Quantum Information Processing (2024) 23:221
https://doi.org/10.1007/s11128-024-04433-7

Secure two-party computation via measurement-based quantum computing

Zeinab Rahmani[1,2,3] · **Armando Humberto Moreira Nolasco Pinto**[1,2] · **Luis Manuel Dias Coelho Soares Barbosa**[3,4,5]

Received: 15 November 2023 / Accepted: 13 May 2024

Abstract
Secure multiparty computation (SMC) provides collaboration among multiple parties, ensuring the confidentiality of their private information. However, classical SMC implementations encounter significant security and efficiency challenges. Resorting to the entangled Greenberger–Horne–Zeilinger (GHZ) state, we propose a quantum-based two-party protocol to compute binary Boolean functions, with the help of a third party. We exploit a technique in which a random Z-phase rotation on the GHZ state is performed to achieve higher security. The security and complexity analyses demonstrate the feasibility and improved security of our scheme compared to other SMC Boolean function computation methods. Additionally, we implemented the proposed protocol on the IBM QisKit and found consistent outcomes that validate the protocol's correctness.

Keywords Quantum secure multiparty computation · Measurement-based quantum computing · Greenberger–Horne–Zeilinger state · IBM Qiskit

✉ Zeinab Rahmani
zeinab.rahmani@ua.pt

Armando Humberto Moreira Nolasco Pinto
anp@ua.pt

Luis Manuel Dias Coelho Barbosa
lsb@di.uminho.pt

1 Department of Electronics, Telecommunications and Informatics, University of Aveiro, Aveiro, Portugal

2 Instituto de Telecomunicações, Aveiro, Portugal

3 International Iberian Nanotechnology Laboratory, Braga, Portugal

4 Department of Computer Science, University of Minho, Braga, Portugal

5 Institute of Systems and Computer Engineering, Technology and Science, Porto, Portugal

Published online: 01 June 2024

1 Introduction

Secure multiparty computation (SMC) is a technology where multiple individuals, each having their own set of confidential data, can calculate a function f that is open to the public without unveiling any details about their data to the other participants. SMC has practical applications in real-life scenarios, such as vehicular networks [1, 2], genomics data mining [3], and secure voting [4]. Despite the wide range of applications, classical SMC implementations confront security and efficiency challenges. These implementations rely on public-key cryptography, which often leads to significant computational and communication overheads. Furthermore, classical SMC protocols that rely on prime number factorization or discrete logarithms are vulnerable to quantum computers attacks, as a consequence of Shor's algorithm [5].

The concept of SMC was initially introduced by Yao [6], to address the millionaire's problem within the realm of two-party secure computation. Later, classical SMC solutions with more than two parties [7–12] were introduced. After the achievement of Quantum Key Distribution [13, 14], significant strides have been taken to enhance the security and efficiency of the classical SMC protocols by utilizing quantum technology. A number of quantum-based SMC protocols have been proposed to carry out computations on various types of functions, such as Boolean functions [15, 16], polynomials [17], and arithmetic operations [18–21]. Despite the remarkable accomplishments of quantum protocols leveraging single qubits or entangled states across various computational scenarios, the use of Measurement-Based Quantum Computing (MBQC) [22–24] remains a relatively unexplored domain. The concept of MBQC was initially introduced by Raussendorf and Briegel [25]. In this quantum computing model, computations are accomplished through a sequence of measurements performed on a highly entangled quantum state, referred to as a cluster state or a resource state. This type of quantum computing is different from the traditional circuit model that involves the manipulation of qubits using quantum gates, similar to how classical computers employ logic gates for bit processing. The MBQC provides a number of advantages as it inherits certain characteristics that make it more fault-tolerant compared to the circuit model. For instance, if the quantum state prepared in the initial step is too imprecise, we can simply discard this state before the computation is carried out and re-prepare it to make sure the output of the computation is accurate [26]. There are two MBQC schemes [22]: One-Way Quantum Computer (1WQC), also referred to as the cluster state model [27], and Teleportation Quantum Computation (TQC) [28]. The 1WQC method utilizes one-qubit measurements on a highly entangled state while the teleportation-based model requires joint (entangled) measurements [23]. It has been demonstrated that the 1WQC offers promising security features, as it leverages Blind Quantum Computation techniques [29]. In [30], authors demonstrate that any quantum algorithm can be implemented using single-qubit measurements on a cluster state. Furthermore, authors discuss how this approach can be used to implement various quantum algorithms, including Shor's and Grover's algorithms. In [31], a series of quantum schemes that exploit quantum entanglements in Greenberger–Horne–Zeilinger (GHZ) states to compute symmetric Boolean functions are proposed. In [32], the security of one of the schemes (scheme C) outlined in [31], which claimed to achieve secure multiparty computation for dishonest majority with

a threshold of $n-1$, was disproved. In [33], an refined SMC protocol based on [31] was proposed that rectifies the security flaws and offers enhanced efficiency.

