# CSI3002 - APPLIED CRYPTOGRAPHY AND NETWORK SECURITY

## THEORY ASSIGNMENT – 01

| | |
|---|---|
| **NAME**: BOYANA RAHUL | **COURSE CODE**: CSI3002 |
| **REG.NO**: 21MID0103 | **DATE**: 25-04-2024 |
| **SLOT**: E2 | **FACULTY**: SASIKALA R |

# FINGERPRINT AUTHENTICATED SECURED ANDROID NOTES

*Abstract. In today's digital landscape, ensuring the security of personal data on smartphones has become increasingly crucial. Traditional security measures like PINs and passwords have shown vulnerabilities, leaving sensitive information at risk. This article addresses the pressing need to enhance smartphone data security and introduces a potential solution: secure Android notes authenticated using fingerprints. The aim is to combat these vulnerabilities by leveraging biometric authentication to secure sensitive data. By introducing the concept of secure notes authenticated through fingerprints, this article presents a more robust approach to protecting personal information on smartphones. This solution emphasizes the critical role of encryption standards like AES and SHA256 in fortifying data protection. AES encryption secures the content of these notes, while SHA256 ensures the integrity of the data, making it resistant to unauthorized alterations. By adopting this comprehensive strategy, the goal is to cater to the evolving security requirements of modern smartphone users. This approach ensures that users' data remains confidential and accessible only through authorized fingerprint verification. This shift towards biometric authentication aims to offer a more secure and user-friendly method for safeguarding sensitive information on smartphones*

*Keywords - Fingerprint authentication and recognition, Data Security, Data Privacy, Biometric Security,*
*Security risks, Symmetric and Asymmetric Encryption, secured Android notes, AES Algorithm, Hashing Algorithms, and Cryptographic Key Management*

## I. INTRODUCTION

A fingerprint authentication system is a security system that verifies a person's identity by comparing their fingerprint to a stored template. It falls under the umbrella of biometric authentication, which uses unique biological traits for identification. There components include - Sensor, Extractor, Comparator, Decision Engine. Sensor- This captures a digital image of the fingerprint. There are different sensor technologies like capacitive, optical, and

ultrasonic. Extractor -This component extracts relevant features from the captured fingerprint image, like ridge patterns and minutiae (branching points and ridge endings). Comparator - This compares the extracted features with a stored fingerprint template (a mathematical representation of the fingerprint) and determines if they match. Decision Engine - Based on the comparison score from the comparator, the system decides whether to grant access or not. Few characteristics of fingerprint authentication system are as follows. Fingerprint scanning is a relatively quick and easy process compared to remembering passwords or carrying keys.
Fingerprints are unique to each individual, making them a more secure way to verify identity compared to something that can be stolen or shared, like a password. Modern fingerprint scanners are highly accurate, although factors like skin condition or sensor limitations can affect recognition. Some systems incorporate liveness detection to ensure the presented fingerprint is from a real person and not a replica.

Fingerprint authentication systems are widely used in various applications like Unlocking smartphones and laptops, Access control for buildings and secure areas, Authorizing financial transactions, Border security checkpoints. But in this paper we're mainly going to concentrate on fingerprint authentication system is used in android notes. Fingerprint authentication in Android notes provides a secure and convenient method for users to access and protect their sensitive information without relying on traditional passwords. It is beneficial in securing personal notes, financial information, and other confidential data stored on smartphones, enhancing data privacy and security. The integration of fingerprint authentication addresses evolving security concerns in the digital landscape, offering a more robust and user-friendly approach to safeguarding information.: Implementing secure encryption keys, utilizing AES encryption, and SHA-256 hashing for data integrity verification are recommended solutions to enhance security. Challenges may arise if

the fingerprint sensor fails or malfunctions, potentially hindering users from accessing their notes without alternative authentication methods or backups.

## II. LITERATURE REVIEW

### A. Visual Cryptography Based Biometric Authentication Mechanism for Online Elections (2022)

**Objective:** The paper aims to enhance biometric authentication in electronic voting systems through visual cryptography and fingerprint matching.
**Mechanism Used:** The mechanism involves utilizing image processing methods and neural networks for precise feature extraction.
**Solution:** The proposed approach provides a partial solution by bolstering security and reducing fraud in electronic voting.
**Advantage:** The advantage of the system lies in fostering increased voter engagement and confidence in the voting process.
**Disadvantage:** A potential drawback could be the complexity of implementation.
**Future Scope:** Future prospects include refining the system for broader application in secure voting processes, potentially integrating additional biometric modalities to enhance security and accuracy and exploring advanced cryptographic techniques to strengthen resilience against potential threats and vulnerabilities.

### B. Establishing security using cryptography and biometric authentication to counter cyber-attacks (2023)

**Objective:** The paper aimed to enhance security in cloud computing through the integration of fingerprint encryption and lightweight ECC encryption for user authentication.

**Mechanism Used:** The study implemented a combined algorithm to reduce computational load and improve system security.
**Solution:** The proposed method demonstrated good performance, identified more attacks, reduced computing load, and enhanced system security.
**Advantages:** Improved security, reduced computational load, and increased attack detection capabilities.
**Disadvantages:** Potential challenges in implementation and scalability.
**Future Scope:** Future research can focus on measuring the robustness of detection systems, assessing deep learning models, and developing a biometric multi-factor identification system for quick and accurate operations.

C. Generation of Cryptographic Keys and Person's Verification Based on face Biometric (2023)

**Objective:** The paper aimed to propose a method for generating cryptographic keys based on face biometric data for user authentication.
**Mechanism Used:** The method utilized a fuzzy extractor scheme with 19 face biometric features to generate cryptographic keys.
**Solution:** A full solution was obtained, demonstrating the operability of the method and achieving error-free authentication of users.
**Advantages:** The method allows for the generation of cryptographic keys up to 232 bits with an average redundancy of about 52%.
**Disadvantages:** Typical biometric authentication system drawbacks include the "fuzziness" of biometric data and potential security vulnerabilities.
**Future Scope:** Future research could focus on enhancing the method's security, scalability, and efficiency for broader real-world applications.

