CATALOGUE

EPFL

104

Lecturers: B. Ford, P. Borsò
CS-438 Decentralized Systems Engineering
18.01.2024
2 hours

SCIPER : **999999** Room : **PO1** Signature :

Do not turn the page before the start of the exam. This document is double-sided, has 16 pages, the last ones possibly blank. Do not unstaple.

- Place your student card on your table.
- **No other paper materials** are allowed to be used during the exam.
- Using a **calculator** or any electronic device is not permitted during the exam.
- Grading:
  - for the **multiple choice** questions, **each answer** is awarded :
    - ☒ This answer is correct : if checked, +1 point, if not -1 point
    - □ This answer is NOT correct : if checked, -1 point, if not +1 point
    - If you give no answer **to the question**, 0 point
  - for the **single choice** questions, we give :

    +3 points if your answer is correct,
    0 points if you give no answer or more than one,
    −1 points if your answer is incorrect.
- Use a **black or dark blue ballpen** and clearly erase with **correction fluid** if necessary.
- If a question is wrong, the teacher may decide to nullify it.

Respectez les consignes suivantes | Observe this guidelines | Beachten Sie bitte die unten stehenden Richtlinien

choisir une réponse | select an answer Antwort auswählen	ne PAS choisir une réponse | NOT select an answer NICHT Antwort auswählen	Corriger une réponse | Correct an answer Antwort korrigieren
☒ ☑ ■	□	□ □

ce qu'il ne faut **PAS** faire | what should **NOT** be done | was man **NICHT** tun sollte

▨ ☒ ○ ⊡ ▨ □

You are designing the communication protocol used by a network of software agents. Each agent is connected to a sensor platform, collecting sensitive data. There may be between 100 and 200 agents in use at any point in time. The main application-level performance criteria is the propagation of sensor values (a few hundred bytes every 5 seconds) with minimal delay to all agents, as values only stay relevant for decision-making during 2 seconds and their correctness is paramount.

This is the context for the following 2 questions.

Question [MCQ-49] Which communication protocols would be appropriate foundations for the application presented above?

■ Rumor mongering
□ BitTorrent
□ DC-nets
□ Mix-nets
□ Tor

Question [MCQ-50] Which quality measures would likely be relevant to assess your communication protocol in the context of the application presented above?

■ Percentage of sensor values delivered to all agents within 1 s
■ Variance in the time to deliver sensor values to the agents
■ Median time to deliver sensor values to the agents
■ Percentage of agents obtaining > 50% of the sensor values within 1 s
□ Time to deliver a sensor value to the last agent

Question [MCQ-01] You are implementing a gossip algorithm in the presence of Byzantine adversaries. What properties do message IDs need to have to ensure the algorithms function correctly?

■ They need to use a cryptographic hash function.
■ They need to be computed based on the message data.
□ They should be generated based on the source's IP address.
□ They need to be monotonically increasing.
□ They need to have a fixed length.

Question [MCQ-02] What problems are death certificates trying to address in gossip algorithms ?

■ Minimizing data storage requirements
■ Preventing the reintroduction of outdated information
■ Decreasing unnecessary bandwidth usage
□ Minimizing communication retries
□ Attesting to the failure/exclusion of a node

Catalogue

Question [SCQ-03] Recall that social media applications need to have globally unique, sortable message IDs, which led to the creation of Snowflake ID. In what order, from most- to least-significant bit, are the following components found in a Snowflake ID to achieve these properties?
(This is a single-choice question)

■ timestamp, host id, counter
□ timestamp, counter, host id
□ counter, timestamp, host id
□ counter, host id, timestamp
□ host id, timestamp, counter

Question [MCQ-04] You are developing a distributed, resilient database system under a crash-recovery failure model. Inspired by existing solutions, you've decided that your replicas will carry out node metadata propagation and failure detection through a gossiping algorithm. Which of the following statements would be accurate regarding the system design?

■ Timestamps are not necessary in the gossip messages for the system's correct functioning.
□ Gossiping removes the need for consensus to exclude a failed node from the database cluster.
□ Death certificates are necessary to ensure failure detection works correctly.
□ For performance reasons, the gossip communication should be separate from application-level traffic.
□ Anti-entropy (pull-based gossip) can't be used as it propagates messages too slowly for the system's needs.

Question [MCQ-05] What are the advantages of BubbleStorm over expanding rings search in systems with a large number of nodes?

