The **Secure Image Steganography with AES Encryption** project combines two key areas: **Cryptography** and **Steganography**. Here's a breakdown of the **algorithms used**:

**🔐 1. AES Encryption (Advanced Encryption Standard)**

**✅ Purpose in the project:**

Used to **encrypt the message** before hiding it in the image. This ensures **confidentiality**, even if someone extracts the hidden data.

**🔎 Algorithm Details:**

* **Type**: Symmetric key block cipher
* **Key Size**: 256-bit key (derived from password using SHA-256)
* **Mode Used**: EAX (Authenticated encryption mode; provides both confidentiality and integrity)
* **Libraries Used**: pycryptodome

**📋 Steps:**

1. Convert password to 256-bit key using **SHA-256**
2. Use **AES-EAX** to encrypt the message
3. Encode the result using **Base64** for safe embedding into the image

**🖼️ 2. LSB Steganography (Least Significant Bit Algorithm)**

**✅ Purpose in the project:**

Used to **embed the encrypted message** into the least significant bits of the image pixels.

**🔎 Algorithm Details:**

* **Type**: Substitution steganography
* **Embedding Capacity**: One bit per color channel (RGB = 3 bits per pixel)
* **Encoding**: 8-bit binary representation of each character
* **EOF Marker**: 11111110 used to signal the end of the message

**📋 Steps:**

1. Convert the encrypted message to a binary string
2. Replace the **LSB** of each pixel's RGB channel with one bit of the message
3. Stop once the entire message (and EOF marker) is embedded
4. To extract, read LSBs until the EOF pattern is found

**🔁 3. Base64 Encoding (for binary safety)**

**✅ Purpose in the project:**

Used to **safely encode encrypted binary data** into a string format that can be embedded into an image.

**🔎 Why needed:**

* Encrypted data may contain special or non-printable characters
* Base64 ensures all bytes are represented as text characters (A-Z, a-z, 0-9, +, /)

**✅ Summary Table**

| **Component** | **Algorithm** | **Role in Project** |
| --- | --- | --- |
| Encryption | **AES (EAX mode)** | Encrypt the secret message |
| Hashing | **SHA-256** | Convert password into a secure key |
| Embedding | **LSB Algorithm** | Hide binary data in image pixels |
| Encoding | **Base64** | Convert binary to text for embedding |