Building upon the foundations outlined in scheme A from Loukopoulos and Browne [31], we propose a quantum-based two-party protocol in which the correlation of the GHZ state is exploited to compute binary Boolean functions, with the help of a third party. Our method introduces an additional random Z-phase rotation to the GHZ qubits to increase the protocol's security. The security and efficiency analyses show that this method achieves a higher security level while utilizing the same quantum resources and preserving the existing complexity. We implemented the proposed scheme on the IBM Qiskit platform and validated its feasibility through consistent and reliable results.

In the reminder of the paper, Sect. 2 overviews the computation of the logical NAND using the entanglement of the GHZ state. In Sect. 3, the computation of Boolean functions based on the AND operation is explained, providing the background for the contribution of the study. In Sect. 4, a quantum-based two-party protocol to compute binary Boolean functions is proposed. In Sect. 5, we implement our protocol on the IBM QisKit platform. In Sect. 6, we conduct analyses on the privacy, security, and efficiency aspects of the proposed approach. Lastly, Sect. 7 provides the conclusion for the paper.

2 Secure NAND computation

This section recalls the idea initially proposed by Anders and Browne [34], to compute the universal NAND function using the entanglements of the GHZ state $|GHZ\rangle = (|001\rangle - |110\rangle)/\sqrt{2}$. The computation is accomplished in a secure manner which means that three parties (say, Alice, Bob, and Charlie) with input bits a, b, and c compute the $NAND(a, b)$, while ensuring that no information about the individual inputs is disclosed to the other parties. Let us consider a scenario where the three qubits of the GHZ state are divided among three parties with each party holding one qubit of the entangled state. The parties measure the qubits in either σ_x or σ_y according to the input bits $a,\ b,\ c \in \{0, 1\}$. The third input is defined as $c = a \oplus b$. If the input bit is 0, they measure the qubit in σ_x basis, and if the bit is 1, the measurement is done in σ_y. There are four independent choices of inputs that form the following stabilizer equations for the GHZ state, initially outlined in [35]:

$$\begin{aligned}
\sigma_x \otimes \sigma_x \otimes \sigma_x \, |GHZ\rangle &= -\,|GHZ\rangle\,, \\
\sigma_x \otimes \sigma_y \otimes \sigma_y \, |GHZ\rangle &= -\,|GHZ\rangle\,, \\
\sigma_y \otimes \sigma_x \otimes \sigma_y \, |GHZ\rangle &= -\,|GHZ\rangle\,, \\
\sigma_y \otimes \sigma_y \otimes \sigma_x \, |GHZ\rangle &= +\,|GHZ\rangle\,.
\end{aligned} \tag{1}$$

The four equations can be expressed in a more concise way as:

$$\sigma_a \otimes \sigma_b \otimes \sigma_{(a\oplus b)} \, |GHZ\rangle = (-1)^{NAND(a,b)} \, |GHZ\rangle\,. \tag{2}$$

Equation (2) implies that the output of $NAND(a, b)$ is encoded into the eigenvalues of Eq. (1). If we assign the eigenvalues $+1$ and -1 with bit values 0 and 1, respectively, we obtain:

$$NAND(a, b) = M_a \oplus M_b \oplus M_{(a \oplus b)}, \tag{3}$$

where $\oplus$ is addition modulo 2; M_a, M_b, and $M_{(a \oplus b)} \in \{0, 1\}$ are the measurement outcome of parties. This result implies that if three parties share the GHZ state and perform measurements determined by their inputs, the parity of their measurement outcomes is equal to $NAND(a, b)$. In [36], the idea of Anders and Browne [34] was expanded so that instead of using different measurement bases (σ_x and σ_y), parties can perform a pre-rotation operation to the GHZ qubits, based on the values of a, b, and $a \oplus b$. Afterwards, by performing the σ_x measurement on the three qubits of the GHZ state, $NAND(a, b)$ is computed. In other words, if we represent the $\pi/2$ rotation along the Z-axis of the Bloch sphere by

$$U = R_z(\pi/2) = e^{-i\pi\,\sigma_z/4}, \tag{4}$$

then, performing the $U^\dagger$ rotation on the GHZ state will encode the parties' inputs in the resource state, leading to

$$|\psi\rangle = U^{\dagger a} U^{\dagger b} U^{\dagger (a \oplus b)} |GHZ\rangle\,. \tag{5}$$

