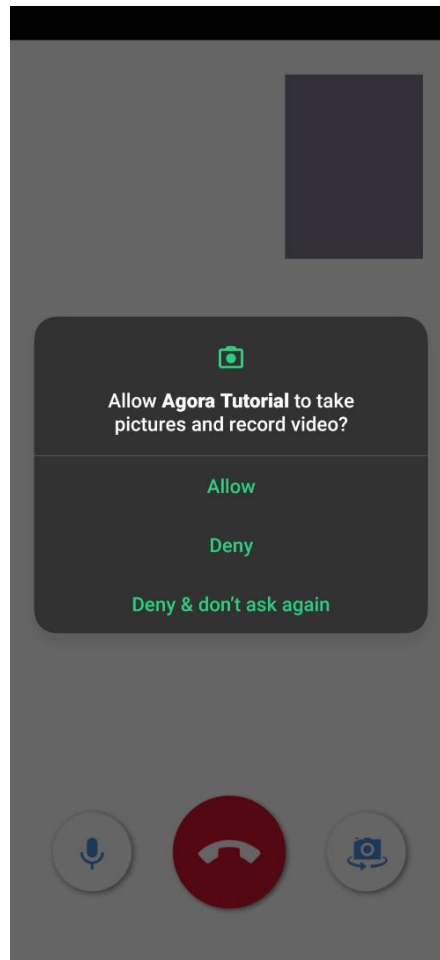
## **CHAPTER 5**

### **RESULTS AND DISCUSSION**

#### **5.1 RESULT**

- The user logs into the application. Then the desired permissions are given and it then connects to the other device to enable the streaming
- RSA Encryption Algorithm uses two keys, the public key and the private key.
- These keys are used in the process of security which are encryption and decryption to provide security to the system.

## 5.2 OUTPUTS & SCREENSHOTS



**Fig 5.1 UI**

**Fig 5.1** shows the UI of the android application which has the buttons on the bottom of the screen to Mute/Unmute the stream, Swap between primary and secondary camera and Cut the stream. It also shows the application asking permissions in order for the stream to take place.

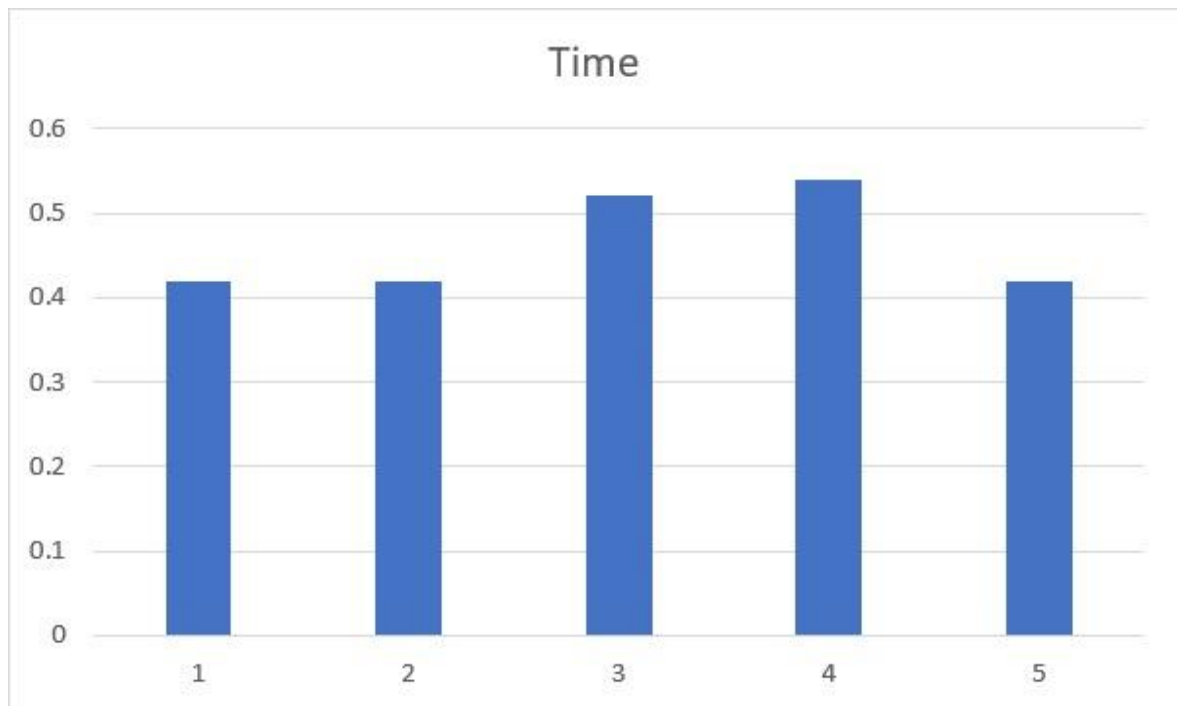
```
Public Key:  
N = 1887516746320483621 E = 3  
  
Private Key:  
D = 1258344495715163400  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

**Fig 5.2 Generation of Public & Private Keys**

