Question 1 Solution:

In network forensics, encryption is often seen as a double-edged sword: it protects privacy but creates a “black box” for investigators.

A key forensic technique to recover encrypted network data is *Memory Forensics* (RAM analysis).

It involves capturing a snapshot (dump) of a computer’s volatile memory while it is still powered on and then analysing that snapshot for forensic artefacts. The investigators capture a system’s RAM to extract encryption keys used by *protocols* such as *TLS, SSH, VPNs, or Tor.* Since encrypted traffic must be decrypted in memory for processing, session keys can sometimes be recovered and used to decrypt previously captured network traffic.

Where it can be used: This technique can be applied when the system is powered on and actively handling encrypted traffic. It is instrumental during incident response, where rapid live acquisition is possible.

Where it cannot be used: It cannot be used if the system is powered off, memory is overwritten, or keys are protected by hardware security mechanisms (e.g., TPM or secure enclaves).

Ease of Application:

In Incident Response, it is moderately practical with specialist tools (e.g., Volatility) and trained personnel, and in criminal investigations, it is harder to apply due to legal constraints and the difficulty of timely live acquisition.

Course Reference:

This technique is relevant to the Week 6 Network and Internet Forensics slides.

Question 2 Solution:

In network forensics and online safety, a widely used technique to identify harmful images is known as **Perceptual Hashing** (sometimes called "Robust Hashing").

Unlike cryptographic hashes, perceptual hashes generate similar outputs for visually similar images, even if the files differ in size, format, or compression. The hash is computed by reducing image resolution or applying transforms such as the Discrete Cosine Transform and then quantising visual features into a compact fingerprint

Deployment on Social Media:

On social media platforms, perceptual hashing is deployed as follows: when a user uploads an image, the platform computes its perceptual hash and compares it with a database of hashes of known harmful images (e.g., CSAM or extremist content). Similarity is measured using *Hamming distance*, and if the distance is below a predefined threshold, the image is blocked or flagged for human review. This enables large-scale, automated proactive moderation

Limitations:

Perceptual hashing suffers from *false negatives*, where modified harmful images (cropped, rotated, recoloured) are not detected, and *false positives*, where visually similar but benign images may be flagged. In addition, perceptual hashes encode visual features, meaning machine-learning techniques can partially reconstruct original images from hashes, creating privacy risks

Ethical Consideration:

Ethically, perceptual hashing raises concerns about *mass surveillance*, lack of transparency in hash databases, and potential framing attacks if benign images are inserted into hash lists. False positives can also cause serious harm to innocent users. Therefore, strong oversight, transparency, and proportional use are necessary when deploying this technology

Course reference:

This technique is mentioned in Week 7, Perceptual hashing and steganography slides.

Question3 Solution:

In network forensics, each TCP/IP layer provides different types of evidence that can support an investigation.

- **Application Layer (HTTP)**
  A relevant artefact is the HTTP User-Agent string in HTTP headers. It can identify the browser, operating system, or automated tool used by a suspect, helping link activity to a specific device or malware family. This is shown in the HTTP request analysis in the Week 6 slides.
- **Transport Layer (TCP/UDP)**
  Port numbers and TCP flags (SYN, FIN, RST) reveal the nature of a connection. For example, repeated SYN packets indicate port scanning, while RST flags may indicate forced connection termination. Port-protocol relationships are discussed in the default ports slide.
- **Internet Layer (IP)**
  The IP address and TTL field are forensically useful. IP addresses help identify communicating hosts, while TTL values can help estimate hop distance and sometimes infer the sender’s operating system. IP header fields are shown in the IP datagram slide.
- **Link Layer (Ethernet)**
  The MAC address identifies the physical network interface. Investigators can map an IP address to a MAC address to locate a specific device within a local network, as shown in the Ethernet frame structure slide.

Course Reference :

This information is mentioned in the Week 6 lecture slides (9 - 16).

Question 4 Solution:

For LSB steganography, the best pictures are **high-detail natural photographs(high entropy/noisy image)** such as landscapes or textured scenes. These images have naturally random LSB bit-planes, so small LSB modifications are visually imperceptible. This makes the hidden encrypted data statistically and visually difficult to detect.

Simple images with large flat areas (e.g., cartoons or logos) are unsuitable because LSB changes introduce detectable patterns. A picture that is **visually complex**, **highly textured**, or **"noisy"** is the best choice for LSB steganography.

Question 5 Solution:

Think of onion routing as a specialized security system designed to hide your digital footprint by separating your identity (your IP address) from what you are doing online. It uses public key cryptography and relies on a journey through a decentralized network using a “layered” approach.

The core principle: Layered Encryption: The system earns its name because it protects data in layers, much like an onion.

- **Wrapping the Data**: Your computer acts as the first architect, wrapping your data in multiple layers of encryption. Each layer is intended for a specific "stop" or node along the path—usually an Entry, Middle, and Exit relay.
- **The "Peeling" Process**: As your data travels, each relay on the circuit uses its own unique private key to "peel" back exactly one layer of encryption.
- **Strategic Ignorance**: This setup ensures that no single participant knows the whole story. The Entry node knows **who** you are but has no idea **where** you are going; conversely, the Exit node knows your **destination** but has no clue **who** sent the request.

The hidden rule: Node independence:

For this privacy shield to work, there is an implicit requirement: **node independence and diversity**. The security of the entire circuit rests on the assumption that an investigator cannot "see" or control both the start (Entry) and the end (Exit) of the path at the same time.

