Lesson 3.1 Quiz: Cryptographic Foundations for Finance

Module 3: The Trust Problem

Prof. Dr. Joerg Osterrieder

Digital Finance — BSc Course (v2026.05)

Question 1

A payment processor stores a SHA-256 hash of every incoming transaction record. A junior auditor asks: "Why not just store the records themselves?" Which answer **best** explains the purpose of hashing in this context?

- (A) Hashing encrypts the records so unauthorized users cannot read them
- (B) Hashing speeds up database queries by replacing text with numbers
- (C) Hashing creates a fixed-size fingerprint that detects any unauthorized modification to the record
- (D) Hashing compresses the records to save storage space

Question 1

A payment processor stores a SHA-256 hash of every incoming transaction record. A junior auditor asks: "Why not just store the records themselves?" Which answer **best** explains the purpose of hashing in this context?

- **A** Hashing encrypts the records so unauthorized users cannot read them
- **B** Hashing speeds up database queries by replacing text with numbers
- **C** Hashing creates a fixed-size fingerprint that detects any unauthorized modification to the record
- **D** Hashing compresses the records to save storage space

[Answer hidden – compile with \solutionstrue to reveal]

Question 2

A blockchain developer tells a client: "SHA-256 is a one-way function." The client asks what "one-way" means. Which explanation is **most accurate**?

- A The hash can only be computed on one specific type of hardware
- B The hash output always has fewer characters than the input
- C The function can only be called once per input
- D Given the hash output, it is computationally infeasible to find the original input

Question 2

A blockchain developer tells a client: "SHA-256 is a one-way function." The client asks what "one-way" means. Which explanation is **most accurate**?

- A The hash can only be computed on one specific type of hardware
- B The hash output always has fewer characters than the input
- C The function can only be called once per input
- D Given the hash output, it is computationally infeasible to find the original input

[Answer hidden – compile with \solutionstrue to reveal]

Question 3

An intern hashes the string `"Transfer $500 to Alice"` and then hashes `"Transfer $500 to alice"` (lowercase ‘a’). The two hashes are completely different. Which property of cryptographic hash functions does this demonstrate?

- A. The avalanche effect
- B. Pre-image resistance
- C. Second pre-image resistance
- D. Collision resistance

Question 3

An intern hashes the string `"Transfer $500 to Alice"` and then hashes `"Transfer $500 to alice"` (lowercase ‘a’). The two hashes are completely different. Which property of cryptographic hash functions does this demonstrate?

- **A** The avalanche effect
- **B** Pre-image resistance
- **C** Second pre-image resistance
- **D** Collision resistance

[Answer hidden – compile with \solutionstrue to reveal]

Question 4

A compliance officer needs to verify that a regulatory filing has not been altered since it was submitted. She compares the hash she computed today with the hash recorded at submission. Which property of hash functions makes this verification **trustworthy**?

- (A) Collision resistance — it is infeasible that a different document produces the same hash
- (B) The avalanche effect — any change would flip output bits
- (C) Pre-image resistance — the original cannot be derived from the hash
- (D) Determinism — the same input always produces the same output

Question 4

A compliance officer needs to verify that a regulatory filing has not been altered since it was submitted. She compares the hash she computed today with the hash recorded at submission. Which property of hash functions makes this verification **trustworthy**?

- (A) Collision resistance — it is infeasible that a different document produces the same hash
- (B) The avalanche effect — any change would flip output bits
- (C) Pre-image resistance — the original cannot be derived from the hash
- (D) Determinism — the same input always produces the same output

[Answer hidden – compile with \solutionstrue to reveal]

Question 5

A blockchain contains 1,024 transactions organized in a Merkle tree. A lightweight mobile wallet needs to verify that a specific transaction is included in the block. How many sibling hashes must the network provide in the inclusion proof?

- A 1,024
- B 10
- C 512
- D 5

Question 5

A blockchain contains 1,024 transactions organized in a Merkle tree. A lightweight mobile wallet needs to verify that a specific transaction is included in the block. How many sibling hashes must the network provide in the inclusion proof?

- A 1,024
- B 10
- C 512
- D 5

[Answer hidden – compile with \solutionstrue to reveal]

Question 6

You are constructing a Merkle tree from four transactions: T_1, T_2, T_3, T_4. You compute $H_1 = \text{Hash}(T_1)$ through $H_4 = \text{Hash}(T_4)$. What is the **next** step?