D. A Cryptography based Face Authentication System for Secured Communication (2022)

**Objective:** The paper aims to enhance secure communication through a system integrating face recognition and RSA cryptography.
**Mechanism Used:** Utilizes LBPH face recognition, RSA encryption, and SSH tunneling for secure data transfer.
**Solution:** Provides a partial solution by improving authentication and encryption processes for secure communication.
**Advantage:** Efficient authentication with face recognition, secure data transfer using RSA, and SSH tunneling.
**Disadvantage:** Limited message size for RSA encryption may pose constraints on data transfer.
**Future Scope:** Future work could focus on implementing digital envelope mechanism for enhanced encryption and decryption processes.

E. A Secure Authentication Framework to Guarantee the Traceability of Avatars in Metaverse (2023)

**Objective:** The paper aims to design a two-factor authentication framework for avatars in the metaverse to ensure virtual-physical traceability and consistency of identities.
**Mechanism Used:** Utilizes a chameleon signature and iris biometrics for secure authentication and avatar tracking.
**Solution:** Provides a partial solution by addressing the challenges of malicious players in the metaverse.
**Advantage:** Offers a decentralized and traceable authentication framework, enhancing security in virtual environments.
**Disadvantage:** Limited to addressing specific security concerns and may require further enhancements for comprehensive protection.
**Future Scope:** Future research could focus on expanding the framework to cover a broader range of security threats and integrating advanced technologies for enhanced protection

in the metaverse.

F. Empirical Studies of TESLA Protocol: Properties,Implementations, and Replacement of Public Cryptography Using Biometric Authentication(2023)

**Objective:** The paper aimed to enhance the TESLA protocol by introducing the TLI-µTESLA protocol, addressing synchronization and delay issues in authentication.

**Mechanism Used:** TLI-µTESLA utilizes a two-level keychain system and biometric authentication for improved security.
**Solution:** The TLI-µTESLA protocol provides a partial solution by enhancing security services and reducing delay in authentication processes.
**Advantages:** Improved security, reduced delay, and enhanced authentication processes.
**Disadvantages:** Partial solution, may require additional validation and testing for real-world implementation.
**Future Scope:** Further research can focus on real-world implementation, scalability, and efficiency testing of the TLI-µTESLA protocol to validate its effectiveness in diverse IoT environments.

## III. RELATED WORK / BACKGROUND

| Paper | A |
|---|---|
| Observation | Utilizes visual cryptography for biometric authentication in online elections.<br><br>Focuses on enhancing security and trust in the election process through biometric verification. |
| Drawback | Potential challenges in scalability and efficiency when implementing visual cryptography in large-scale online election systems. |

| Paper | B |
|---|---|
| Observation | Emphasizes the use of cryptography and biometric authentication to enhance security against cyber-attacks.<br><br>Addresses the need for robust authentication mechanisms to mitigate cybersecurity threats. |
| Drawback | Limited discussion on the potential vulnerabilities of combining cryptography and biometrics for security. |

| Paper | C |
|---|---|
| Observation | Proposes the generation of cryptographic keys based on face biometrics for person verification.<br><br>Focuses on enhancing security through biometric-based cryptographic key generation. |
| Drawback | Potential challenges in ensuring the uniqueness and reliability of face biometric data for key generation |

| Paper | D |
|---|---|

| | |
|---|---|
| Observation | Introduces a face authentication system based on cryptography for secure communication.<br><br>Highlights the importance of secure communication channels using biometric authentication. |
| Drawback | Potential limitations in the speed and accuracy of face authentication for real-time secure communication. |

| Paper | E |
|---|---|
| Observation | Presents a secure authentication framework for ensuring traceability of avatars in the metaverse.<br><br>Addresses the need for secure and accountable authentication mechanisms in virtual environments. |
| Drawback | Potential complexities in implementing traceability features without compromising user privacy in the metaverse. |

| Paper | F |
|---|---|
| Observation | Conducts empirical studies on the TESLA protocol and its properties.<br><br>Explores the potential of replacing public cryptography with biometric authentication in TESLA. |
| Drawback | Limited discussion on the practical challenges and adoption barriers of replacing public cryptography with biometric authentication in TESLA. |

Through these we could obtain a conclusion that there is a need for Projection in Real-Time Application. Which means Implementing a secure multi-factor authentication system for IoT devices. Robust authentication mechanisms to protect sensitive data and ensure secure access to IoT devices. Developing a biometric multi-factor identification system that combines biometric factors for quick and accurate authentication in real-time IoT applications, addressing drawbacks such as scalability, reliability, and privacy concerns.

## IV. EXISTING SYSTEM

To address the drawbacks identified in the papers while aligning with the scope of application in real-time scenarios provided, we can propose enhancements and solutions

A. **Scalability Enhancement:** Implementing visual cryptography for biometric authentication in online elections could face scalability challenges due to the need to process a large volume of authentication requests. To mitigate this, we can introduce distributed authentication servers that handle authentication requests in parallel, ensuring scalability without compromising security. Additionally, optimizing the visual cryptography algorithms for efficiency can improve the authentication speed.

- **Real-time Integration**: Integrate the visual cryptography-based authentication mechanism with real-time data processing systems to ensure timely authentication during online elections. This integration would involve optimizing algorithms and infrastructure to handle authentication requests in milliseconds, ensuring a seamless and efficient election process.

