**1. Introduction**

The system is designed to enhance **confidential and covert communication** by combining two powerful techniques: **cryptography** and **steganography**. While encryption protects the content of the message, steganography conceals the existence of the message itself.

This dual-layered security model is useful in cybersecurity, where the goal is to **protect sensitive data from both interception and detection**.

**🧠 2. Existing System**

Traditional communication systems rely on **encryption** to secure data. However, even when encrypted, the presence of sensitive content can attract attackers or raise suspicion.

Similarly, **basic steganography** hides data without encryption, making it vulnerable if extracted.

**🔴 Limitations of the Existing Systems:**

* Lack of **confidentiality** in pure steganography.
* **Suspicious communication patterns** in encrypted-only systems.
* Limited resistance to **attacks or steganalysis**.

**💡 3. Proposed System**

The proposed system enhances data security by first **encrypting the message using AES**, and then **hiding the encrypted data inside an image using LSB steganography**.

**✅ Key Advantages:**

* Ensures **confidentiality** through encryption.
* Achieves **concealment** using steganography.
* **Password-protected access** for both hiding and extraction.
* Can bypass conventional detection systems (e.g., firewalls, sniffers).

**⚙️ 4. System Objectives**

* 🔐 To securely **encrypt** messages using AES algorithm.
* 🖼️ To **embed and extract** data using Least Significant Bit (LSB) steganography.
* 🔎 To **prevent unauthorized access** through password-based key derivation.
* 📤 To allow secure **image-based transmission** of hidden information.

**📦 5. System Features**

| **Feature** | **Description** |
| --- | --- |
| AES Encryption | Encrypts the message using a symmetric key |
| SHA-256 Key Generation | Converts password into a 256-bit key |
| LSB Image Embedding | Hides encrypted message in image pixels |
| Base64 Encoding | Ensures encrypted binary is embedding-safe |
| EOF Marker Detection | Helps accurately extract hidden data |
| Secure Message Extraction | Allows only authorized users to decrypt content |

**🏗️ 6. Modules of the System**

1. **User Input Module**: Accepts message, image, and password.
2. **Encryption Module**: Performs AES encryption.
3. **Steganography Module**: Embeds/extracts data in image using LSB.
4. **Decryption Module**: Reconstructs original message from the encrypted payload.
5. **Output Module**: Displays the recovered message.

**🧪 7. Feasibility Study**

**✅ Technical Feasibility**

* Uses widely supported Python libraries (Pillow, pycryptodome).
* Works on all major OS (Windows, Linux, macOS).

**✅ Operational Feasibility**

* Easy-to-use CLI or GUI interface.
* Can be integrated into secure messaging platforms.

**✅ Economic Feasibility**

* Open-source libraries reduce cost.
* Can be deployed without additional hardware.

**📌 8. Applications**

* Secure messaging for **cybersecurity analysts or whistleblowers**.
* **Stego-forensics** or anti-forensics research.
* Military or defense communication systems.
* Digital watermarking and copyright protection.