- (A) XOR all four hashes to produce the root
- (B) Concatenate $H_1 \| H_2$ and hash them; concatenate $H_3 \| H_4$ and hash them
- (C) Hash all four leaf hashes together in one operation to get the Merkle root
- (D) Sort the hashes alphabetically, then hash the sorted list

Question 6

You are constructing a Merkle tree from four transactions: T_1, T_2, T_3, T_4. You compute $H_1 = \text{Hash}(T_1)$ through $H_4 = \text{Hash}(T_4)$. What is the **next** step?

- (A) XOR all four hashes to produce the root
- (B) Concatenate $H_1 \| H_2$ and hash them; concatenate $H_3 \| H_4$ and hash them
- (C) Hash all four leaf hashes together in one operation to get the Merkle root
- (D) Sort the hashes alphabetically, then hash the sorted list

[Answer hidden – compile with \solutionstrue to reveal]

Question 7

Alice wants to send Bob an encrypted message using public-key cryptography. She has Bob's public key and her own key pair. Which key does she use to **encrypt** the message?

A. Her own public key
B. Bob's public key
C. Bob's private key
D. Her own private key

Question 7

Alice wants to send Bob an encrypted message using public-key cryptography. She has Bob's public key and her own key pair. Which key does she use to **encrypt** the message?

- Her own public key
- Bob's public key
- Bob's private key
- Her own private key

[Answer hidden – compile with \solutionstrue to reveal]

Question 8

A cryptocurrency wallet shows a user the following signing process: (1) hash the transaction data, (2) sign the hash with the private key, (3) broadcast the transaction + signature. Why is the **hash** signed rather than the raw transaction data?

- (A) The hash adds an extra layer of encryption to the signature
- (B) Signing the raw data would reveal the private key
- (C) Hashing first converts the variable-length transaction into a fixed-size input, making the signing operation efficient
- (D) Signing the hash hides the transaction details from the network

Question 8

A cryptocurrency wallet shows a user the following signing process: (1) hash the transaction data, (2) sign the hash with the private key, (3) broadcast the transaction + signature. Why is the **hash** signed rather than the raw transaction data?

- A The hash adds an extra layer of encryption to the signature
- B Signing the raw data would reveal the private key
- C Hashing first converts the variable-length transaction into a fixed-size input, making the signing operation efficient
- D Signing the hash hides the transaction details from the network

[Answer hidden – compile with \solutionstrue to reveal]

Question 9

A bank implements a commitment scheme for sealed-bid auctions: each bidder submits Hash(bid||nonce) before the deadline, then reveals the bid and nonce afterward. Which hash property ensures a bidder **cannot change their bid** after submission?

- A. Pre-image resistance — the auctioneer cannot reverse-engineer the bid
- B. The avalanche effect — changing the bid changes the hash
- C. Determinism — the same bid always produces the same hash
- D. Collision resistance — the bidder cannot find a different bid with the same hash

Question 9

A bank implements a commitment scheme for sealed-bid auctions: each bidder submits Hash(bid||nonce) before the deadline, then reveals the bid and nonce afterward. Which hash property ensures a bidder **cannot change their bid** after submission?

- A. Pre-image resistance — the auctioneer cannot reverse-engineer the bid
- B. The avalanche effect — changing the bid changes the hash
- C. Determinism — the same bid always produces the same hash
- D. Collision resistance — the bidder cannot find a different bid with the same hash

[Answer hidden – compile with \solutionstrue to reveal]

Question 10

An exchange publishes a Merkle tree of all customer balances for a proof-of-reserves audit. Customer X wants to verify that their balance of 2.5 BTC is included. What information does Customer X need from the exchange?

- A Only the Merkle root
- B Their leaf hash plus the sibling hashes along the path to the root
- C The complete list of all customer balances
- D The private key used to construct the tree

Question 10

An exchange publishes a Merkle tree of all customer balances for a proof-of-reserves audit. Customer X wants to verify that their balance of 2.5 BTC is included. What information does Customer X need from the exchange?

- (A) Only the Merkle root
- (B) Their leaf hash plus the sibling hashes along the path to the root
- (C) The complete list of all customer balances
- (D) The private key used to construct the tree

[Answer hidden – compile with \solutionstrue to reveal]

Question 11

A node on the Bitcoin network receives a transaction claiming to transfer 0.5 BTC from address `1A3x...` to address `1B7y...`. The transaction includes an ECDSA signature. What does the node verify **first**?

- (A) That the sender has sufficient balance (0.5 BTC or more)
- (B) That the recipient address `1B7y...` exists on the blockchain
- (C) That the signature was produced by the private key corresponding to `1A3x...`
- (D) That the transaction amount is below the network maximum

Question 11

A node on the Bitcoin network receives a transaction claiming to transfer 0.5 BTC from address `1A3x...` to address `1B7y...`. The transaction includes an ECDSA signature. What does the node verify **first**?