B**. Vulnerability Assessment**: Conduct a thorough analysis of potential vulnerabilities associated with combining cryptography and biometric authentication. This assessment should include scenarios such as spoofing attacks, replay attacks, and compromised biometric data. By identifying and addressing these vulnerabilities proactively, the system can ensure robust security against cyber-attacks.

-**Dynamic Security Measures:** Implement dynamic security measures such as adaptive biometric templates and cryptographic key rotation to mitigate the impact of potential breaches. Adaptive biometric templates can adjust to changes in biometric data over time, while cryptographic key rotation can limit the exposure of sensitive cryptographic keys, enhancing overall security.

C. **Biometric Data Integrity**: Address concerns regarding the uniqueness and reliability of face biometric data by incorporating multi-factor authentication. In addition to face biometrics, integrate other biometric modalities such as fingerprint or iris recognition to enhance the reliability of biometric-based cryptographic key generation. This multi-factor approach ensures greater accuracy and reduces the risk of false positives or negatives.

-**Privacy Preservation:** Implement privacy-preserving techniques such as secure multi-party computation or homomorphic encryption to protect sensitive biometric data during cryptographic key generation. These techniques enable cryptographic operations to be performed on encrypted biometric data without compromising privacy, ensuring that biometric information remains secure throughout the authentication process.

D. **Real-time Performance Optimization**: Address potential limitations in the speed and accuracy of face authentication by optimizing the face recognition algorithms for real-time performance. Utilize hardware acceleration, parallel processing, and algorithmic optimizations to reduce authentication latency and improve accuracy, ensuring seamless and secure communication.

- **Continuous Authentication:** Implement continuous authentication mechanisms to enhance security during communication sessions. By continuously verifying the user's identity through periodic biometric re-authentication, the system can detect and prevent unauthorized access or session hijacking attempts in real-time, ensuring end-to-end security.

E. **Privacy-Preserving Traceability**: Develop privacy-preserving traceability mechanisms that guarantee the traceability of avatars in the metaverse without compromising user privacy. Utilize cryptographic techniques such as zero-knowledge proofs or anonymous credentials to enable traceability while protecting user anonymity. This ensures accountability and traceability while respecting users' privacy rights in virtual environments.

-**Decentralized Authentication**: Implement decentralized authentication protocols based on blockchain technology to distribute trust and ensure the integrity of authentication processes in the metaverse. By leveraging blockchain's immutability and consensus mechanisms, the system can establish secure and tamper-resistant authentication mechanisms that are resilient to attacks and manipulation.

F. **Practical Implementation Challenges**: Conduct empirical studies to identify practical challenges and adoption barriers associated with replacing public cryptography with biometric authentication in TESLA. Evaluate factors such as performance overhead, compatibility with existing infrastructure, and user acceptance to assess the feasibility and effectiveness of the proposed approach. By addressing these challenges, the system can pave the way for successful adoption and implementation of biometric authentication in TESLA protocols.

-**Interoperability and Standards:** Ensure interoperability and adherence to industry standards when integrating biometric authentication into TESLA protocols. Collaborate with standardization bodies and industry stakeholders to develop interoperable biometric authentication frameworks that seamlessly integrate with existing cryptographic protocols. This approach facilitates compatibility and interoperability across different systems and platforms, promoting widespread adoption and usability of biometric authentication in TESLA.

## V. PROPOSED SYSTEM

### 1) NEED FOR THE SYSTEM

Personal Note-Taking: For personal note-taking, users can employ fingerprint authentication to secure their notes, thoughts, and sensitive information. This ensures that only authorized individuals with the registered fingerprint can access and modify the content, adding an extra layer of privacy and security to personal data.

• **AES Encryption**: Utilize the AES (Advanced Encryption Standard) algorithm for encrypting the content of personal notes. AES is a symmetric encryption algorithm that provides a high level of security. Each note is encrypted using a secret key, and the encrypted data is stored securely.

• **SHA Hashing**: Apply the SHA-256 (Secure Hash Algorithm 256-bit) for hashing the encrypted notes. SHA-256 produces a fixed-size hash value that uniquely represents the content of the note. Storing and comparing these hash values can help detect any unauthorized modifications to the notes.

Business Meetings and Conferences: In business meetings and conferences, professionals can enhance privacy by using fingerprint authentication to secure meeting notes and confidential information. This added layer of security ensures that only authorized individuals with a registered fingerprint can access and safeguard sensitive business discussions.

• **AES Encryption** for Meeting Notes Utilize AES (Advanced Encryption Standard) encryption to secure meeting notes. Encrypt the content of the notes using a symmetric key tied to the user's fingerprint. This ensures that only the authorized user can decrypt and access the confidential meeting information.

• **SHA Hashing** for Content Integrity: Implement SHA-256 hashing to create a unique hash value for the encrypted meeting notes. This hash can serve as a digital

fingerprint of the content, enabling the system to detect any alterations or unauthorized modifications to the meeting notes.

Financial Transactions and Budgeting: For financial transactions and budgeting, individuals can employ the system to secure financial notes, budget details, and investment plans. This adds an extra layer of security, ensuring that sensitive financial information is protected and accessible only to authorized users with a registered fingerprint.

• **AES Encryption** for Financial Notes: Employ AES (Advanced Encryption Standard) encryption to secure financial notes, including budget details and investment plans. Encrypt the content using a symmetric key associated with the user's fingerprint, ensuring that only the authorized user can decrypt and access financial information.

• **SHA Hashing** for Data Integrity: Implement SHA-256 hashing to generate a unique hash value for the encrypted financial notes. This hash acts as a digital fingerprint, allowing the system to detect any unauthorized modifications or tampering with the financial data.