■ BubbleStorm searches incur less total traffic to satisfy a search query.
■ BubbleStorm searches incur less latency in a search query.
□ BubbleStorm searches always succeed eventually while expanding-ring searches may not.
□ BubbleStorm requires less storage.
□ BubbleStorm offers Byzantine fault tolerance.

Question [MCQ-06] What are the advantages of using Chord over BubbleStorm-based search?

■ It offers lower asymptotic search complexity.
□ It uses a global index to store all keys, reducing search latency.
□ It replicates data across all nodes to ensure high availability and fault tolerance.
□ It decouples the search process from the network topology.
□ It offers better resilience to network churn.

Question [SCQ-07] Recall that the success rate of a BubbleStorm query is $1 - e^{-\frac{d \cdot q}{n}}$, where d and q are the data and query replication factors, respectively. Assume that half of the nodes of a BubbleStorm network fail following a random uniform distribution. In the immediate aftermath, a node issuing a query is not yet aware that this massive failure has occurred, but the queries it issues via random walks traverse only non-failed nodes, because each node immediately knows which of its peers have failed. How does the failure impact the querying node's success rate?
(This is a single-choice question)

■ The success rates stays the same.
□ The success rate increases.
□ The success rate decreases.

Question [MCQ-08] Which properties of Ad-hoc On-demand Distance-Vector (AODV) routing make it particularly well-suited to usage in low-power, embedded systems ?

■ AODV requires minimal path maintenance traffic.
■ Route discovery occurs only when needed.
■ AODV requires limited memory resources.
□ It produces loop-free routing paths.
□ Nodes retain complete network topologies.

Question [MCQ-09] What is the purpose of the finger table in Chord?

■ To help locate a node responsible for a given key
□ To store the DHT's key/value data efficiently
□ To maintain a database of IP addresses for all peers
□ To manage the peer IDs such that each node has a unique identifier
□ To encrypt the data for confidentiality

Question [MCQ-10] How does a typical search query work in the Chord paper?

■ The query is forwarded to a successor node that is closer to the target key.
□ The query is matched against a replicated, global index, and then forwarded directly to the correct node.
□ The query is sent to a randomly-selected rendezvous node to achieve O(sqrt(n)) query complexity.
□ The query is processed along parallel paths to ensure resiliency.
□ The query is gossiped using rumour mongering.

Question [MCQ-11] Where is new data stored in a content-addressed Chord DHT?

■ A new item is stored at the node whose identifier precedes the item's hash in the identifier space.
■ Statistically, new data items will be evenly distributed among all nodes.
□ The data is stored at a randomly selected node to balance the load.
□ The data is assigned to the node with the most available storage capacity.
□ Chord splits the data into fixed-size blocks, each block's location depending on the block's hash.

Question [MCQ-12] How does Chord guarantee lookup correctness in the face of churn?

■ Nodes periodically update their finger table.
■ The Chord-based application may replicate data on N nodes succeeding a key.
■ The finger table remains substantially correct in the presence of simultaneous, uniformly distributed node failures.
□ Nodes replicate their finger table on the N successive nodes.
□ Nodes periodically advertise their presence to immediate predecessors as part of the stabilize operation.

Question [MCQ-13] An adversary is trying to censor some data on a content-addressed network by mounting an eclipse attack. What would increase the cost to the adversary?

- ■ Executing parallel routing lookups along multiple disjoint paths
- ■ Requiring proof-of-work for nodes joining the DHT
- □ Periodically removing inactive nodes in the DHT
- □ Limiting the number of successor nodes in the DHT
- □ Using static, non-changing neighbor lists for routing

Question [MCQ-14] Which of the following statements accurately reflect the modern understanding between consistency (C) and availability (A) under partitions (P) in distributed systems?

- ■ The choice between consistency and availability can be user-specific.
- ■ The choice between consistency and availability is only necessary under partition.
- ■ Operationally, a partition is a timeout on communication.
- □ Achieving partition tolerance requires a mechanism for all nodes to become aware of a network partition.
- □ AP systems using CRDTs don't need recovery and compensation mechanisms after a partition.

Question [MCQ-15] You are tasked with developing a new online banking application, supporting common operations. As you expect a high peak loader, you plan to replicate the application across 10 servers while keeping it strongly consistent. Which of the following are achieved by replicating data across multiple nodes?

