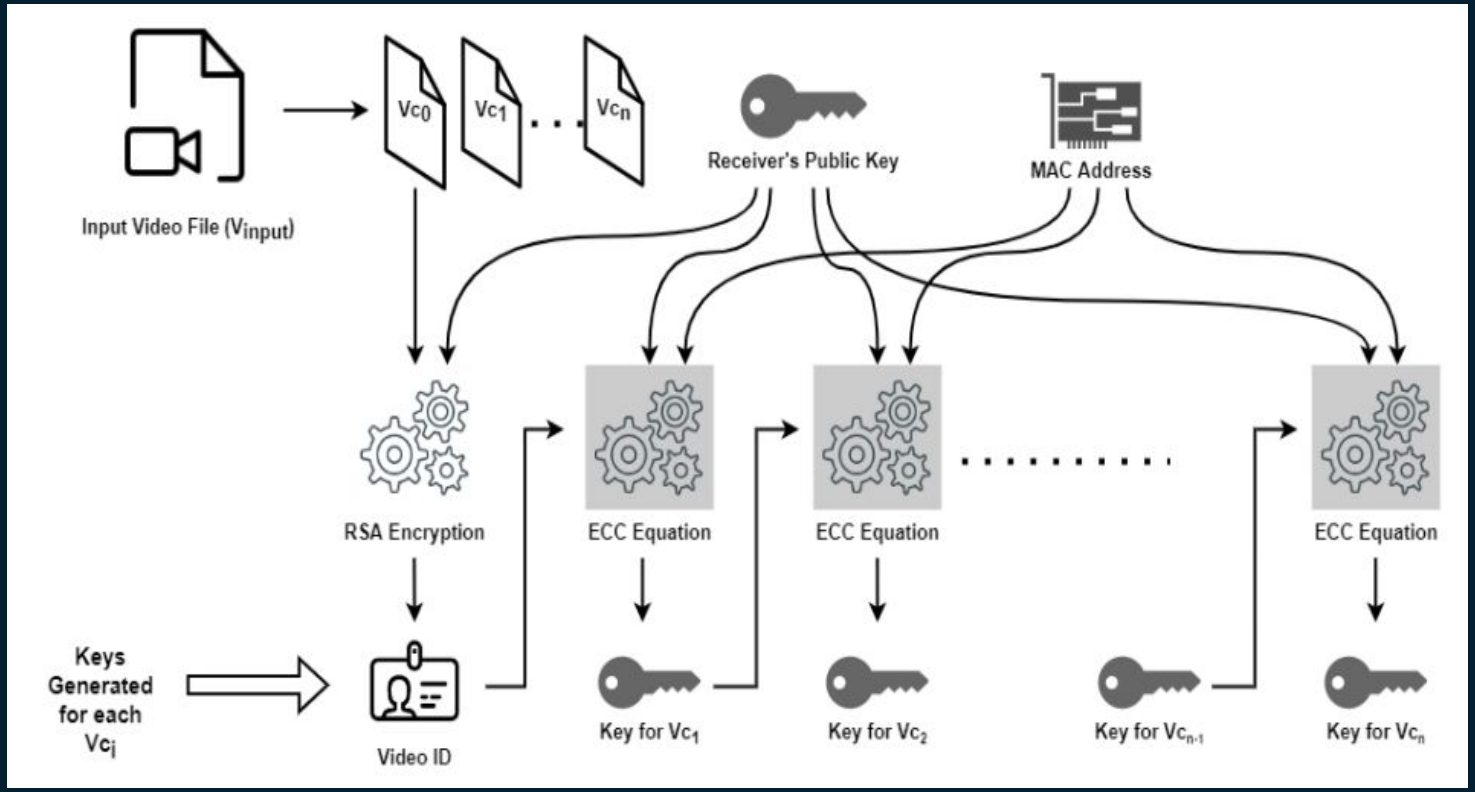
A NOVEL HYBRID MULTI-KEY CRYPTOGRAPHY FOR VIDEO COMMUNICATION

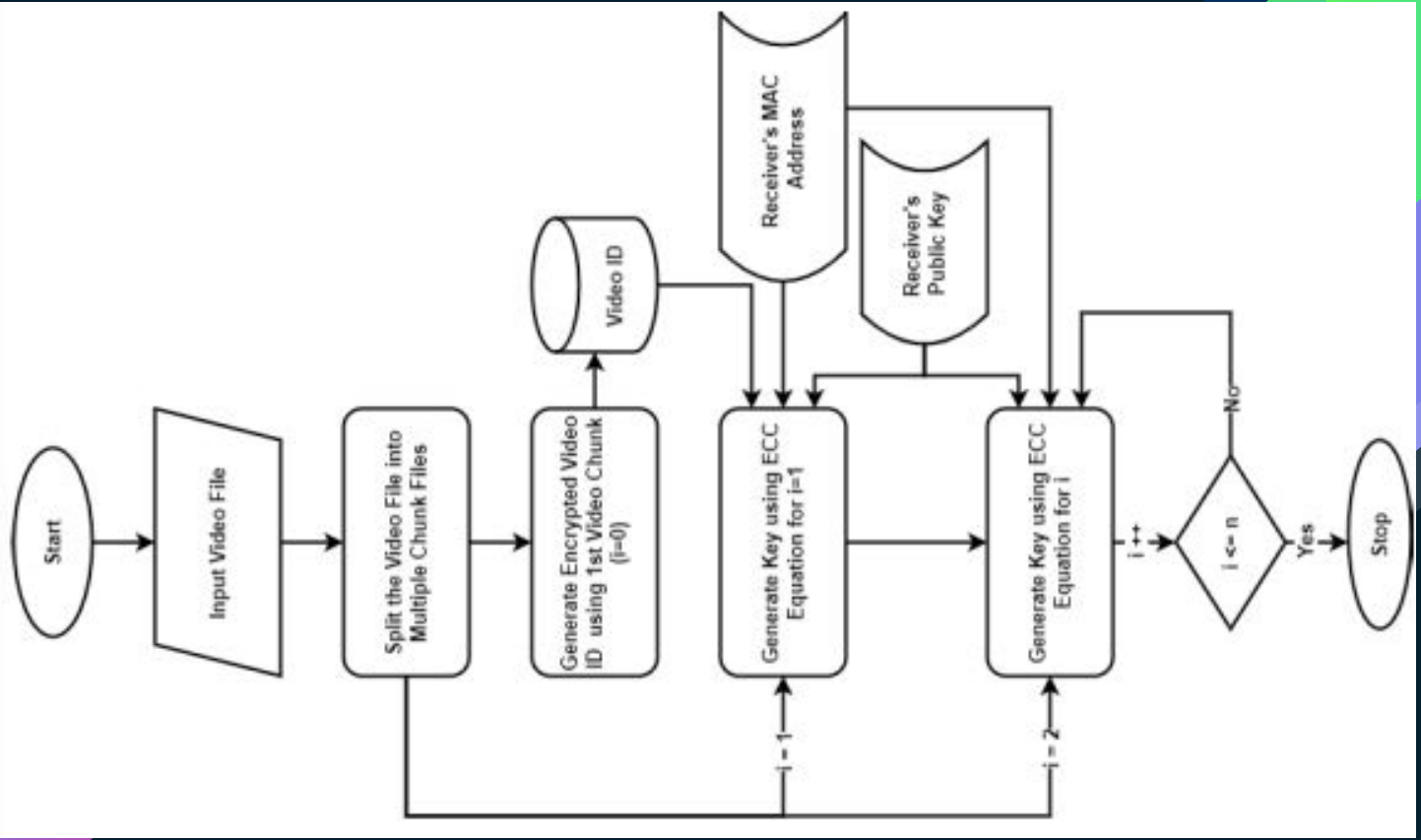
Abstract

Online video streaming is becoming more widespread in people's everyday entertainment routines. Protecting copyright and piracy has become a key concern in real-time video streaming systems.This research provides a revolutionary multi-key and hybrid cryptography approach to offer security. This work describes the software implementation of video encryption and decryption employing continuous systems based on the Elliptic Curve Cryptography approach as pseudo random encryption key generators.This approach creates several keys to encrypt and decode small chunks of video files that are produced dynamically based on the video data. The suggested approach was implemented on the Android platform, where applications for sender and recipients had been created to enable streaming. The security and performance of the proposed system have been examined by implementing it on devices and streaming videos. The outcomes demonstrate superiority in terms of performance and security.