Research and Development: In research and development, the system can safeguard valuable information such as research notes, code snippets, and intellectual property. Researchers and developers can use fingerprint authentication to protect their work, ensuring that only authorized individuals with registered fingerprint can access and secure sensitive R&D data.

• **AES Encryption** for Research Notes:Implement AES (Advanced Encryption Standard) encryption to secure research notes and code snippets. Encrypt the content using a symmetric key associated with the user's fingerprint. This ensures that only the authorized user can decrypt and access the R&D information.

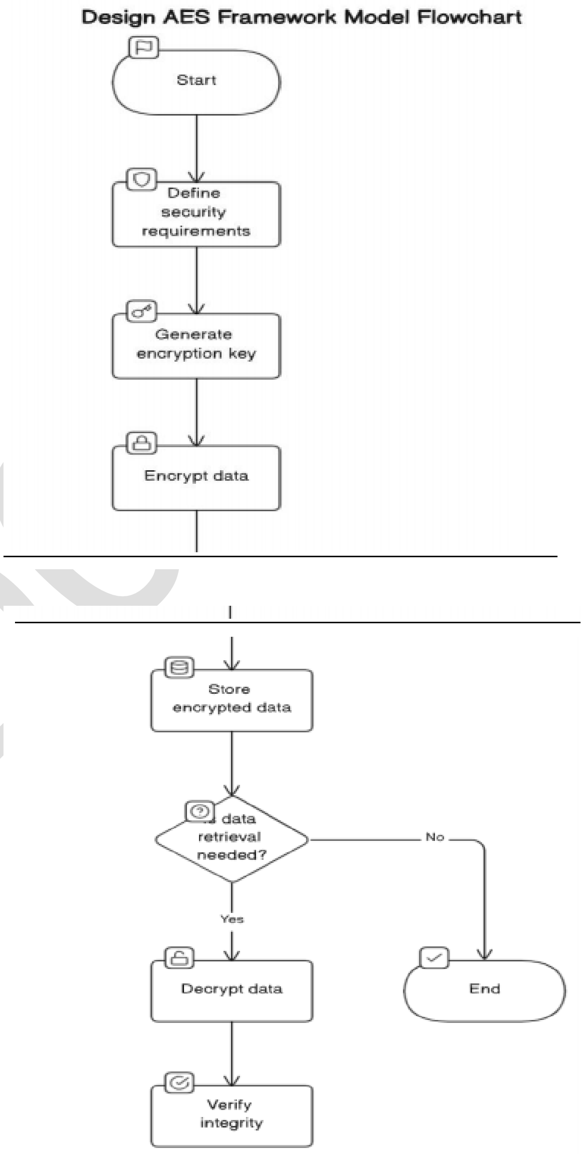
• **SHA Hashing** for Content Integrity:Apply SHA-256 hashing to generate a unique hash value for the encrypted research notes. This hash serves as a digital fingerprint, allowing the system to detect any unauthorized modifications or tampering with the R&D data.

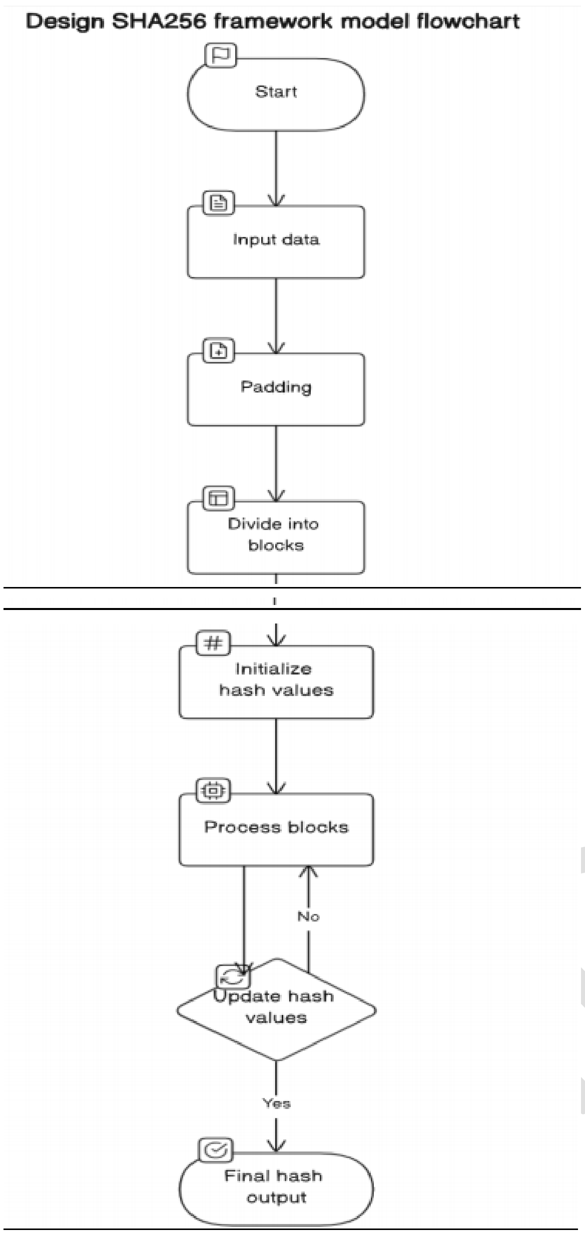
Task and Project Management: In task and project management, professionals can enhance confidentiality by securing project notes, timelines, and important details with fingerprint authentication. This ensures that only authorized individuals with a registered fingerprint can access and maintain the privacy of critical project information.

• **AES Encryption** for Project Notes:Utilize AES (Advanced Encryption Standard) encryption to secure project notes. Encrypt the content using a symmetric key tied to the user's fingerprint, ensuring that only the authorized user can decrypt and access the project information.

