Consider the following 2-party protocol that realizes the functionality F(X,Y):= X+Y.

1. Party 1 sends X
2. Party 2 sends Y
3. Both outputs X+Y

Question 1: Recall that in the semi-honest (a.k.a. honest but curious) setting we assumed private Point-to-Point channels between parties. Does the above protocol securely realizes F in the semi-honest setting?

1. Yes
2. No

Question 2: Recall that in the malicious-setting we assumed private and authenticated Point-to-Point channels between parties. Does the above protocol securely realizes F in the malicious setting?

1. Yes
2. No

Question 3: Assume that the parties do **not** have private Point-to-Point channels. How can we modify the protocol to securely realizes F in presence of semi-honest adversaries?

Question 4: A protocol Pi realizes a functionality F in presence of semi-honest adversaries if:

1. For any real world PT adversary A there exists a PT distinghuisher D such that the real world and the ideal world are computationally indistinguishable
2. For any ideal adversary PT adversary ~A there exists a real adversary A such that the ideal world (with parameters ~A and F) and the real world (with parameters A and Pi) are indistinguishable
3. There exists a PT simulator S such that for any subset of corrupted parties produces a view indistinguishable that is indistinguishable to the view of such subset of corrupted parties honestly executing the protocol.

Question 5: Intuitively, an MPC protocol realizes a functionality F in presence of semi-honest adversaries if the inputs remain private after the protocol execution (namely, no information about the inputs is revealed).

1. True
2. False

Question 6: The Oblivious Transfer ideal functionality (multiple options might be true):

1. Take in input two messages m0,m1 from the receiver and send one message to the sender
2. Take in input two messages m0,m1 from the sender and send one message to the receiver
3. Take in input one bit b from the receiver
4. Output a random message.
5. Output the b-th message to both the sender and the receiver

Question 6: We saw the following protocol realizing the OT functionality in presence of semi-honest adversaries:

- Receiver sample (PK_b,SK_b) valid key pair of a Public Key Encryption scheme and sample a random public key PK_{1-b} and sends PK_0,PK_1 to the sender
- Sender sends C_0 = Enc(PK_0, message_0), C_1 = Enc(PK_1, message_1)
- …. ? …..

----

1. What is the last step?

2. What are the security properties the **two** security properties that are necessary from the PKE?

Question 7: Consider the following procedure:

- Sample random keys K0A,K1A,K0B,K0A
- Compute C00 = E(K0A,E(K0B,0)
- Compute C01 = E(K0A,E(K1B,1)
- Compute C10 = E(K1A,E(K0B,1)
- Compute C11 = E(K1A,E(K1B,0)
- Send (C00,C01,C10,C11)

Which of the following statements is true:

1. The procedure is a valid garbling circuit of a NAND
2. The procedure is a valid garbling circuit of a XOR
3. The procedure is not a valid garbling

Question 8: Let C be a circuit with 10 input binary-gates and 90 middle/output binary-gates, for each of these binary gates the party 1 contributes with the first input and the party 2 contributes with the second input (thus, C: {0,1}^10 x {0,1}^10 → {0,1}).
Assume we have a protocol realizing the 1-out-4 OT functionality that has 3 messages.
How many many messages has the "Yao's GC protocol" that realizes the functionality of Circuit Evaluation for the circuit C?

1. 120
2. 32
3. 3

Question 9: Let m<m' and, for any k, let OT^k be the ideal functionality realizing k independent instances of the standard 1-out-2 OT functionality. Which of the following statements is true:

1. Any OT-extension protocol uses "Yao's GC protocol" to realize OT^m functionality
2. An OT-extension protocol can use "Yao's GC protocol" to realize the OT^m' functionality using a circuit with 2m+1 inputs.
3. An OT-extension protocol realizes the OT^m functionality internally using m' instances of an OT protocol.
4. An OT-extension protocol realizes the OT^m' functionality internally using m instances of an OT protocol.

Question 10: A secret sharing is t-private if for any subset R of the parties of cardinality t the parties cannot compute the secret.

