**Project Architecture: Secure Image Steganography with AES Encryption**

pgsql

CopyEdit

+--------------------+ +--------------------------+

| User Input Module | | User Input Module |

|--------------------| |--------------------------|

| - Message (Text) | | - Stego Image |

| - Password | | - Password |

| - Cover Image | +-----------+--------------+

+----------+---------+ |

| |

| v

v +-----------------------------+

+----------------------+ | Data Extraction Module |

| Encryption Module | |-----------------------------|

|----------------------| | - Extract binary from LSBs |

| - AES Encryption | | - Detect EOF marker |

| - SHA-256 Key Derive | +-------------+---------------+

+----------+-----------+ |

| v

v +------------------------------+

+----------------------+ | Decryption Module |

| Base64 Encoding | |------------------------------|

+----------+-----------+ | - Base64 Decode |

| | - AES Decryption |

v | - Retrieve original message |

+----------------------+ +--------------+---------------+

| LSB Embedding Module | |

|----------------------| v

| - Embed bits in LSBs | +------------------------------+

| - Use EOF delimiter | | Output Module |

+----------+-----------+ |------------------------------|

| | - Display extracted message |

v +------------------------------+

+----------------------+

| Stego Image File |

+----------------------+

**🧩 Component Descriptions**

**1. User Input Module**

* Collects:
  + Text message to hide
  + Password for encryption
  + Cover image file
  + (On the receiver side, collects the stego image and password)

**2. Encryption Module**

* Converts password into a **256-bit key** using **SHA-256**
* Encrypts message using **AES (EAX mode)** for security and integrity

**3. Base64 Encoding Module**

* Ensures encrypted data is text-safe for LSB embedding
* Converts encrypted binary into a Base64 string

**4. LSB Embedding Module**

* Converts base64 text to binary
* Hides binary data inside the **least significant bits** of each RGB pixel
* Appends **EOF marker** (11111110) to signal the end of the message

**5. Stego Image File**

* The final image that contains the encrypted message hidden in it

**6. Data Extraction Module**

* Reads the LSBs of each pixel
* Reconstructs the hidden binary string until the **EOF** marker is found

**7. Decryption Module**

* Converts extracted Base64 to binary
* Decrypts the ciphertext using AES (with the same password)

**8. Output Module**

* Displays or returns the original hidden message

**🛡️ Cybersecurity Strengths in Architecture**

* **Confidentiality**: Achieved with AES encryption
* **Obfuscation**: Done with steganography (LSB)
* **Password Protection**: Prevents unauthorized decryption
* **Data Integrity**: Using AES-EAX mode ensures message is not tampered