• **SHA Hashing** for Data Integrity:Implement SHA-256 hashing to generate a unique hash value for the encrypted project notes. This hash acts as a digital fingerprint, enabling the system to detect any unauthorized modifications or tampering with the project data.

## 2) PROPOSED MODEL / ARCHITECTURE





protocol steps for AES

1. User Authentication:
   - Step 1: User initiates the authentication process through the Android app.
   - Step 2: Fingerprint authentication system validates the fingerprint.
   - Step 3: If successful, the app generates or retrieves an encryption key associated with the user.

2. Note Creation and Encryption:
   - Step 1: User creates a new note through the app.
   - Step 2: App generates a unique encryption key for the note.
   - Step 3: App encrypts the note content using AES with the generated key.

3. Note Retrieval and Decryption:
   - Step 1: User requests to view a specific note.
   - Step 2: App retrieves the encrypted note and associated metadata.
   - Step 3: App uses the stored or regenerated encryption key to decrypt the note content.

4. Secure Communication:
   - Step 1: App communicates securely with any backend servers or databases using HTTPS.
   - Step 2: Data transmitted between the app and server is encrypted using secure transport protocols.

5. Backup and Restore:
   - Step 1: User initiates a backup operation.
   - Step 2: App encrypts the notes and associated data for backup.
   - Step 3: Encrypted backup is stored securely, either locally or on a server.
   - Step 4: During restore, the app decrypts the backup data and restores the notes.

6. Error Handling and Logging:
   - Step 1: App includes error-handling mechanisms at each step of the process.
   - Step 2: Security-relevant events and errors are logged securely for auditing.

7. Security Compliance:
   - Step 1: App follows security best practices and standards.
   - Step 2: Regular security assessments and updates are conducted.

8. Testing:
   - Step 1: Rigorous testing is performed, including unit testing, integration testing, and security testing.
   - Step 2: Test the app under various scenarios, including edge cases and stress testing.

9. Documentation:
   - Step 1: Thorough documentation of the entire implementation, including data formats, encryption algorithms, and protocols used.
   - Step 2: Documentation is regularly

updated to reflect any changes in the implementation.

SHA256

1. User Authentication:
   - Step 1: User initiates the authentication process through the Android app.
   - Step2: The fingerprint authentication system validates the fingerprint.
   - Step 3: If successful, the app proceeds to the next steps.
2. Note Creation and SHA Hashing:
   - Step 1: User creates a new note through the app.
   - Step 2: App generates a unique identifier for the note.
   - Step 3: Note content is combined with the unique identifier.
   - Step 4: App calculates the SHA hash of the combined data.
   - Step 5: The SHA hash is associated with the note for integrity verification.
3. Note Retrieval and Integrity Verification:
   - Step 1: User requests to view a specific note.
   - Step 2: App retrieves the note and associated SHA hash.
   - Step 3: App recalculates the SHA hash of the stored note content.
   - Step 4: Compare the recalculated SHA hash with the stored SHA hash to verify note integrity.

3) PSEUDOCODE

**AES ENCRYPTION**

```
public void encrpyt(View view) {
 try {
 String encrypted =
AESCrypt.encrypt(key.getText().toString()
,message_et.getText().toString());
 ClipboardManager clipboardManager =
(ClipboardManager)getSystemService(CLI
PBOARD_SERVICE);
 ClipData clipData =
ClipData.newPlainText("label",encrypted);
 clipboardManager.setPrimaryClip(clipData);
 key.setText("");
 message_et.setText("");
 message.setText(encrypted);
 Toast.makeText(this,"your message is copied
to clipborad",
Toast.LENGTH_SHORT).show();
 } catch (GeneralSecurityException e) {
 throw new RuntimeException(e);
 }
}
```

**AES DECRYPTION**

```
public void decrypt(View view) {
 try {
 String encrypted =
AESCrypt.decrypt(key.getText().toString(),me
ssage_et.getText().toString());
 ClipboardManager clipboardManager =
(ClipboardManager)getSystemService(CLIPB
OARD_SERVICE);
 ClipData clipData =
ClipData.newPlainText("label",encrypted);
 clipboardManager.setPrimaryClip(clipData);
 key.setText("");
 message_et.setText("");
 message.setText(encrypted);
 Toast.makeText(this,"Your message was
copied to
clipboard",Toast.LENGTH_SHORT).show();
 } catch (GeneralSecurityException e) {
 throw new RuntimeException(e);
 }
```

**SHA 256**

```
btnCreateSHA256.setOnClickListener(v -> {
 String data = tilData.getText().toString();
 if (!TextUtils.isEmpty(data)) {
 result = encryptionManager.getSHA256();
 txtResult.setText(result);
 if
(result.equals(encryptionManager.getSHA256(
))) {
 Log.d("TAG", "same");
 } else {
 Log.d("TAG", "NOT same");
 }
 } else {
 Toast.makeText(Activity2.this, "Field is
empty",
 Toast.LENGTH_SHORT).show();
 }
 });
encryptionManager =
EncryptionManager.getInstance();
 btnEncryptData.setOnClickListener(v -> {
 String data = tilData.getText().toString();
 if (!TextUtils.isEmpty(data)){
 result =
Objects.requireNonNull(EncryptionManag
er.encrypt(data));
 txtResult.setText(result);
 }else {
 Toast.makeText(Activity2.this,"Field is
empty",
 Toast.LENGTH_SHORT).show();
 }
 });
 btnDecryptData.setOnClickListener(v -> {
 if (!TextUtils.isEmpty(result)){
 result =
EncryptionManager.decrypt(result);
 txtResult.setText(result);
 } else {
 Toast.makeText(Activity2.this,"No data to
decrypt",Toast.LENGTH_SHORT).show()
;
 }
 });
```

## 4) MODULE DESCRIPTION

**Fingerprint Authentication:**
Objective: Develop a robust Android note-taking module that ensures data security, integrity, and access control using fingerprint authentication.