```
Public Key:  
N = 1788633124200636949 E = 5  
  
Private Key:  
D = 715453248610336465  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

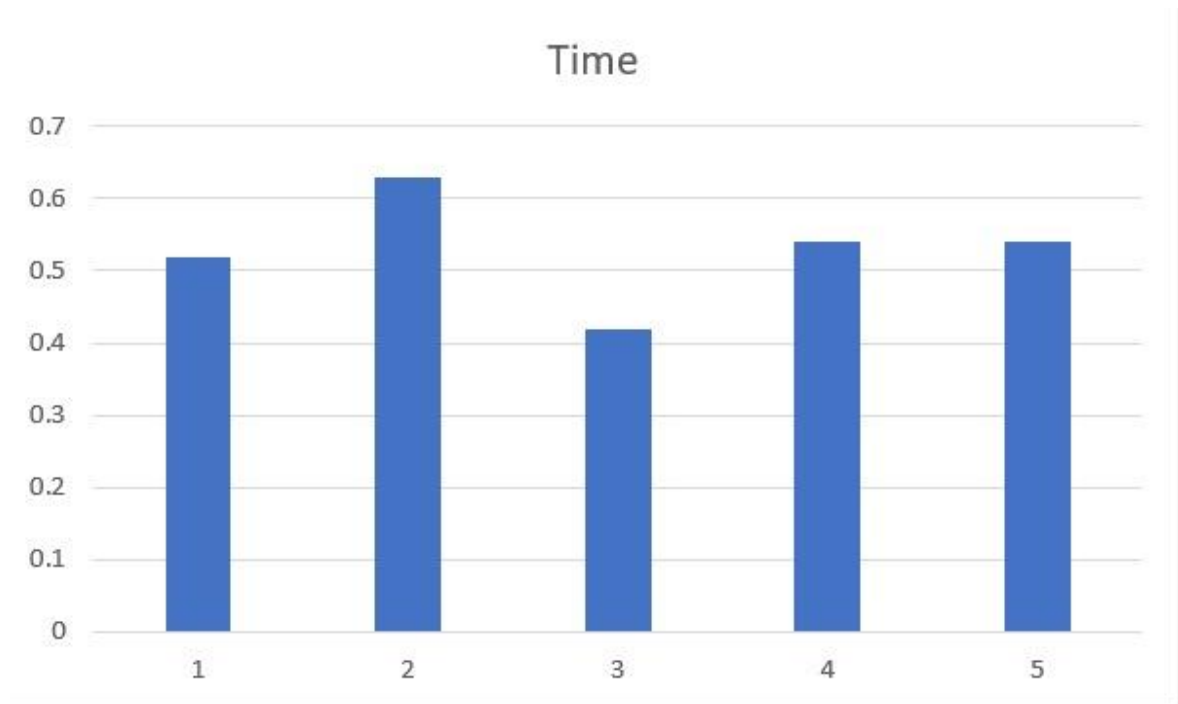
**Fig 5.3 Generation of Public & Private Keys**

**Fig 5.2 & Fig 5.3** shows the output of the public key and private key being generated for the encryption and decryption. The outputs show different keys generated as it contains a randomizer function to ensure different keys are generated each time.



***Fig 5.4 Encryption Graph***

**Fig 5.4** shows the graph for the time taken for the generation of the public key used for the encryption process.



***Fig 5.5 Decryption Graph***

**Fig 5.5** shows the graph for the time taken for the generation of the private key used for the decryption process.

## **CHAPTER 6**

### **CONCLUSION AND FUTURE WORK**

#### **6.1 CONCLUSION**

The system has the user module will gives the ability to monitor, mute and unmute and switch between cameras which in turn enables better and extra views for surveillance

This system will make the surveillance to another dimension by allowing enhanced security and portability to the existing systems. The user does not have to worry about anyone else breaching into their private space as this is encrypted with RSA encryption. RSA provides a strong and hard to break encryption system

#### **6.2 FUTURE WORK**

With respect to the future work of this project, adding of auto detection of people and/or alarm system based on our needs can be set. The UI and the experience of the user can be enhanced by a more intuitive and informative user interface.