1. True
2. False

Question 11: Consider the following 3-PC protocol:

- Party 1 shares X1 with a 1-out-of-3 linear secret sharing scheme over ZZ_q (the field of integers modulo a prime q) obtaining [X1]= (X11,X12,X13) and sends the shares X12 to Party 2 and X13 to Party 3.
- Party 2 shares X2 with a 1-out-of-3 linear secret sharing scheme obtaining [X2]= (X21,X22,X23) and sends the shares X21 to Party 1 and X23 to Party 3.
- Party 1,2,3 computes a multiplication protocol with inputs [X1] and [X2] obtaining shares Z1,Z2,Z3
- Party 1 receives Z2,Z3 from Party 2 and Party 3 respectively and outputs the reconstruction of the shares Z1,Z2,Z3.

Which one of the following statements is true:

1. The protocol is secure.
2. The protocol realizes F(X1,X2) = X1 * X2
3. The protocol realizes F(X1,X2) = X1 * X2 mod q
4. The protocol realizes F(X1,X2) = (X1 * X2 mod q, null, null)
5. The protocol is secure if at least 2 parties are honest.

Question 12: Let (S1,S2,…,Sn) be 2-out-of-n Shamir’s secret sharing of a secret message. Describe an algorithm that receives in input shares S1, S3, S9 (i.e. the shares for the indexes 1,3,9) and outputs the shared secret.

Question 13: Assume you have a linear secret sharing scheme (Share, Rec) over ZZ_q where q is prime. Describe a protocol that securely realizes F(X1,X2) = X1 + X2 mod q in presence of semi-honest adversaries.

Question 14: Which of the following statements are true:

1. The GMW protocol securely realizes the Circuit Evaluation ideal functionality in presence of semi-honest adversaries.
2. The GMW protocol uses an OT-protocol internally
3. The number of messages of the GMW protocol is proportional to the number of input gates of the evaluated circuit
4. The GMW protocol uses Garbled Circuit when the number of parties is two
5. The GMW protocol securely realizes the Circuit Evaluation ideal functionality in presence of semi-honest adversary corrupting at least n-1 parties.

Question 15: Let (X1,…, Xn) be shares of the secret X for a t-out-of-n Shamir's secret sharing scheme.
Let (Y1,…, Yn) be shares of the secret Y for a t-out-of-n Shamir's secret sharing scheme.
Which one of the following is true:

1. The shares (X1*Y1, …. , Xn*Yn) are uniformly random Shamir's secret sharing of a secret X*Y

2. The shares (X1*Y1, …. , Xn*Yn) are uniformly random Shamir’s secret sharing of a secret X*Y if the threshold is t < n/2.
3. None of the above

Question 16: An interactive proof system is Zero-Knowledge if for any distinghuisher there exists a simulator such that the view produced by the simulator is indistinguishable from the real execution of the proof system.

1. True
2. False

Question 17: An interactive proof system for a relation R= { (x,w) : Predicate(x,w) } is honest-verifier zero-knowledge if there exists a simulator that on input x outputs w.

1. True
2. False

Question 18: Let R = { (x,w) : 2*x = w and x and w are natural numbers }.
Describe a Zero-Knowledge proof system for the relation R.

Question 19: Complete the following Sigma-Protocol for the relation R = { (H, x) : G^x = H } where (G,*) is a group in multiplicative notation.

- Prover has in input (G,H,x) while Verifier input is (G,H).
- Prover samples random y and sends “C = G^y”
- Verifier sends random “a” in ZZ_q
- Prover sends “ z = …..?….. ”
- Verifier checks that G^z = H * C^a mod q

Question 20: Consider the protocol below:

- Party 1 samples random string r1 and commit using a protocol that securely realizes the commitment functionality.
- Party 2 sends a random string r2
- Party 1 opens to Party 2 the commitment to the random string r1
- Both party compute r1 \xor r2.

Let the Coin Tossing ideal functionality, be the ideal functionality that:

- It does not receive any input
- It samples a random string r and it sends the random string to P1 and P2

Which one of the following statements is true:

1. The protocol above realizes the Coin Tossing ideal functionality in presence of semi-honest adversaries.
2. The protocol above realizes the Coin Tossing ideal functionality in presence of malicious adversaries.
3. The protocol above realizes the Coin Tossing ideal functionality in presence of malicious adversaries **that can only corrupt** the party P2.
4. The protocol above realizes the Coin Tossing ideal functionality in presence of malicious adversaries **that can only corrupt** the party P1.