The execution of the $U^\dagger$ operation on each qubit relies on the inputs of the respective parties (either a, b, $a \oplus b$). Specifically, if the input of a party is 1, the $U^\dagger$ operation is performed on the corresponding qubit. Conversely, if the input is 0, $U^\dagger$ is skipped for the corresponding qubit, resulting in the qubit retaining its initial state. This flexibility allows each party to decide whether or not to rotate its qubit based on the corresponding input. Afterwards, by measuring the three qubits of the GHZ state in σ_x basis and performing XOR among the measurement results, $NAND(\vec{a}, \vec{b})$ is obtained as shown in Eq. (3).

3 Computing boolean functions

In this section, we explain how binary Boolean functions can be computed using the secure NAND computation technique outlined in Sect. 2. We start by considering that any Boolean function $f(\vec{a}, \vec{b}) : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$, which operates on two n-bit strings $\vec{a}$ and $\vec{b}$ as inputs and returns a single binary output, can be computed by taking the inner product of two polynomial vectors $P_i(\vec{a})$ and $K_i(\vec{b})$ as follows [37]:

$$f(\vec{a}, \vec{b}) = \bigoplus_{i=1}^{m} P_i(\vec{a}) \cdot K_i(\vec{b}), \tag{6}$$

where $\vec{a} = (a_1, ..., a_n)$ and $\vec{b} = (b_1, ..., b_n)$ correspond to Alice's and Bob's input data, respectively. Equation (6) implies that to compute the Boolean function $f(\vec{a}, \vec{b})$,

m series of AND operations are required. Therefore, by resorting to the secure NAND computation technique outlined in Sect. 2 and subsequently converting it to an AND operation through a NOT operator, we can compute $f(\vec{a}, \vec{b})$. The maximum number of terms required for Eq. (6), denoted as m, is limited to 2^n, where n represents the input length.

According to Eq. (6), to compute $f(\vec{a}, \vec{b})$, three terms should be taken into account: $P_i(\vec{a})$, $K_i(\vec{b})$, and $P_i(\vec{a}).K_i(\vec{b})$. The first two polynomials $P_i(\vec{a})$ and $K_i(\vec{b})$ can be calculated locally by Alice and Bob, respectively. To compute the third term $P_i(\vec{a}).K_i(\vec{b})$, we use the scheme presented in Sect. 2. All that is needed is to find the result of $NAND\big(P_i(\vec{a}), K_i(\vec{b})\big)$ for each value of i. Afterwards, we obtain $P_i(\vec{a}) \cdot K_i(\vec{b}) = \neg NAND(P_i(\vec{a}), K_i(\vec{b}))$ by performing a NOT.

Note that the particular form of polynomials in Eq. (6) depend on the Boolean function being evaluated. For example, let us obtain the polynomials that are required to compute $OR(\vec{a}, \vec{b})$. The 2-bit OR function can be represented as [37]:

$$OR(\vec{a}, \vec{b}) = \vec{a} + \vec{b} + \vec{a} \cdot \vec{b}\,, \tag{7}$$

leading to:

$$\begin{aligned} OR(\vec{a}, \vec{b}) &= (a_1\ OR\ b_1) \cdot (a_2\ OR\ b_2) \\ &= (a_1 + b_1 + a_1.b_1) \cdot (a_2 + b_2 + a_2.b_2) \\ &= a_1a_2 + a_1b_2 + a_1a_2b_2 + b_1a_2 + b_1b_2 + b_1a_2b_2 + a_1b_1a_2 + a_1b_1b_2 \\ &\quad + a_1b_1a_2b_2 \\ &= \underbrace{a_1a_2}_{P_1} \cdot \underbrace{1}_{K_1} + \underbrace{(a_1 + a_1a_2)}_{P_2} \cdot \underbrace{b_2}_{K_2} + \underbrace{(1 + a_1 + a_2 + a_1a_2)}_{P_3} \cdot \underbrace{b_1b_2}_{K_3} \\ &\quad + \underbrace{(a_2 + a_1a_2)}_{P_4} \cdot \underbrace{b_1}_{K_4} \\ &= \sum_{i=1}^{4} P_i(a_1, a_2) \cdot K_i(b_1, b_2). \end{aligned} \tag{8}$$