Though the app is secure, ensuring more testing might help us detect loop holes or zero-day vulnerabilities and improve the security to the highest level.

We can also bring in a device that can provide movement or a tripod stand which could enable more use cases at almost no cost, as the devices used are mostly old and/or unused smartphones

## REFERENCES

- [1] S. A. Ahirrao, Sayali S. Ballal, Divya K. Sawant, Harsha H. Chahira, Amisha A. Savaliya “Android Based Remote Surveillance System andContent Sharing Between PC And Mobile” (2015)
- [2] Aman Chadha, Sushmit Mallik, Ankit Chadha, Ravdeep Johar, M. Mani Roja “Dual-Layer Video Encryption using RSA Algorithm” (2015)
- [3] Aswin Vishal Kumar, Kiran, Sandeep, Santhiya “Intelligent Surveillance Robot for Monitoring Outdoor Security ” (2020)
- [4] [What is RSA encryption and how does it work? | Comparitech](#)
- [5] <https://developer.android.com/training/connect-devices-wirelessly/wifi-direct>
- [6] <http://cagneymoreau.com/stream-video-android/>
- [7] [www.agora.io](http://www.agora.io)
- [8] <https://www.geeksforgeeks.org/rsa-algorithm-cryptography/>
- [9] <https://docuserve.com/blog-post/video-encryption-how-to-secure-your-digital-content/>

## RESULTS



Completed: 100% Checked



Plagiarism



Unique



Sentence Wise Result



Document View



Matched Sources

|        |                                                                                                              |
|--------|--------------------------------------------------------------------------------------------------------------|
| Unique | UG Student, UG Student, Professor, Department of Computer Science & Engineering                              |
| Unique | Today's fast-growing world has adapted itself to the convenience of the smartphone and has enabled ...       |
| Unique | to do virtually any task at the tip of their finger.                                                         |
| Unique | This growth has influenced a change in the way day to day                                                    |
| Unique | Security and surveillance have also been improved and enhanced over the years. This                          |
| Unique | project aims to create an android application that provides a cheaper/simpler alternative to the traditio... |

# VIDEO SURVEILLANCE USING A SMARTPHONE

Aditya Hari<sup>1\*</sup>, Anish J<sup>2\*</sup>, Dr. S. Murugan<sup>3\*</sup>

<sup>1\*</sup>UG Student, <sup>2\*</sup>UG Student, <sup>3\*</sup>Professor, Department of Computer Science & Engineering

Sathyabama Institute of Science & Technology

---

## ABSTRACT:

Today's fast-growing world has adapted itself to the convenience of the smartphone and has enabled everyone to do virtually any task at the tip of their finger. This growth has influenced a change in the way day to day activities are undertaken. Security and surveillance have also been improved and enhanced over the years. This project aims to create an android application that provides a cheaper/simpler alternative to the traditional CCTV setup that is used today. This is achieved by using a smartphone as the surveillance device.

**Key Words:** Surveillance, Android Application, RSA Encryption Algorithm, Agora API

## 1. INTRODUCTION

Surveillance is a part of almost every aspect of our daily lives. Every retail store in the world possesses a security surveillance system to monitor the comings and goings of the store. This app aims to provide a simple and cheaper alternative to establishing a secure surveillance system which can be easily set up by using a smartphone. A connection is set up between two phones to emulate the process of surveillance.

user – friendly means of setting up a home surveillance system. The application sets up a connection between two devices wherein one device is a sender and one is a receiver.

## b. OBJECTIVE

Surveillance is a part of almost every aspect of our daily lives. Every retail store in the world possesses a security surveillance system to monitor the comings and goings of the store. The above tech is usually done with the help of separate cameras and also limits the use cases and portability. What if we need to have it for a specific time frame? What if the setup requires to be moved from one place? That could open up ideas and various

## 2. AIM & OBJECTIVE

### a. AIM

The main goal of this application is to be able to provide a simple, feasible,

ways of implementing which in turn makes our lives easier.

This app looks to use an extra smartphone to act as the camera used to perform the surveillance process. This will solve the above-mentioned issues like temporary need and portability. The application could be niche, but the use cases are unthinkable based on each and every user's need.

