SECURED DIGITAL SIGNATURE SYSTEM FOR ID CARD VERIFICATION

PRANAV GHANTE – 21CSB0A75

UTKARSH RAJ – 21CSB0A63

**Introduction**

In the modern era of digitalization, the need for secure and efficient systems for handling sensitive information has become paramount. One such area where security is of utmost importance is the verification and certification of identity documents, such as ID cards. The problem addressed in this assignment revolves around the secure transmission, verification, certification, and digital signing of ID cards using cryptography techniques.

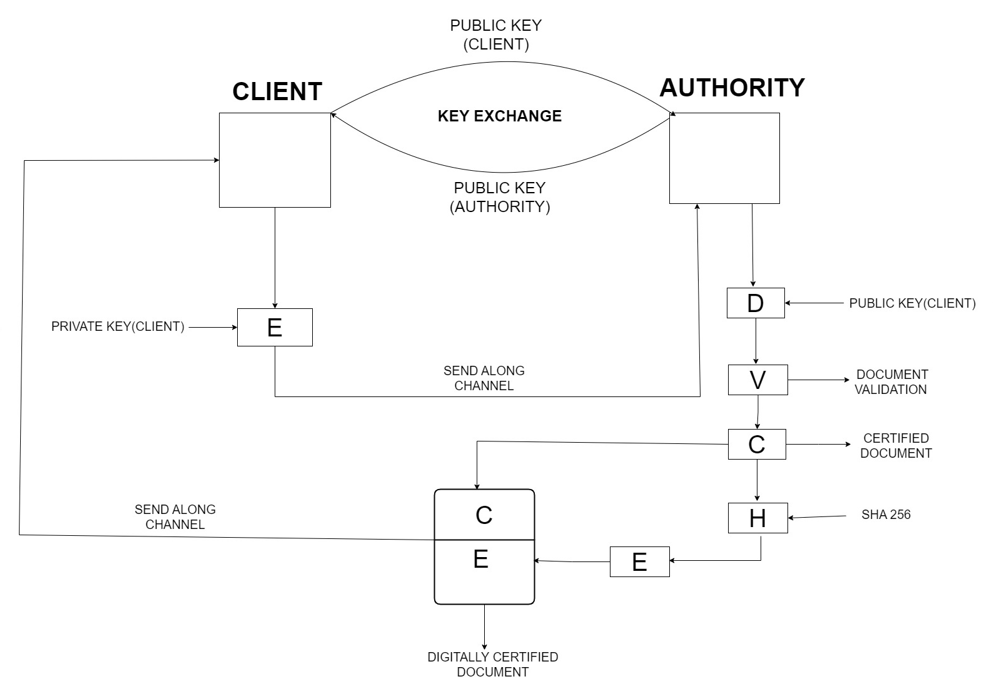
Ensuring the authenticity of ID cards is crucial in various domains, including border control, access management, and financial transactions. However, traditional methods of ID verification are often prone to forgery and tampering, leading to security breaches and identity theft. Thus, there is a pressing need for robust solutions that can mitigate these risks and provide reliable verification mechanisms.

Existing solutions for ID card verification typically rely on manual inspection or barcode scanning, which are susceptible to human error and manipulation. While some systems incorporate cryptographic techniques for secure transmission, they often lack comprehensive verification and certification mechanisms, leaving room for exploitation by adversaries.

The key contributions of our project lie in the integration of advanced cryptography techniques, such as RSA encryption and digital signatures, with image recognition and OCR technologies for comprehensive ID card verification. By leveraging these techniques, we aim to provide a secure and automated system for verifying the authenticity of ID cards, thereby reducing the risk of fraud and ensuring trust in digital identity management.

**Objectives**

* Develop a secure system for the transmission, verification, certification, and digital signing of ID cards using cryptography techniques.
* Implement image recognition and OCR algorithms to automate the verification process and enhance the accuracy and reliability of ID card authentication.

**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.

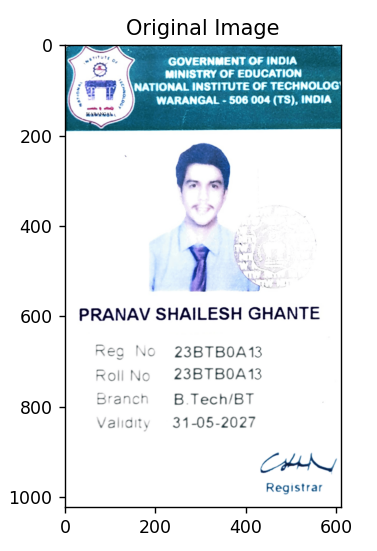
**Step4** - Through Optical Character Recognition the details of the ID card are extracted and verified by comparing with data in the database. The image of student in ID card is verified uniquely by hashing technique.

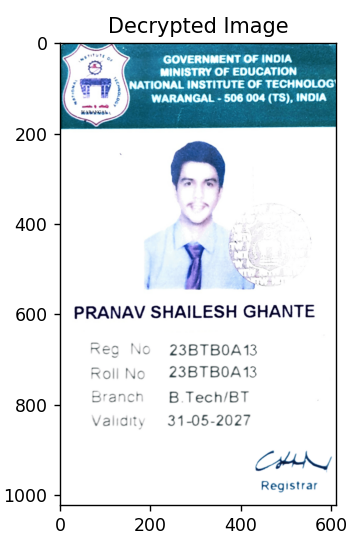
**Step5** - If the details do not match a message saying “Tampered Document “is sent to client.

**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.

**Output:**

The Image of the ID card sent to the Authority

The Encrypted Id card sent to the Authority The Id card after Decryption

The final Digitally Certified and Signed Document

**Result Analysis**

**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.

**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

**Conclusion**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

**Learning outcomes**

* 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.

**Source code:**

# 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")

**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>

**Drive Link to full code**

<https://github.com/PranavGhante/SECURED-DIGITAL-SIGNATURE-SYSTEM-FOR-ID-CARD-VERIFICATION-.git>