Yao's Protocol (2PC):

Key ingredient: "garbling" protocol (garbled circuits)

1) Associate a pair of keys $(k_{i}^{0}, k_{i}^{1})$ with each wire $i$ in the circuit

2) Prepare garbled truth table for the gate

- Replace each entry of truth table with corresponding key
- Encrypt output key with each of the input keys

3) Construct decoding table for output values

General garbling transformation: construct garbled table for each gate in the circuit, prepare decoding table for each output wire in the circuit

Evaluating a garbled circuit:

Invariant: given keys for input wires of a gate, can derive key corresponding to output wire $\Rightarrow$ enables gate-by-gate evaluation of garbled circuit

Requirement: Evaluator needs to obtain keys (labels) for its inputs (but without revealing which set of labels it requested)
Yao's garbled circuit protocol:

1. Prepare garbled circuit for C
2. Prepare OT responses for Bob's inputs. Messages correspond to wire labels.
3. Evaluate garbled circuit to learn C(x, y)

Correctness: Follows by correctness of OT and of the garbling construction.

Security: Relies on security of OT and garbling transformation. Relies on OT simulator to simulate OT responses.

- Simulate Bob's view given output of computation (using the garbled circuit simulator)
- Simulate Alice's view using OT simulator

Variants:
1. If both parties should learn output, Bob can send it to Alice.
2. If Alice and Bob should learn distinct outputs, Alice can have the functionality output a blinded/encrypted version of her output.
3. Can extend to malicious security (need additional rounds and some modifications).

Many optimizations possible:
1. free XOR - no need to send garbled tables for XOR gates in circuit
2. half gates - only need two ciphertexts for each AND gate (not 4)
3. no need to double encrypt - can "encrypt" once using key derived from input keys