### **3. LITERATURE SURVEY**

S. A. Ahirrao et al [1] discussed about how we can use a phone camera and send feed to a PC, through peer-to-peer connection. This enabled a surveillance system using the phone camera. This approach could be used for a peer-to-peer network, not for remote surveillance which restricts its use while the user is out of the network.

Aman Chada et al [2] speaks about the use of RSA to perform the task of encryption. Using RSA, we can secure and encrypt the video frames. This approach enables an improved and enhanced security system. It proposes the use of pseudo noise while splitting of the video into R, G, B frames.

Aswin Vishal Kumar et al [3] tells about how conventional methods can require human intervention and restrict also easy and convenient access. This project stresses on the latter part of how the application can be easy and enable portability in turn improving the convenience.

## **4. SOFTWARE & METHODOLOGY**

### **a. EXISTING SYSTEMS**

The old technique of monitoring or surveillance uses IP camera which are bulky and fixed to one spot.

These devices are permanent or intended to be for long term use. The proposed system enables the users to have the device portable to any location without having to allocate a particular space, unlike the IP camera.

The few other existing systems of this portable nature does not allow users to connect to the devices that are outside of the network. This is solely due to the fact that the existing system uses Peer to Peer connections where in the devices must be connected to the same network.

The connection is established based on the network. This makes it difficult for the user to move away more than the distance allowed by the wireless local network. There are few other existing systems, like one being through wired medium which again limits the distance and increases hassle to use the device.

allows us to use it for our own personal needs which in turn increases the number of use cases. An illustration that satisfies a use case with the proposed system is for toddler or infant parents who can monitor their children who will be sleeping away from the parent who can monitor their movements and act accordingly without the need for a permanent CCTV in the room.

## b. PROPOSED SYSTEMS

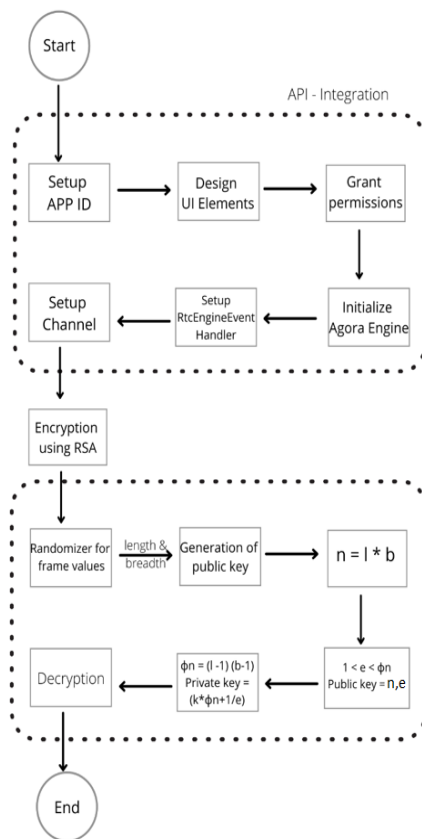


Figure 1: Proposed New System

The proposed system enables the user to connect the device remotely through the internet as this is enabled by the Real Time Communication API. This will make it less difficult to operate, enhance the portability and at the same time

The above-mentioned portable systems also lack in the security aspects. In the system of Peer-to-Peer networks, anyone in the network can access the feed, because it is unprotected and not encrypted. To overcome the issue of security we have implemented the RSA 2048 encryption

The RSA 2048 is a strong asymmetric encryption algorithm. The idea of using the RSA algorithm is that finding the factor of a very large number and this improves the overall security of the system designed. The public key as in the name, is public and is used to encrypt the data being sent. The private is unique and varies from person to person/ device to device. It is used for decrypting the data that is received.

From the above diagram, the RSA algorithm uses the dimensions of the frame of the video in order to calculate and generate both keys. In

order to remove the repeatedly occurring frame values and ensure that each frame has a unique pair of keys, a random number is generated and added to the frame value to slightly alter them.

## 5. AGORA

Agora is a platform-as-a-service (PaaS) company. They provide API for real time communication on various modes like text, audio, video, live streaming services etc. The API used in the project is one-on-one video chat API for android.