- ■ Protection against complete data loss in the case of single-node failure.
- ■ Increased data availability by tolerating crash failures of a subset of nodes.
- ■ Improved system scalability by balancing read requests across nodes.
- □ Improved system scalability by distributing writes across all nodes.
- □ Increased network partition tolerance, allowing uninterrupted operations across all nodes.

Question [MCQ-16] What are the properties of leaderless Paxos, with every node acting simultaneously as a Proposer, Acceptor and Learner?

- ■ Liveness can break if multiple proposers repeatedly try to propose simultaneously.
- □ The replicated state machine will behave correctly even in the presence of equivocating nodes.
- □ Safety can break if the acceptors crash.
- □ Leaderless Paxos requires a view change protocol.
- □ We need a Pre-Prepare phase to handle the concurrent proposals.

Question [MCQ-17] Assume that you are dealing with a system relying on leader-based Paxos consensus. Which of the following statements are true about the consensus protocol?

- ■ It can temporarily lose liveness if the leader crashes.
- ■ Its performance is limited by the leader.
- ■ Consensus validity is always guaranteed.
- ■ It can achieve liveness due to the absence of contention.
- □ It can lose safety if the leader crashes.

Question [MCQ-18] You are tasked with enhancing the performance of a Paxos-based consensus system by making transaction data dissemination more efficient. You decide to switch to gossip-based communication, instead of broadcast, for data dissemination. Which of the following are benefits you expect from using a gossip-based push-pull protocol in this way?

■ More scalable to a large number of replicas

■ Lower total network traffic for data dissemination

■ Improved ability to route around transient network failures between two nodes

□ Improved latency in the common-case deployment of $n = 3$ replicas

□ Guaranteed delivery of messages in a predictable amount of time

Question [MCQ-19] BitTorrent is a robust protocol that has stood the test of time. Which of the following statements are correct about its trust model and security properties?

■ Torrent indexers/search engines are trusted.

■ Data integrity is guaranteed against malicious trackers.

■ Data integrity is guaranteed against malicious peers.

■ BitTorrent is subject to Sybil attacks.

□ Attackers can never consume more than their fair share of honest peers' resources.

Question [MCQ-20] Which of the following approaches are employed by the BitTorrent protocol to optimize the distribution and download speed of files in a peer-to-peer network?

■ Use of an "anti-snubbing" heuristic to accelerate the discovery of better peers.

■ The "rarest first" strategy to prioritize downloading of the least available pieces of the file across the network.

■ Use of a "tit-for-tat" strategy to incentivize peers to upload.

□ Use of "predictive caching" to prefetch frequently accessed files across nodes.

□ The "lazy loading" technique to defer loading of file parts until requested by multiple peers.

Question [MCQ-21] Recall that BitTorrent splits files into pieces, and pieces into sub-pieces for transfer. How are file pieces and sub-pieces selected for download, in order to optimize resource utilization and total download time?

■ In endgame mode, all missing sub-pieces are requested from all peers.

■ BitTorrent clients use randomness in piece selection.

■ The download prioritizes obtaining full pieces before downloading other pieces.

□ The file download starts with the rarest file piece first.

□ The BitTorrent client considers peers' upload speed as part of the piece selection.

Question [MCQ-22] You've been tasked with building a decentralized wiki system. As in any wiki, pages can be updated and they may include various media documents, large and small. After looking into BitTorrent and IPFS as a storage layer, you've decided on using IPFS (and against BitTorrent) to implement the wiki. What made you think that IPFS is better than BitTorrent for this application?

■ IPFS has a built-in solution to reference the latest versions of files.

■ IPFS deduplicates copies of identical files.

□ IPFS is more efficient for large files.

□ IPFS is content-addressed, hence guaranteeing data integrity.

□ IPFS ensures the immutability of data for a given page version.

Question [MCQ-23] Which of the following design principles of the InterPlanetary File System (IPFS) contribute to its ability to distribute and retrieve content across a decentralized network efficiently?

■ Using a Merkle directed acyclic graph to provide verifiable data integrity.
■ The filesystem is content-addressed, yet allows the naming of individual files in directories.
□ Employing centralized metadata servers to maintain file location information.
□ Byzantine Fault Tolerance (BFT) consensus on file versions.
□ Use of a credit/debt system to avoid freeloading peers (leeches).

Question [MCQ-24] What are the key components of a state-based Conflict-Free Replicated Data Type (CRDT), also known as a convergent replicated data-type (CvRDT) ?