In Eqs. (7) and (8), the symbols '+' and '·' represent the XOR and the logical AND, respectively. Equation (8) indicates that a 2-bit $OR(\vec{a}, \vec{b})$ function can be computed using the following vector of polynomials:

$$P(\vec{a}) = \begin{bmatrix} a_1a_2 \\ a_1 + a_1a_2 \\ 1 + a_1 + a_2 + a_1a_2 \\ a_2 + a_1a_2 \end{bmatrix}, \quad K(\vec{b}) = \begin{bmatrix} 1 \\ b_2 \\ b_1b_2 \\ b_1 \end{bmatrix}. \tag{9}$$

4 The Z-phase rotation two-party protocol for binary boolean function computation

Built upon the methodology outlined in [31], we propose a quantum-based two-party protocol to compute binary Boolean functions, with the help of a third party. Using the proposed protocol, two parties, referred to as Alice and Bob, can compute a binary Boolean function without disclosing any information about their private inputs. In [38], it was proven that attaining unconditionally secure two-party computations is not feasible. Consequently, the participation of a third party, referred to as Charlie, becomes necessary [38]. To address this security requirement, our protocol, originally designed for two-party scenarios, is extended to include the collaborative participation of Charlie.

In [31], a nearly private computation protocol (Scheme A) was proposed in which three parties share a GHZ state to perform SMC. Their scheme is described as "nearly private" because there are certain circumstances in which the third party, Charlie, can acquire information about the inputs of other participants. The primary source of information leakage in this scheme is that Charlie can simultaneously learn about the parity of Alice and Bob's inputs as well as the outcome of the protocol. The security of this protocol can be improved by preventing Charlie from gaining knowledge of the parity of Alice and Bob's private inputs, or the final output of the protocol, or both. We utilize a technique where we introduce an additional random Z-phase rotation on the GHZ qubits to obscure the outcome from Charlie. Using this technique, we reach a higher level of security, while using the same quantum resources and maintaining the existing complexity.

The proposed protocol advances through the following steps. First, Alice and Bob agree on the specific Boolean function and compute the required polynomial vectors P and K locally, using their inputs. Afterwards, the protocol is executed over m rounds. The number of rounds corresponds to size of polynomial vectors P and K. In each round i ($1 \leq i \leq m$), the secure computation of the AND operation between P_i and K_i is carried out as follows. First, three qubits that form a GHZ state are distributed among the three parties. An intriguing characteristic of the GHZ state is that the computation can take place even when the qubits are located in different places, allowing for a secure computation among the distributed parties. Next, each party performs $U^\dagger$ (i.e. a $-\pi/2$ rotation around the Z axis of the Bloch sphere) on its qubit, considering P_i, K_i, and $(P_i \oplus K_i)$:

$$|\psi_i\rangle = U^{\dagger P_i} \otimes U^{\dagger K_i} \otimes U^{\dagger (P_i \oplus K_i)} |GHZ_i\rangle . \tag{10}$$

Since all the information is encoded in the phase of the quantum state, performing an additional Pauli-Z rotation on the one of the qubits will obscure the outcome of the computation. Therefore, if Alice and Bob intend to obscure the output from Charlie, one of them (say Alice) has to perform a Pauli-Z rotation on its qubit considering a random bit. To share a random bit, a Bell state $|\varphi_i\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ is distributed between the two parties. Afterwards, each of them measures a qubit of the Bell state and stores the result in r_i. Next, Alice performs the Z-rotation on the first qubit leading to

$$\left|\psi_i'\right\rangle = {\sigma_z}^r \, U^{\dagger P_i} \otimes U^{\dagger K_i} \otimes U^{\dagger (P_i \oplus K_i)} \left|GHZ_i\right\rangle . \tag{11}$$

If $r_i = 0$, the σ_z operator is not applied to the qubit, whereas if $r = 1$, σ_z is applied to the qubit. The additional Z rotation enhances security by concealing the protocol outcome from Charlie. However, Alice and Bob can easily decode the actual output by executing a bit-flip. In the next step, parties measure their qubit in Pauli-X basis and store the results in classical bits m_{P_i}, m_{K_i}, and $m_{(P_i \oplus K_i)}$. Subsequently, the parties apply a NOT operator to their results and proceed to the next round. Once the protocol is executed for m rounds, the three parties compute $M_1 = \bigoplus_{i=1}^{m} \neg m_{P_i}$, $M_2 = \bigoplus_{i=1}^{m} \neg m_{K_i}$ and $M_3 = \bigoplus_{i=1}^{m} \neg m_{(P_i \oplus K_i)}$. Alice and Bob send their result to Charlie, who sums up all the classical bits as