### a. How to implement the Agora System:

- Set up the Application ID and add the necessary SDK's
- UI consists of simple/basic elements such as Mute/Unmute, Switch Camera and disconnect
- Ensure permissions for the following are enabled in order to establish connections:
  - viii. Phone State
  - ix. Internet
  - x. Camera
  - xi. Audio
  - xii. Modify Audio Settings
  - xiii. Bluetooth
  - xiv. Access WIFI State
- Check if the user has given permission for the above mentioned using the following functions:

onRequestPermissionsResult() and onCreate()

- Initialize the Agora engine and connect the video from the local feed to the receiver feed in order to establish connection.
- Set up the event handler using: RtcEngineEventHandler
- The channel is now set up and can be joined with another device.

## 6. RSA

RSA (Rivest – Shamir – Adleman) is an asymmetric encryption cryptography algorithm. Asymmetric suggests that the encryption is done with the help of two different keys i.e., Public Key and Private Key. The public key is used for the encryption and is available to everyone, whereas the private key is unique for each user and decryption cannot be done without it. This algorithm works on the principle that factorizing a very large composite number is difficult and hence offers a high level of security.

### a. ALGORITHM

The RSA algorithm works on the principle that a very large composite number is difficult and an extremely time taking process to do so. It revolves around two random values undergoing various arithmetic operations in the generation of the public key and the private key. The public key consists of two numbers

where one number is multiplication of two large prime numbers and private key is also derived from the same two prime numbers.

The following is the algorithm for the generation of keys using RSA:

- i. Initialize two variables p & q which hold two random prime numbers
- ii. Calculate  $n = p \cdot q$
- iii. Initialize a small exponent e, such that e is not a factor of n i.e., e & n are co-primes
- iv. The public key consists of n & e
- v. Initialize k as any arbitrary constant
- vi. Calculate  $O(n) = (p-1)(q-1)$
- vii. Calculate  $d = (k \cdot O(n) + 1)/e$
- viii. The private key consists of d

```
Public Key:
N = 1887516746320483621 E = 3

Private Key:
D = 1258344495715163400

...Program finished with exit code 0
Press ENTER to exit console.
```

Figure 2: Generation of Keys

```
Public Key:
N = 1788633124200636949 E = 5

Private Key:
D = 715453248610336465

...Program finished with exit code 0
Press ENTER to exit console.
```

Figure 3: Generation of Public & Private Key  
(different from "Figure 2" to show random function)

## 7. RESULT & CONCLUSION

The system has the user module will gives the ability to monitor, mute and unmute and switch between cameras which in turn enables better and extra views for surveillance

This system will make the surveillance to another dimension by allowing enhanced security and portability to the existing systems. The user does not have to worry about anyone else breaching into their private space as this is encrypted with RSA encryption. RSA provides a strong and hard to break encryption system

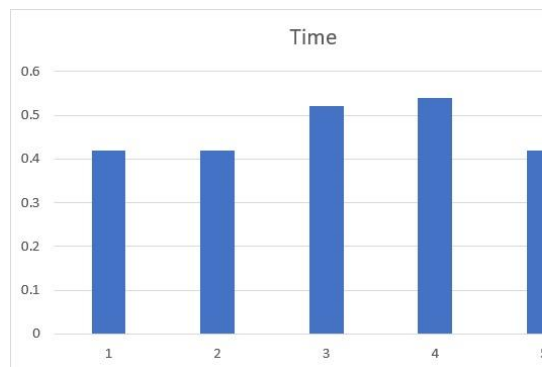


Figure 3: Graph showing time taken to perform/generate the public key for encryption

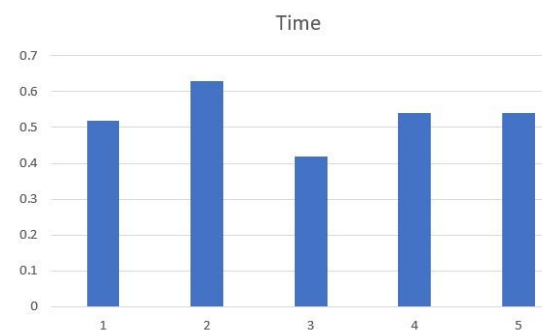


Figure 4: Graph showing time taken to perform/generate private key for decryption

## 8. FUTURE WORK

With respect to the future work of this project, adding of auto detection of people and/or alarm system based on our needs can be set. The UI and the experience of the user can be enhanced

by a more intuitive and informative user interface. Though the app is secure, ensuring more testing might help us detect loop holes or zero-day vulnerabilities and improve the security to the highest level.

