SECURED DIGITAL SIGNATURE SYSTEM FOR ID CARD VERIFICATION

PRANAV GHANTE – 21CSB0A75

UTKARSH RAJ – 21CSB0A63

1. **Introduction**

In today's digitized world, the importance of secure and reliable identification systems cannot be overstated. However, traditional methods of verifying identity, such as physical ID cards, are susceptible to fraud and manipulation. In response to this challenge, the field of cryptography offers promising solutions to enhance the security and integrity of identity verification processes.

The problem addressed in this assignment is the need for a robust system that can efficiently verify the authenticity of ID cards through a combination of Optical Character Recognition (OCR) and image recognition techniques, while also generating digital certificates with RSA digital signatures.

The significance of this problem lies in the growing reliance on digital transactions and online services, where the need for trustworthy identity verification is paramount. With the proliferation of identity theft and fraudulent activities, there is an urgent need for solutions that can reliably authenticate individuals and safeguard sensitive information.

Existing solutions have attempted to address this problem through various means, including manual verification processes, barcode scanning, and biometric authentication. While these methods offer certain advantages, they often lack the comprehensive approach required to detect sophisticated forms of fraud and tampering.

The key contributions of our project lie in the integration of OCR and image recognition technologies to extract and validate information from ID cards, coupled with the generation of digital certificates using the RSA algorithm for enhanced security. By combining these techniques, our system aims to provide a robust and efficient solution for identity verification in digital environments.

1. **Objectives**

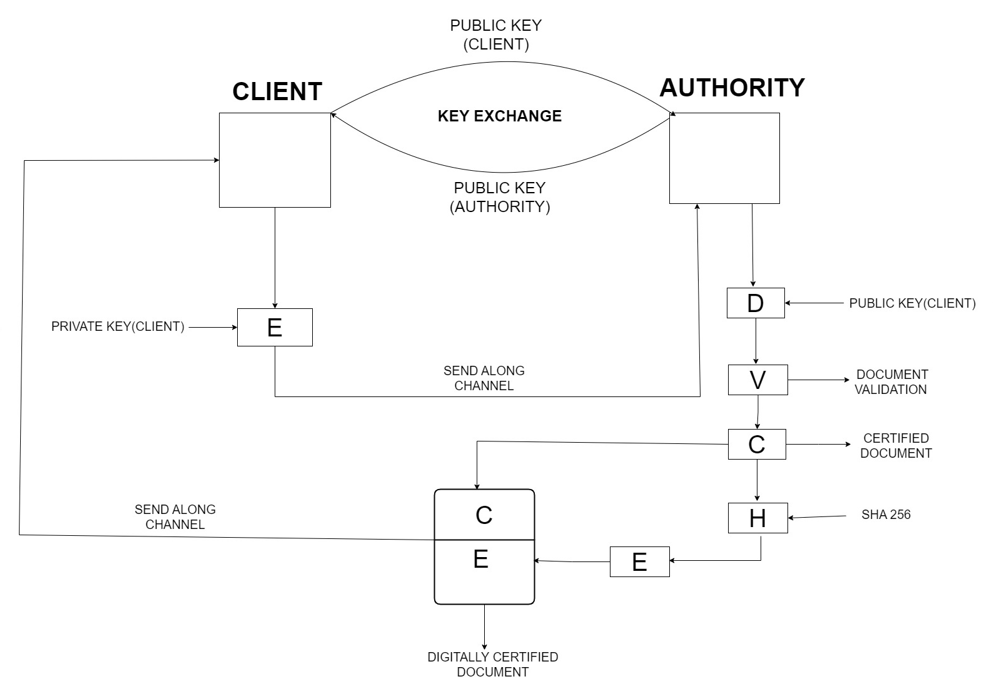
* **Confidentiality and Integrity** – Confidentiality ensure that the ID card image remains confidential during transmission from the client to the server by encrypting it. This prevents unauthorized parties from intercepting and accessing the sensitive information contained within the ID card.

Integrity of the ID card is guaranteed by verifying its authenticity upon receipt at the server. This involves confirming that the ID card has not been tampered with or altered during transmission or processing.