**Components:** Fingerprint Authentication: Implement biometric fingerprint authentication for accessing and modifying notes. Utilize Android's fingerprint API to authenticate users before
granting access to notes.
**Note Encryption (AES):** Employ the Advanced Encryption Standard (AES) algorithm to encrypt note content. Encrypt notes upon creation or modification to ensure confidentiality.
**SHA256 Fingerprint Generation:** Utilize the SHA256 hashing algorithm to generate unique
fingerprints (hashes) for each encrypted note. These fingerprints serve as digital signatures, allowing verification of note integrity.
**Access Control:**
a. Authentication: Integrate fingerprint-based authentication to control access to encrypted notes.
b. Decryption: Decrypt notes using AES upon successful fingerprint authentication to allow authorized access.
Security Measures:
• **Key Management:** Securely manage encryption keys and fingerprint data to prevent unauthorized access.
• **Secure Storage:** Ensure encrypted notes and fingerprint data are stored securely within
the application.

<u>FINGERPRINT AUTHENTICATION USING AES:</u>

**ENCRYPTION :**
AES Encryption: Implement the Advanced Encryption Standard (AES) algorithm for securing note content. AES ensures data confidentiality by encrypting the content of notes upon creation or modification.
Key Length Options
Block Cipher
Rounds of Encryption
Standardization and Adoption
**Data Confidentiality:** Utilize AES encryption to protect the sensitive content of notes,

ensuring that unauthorized users cannot access or decipher the information without proper decryption.
**Encryption Key Management:** Maintain secure management of encryption keys used in the AES algorithm to prevent unauthorized decryption and access to the encrypted notes.
**Enhanced Security:** AES encryption significantly enhances the security posture of the Android note-taking application, safeguarding sensitive data from unauthorized access or exposure.

**DECRYPTION :**
**Access Control Integration**: Integrate AES decryption with fingerprint-based access

control mechanisms. Allow decryption of notes only upon successful fingerprint authentication, ensuring access to decrypted content by authorized users.

**Secure Note Retrieval:** Implement AES decryption to retrieve and present the original content of notes, ensuring confidentiality and preventing unauthorized access to sensitive information.

**Fingerprint Verification for Decryption:** Use AES decryption processes within the module to prompt fingerprint verification before decrypting notes, ensuring that only authenticated users with verified fingerprints can access the decrypted content.

**Data Confidentiality Maintenance**: Utilize AES decryption to maintain data confidentiality, ensuring that decrypted notes remain accessible only to authenticated users and protecting sensitive information from unauthorized access.

**Decryption Key Management:** Manage decryption keys securely within the module, allowing authorized decryption while preventing unauthorized access to encrypted notes.

• Key Generation: Creation of Decryption Keys

• Key Storage: Secure Storage of Decryption Keys

• Access Control: Restricted Access to Decryption Keys

• Key Rotation: Periodic Changing of Decryption Keys

• Key Destruction: Secure Disposal or Deletion of Decryption Keys

<u>FINGERPRINT AUTHENTICATION USING SHA256:</u>

SHA256 plays a pivotal role in fingerprint-authenticated, secured Android notes by serving as the backbone for data integrity verification. Here's how it intertwines with this system:

**Fingerprint Generation:** When a note is created or modified, its content undergoes the SHA256 hashing process, generating a unique fingerprint or hash value. This fingerprint encapsulates the essence of the note's content in a secure, fixed-length representation.

**Integrity Check:** Upon accessing a note, the stored SHA256 fingerprint is recalculated based on the current content. This new fingerprint is then compared with the stored one. A match signifies that the note hasn't been tampered with since the last save, ensuring its integrity.

• **Hash Functions**: Like SHA256, hash functions generate unique fixed-size strings (hashes) from data. Verifying integrity involves recalculating the hash and comparing it to the original. Any change in data would result in a different hash, signaling tampering.

• **Parity Checking**: Often used in computer memory, parity bits help detect errors by ensuring the number of set bits in data meets a specific requirement. Any change in data alters the parity bit, indicating an error.

**HASH FUNCTION:**

• Cryptographic Hash Function
• Data Integrity Verification
• Fixed Output Length

**Security Assurance:** SHA256-based fingerprinting assures the note's authenticity and integrity. Even if the note's content is encrypted or secured otherwise, SHA256 ensures swift identification of any unauthorized modifications.

5) IMPLEMENTATION PLATFORM

<u>**AES Algorithm:**</u>

1. **Key Generation:**

• Use the Android Keystore system for secure key storage.
• Generate an AES key using the KeyGenerator class within the Keystore.

2. **Encryption:**

• Obtain plaintext data for encryption (e.g., sensitive information to be protected).
• Use the Cipher class with AES transformation and the generated key to perform encryption.