We can also bring in a device that can provide movement or a tripod stand which could enable more use cases at almost no cost, as the devices used are mostly old and/or unused smartphones

## REFERENCES

- [1] S. A. Ahirrao, Sayali S. Ballal, Divya K. Sawant, Harsha H. Chahira, Amisha A. Savaliya "Android Based Remote Surveillance System and Content Sharing Between PC and Mobile" (2015)
- [2] Aman Chadha, Sushmit Mallik, Ankit Chadha, Ravdeep Johar, M. Mani Roja "Dual-Layer Video Encryption using RSA Algorithm" (2015)
- [3] Aswin Vishal Kumar, Kiran, Sandeep, Santhiya "Intelligent Surveillance Robot for Monitoring Outdoor Security " (2020)
- [4] [What is RSA encryption and how does it work? | Comparitech](#)
- [5] <https://developer.android.com/training/connect-devices-wirelessly/wifi-direct>
- [6] <http://cagneyemoreau.com/stream-video-android/>
- [7] [www.agora.io](http://www.agora.io)
- [8] <https://www.geeksforgeeks.org/rsa-algorithm-cryptography/>
- [9] <https://docuserve.com/blog-post/video-encryption-how-to-secure-your-digital-content/>

## APPENDIX

### RSA Encryption Code

```
#include<iostream>
#include<math.h>
#include <cstdlib>
using namespace std;
int gcd(int a, int h)
{
    int temp;
    while(1)
    {
        temp = a%h;
        if(temp==0)
            return h;
        a = h;
        h = temp;
    }
}
int main()
{
    long long int p = 1980, q = 1080,n;
    srand(time(0));
    int ran = rand();
    p+=ran;
    q+=ran;
    n=p*q;
    long long int count;
    long long int phi = (p-1)*(q-1);

    //public key
    long long int e=2;
```

```

while(e<phi){
count = gcd(e,phi);
if(count==1)
    break;
else
    e++;
}

//private key
long long int d;
long long int k = 2;
d = (1 + (k*phi))/e;
double msg = 12;
long long int c = pow(msg,e);
long long int m = pow(c,d);
c=fmod(c,n);
m=fmod(m,n);

//printing the generated keys
cout<<"Public Key:\n N = "<<n<<" E = "<<e;
cout<<"\n\nPrivate Key: \n D = "<<d;
return 0;
}

```

### **VideoChatViewActivity.java**

```

package io.agora.tutorials1v1vcall;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;

```

```

import android.text.TextUtils;
import android.util.Log;
import android.view.SurfaceView;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewParent;
import android.widget.FrameLayout;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.Toast;

import io.agora.rtc.IRtcEngineEventHandler;
import io.agora.rtc.RtcEngine;
import io.agora.rtc.video.VideoCanvas;
import io.agora.rtc.video.VideoEncoderConfiguration;
import io.agora.uikit.logger.LoggerRecyclerView;

public class VideoChatViewActivity extends AppCompatActivity {
    private static final String TAG = VideoChatViewActivity.class.getSimpleName();

    private static final int PERMISSION_REQ_ID = 22;

    private static final String[] REQUESTED_PERMISSIONS = {
        Manifest.permission.RECORD_AUDIO,
        Manifest.permission.CAMERA,
        Manifest.permission.WRITE_EXTERNAL_STORAGE
    };

    private RtcEngine mRtcEngine;
    private boolean mCallEnd;
    private boolean mMuted;

    private FrameLayout mLocalContainer;

```

```

private RelativeLayout mRemoteContainer;
private VideoCanvas mLocalVideo;
private VideoCanvas mRemoteVideo;

private ImageView mCallBtn;
private ImageView mMuteBtn;
private ImageView mSwitchCameraBtn;

// Customized logger view
private LoggerRecyclerView mLogView;

private final IRtcEngineEventHandler mRtcEventHandler = new
IRtcEngineEventHandler() {

    @Override
    public void onFirstRemoteVideoDecoded(final int uid, int width, int height, int
elapsed) {
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                mLogView.logI("First remote video decoded, uid: " + (uid &
0xFFFFFFFFFL));
                setupRemoteVideo(uid);
            }
        });
    }

    @Override
    public void onUserOffline(final int uid, int reason) {
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                mLogView.logI("User offline, uid: " + (uid & 0xFFFFFFFFFL));

```

```

        onRemoteUserLeft(uid);
    }
    });
}
};

```

```

private void setupRemoteVideo(int uid) {
    ViewGroup parent = mRemoteContainer;
    if (parent.indexOfChild(mLocalVideo.view) > -1) {
        parent = mLocalContainer;
    }
}