- That the sender has sufficient balance (0.5 BTC or more)
- That the recipient address `1B7y...` exists on the blockchain
- That the signature was produced by the private key corresponding to `1A3x...`
- That the transaction amount is below the network maximum

[Answer hidden – compile with \solutionstrue to reveal]

Question 12

ECDSA uses 256-bit keys while RSA requires 3,072-bit keys for equivalent security. A mobile banking app is choosing between the two. What is the **primary practical advantage** of ECDSA in this context?

- ECDSA is a symmetric algorithm, which is always faster
- ECDSA signatures are impossible to forge, while RSA signatures can be
- ECDSA does not require a private key, simplifying key management
- ECDSA's smaller keys mean less storage, bandwidth, and computation — critical for mobile devices

Question 12

ECDSA uses 256-bit keys while RSA requires 3,072-bit keys for equivalent security. A mobile banking app is choosing between the two. What is the **primary practical advantage** of ECDSA in this context?

- A ECDSA is a symmetric algorithm, which is always faster
- B ECDSA signatures are impossible to forge, while RSA signatures can be
- C ECDSA does not require a private key, simplifying key management
- D ECDSA's smaller keys mean less storage, bandwidth, and computation — critical for mobile devices

[Answer hidden – compile with \solutionstrue to reveal]

Question 13

An attacker modifies transaction T_3 in a Merkle tree of 8 transactions. Which of the following hashes in the tree will **change** as a result?

- (A) The leaf hash of T_3 and all hashes on the path from T_3 to the root
- (B) Only the Merkle root
- (C) Every hash in the entire tree
- (D) Only the leaf hash of T_3

Question 13

An attacker modifies transaction T_3 in a Merkle tree of 8 transactions. Which of the following hashes in the tree will **change** as a result?

- **A** The leaf hash of T_3 and all hashes on the path from T_3 to the root
- **B** Only the Merkle root
- **C** Every hash in the entire tree
- **D** Only the leaf hash of T_3

[Answer hidden – compile with \solutionstrue to reveal]

Question 14

A digital signature provides three guarantees: authentication, integrity, and non-repudiation. A disgruntled employee signs a fraudulent transaction and later claims they did not authorize it. Which property of digital signatures **directly refutes** this claim?

- (A) Integrity — the transaction has not been modified
- (B) Confidentiality — the transaction was encrypted during transmission
- (C) Authentication — the signature proves the identity of the signer
- (D) Non-repudiation — the signer cannot deny having signed, since only their private key could have produced the signature

Question 14

A digital signature provides three guarantees: authentication, integrity, and non-repudiation. A disgruntled employee signs a fraudulent transaction and later claims they did not authorize it. Which property of digital signatures **directly refutes** this claim?

- (A) Integrity — the transaction has not been modified
- (B) Confidentiality — the transaction was encrypted during transmission
- (C) Authentication — the signature proves the identity of the signer
- (D) Non-repudiation — the signer cannot deny having signed, since only their private key could have produced the signature

[Answer hidden – compile with \solutionstrue to reveal]

Question 15

Alice signs a transaction "Pay Bob 1 BTC" and broadcasts it. An attacker intercepts the signed transaction and changes it to "Pay Eve 1 BTC." Will the attack succeed?

A. No — but only because the network rejects all transactions from intercepted broadcasts

B. Yes — the attacker can compute Alice's private key from the signature

C. No — modifying the transaction invalidates the signature because the hash of the modified message will not match

D. Yes — the attacker can modify the transaction and reuse Alice's signature

Question 15

Alice signs a transaction "Pay Bob 1 BTC" and broadcasts it. An attacker intercepts the signed transaction and changes it to "Pay Eve 1 BTC." Will the attack succeed?

A. No — but only because the network rejects all transactions from intercepted broadcasts

B. Yes — the attacker can compute Alice's private key from the signature

C. No — modifying the transaction invalidates the signature because the hash of the modified message will not match

D. Yes — the attacker can modify the transaction and reuse Alice's signature

[Answer hidden – compile with \solutionstrue to reveal]

Question 16

In the double-spend problem, Alice sends 1 BTC to Bob and simultaneously sends the *same* 1 BTC to Charlie. Both transactions have valid digital signatures. Why does cryptography **alone** fail to prevent this?