■ A query function, retrieving the CRDT value based on its internal state
■ An initial default state for the CRDT.
■ An update function, allowing local changes to the state.
■ A merge function, integrating external changes into the current internal state
□ A conflict resolution function, to deal with conflicting concurrent changes.

Question [MCQ-25] Which of the following are advantages of Conflict-Free Replicated Data Types compared to Paxos in distributed data management systems?

■ Improved availability in the presence of network partitions without requiring quorum-based consensus.
■ Reduced coordination overhead by relying on monotonic state growth instead of repeated consensus rounds.
□ Stronger consistency guarantees under high churn conditions due to deterministic state convergence.
□ Enhanced support for linearizability in highly dynamic, geographically distributed environments.
□ Lower storage costs through optimized garbage collection of historical state changes.

Question [MCQ-26] In a decentralised social media application, you notice that a certain user's posts are no longer visible to the majority of their followers, even though the posts appear to have been published successfully. This user is unable to reach their audience, and their posts are being silently dropped or redirected to irrelevant nodes in the network. What kind of security attack could have caused this behavior?

■ Eclipse Attack
■ Man-in-the-Middle Attack
□ Privacy attack
□ Sybil Attack
□ Authentication attack

Question [MCQ-27] After a period of normal activity, you observe that a social media's network is flooded with new, seemingly legitimate users. However, these new users begin to distort discussions, manipulate voting mechanisms, and disrupt the normal functioning of the application. Further investigation reveals that these accounts are controlled by a small group of attackers. What kind of security attack could have caused this behavior?

■ Sybil Attack
□ Replay Attack
□ Data Poisoning Attack
□ Brute Force Attack
□ Man-on-the-side attack

Question [MCQ-28] Users in a decentralized social media application notice that sensitive data, such as private messages and user profiles, are being accessed and shared without their consent. This behavior occurs despite users following all the necessary privacy precautions, such as encryption and secure key management. What kind of security attack could have caused this behavior?

- ■ Man-in-the-Middle Attack
- □ Sybil Attack
- □ Denial of Service (DoS) Attack
- □ Data Corruption Attack
- □ Traffic Analysis Attack

Question [MCQ-29] In a decentralized social media application, users have always relied on the system to keep their identities anonymous. However, some users begin to report that they are being targeted or harassed based on their activities and opinions shared within the platform. Upon investigation, it appears that their real identities or patterns of behavior have been linked to their anonymous profiles. What kind of security attack or vulnerability could have caused this behavior?

- ■ Traffic Analysis Attack
- ■ De-anonymization Attack
- □ Brute Force Attack
- □ Eclipse Attack
- □ Sybil Attack

Question [MCQ-30] How do mixnets, such as Mixminion, address the challenge of anonymous communication?

- ■ Clients can provide encrypted return paths to receivers, thereby protecting their identity.
- ■ Mixnets pad the content of messages, to minimize message correlation based on content size or similarity.
- ■ Mixnet nodes shuffle messages to make it difficult to match incoming with outgoing messages.
- ■ Mixnet nodes delay and batch messages to complicate traffic analysis.
- □ Clients choose routing paths through mixnet nodes to minimize latency.

Question [MCQ-31] Which problems is Tor, the onion router, solving?

- ■ Offering location-hidden services
- ■ Enabling anonymous low-latency communication
- ■ Guaranteeing perfect forward secrecy
- □ Offering anonymous communication that resists traffic analysis
- □ Guaranteeing anonymity on a fully P2P network

Question [MCQ-32] In a decentralized communication network, a user wants to send a private message to another user without revealing their identity or the identity of the recipient. However, without proper precautions, intermediaries or adversaries on the network can easily trace the message back to the sender or determine the final recipient by analyzing the traffic flow. How does onion routing solve the problem of preserving anonymity for both the sender and the receiver?

■ A node in the onion routing path learns only where it received the packet from and the next hop.

■ The sender creates a multi-layered message, where each layer targets a specific node.

□ Nodes never know if the next hop is the destination.

□ Nodes never know if the previous hop is the sender.

□ Each node on the onion routing path decrypts the message and re-encrypts it with the next hop's key.

Question [MCQ-33] In a Dining Cryptographers network (DC-net), how is the anonymity of the sender achieved?

■ Each participant contributes random bits that collectively cancel out except for the sender's contribution.

■ DC-nets achieves ideal anonymity if all honest nodes are connected (directly or indirectly) by shared random keys.