```

```

if (mRemoteVideo != null) {
    return;
}

```

```

    SurfaceView view = RtcEngine.CreateRenderView(getBaseContext());
    view.setZOrderMediaOverlay(parent == mLocalContainer);
    parent.addView(view);
    mRemoteVideo = new VideoCanvas(view,
VideoCanvas.RENDER_MODE_HIDDEN, uid);
    // Initializes the video view of a remote user.
    mRtcEngine.setupRemoteVideo(mRemoteVideo);
}

```

```

private void onRemoteUserLeft(int uid) {
    if (mRemoteVideo != null && mRemoteVideo.uid == uid) {
        removeFromParent(mRemoteVideo);

        mRemoteVideo = null;
    }
}
}

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_video_chat_view);
    initUI();

    // Ask for permissions at runtime.
    // This is just an example set of permissions. Other permissions
    // may be needed, and please refer to our online documents.
    if (checkSelfPermission(REQUESTED_PERMISSIONS[0],
        PERMISSION_REQ_ID) &&
        checkSelfPermission(REQUESTED_PERMISSIONS[1],
        PERMISSION_REQ_ID) &&
        checkSelfPermission(REQUESTED_PERMISSIONS[2],
        PERMISSION_REQ_ID)) {
        initEngineAndJoinChannel();
    }
}

private void initUI() {
    mLocalContainer = findViewById(R.id.local_video_view_container);
    mRemoteContainer = findViewById(R.id.remote_video_view_container);

    mCallBtn = findViewById(R.id.btn_call);
    mMuteBtn = findViewById(R.id.btn_mute);
    mSwitchCameraBtn = findViewById(R.id.btn_switch_camera);

    mLogView = findViewById(R.id.log_recycler_view);

    // Sample logs are optional.
    showSampleLogs();
}

```

```

private void showSampleLogs() {
    mLogView.logI("Welcome to Agora 1v1 video call");
    mLogView.logW("You will see custom logs here");
    mLogView.logE("You can also use this to show errors");
}

private boolean checkSelfPermission(String permission, int requestCode) {
    if (ContextCompat.checkSelfPermission(this, permission) !=
        PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, REQUESTED_PERMISSIONS,
requestCode);
        return false;
    }

    return true;
}

@Override
public void onRequestPermissionsResult(int requestCode,
                                     @NonNull String[] permissions, @NonNull int[]
grantResults) {
    if (requestCode == PERMISSION_REQ_ID) {
        if (grantResults[0] != PackageManager.PERMISSION_GRANTED ||
            grantResults[1] != PackageManager.PERMISSION_GRANTED ||
            grantResults[2] != PackageManager.PERMISSION_GRANTED) {
            showLongToast("Need permissions " +
Manifest.permission.RECORD_AUDIO +
                "/" + Manifest.permission.CAMERA + "/" +
Manifest.permission.WRITE_EXTERNAL_STORAGE);
            finish();
            return;
        }
    }
}

```

```

        // Here we continue only if all permissions are granted.
        // The permissions can also be granted in the system settings manually.
        initEngineAndJoinChannel();
    }
}

private void showLongToast(final String msg) {
    this.runOnUiThread(new Runnable() {
        @Override
        public void run() {
            Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_LONG).show();
        }
    });
}

private void initEngineAndJoinChannel() {
    // This is our usual steps for joining
    // a channel and starting a call.
    initializeEngine();
    setupVideoConfig();
    setupLocalVideo();
    joinChannel();
}

private void initializeEngine() {
    try {
        mRtcEngine = RtcEngine.create(getBaseContext(),
getString(R.string.agora_app_id), mRtcEventHandler);
    } catch (Exception e) {
        Log.e(TAG, Log.getStackTraceString(e));
        throw new RuntimeException("NEED TO check rtc sdk init fatal error\n" +
Log.getStackTraceString(e));
    }
}