3. **Decryption:**

• Use the same AES key and Cipher instance to decrypt the data.
• Obtain the plaintext result after successful decryption.
4. **Error Handling:**
• Implement robust error handling mechanisms for biometric authentication and encryption/decryption processes

**SHA256 Algorithm:**
1**. SHA-256 Hah Generation:**
• Capture the fingerprint data and generate a SHA-256 hash
2**. Secure Data Storage:**
• Store the SHA-256 hash securely, considering Android's recommended secure storage practices.
3. **Structure Data Comparision:**
• During authentication, compare the generated SHA-256 hash with the stored reference hash.
4. **Result Handling:**
• Make authentication decisions based on the comparison result

6) PERFORMANCE METRICS AND RUN RESULTS:

Creating a fingerprint-authenticated Android notes application with AES (Advanced Encryption Standard) and SHA-256 (Secure Hash Algorithm 256-bit) involves several components. Let's break down the key aspects, performance metrics, and potential run results for such an application:

**Components:**

**Fingerprint Authentication:**
Utilize the Android Fingerprint API for secure and convenient user authentication.
Implement proper error handling and fallback mechanisms for devices without fingerprint
sensors.
1. AES Encryption:
Use AES for encrypting the notes to ensure confidentiality.
Choose a secure key management strategy, such as the Android Keystore system, to store and protect encryption keys.
2. SHA-256 Hashing:
Apply SHA-256 hashing for integrity verification.
Hash passwords or other sensitive data to generate keys or ensure data integrity.
Performance Metrics:
3. Authentication Speed:
Measure the time it takes for the fingerprint authentication process.
Evaluate the responsiveness of the application during authentication.
4. Encryption/Decryption Speed:
Benchmark the time required for encrypting and decrypting notes using AES.ptimize cryptographic operations to minimize any noticeable delays.
5. Resource Usage:
Monitor CPU and memory usage during fingerprint authentication and cryptographic operations.Ensure the application doesn't excessively consume system resources.
6. Key Generation and Management:
Assess the efficiency of key generation and management using the Android Keystore or another secure method.Verify that keys are securely stored and managed to prevent unauthorized access.
7. Hashing Performance:
Measure the time it takes to generate SHA-256 hashes for data integrity.Confirm that hashing operations do not introduce significant performance overhead.
**Run Results:**
8. User Experience:
Successful and smooth fingerprint authentication.Notes should be seamlessly encrypted and decrypted without noticeable delays.
9. Security:
Verify that encrypted notes remain secure and are not vulnerable to attacks.Ensure that the implementation follows best practices for cryptography and key management.
10. Error Handling:
Properly handle authentication failures, encryption errors, and other potential issues.

Display clear error messages to users when needed.
11. Compatibility:
Confirm the application works across a range of

Android devices.Test on various Android versions to ensure compatibility.

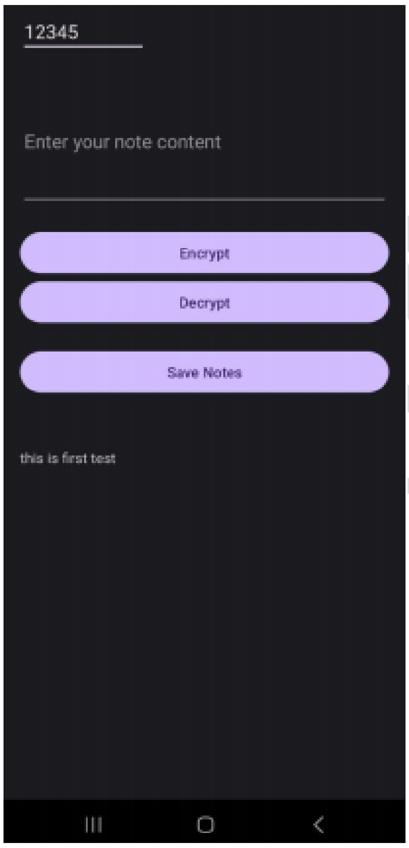
12. Battery Impact:

Evaluate the impact of the application on device battery life.Optimize the application to minimize unnecessary resource usage.

13. Scalability:

Assess how the application performs as the number of notes or the amount of data increases.Ensure scalability for a growing user base.By focusing on these components, performance metrics, and run results, you can create a robust and secure Android notes application with fingerprint authentication, AES encryption, and SHA-256 hashing. Regular testing and updates will be essential to address potential security vulnerabilities and improve overall performance