□ The DC-net allows multiple participants to communicate at the same time, creating a correspondingly large anonymity set.

□ The DC-net assigns a single shared key to all participants, ensuring nobody knows who transmits.

□ The DC-net nodes successively mix the messages, thereby ensuring that identifying information has been lost before decryption.

Question [MCQ-34] Which of the following represent potential techniques to resist Sybil attacks?

■ Proof of Work

■ Proof of Stake

■ Proof of Space

■ Web of Trust

□ E-mail verification

Question [MCQ-35] What are the key properties of a Proof-of-Personhood obtained through pseudonym parties?

■ It guarantees a "One physical body, one attendance token" property.

■ The Proof-of-Personhood is privacy preserving, as it doesn't reveal anything about the person.

■ It allows for simple geographical federation of identities, if parties are simultaneous.

■ Proof-of-Personhood could be used to create unlinkable user accounts on third party services.

□ Identities based on Proof-of-Personhood are disposable.

Question [MCQ-36] What social graph analysis methods can be used to achieve Sybil resistance in a distributed system?

■ Using trust scores derived from the network topology to identify and limit Sybil nodes.

■ Applying community detection algorithms to isolate clusters of Sybil nodes.

■ Implementing random walk-based algorithms to evaluate the probability of a node being Sybil.

□ Analyzing how long each identity has been in the system and how many connections it has to identify Sybil nodes.

□ Using the shortest path algorithm to directly identify Sybil nodes based on graph distance.

Question [MCQ-37] Contrast and compare the key properties do Proof of Work (PoW) and Proof of Stake (PoS) sybil defenses in permissionless consensus. Which of the following statements accurately distinguish their implications in blockchains?

■ PoW provides strong security by imposing a computational cost on Sybil nodes.

■ PoS reduces overall energy consumption compared to PoW.

■ PoS rewards participants based on their staked assets.

□ PoW allows faster transaction finality than PoS.

□ PoW, unlike PoS, ensures that wealth can't be used to wield influence in the blockchain.

Question [MCQ-38] You're designing an offline-first system designed for concurrent, asynchronous editing of structured data in a decentralized context. Given the application's specifications, you expect most edits not to be in conflict. However, when editing conflicts do happen, the system must exclusively keep the edits that happened-before (causally ordered), and ignore entirely concurrent edits. Which of the following form of clocks could you associate with each edit to achieve the desired behavior?

■ Vector Clock

□ Threshold Logical Clock

□ Wall Clock

□ Lamport Clock

□ System Clock

Question [MCQ-39] In the context of blockchain systems, there are two prevalent models for tracking ownership: the account-based model and the unspent transaction output (UTXO) model. Which of the following statements accurately distinguish the characteristics and implications of these two models?

■ In the UTXO model, the execution of new transactions requires validating references to the output of previous transactions.

■ In the UTXO model, programmable logic is associated with a specific coin rather than an account.

■ In the account-based model, the system tracks balances directly associated with accounts.

□ In the UTXO model, on-chain data does not allow the derivation of the user's balance.

□ The account-based model is easier to reason with and inherently more parallelizable.

Question [MCQ-40] The Bitcoin blockchain favors liveness over safety in its consensus protocol. What fundamental design principles can the nodes and clients rely on to reach an eventually-consistent state?

■ The nodes favor adding blocks to the heaviest (longest) chain they are aware of.

■ Bitcoin's consensus is designed to resist attempted tampering provided more than 50% of the mining power is honest.

■ Nodes validate all the transactions in a block they receive to prevent double spending.

□ Finality is guaranteed as long as the client waits 6 blocks for confirmation.

□ A Bitcoin miner disconnected from the rest of the network for a long period of time cannot proceed or commit transactions.

Question [MCQ-41] In public blockchain environments, it is critical to ensure that on-chain contract code always completes within a predictable and finite amount of time and produces deterministic output across all nodes. Which of the following approaches can be used to enforce these constraints?

■ Use a resource-bounded deterministic virtual machine that halts on resource depletion.

□ Implement threshold signatures with a fixed execution schedule to control runtime.

□ Run the contract under a real-time operating system that terminates code after a fixed timeout.

□ Restrict input size so the code will always finish quickly in practice.

□ Ensure each node relies on a hardware timer to kill any contract running beyond a set interval.

Question [MCQ-42] When combining Proof-of-Work-based consensus with Byzantine Fault Tolerance (BFT) consensus mechanisms in a blockchain system, such as Byzcoin, how does the mining power influence the BFT voting process?