Key Generation



FLOW DIAGRAM

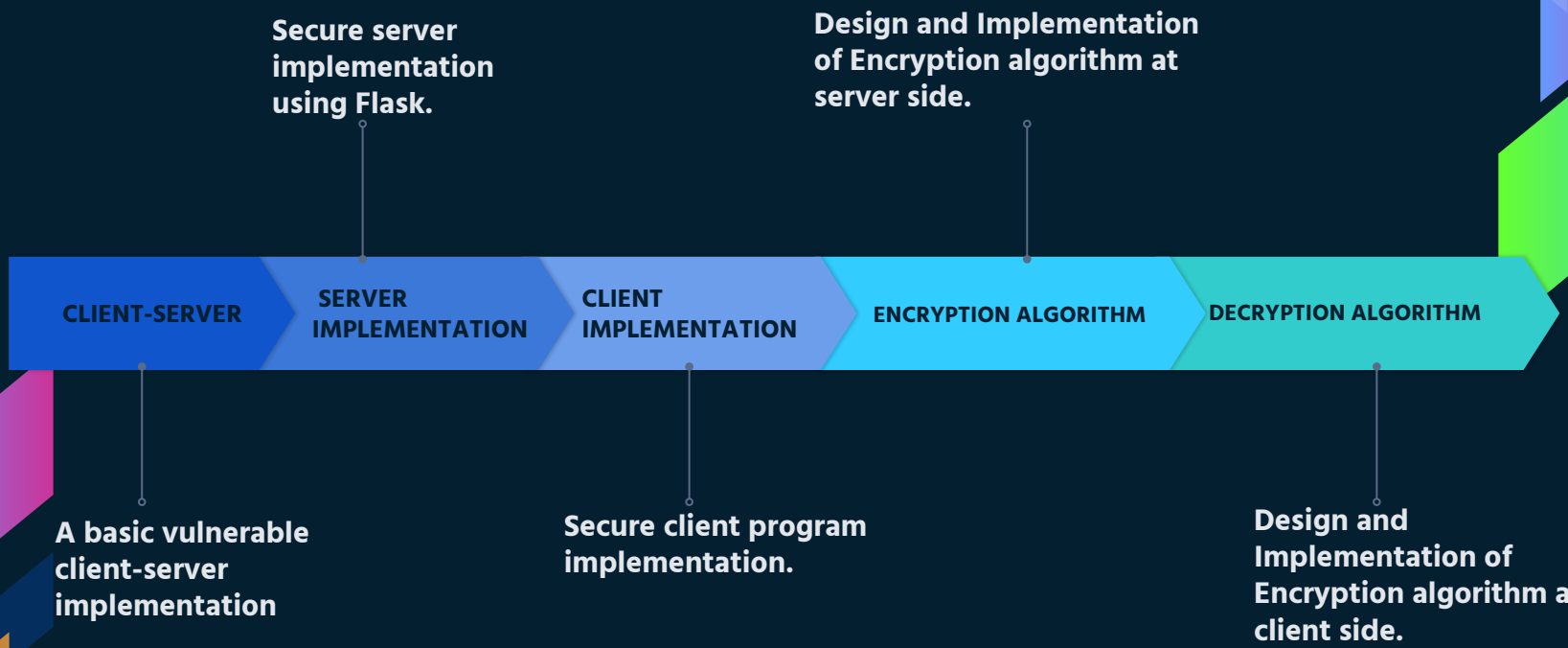


ENCRYPTION ALGORITHM

1: Input video file V_input.
2: Generate video chunks V_ci from V_input.
3: Fetch the receiver's public key of the R_Pkey.
4: Collect receiver's MAC address R_mac.
5: Generate V_ID using Vc_0.
6: Store V_ID in a temporary file.
7: Encrypt Vc0 using RSA.
8: Generate Key_a ← x^3 + V_ID * x + R_mac.
9: Encrypt Vc_1 using Key_a and AES.
10: for i:=2 don.
11: Generate Key_a ← x^3 + Key_a * x + R_mac.
12: Encrypt V_ci using Key_a and AES.
13: end for.

Project Life Cycle

Project Flow Diagram



PHASE - 1

- **Developing a vulnerable video file streaming application**.
- Technologies can be used :
    - Simple **Client - Server** model using **Socket Programming**
    - Django Web Framework
    - Flask Web Framework

PHASE - 2

- **Secured Server Implementation**

  - Server can be developed by **Flask Web Framework.**
  - Server will run the **Encryption** in the background.
  - The Flask framework can be used with **Sqlite3/MySQL** database to store user attributes (*uname, pass, MAC,etc.*)

PHASE - 2

- Server will only hosts one endpoint ***/signup***
- After the signup the **http://server/signup** return a **client** application which has decryption function.

PHASE - 3

- **Client Implementation**
  - Client program will be provided to the user once they successfully completed sign up process.
  - The client program will be installed on user system.
  - The client program will access the locally available **Private key** , **MAC** and other attributes.

PHASE - 3

- The client program will send a request to the server with these attributes.
- Then the server will authenticate the client, if they are valid users, the requested video will be streamed on the client program.

PHASE - 4

Design and Implementation of Encryption algorithm at server side.

- The FFmpeg module can be used divide videos into small chunks.
- ECC is used to generate key stream
- AES is used to encrypt video chunks with key

PHASE - 5

Design and Implementation of Decryption algorithm at client side.

- The encrypted chunk is decrypted with the private key and other attributes.
- ECC is used to generate key stream
- AES is used to decrypt video chunks with key

Thank You

...Our Team 1337...