```

```

}

private void setupVideoConfig() {
    // In simple use cases, we only need to enable video capturing
    // and rendering once at the initialization step.
    // Note: audio recording and playing is enabled by default.
    mRtcEngine.enableVideo();

    mRtcEngine.setVideoEncoderConfiguration(new VideoEncoderConfiguration(
        VideoEncoderConfiguration.VD_640x360,
        VideoEncoderConfiguration.FRAME_RATE.FRAME_RATE_FPS_15,
        VideoEncoderConfiguration.STANDARD_BITRATE,
        VideoEncoderConfiguration.ORIENTATION_MODE.ORIENTATION_MODE_FIXED_PORTRAIT));
}

private void setupLocalVideo() {
    SurfaceView view = RtcEngine.CreateRenderView(getBaseContext());
    view.setZOrderMediaOverlay(true);
    mLocalContainer.addView(view);
    dimension of the video may have clipped contents.
    mLocalVideo = new VideoCanvas(view,
        VideoCanvas.RENDER_MODE_HIDDEN, 0);
    mRtcEngine.setupLocalVideo(mLocalVideo);
    mLogView.logl("" +
        ".video call");
}

private void joinChannel() {
    String token = getString(R.string.agora_access_token);
    if (TextUtils.isEmpty(token) || TextUtils.equals(token,
        "0062fec813db68549f6b583a8dfc7d3a196IABNnE2+2LLbY09p8vKzmqPxaX38pD
        0jxo6M0Nwhqjuk9wx+f9gAAAAEABFd1n8wGQuYAEAAQC/ZC5g")) {

```

```

        token = null; // default, no token
    }
    mRtcEngine.joinChannel(token, "demoChannel1", "Extra Optional Data", 0);
}

```

```

@Override
protected void onDestroy() {
    super.onDestroy();
    if (!mCallEnd) {
        leaveChannel();
    }
}
/*
    Destroys the RtcEngine instance and releases all resources used by the
    Agora SDK.

```

This method is useful for apps that occasionally make voice or video calls, to free up resources for other operations when not making calls.

```

*/
RtcEngine.destroy();
}

```

```

private void leaveChannel() {
    mRtcEngine.leaveChannel();
}

```

```

public void onLocalAudioMuteClicked(View view) {
    mMuted = !mMuted;
    // Stops/Resumes sending the local audio stream.
    mRtcEngine.muteLocalAudioStream(mMuted);
    int res = mMuted ? R.drawable.btn_mute : R.drawable.btn_unmute;
    mMuteBtn.setImageResource(res);
}

```

```

public void onSwitchCameraClicked(View view) {

```

```

        // Switches between front and rear cameras.
        mRtcEngine.switchCamera();
    }

    public void onCallClicked(View view) {
        if (mCallEnd) {
            startCall();
            mCallEnd = false;
            mCallBtn.setImageResource(R.drawable.btn_endcall);
        } else {
            endCall();
            mCallEnd = true;
            mCallBtn.setImageResource(R.drawable.btn_startcall);
        }

        showButtons(!mCallEnd);
    }

    private void startCall() {
        setupLocalVideo();
        joinChannel();
    }

    private void endCall() {
        removeFromParent(mLocalVideo);
        mLocalVideo = null;
        removeFromParent(mRemoteVideo);
        mRemoteVideo = null;
        leaveChannel();
    }

    private void showButtons(boolean show) {
        int visibility = show ? View.VISIBLE : View.GONE;
        mMuteBtn.setVisibility(visibility);
    }

```

```

        mSwitchCameraBtn.setVisibility(visibility);
    }

    private ViewGroup removeFromParent(VideoCanvas canvas) {
        if (canvas != null) {
            ViewParent parent = canvas.view.getParent();
            if (parent != null) {
                ViewGroup group = (ViewGroup) parent;
                group.removeView(canvas.view);
                return group;
            }
        }
        return null;
    }

    private void switchView(VideoCanvas canvas) {
        ViewGroup parent = removeFromParent(canvas);
        if (parent == mLocalContainer) {
            if (canvas.view instanceof SurfaceView) {
                ((SurfaceView) canvas.view).setZOrderMediaOverlay(false);
            }
            mRemoteContainer.addView(canvas.view);
        } else if (parent == mRemoteContainer) {
            if (canvas.view instanceof SurfaceView) {
                ((SurfaceView) canvas.view).setZOrderMediaOverlay(true);
            }
            mLocalContainer.addView(canvas.view);
        }
    }

    public void onLocalContainerClick(View view) {
        switchView(mLocalVideo);
        switchView(mRemoteVideo);
    }

```

```
}  
}
```

## Strings.xml

```
<resources>  
    <string name="app_name">Agora Tutorial</string>  
  
    <string name="agora_app_id">2fec813db68549f6b583a8dfc7d3a196</string>  
  
    <string  
name="agora_access_token">0062fec813db68549f6b583a8dfc7d3a196IABNnE2  
+2LLbY09p8vKzmqPxaX38pD0jxo6M0Nwhqjuk9wx+f9gAAAAAEABFd1n8wGQuY  
AEAAQC/ZC5g</string>  
</resources>
```