- A Cryptography ensures each signature is valid but cannot determine the global ordering of transactions
- B Hash functions cannot detect duplicate transactions
- C Digital signatures cannot verify the sender's identity
- D Public-key encryption prevents nodes from seeing both transactions

Question 16

In the double-spend problem, Alice sends 1 BTC to Bob and simultaneously sends the *same* 1 BTC to Charlie. Both transactions have valid digital signatures. Why does cryptography **alone** fail to prevent this?

- (A) Cryptography ensures each signature is valid but cannot determine the global ordering of transactions
- (B) Hash functions cannot detect duplicate transactions
- (C) Digital signatures cannot verify the sender's identity
- (D) Public-key encryption prevents nodes from seeing both transactions

[Answer hidden – compile with \solutionstrue to reveal]

Question 17

A startup proposes using MD5 (128-bit output) instead of SHA-256 (256-bit output) for their blockchain to "save storage space." An advisor warns against this. What is the **strongest** technical argument against MD5?

- A MD5's collision resistance is broken — practical collisions have been demonstrated, making the blockchain vulnerable to transaction substitution
- B MD5 is slower to compute than SHA-256
- C MD5 is not supported by modern programming languages
- D MD5 produces output that is too short to be displayed in a user interface

Question 17

A startup proposes using MD5 (128-bit output) instead of SHA-256 (256-bit output) for their blockchain to "save storage space." An advisor warns against this. What is the **strongest** technical argument against MD5?

- **A** MD5's collision resistance is broken — practical collisions have been demonstrated, making the blockchain vulnerable to transaction substitution
- **B** MD5 is slower to compute than SHA-256
- **C** MD5 is not supported by modern programming languages
- **D** MD5 produces output that is too short to be displayed in a user interface

[Answer hidden – compile with \solutionstrue to reveal]

Question 18

A user loses their private key but still has their public key and blockchain address. Which of the following is true?

- A They can contact the blockchain's central authority to reset their key
- B They can use their public key to sign transactions temporarily
- C They can derive the private key from the public key using elliptic curve math
- D They can still receive funds but can never spend them, because signing transactions requires the private key

Question 18

A user loses their private key but still has their public key and blockchain address. Which of the following is true?

- **A** They can contact the blockchain's central authority to reset their key
- **B** They can use their public key to sign transactions temporarily
- **C** They can derive the private key from the public key using elliptic curve math
- **D** They can still receive funds but can never spend them, because signing transactions requires the private key

[Answer hidden – compile with \solutionstrue to reveal]

Question 19

A government regulator proposes requiring all cryptocurrency users to register their public keys with a central authority, which would hold backup copies of private keys. Evaluate this proposal from a cryptographic security perspective. Which concern is **most critical**?

- (A) The public keys would become visible to everyone, reducing privacy
- (B) The registration process would slow down transaction processing
- (C) A single breach of the central authority would compromise every user's funds simultaneously
- (D) The central authority would need too much storage for all the keys

Question 19

A government regulator proposes requiring all cryptocurrency users to register their public keys with a central authority, which would hold backup copies of private keys. Evaluate this proposal from a cryptographic security perspective. Which concern is **most critical**?

- (A) The public keys would become visible to everyone, reducing privacy
- (B) The registration process would slow down transaction processing
- (C) A single breach of the central authority would compromise every user's funds simultaneously
- (D) The central authority would need too much storage for all the keys

[Answer hidden – compile with \solutionstrue to reveal]

Question 20

A financial institution currently uses a trusted third party (a bank) to prevent double-spending. A consultant proposes replacing this with a system using only cryptographic hash functions, Merkle trees, and digital signatures — but **no consensus mechanism**. Is this system viable?

- A. No — hash functions are too slow for real-time transaction processing
- B. No — without a consensus mechanism, there is no way to establish which of two conflicting transactions came first across a distributed network
- C. Yes — digital signatures alone can prevent double-spending because they prove ownership
- D. Yes — Merkle trees can detect conflicting transactions and automatically reject the second one

Question 20

A financial institution currently uses a trusted third party (a bank) to prevent double-spending. A consultant proposes replacing this with a system using only cryptographic hash functions, Merkle trees, and digital signatures — but **no consensus mechanism**. Is this system viable?

- (A) No — hash functions are too slow for real-time transaction processing
- (B) No — without a consensus mechanism, there is no way to establish which of two conflicting transactions came first across a distributed network
- (C) Yes — digital signatures alone can prevent double-spending because they prove ownership
- (D) Yes — Merkle trees can detect conflicting transactions and automatically reject the second one

[Answer hidden – compile with \solutionstrue to reveal]