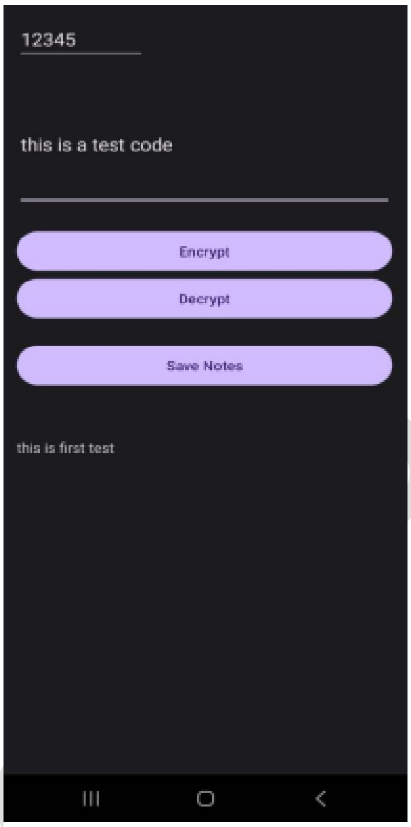
## 7) DEMO SCREENSHOTS & IMPLEMENTATION



12345

Enter your note content

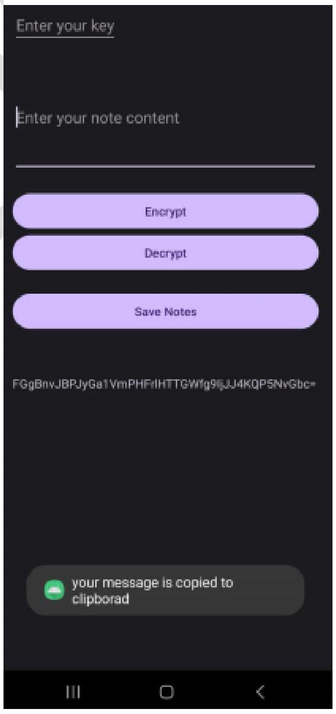
this is first test



12345

this is a test code

this is first test

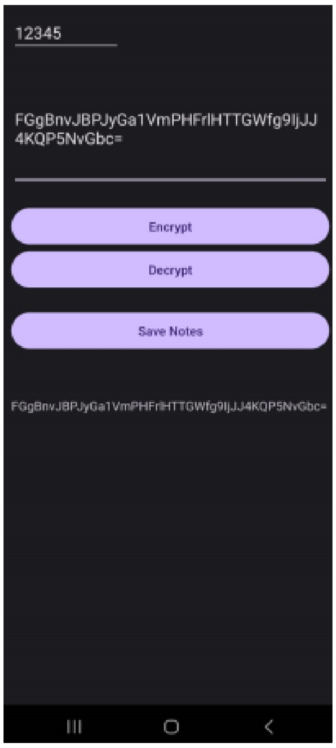


Enter your key

Enter your note content

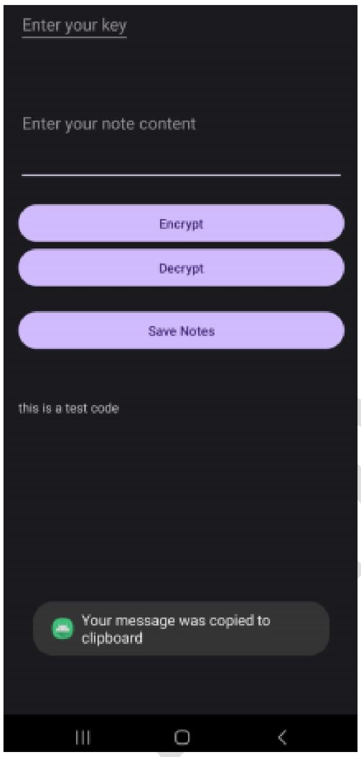
FGgBnvJBPJyGa1VmPHFrlHTTGWfg9ljJJ4KQP5NvGbc=

your message is copied to clipborad


12345

FGgBnvJBPJyGa1VmPHFrlHTTGWfg9IjJJ4KQP5NvGbc=

FGgBnvJBPJyGa1VmPHFrlHTTGWfg9IjJJ4KQP5NvGbc=

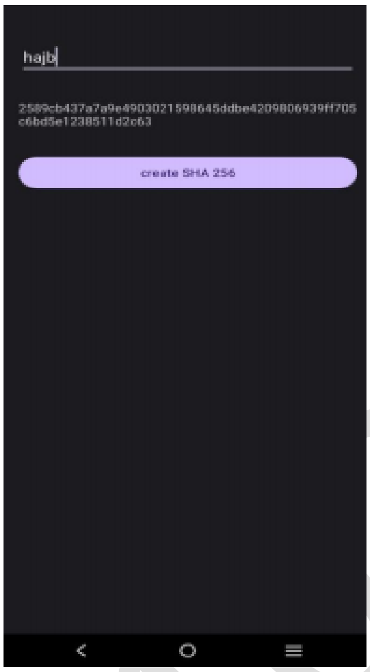

Enter your key

Enter your note content

this is a test code

Your message was copied to clipboard

SHA 256


hajb

2589cb437a7a9e4903021598645ddbe4209806939ff705c6bd5e1238511d2c63

## VI. CONCLUSION

The integration of fingerprint authentication, AES encryption, and SHA-256 hashing in securing Android notes represents a sophisticated and comprehensive approach to data security.

By leveraging biometric authentication through fingerprint recognition, the system ensures a seamless and user-friendly experience while fortifying access control. The implementation of the AES encryption algorithm adds an additional layer of security by transforming sensitive information into a secure, unreadable format, mitigating the risk of unauthorized access.Furthermore, the utilization of SHA-256 as a hashing algorithm plays a pivotal role in maintaining data integrity. The hashed representation of data provides a fixed-size output that is unique to each set of information, serving as a digital fingerprint. This enhances the system's ability to detect any tampering or unauthorized alterations to stored notes.

The use of AES encryption aligns with industry standards for securing sensitive information, providing confidentiality and preventing unauthorized individuals from deciphering the content even if they gain access to the stored data.

SHA-256's cryptographic strength reinforces the overall security posture by creating a hash
that is computationally infeasible to reverse. This ensures the integrity of the stored notes,
allowing users to trust the accuracy and authenticity of their information.In conclusion, the implementation of fingerprint-authenticated, AES-encrypted, and SHA-256-
secured Android notes exemplifies a state-of-the-art approach to mobile data protection.

This comprehensive security framework not only safeguards user privacy and confidential
information but also provides a user-friendly and reliable solution for securing Android notes in the face of evolving cybersecurity challenges.

## VII. REFERENCES:

https://www.ijsr.net/get_count.php?paper_id=ART20171631

https://inha.elsevierpure.com/en/publications/security-analysis-and-improvement-of-fingerprint-authentication-f

https://ieeexplore.ieee.org/document/9104992

https://link.springer.com/referenceworkentry/10.1007/978-1-4419-5906-5_880

https://link.springer.com/chapter/10.1007/11507840_32

https://www.ijstr.org/paper-references.php?ref=IJSTR-1215-13106

https://www.ijsr.net/archive/v10i5/SR21405101230.pdf

https://koreascience.kr/article/JAKO202105254964213.page

https://ijettjournal.org/archive/ijett-v10p276

https://dl.acm.org/doi/10.1147/sj.403.0614

https://ieeexplore.ieee.org/document/7185307

https://sites.google.com/site/aronowitzh/publicationlist

(PDF) Fingerprint: A Unique and Reliable Method for Identification (researchgate.net)

https://www.researchgate.net/publication/286663122_Multi-modal_biometrics_for_mobile_authentication

https://pubmed.ncbi.nlm.nih.gov/17299214/

https://www.sciencedirect.com/science/article/abs/pii/S0031320305001445