LSB Steganography Project - Extended Interview Q&A

1. What is steganography?

Steganography is the practice of concealing a message, file, image, or video within another file or message. The goal is to hide the very existence of the communication, making it harder to detect.

2. How is steganography different from cryptography?

Cryptography encrypts the content to make it unreadable without a key, but its presence is obvious. Steganography hides the message within another medium so that it doesn't attract attention in the first place.

3. What are some common applications of steganography?

Common uses include covert communication, copyright protection (watermarking), digital signatures, and in some cases, malicious activity like hiding malware.

4. What is the Least Significant Bit (LSB) technique in image steganography?

LSB is a method where the least significant bits of the image's pixel values are modified to store secret data. It works well in images because changing the LSB does not noticeably affect the image's appearance.

5. How do you convert a message into a format that can be embedded into an image using LSB?

The message is first converted into a binary representation. Each bit is then inserted into the LSB of a pixel's RGB value.

6. Why did you choose Python for the initial implementation?

Python offers simplicity and powerful libraries like Pillow for image manipulation. This allowed for fast development and easy debugging.

7. How did you read and modify pixel values in your Python implementation?

I used the Pillow library to load and access pixel data. I manipulated the RGB values at the bit level

and used the modified values to save the new image.

8. How do you handle decoding or extracting the message?

The decoder reads LSBs from the image pixels, reassembles the bits into bytes, and then converts the bytes back into characters until a delimiter or length limit is reached.

9. What image formats are best suited for LSB steganography?

Lossless formats like PNG or BMP are ideal because they do not compress pixel data. Formats like JPEG use lossy compression, which can corrupt the hidden data.

10. How did you ensure the hidden message didnt degrade image quality?

I limited the number of bits modified and only used one color channel per pixel to minimize visual changes.

11. What limitations did you face in your implementation?

The method is not robust against image editing, compression, or resizing. Also, the message is not encrypted, so if discovered, it's readable.

12. How can you improve the security of your steganography tool?

By encrypting the message before embedding and using random pixel selection or using more advanced embedding techniques like DCT or spread spectrum.

13. How can steganography be detected or broken?

Through steganalysis analyzing statistical anomalies in pixel values or detecting patterns in LSBs that wouldnt naturally occur in real images.

14. Can this method be used to hide files instead of just text?

Yes, but it requires converting the file to binary and often compressing or encoding it to fit within the image capacity.

15. What is capacity in the context of image steganography?

It refers to the maximum amount of data that can be hidden in an image without significant

degradation or detection.

16. What ethical concerns exist around steganography?

While it can be used for privacy and security, it can also be misused for hiding illegal content, communicating malicious intent, or spreading malware undetected.

17. How would you build a GUI for your tool in the future?

In Python, using Tkinter or PyQt. In Java, using Swing or JavaFX, which would allow users to load an image, enter a message, and perform encoding/decoding with a visual interface.

18. Why might you re-implement this project in Java?

Java offers better performance for large-scale applications, better structure for maintainability, and is easier to integrate with Android or enterprise applications.

19. How would you extend your project to support audio or video steganography?

By modifying the LSBs in audio samples or video frames. This would require handling formats like WAV for audio or MP4 for video and understanding their data encoding.

20. What did you learn from this project?

I learned practical image processing, bit-level manipulation, and the importance of data security. It also strengthened my skills in Python and cybersecurity fundamentals.