$$f'(\vec{a}, \vec{b}) = M_1 \oplus M_2 \oplus M_3, \tag{12}$$

and sends the results to Alice and Bob. Following this, Alice and Bob derive the actual output by executing XOR between the random bit $r = \oplus_{i=1}^{m} \; r_i$ and the classical bit received from Charlie, as follows:

$$f(\vec{a}, \vec{b}) = r \oplus f'(\vec{a}, \vec{b}). \tag{13}$$

As outlined in the protocol description, the demand for quantum resources increases with n, which means that as the length of the parties' input bits extends, a greater amount of quantum resources becomes necessary. Protocol 1 provides an overview of these procedures. In the next section, the computation of a 2-bit OR function is explained through an implementation in the IBM QisKit to illustrate the protocol.

5 QisKit implementation

Universal fault-tolerant quantum computers are not available. Therefore, simulation platforms such as IBM QisKit [39] and Paddle Quantum [40] are employed for the design and implementation of quantum algorithms. Qiskit is a software framework developed by IBM that enables users to simulate and execute quantum programs on both simulation platforms and real quantum computers. In [41], authors utilized Qiskit to implement several quantum gates, such as the Hadamard gate, the Controlled-Not (CNOT) gate, and the $\pi/2$-phase gate, based on MBQC approach. In this section, we design a circuit for the proposed protocol and explain its implementation in the Qiskit IBM platform for a particular scenario involving a 2-bit $OR(\vec{a}, \vec{b})$ function.[1]

Figure 1 depicts the quantum circuit implementing the proposed protocol. Circuit preparation includes three steps: A, B, and C. In the first step, A, a GHZ state is prepared starting from three qubits q_0, q_1, and q_2 with the initial state $|0\rangle$. In step B,

[1] The implementation code for the proposed quantum SMC protocol is accessible in GitHub repository "Quantum-SMC," located at https://github.com/Quantum-SMC.

Protocol 1 Z-Phase Rotation Quantum SMC Protocol

Inputs: Input strings $\vec{a}$ for Alice and $\vec{b}$ for Bob.
Outputs: $f(\vec{a}, \vec{b})$ for Alice and Bob.

1. Starting from $i = 1$, repeat steps 2-9 for each term.
2. Given the particular function being computed, Alice and Bob locally calculate $P_i(\vec{a})$ and $K_i(\vec{b})$.
3. In order to generate a privately shared random bit r_i, a Bell state $|\varphi_i\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ is distributed between Alice and Bob. Subsequently, each measures a qubit of the Bell state and stores the result as r_i.
4. Alice and Bob send to Charlie the bit values $P_i(\vec{a}) \oplus r_i$ and $K_i(\vec{b}) \oplus r_i$, through a secure classical channel, respectively.
5. Charlie computes $(P_i(\vec{a}) \oplus r_i) \oplus (K_i(\vec{b}) \oplus r_i) = P_i(\vec{a}) \oplus K_i(\vec{b})$.
6. A three-qubit GHZ state is distributed among the parties as $|GHZ_i\rangle = (|001\rangle - |110\rangle)/\sqrt{2}$.
7. Alice, Bob, and Charlie individually apply the operations ${\sigma_z}^{r_i}\, {U^\dagger}^{P_i(\vec{a})}$, ${U^\dagger}^{K_i(\vec{b})}$, and ${U^\dagger}^{P_i(\vec{a}) \oplus K_i(\vec{b})}$ to their respective qubits in the GHZ state.
8. Next, parties measure their qubits in Pauli-X basis and store the measurement results m_{P_i}, m_{K_i}, and $m_{(P_i \oplus K_i)}$.
9. The three parties perform a NOT operator on the classical results to compute $\neg m_{P_i}$, $\neg m_{K_i}$, and $\neg m_{(P_i \oplus K_i)}$.
10. Once $i = m$, Alice and Bob individually perform the XOR operation on their measurement results ($M_1 = \bigoplus_{i=1}^{m} \neg m_{P_i}$ and $M_2 = \bigoplus_{i=1}^{m} \neg m_{K_i}$) and send them to Charlie.
11. Charlie then sums the XOR of Alice and Bob's outcomes with his own measurement results. He then reveals the value of $f'(\vec{a}, \vec{b}) = M_1 \oplus M_2 \oplus M_3$, where $M_3 = \bigoplus_{i=1}^{m} \neg m_{(P_i \oplus K_i)}$.
12. Alice and Bob perform the last XOR operation to retrieve the result of the computation as $f(\vec{a}, \vec{b}) = r \oplus f'(\vec{a}, \vec{b})$, with $r = \bigoplus_{i=1}^{m} r_i$.