* **Authentication and Non-Repudiation** - Authenticate the identity of the client and server to each other to establish trust in the communication. This ensures that both parties are legitimate entities and not imposters attempting to gain unauthorized access or manipulate data.

Achieved non-repudiation by creating a digital signature of the ID card data at the server. This ensures that the server cannot deny having verified the ID card or generated the signature, providing accountability for its actions.

1. **Implementation and Results analysis**

**CONCEPTUAL DIAGRAM**

**ALGORITHM**

**Step1 -** Public Key exchange is performed between the client and Authority.

**Step2** - Client then encrypts the ID card image with its private key and sends it to the Authority for generation of digital signature

**Step3** - After receiving the ID card the Authority decrypts the ID card with the public key of the Client.

**Step4** - Through Optical Character Recognition the details of the ID card are extracted and verified by comparing with data in the database. The image of student in ID card is verified uniquely by hashing technique.

**Step5** - If the details do not match a message saying “Tampered Document “is sent to client.

**Step6** – If the details match the digital signature of the ID card is created through RSA. This is done by calculating the hash through SHA\_256 and then encrypting this hash with the private key of the Authority.

**Step7** – This Digitally Certified ID card is then sent back to the Client.

1. **Conclusion (100 words)**

In conclusion, this project presents a comprehensive solution to the challenges of identity verification in digital environments. By leveraging OCR, image recognition, and RSA cryptography, the system ensures the authenticity and integrity of ID cards while generating secure digital certificates. This approach enhances trust in digital transactions, mitigates the risks of fraud and identity theft, and strengthens overall cybersecurity measures

1. **Learning outcomes (In bullet wise)**

* Understood and Implemented RSA encryption and decryption, digital signatures, and hashing algorithms (SHA-256)
* Implemented socket programming for client-server communication facilitating practical application of secure communication concepts
* Understood processing of image data using libraries such as OpenCV and NumPy. This includes loading, decrypting, displaying, and analyzing image content
* Integrated external APIs (OCRSpace) for image text extraction
* Developed a unique way of verifying the persons image using the hashing concept. This method efficiently verifies the image without comparing it with the actual image.

1. **Source code: (Keep only the essential source code of the Assignment)**

# Step 5: Encryption

for i in range(row):

    for j in range(col):

        r, g, b = my\_img[i, j]

        C1 = power(r, E\_server, N\_server)

        C2 = power(g, E\_server, N\_server)

        C3 = power(b, E\_server, N\_server)

        enc[i][j] = [C1, C2, C3]

        my\_img[i, j] = [C1 % 256, C2 % 256, C3 % 256]

# Step 6: Decryption

for i in range(row):

    for j in range(col):

        r, g, b = enc\_matrix[i][j]

        M1 = power(r, D, N)

        M2 = power(g, D, N)

        M3 = power(b, D, N)

        original\_img[i, j] = [M1 % 256, M2 % 256, M3 % 256]

# Extracting text from Image

import ocrspace

api = ocrspace.API()

info = api.ocr\_file("input.png")

words = info.split()

# Computing the hash of the document

convert\_bytes = bytes(original\_img)

msg\_hash = hashlib.sha256(convert\_bytes)

msg = msg\_hash.hexdigest()

# Sending the Document

img\_data = pickle.dumps(original\_img)

img\_size = struct.pack("=L", len(img\_data))

client.sendall(img\_size)

client.sendall(img\_data)

# Establish a socket connection

server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_address = ("192.168.0.217", 1234)

server.bind(server\_address)

server.listen()

print("Server started listening")

# Accept a connection from the client

client, client\_address = server.accept()

print("Connected to client")

1. **References**

* Python Documentation: <https://docs.python.org/>
* NumPy Documentation : [https://numpy.org/doc/](https://numpy.org/doc/%20)
* Matplotlib Documentation : <https://matplotlib.org/stable/index.html>
* OpenCV Documentation: <https://docs.opencv.org/4.x/index.html>
* Research Paper: <https://www.researchgate.net/publication/3227862_Digital_signatures>