■ Mining power influences voting by determining which nodes are eligible to participate in the BFT consensus committee.

■ The PoW-based blockchain is used to constitute and rotate the BFT consensus committee.

□ In PoW with BFT consensus, all nodes have equal voting power regardless of their mining power, ensuring a fully democratic process.

□ The BFT consensus process can be bypassed once a PoW miner succeeds in mining a block, finalizing it immediately.

□ The PoW consensus takes over only when the BFT consensus leader crashes and a new one needs to be chosen.

Question [MCQ-43] Recall that Algorand randomly selects a subset of users to achieve consensus on the next block using Verifiable Random Functions (VRF). In Algorand each node has a private and secret key pair (pk, sk). The random selection is based on verifiable random functions. Verifiable random functions are such that for an input x, $VRF(sk, x)$ returns a hash h of x and a proof. The proof allows anyone who knows the public key pk to verify that h is a hash of x and sk without knowing the secret key sk.
Select all valid assertions:

■ For a given input x it is not possible to predict the selected nodes without knowing their secret keys sk.

■ A node can verify that a proposed block is issued by an elected peer using its proof and public key pk.

□ Algorand relies on verifiable random functions to implement secure multi-party computation.

□ The security of Algorand does not depend on the choice of the input seed x. In particular any node can propose a seed.

□ An adversary can not take control over the Algorand blockchain by flooding the network with new nodes because elected nodes are sampled using a uniform distribution over the hashes.

Question [MCQ-44] One of the key ideas in modern software quality assurance is to "shift left" the quality assurance (QA) efforts. What does this mean?

■ QA efforts should be planned to minimize overall project costs and risks.

■ QA efforts should be designed to catch issues as soon as possible.

□ QA efforts should be shifted to a dedicated, specialized QA team.

□ Tests should be written before the coresponding implementation.

□ Tests should be carried out in both testing and staging environments.

Question [MCQ-45] Netflix introduced Chaos Monkey in 2011, as part of the efforts which later became known as chaos engineering. What is Chaos Monkey's purpose ?

■ Improving the quality of the services

■ Growing a culture of responsibility and resilience in engineering teams

□ Randomly disrupting the network layer to assess resiliency

□ Removing the need for microservices' downtime / maintenance windows

□ Continuously fuzzying microservices in production to catch new bugs

Question [MCQ-46] Recall that industry practices moved past Chaos Monkey and Chaos Kong to adopt Fault-Injection Testing (FIT) as a classic approach to chaos engineering. What are the prerequisites for successful fault-injection testing?

■ A constant level of system activity (steady state)

■ Stable software, well-tested through conventional means

■ A clear experimental protocol with a control group

■ Key metrics that capture the system's correct functioning

□ Specialized hardware to carry out voltage glitching and inject hardware faults

Question [MCQ-47] The Scantegrity E-voting system is based on a switchboard: a circuit that maps voting boxes on ballots to candidates. The circuit is generated along with a public cryptographic commitment proof that can be used to verify the circuit integrity. To ensure voting integrity, the public must be able to verify that ticks on the ballots are transmitted to the appropriate candidate.
What solutions can be used to prove that the circuit maps the boxes to the candidates properly without compromising voters' anonymity?

■ Generate a switchboard twice as big, randomly select half of the ballots and reveal the corresponding part of the circuit. The selected ballots are discarded and not used during the vote.

■ Use two switch boards and route the votes through both. After the election, for each ballot randomly reveal the routing of either the first or second switch board.

□ Reveal the secret used to generate the switch board after the election. Each voter can verify that their vote was router correctly by using their ballot ID.

□ After the election each voter receives a cryptographic proof of which candidate they voted for. The proof can be verified from the public cryptographic commitment.

□ After the election, the mapping from boxes to candidates is released, allowing everyone to verify that it matches the original cryptographic commitment.

Question [MCQ-48] Following an election carried out with an internet-based voting system, some users reported that they were coerced to vote in a certain way. As a designer, what measures could you implement to help mitigate this issue in the future?

■ Allow re-voting, ensuring that only the last ballot is tallied.

■ Allow users to vote with indistinguishable, fake credentials.

□ Allow re-voting, ensuring that only the first ballot is tallied.

□ Remind users that voter coercion is illegal and will be prosecuted.

□ Send a delayed confirmation request to verify the voter’s intent.