“Operation Liberty Lane”

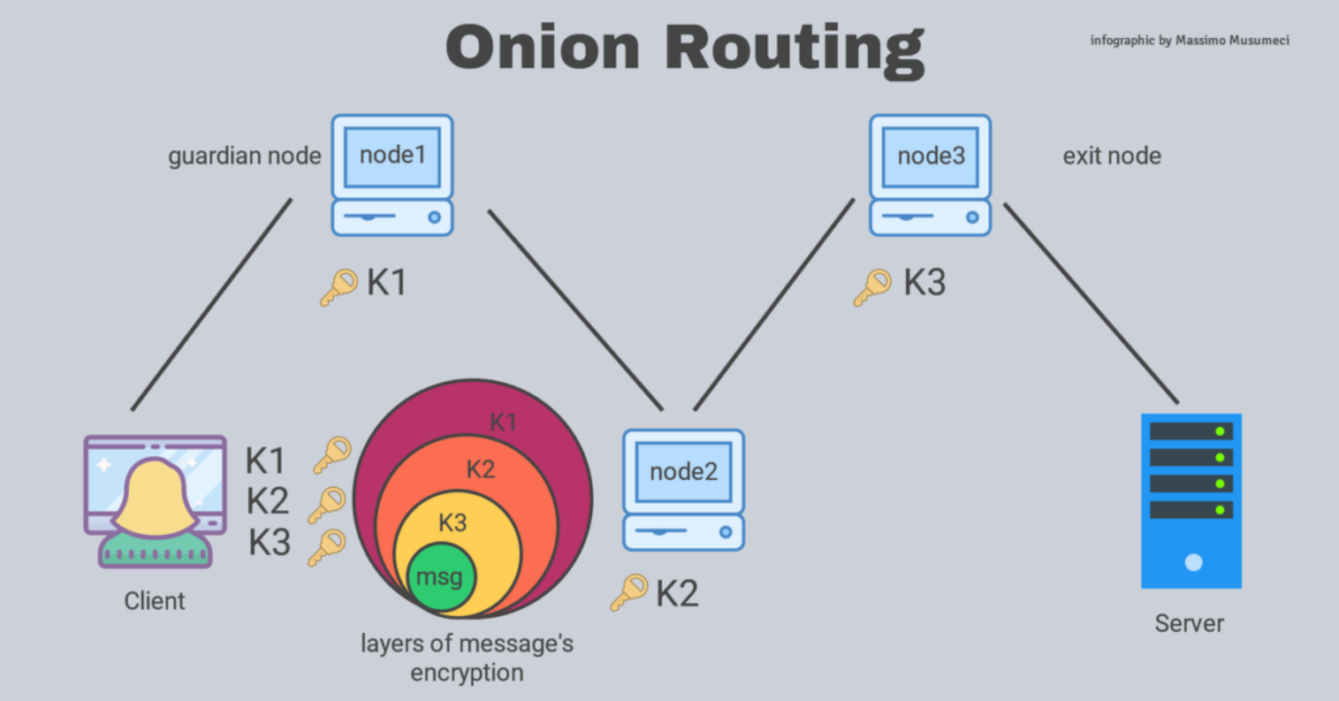
In the real-world case of **Operation Liberty Lane**, law enforcement allegedly bypassed these protections by exploiting **Traffic Correlation**, also known as a Timing Analysis. Rather than trying to "crack" the encryption, they focused on the patterns of the traffic itself.

By allegedly controlling or monitoring a massive portion of the network's relays—such as the "KAX17" cluster—investigators were able to perform a digital stakeout:

- **Watching the Entrance**: They monitored the exact timing and the size of data packets entering the network from a specific user at a "malicious" guard node.

- **Watching the Exit**: They then looked for a matching pattern of data leaving the network at an exit node or the onion service's guard.
- **The Connection**: By "matching the clocks" between these two points, investigators could definitively link a specific person's identity to their supposedly anonymous destination.

This was reportedly made possible through a combination of spinning up their own malicious relays and collaborating with ISPs to watch user connections in real-time.



Question 6 Solution:

Volatility includes two different commands for listing processes because *Windows process information can be manipulated or partially hidden by malware*, and no single method is sufficient to reliably identify all processes from memory.

The two plugins rely on different internal mechanisms, allowing investigators to cross-validate results and detect stealthy malware.

pslist — Kernel process list traversal

pslist extracts processes by walking the ActiveProcessLinks doubly-linked list maintained by the Windows kernel. Each running process has an EPROCESS structure, and Windows links these together so the operating system can manage scheduling and resources. It shows what the OS believes in running

Inner-working:
It starts from the known list head in kernel memory and follows forward and backward pointers to enumerate processes.

psscan — Signature-based memory scanning

psscan does not rely on the kernel list. Instead, it performs a raw scan of physical memory, searching for byte patterns that match the structure of an EPROCESS object**.** It can find:

- Hidden or unlinked malware processes
- Processes that started and exited quickly
- Remnants of terminated processes

Inner-working:
It identifies processes based on memory signatures, even if they are unlinked or partially overwritten.

Volatility provides both commands to overcome malware evasion techniques and to ensure reliable forensic reconstruction.

pslist reflects the OS view of running processes, while psscan reflects the memory reality. Together, they enable the detection of hidden, terminated, or manipulated processes, which is essential for incident response and malware analysis.

Course reference:

This behaviour and investigative workflow are described in Week 10 Memory Forensics slides**.**