Question 21: The GMW protocol we saw in class is an MPC protocol based on Linear Secret Sharing Scheme. Mark all the true statements in the list below:

1. The linear secret sharing scheme is an n-out-of-n secret sharing scheme.
2. The linear secret sharing scheme is a verifiable secret sharing scheme.
3. The linear secret sharing scheme works over ZZ_q for a prime q>2.
4. The multiplication protocol computes logical and of the secret shared messages in input

Question 22: Describe the ideal functionality realized by the multiplication protocol of the BGW protocol.

Question 23: The Beaver's Multiplication Triples technique allows for a multiplication protocol in the pre-processing model. The multiplication protocol works as follow:

- The parties have in input shares [x] and [y] of x,y and [a],[b],[c] of a,b, and c= a*b. Namely, party P_i has shares x_i, y_i and shares a_i,b_i,c_i.
- The parties compute [d]= [x] – [a] and [e]= [y]-[b]
- …… ? …...
- The parties compute [z] = de+d *[b] + e * [a] + [c]

What is the missing step of the protocol?

1. Each party broadcast its share of [a] and [b] and they reconstruct a and b
2. Each party broadcast its share of [x] and [y] and they reconstruct x and x
3. Each party broadcast its share of [d] and [e] and they reconstruct d and e

Question 24: Recall that the GMW compiler is a method that converts an MPC protocol PI that securely realizes a functionality F in presence of semi-honest adversaries to an MPC protocol PI' that securely realizes the same functionality F in to malicious security.
The GMW protocol additionally uses a ZK proof protocol for NP and a commitment scheme COM.

Let EXCHANGE be the ideal functionality that on input X1 from Party P1 e X2 from Party P2 outputs X2 to P1 and X1 to P2 (unless the adversary aborts). The protocol below realizes EXCHANGE in presence of semi-honest adversaries.

- The party P1 sends X1 to P2
- The Party P2 sends X2 to P1

Use the GMW compiler to compile the above protocol from semi-honest security to malicious security.

Question 25: What of the following statements of the SPDZ protocol are true:

- The SPDZ protocol realizes the Circuit Evaluation (with abort) ideal functionality in presence of malicious adversaries
- The SPDZ protocol uses preprocessing to evaluate the circuit on input Xi of the party Pi for each party i=1… n
- The SPDZ protocol (that we saw in class) uses Oblivious Transfer for the multiplication protocol
- The SPDZ protocol uses Baver’s Triples Multiplication Triples
- The preprocessing of the SPDZ protocol realizes the Beaver’s Triples Multiplication protocol in presence of semi-honest adversaries using Fully Homomorphic Encryption

Question 26: Mark the correct statement. Let SHA256 a Cryptographic Hash Function, an Hash-Chain is a data structure B_0, …, B_n where B_0 is called “the genesis block” and for any B_i = < Data_i, H_i > where H_i :

- H_i = SHA256(B_i+1)
- H_i = SHA256(B_0)
- H_i = SHA256(B_i)
- H_i = SHA256(B_i-1)

Question 27: Mark all the valid statements. In the Permission-less model :

- The number of parties involved in the protocol is not known apriori
- The parties have point-to-point authenticated channels
- Sybil attacks are unavoidable without cryptographic assumptions
- there exist MPC protocols that securely realizes the Sybil functionality

Question 28: Mark the fundamental properties that a Blockchain protocol should have:

- The protocol should have common-prefix property, namely, if you prune enough the chain the views of any two honest parties are full of cryptocats
- The protocol should have common-prefix property, namely, if you prune enough the chain the views of any two honest parties are the same
- The protocol should have collision resistance
- The protocol should realize the sybil ideal functionality
- the protocol should have chain quality property, only a small ratio of blocks in the blockchain were added by the adversary
- The protocol should have chain growth, namely the value of the bitcoins the honest parties have should grow exponentially
- The protocol should have Persistence, if a certain round an honest parties add a transaction tx (a piece of data) in a block B i more than k block away from the end of the ledger, then tx will always be reported in the same block in the ledger.
- The protocol should have chain growth, namely the number of blocks in the chain followed by the honest parties grow at a steady speed

Question 29: Mark the correct answer. The selfish-mining attack shows that :

- The PoW-based Blockchain protocol is completely insecure and should not be used
- The PoW-based Blockchain protocol does not posses the common-prefix property
- The PoW-based Blockchain protocol does not posses the chain growth property when the adversary has 49% of the computing power
- The PoW-based Blockchain protocol does not posses the chain quality property with parameter ½ when the adversary has 49% of the computing power