the qubits are rotated according to the parties' private inputs and a random bit r. The rotational operations $R_z(\pi)$ and $R_z(-\pi/2)$ correspond to the V and $U^\dagger$ operations, respectively. Next, in step C, qubits are measured in the Hadamard basis. The default measurement in QisKit is performed in the Z-basis. However, by incorporating an H gate prior to the measurement operator, we can measure the qubit in the X-basis. The measurement result of each qubit is stored in a classical register of the QisKit environment. To store the measurement outcomes, three classical registers, C_0, C_1, and C_2, are used to store the measurement results of q_0, q_1, and q_2, respectively.

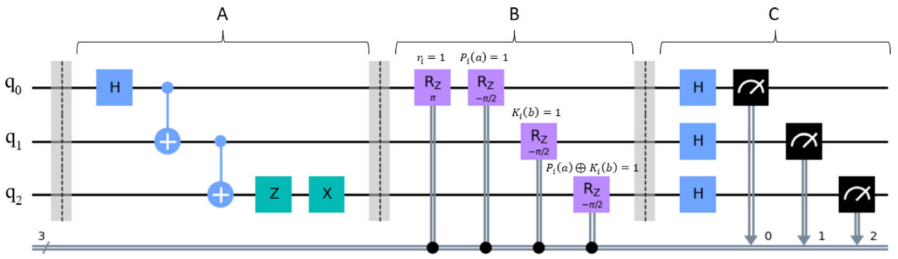

Fig. 1 Quantum circuit for the proposed protocol in each round i. The q_0, q_1, and q_2 are three initial qubits with state $|0\rangle$. The label A represents the preparation of the GHZ state. Label B indicates the rotation of qubits with respect to the bits r, P_i, K_i, and $P_i \oplus K_i$. Note that the rotation gates in label B are exclusively applied when the bit values are equal to 1; otherwise, they are omitted from the circuit. Label C represents qubit measurements in the Hadamard basis. Labels 'H', '+', Z, 'X', and 'R_z' identify the Hadamard, controlled-X, Pauli-Z, Pauli-X, and Z-rotation gates, respectively

In our simulated experiments, we assume with no loss of generality that the three parties, Alice, Bob, and Charlie, with 2-bit inputs $\vec{a} = (1, 0)$, $\vec{b} = (1, 0)$, and $\vec{a} \oplus \vec{b} = (0, 0)$ intend to compute $OR(\vec{a}, \vec{b}) = OR\,(OR(1, 1), OR(0, 0))$, which yields to output 1. Alice and Bob share random bits $\vec{\mathbf{r}} = (0, 1, 1, 0)$ leading to $r = \bigoplus_{i=1}^{4} r_i = 0$. Considering Eq. (8), Alice and Bob compute the polynomials $\vec{P} = (1, 0, 1, 0)$ and $\vec{K} = (0, 1, 1, 0)$, which correspond to the OR function. Figure 2 illustrates the measurement outcomes in four rounds of execution. For a three-qubit GHZ state, the possible measurement outcomes are 000, 001, 010, 011, 100, 101, 110, 111. Note that by performing measurement, the qubits collapse from a superposition state into a classical state with the highest probability (either 0 or 1). The eight potential outcomes occur with nearly equal probabilities, a logical outcome of the randomness of quantum measurement. To compute 2-bit OR, four rounds of computation are carried out, and within each round 1000 shots are executed. As shown in Fig. 2, the measurement outcomes with the highest probabilities for rounds 1 to 4 are 001, 011, 110, and 000, respectively, leading to following outcomes for each party:

$$Outcomes \begin{cases} M_1 = \neg 0 \oplus \neg 0 \oplus \neg 1 \oplus \neg 0 = 1 & \text{Alice} \\ M_2 = \neg 0 \oplus \neg 1 \oplus \neg 1 \oplus \neg 0 = 0 & \text{Bob} \\ M_3 = \neg 1 \oplus \neg 1 \oplus \neg 0 \oplus \neg 0 = 0 & \text{Charlie} \end{cases} \tag{14}$$

Alice and Bob send their summation bits to Charlie, who then performs an XOR operation on the obtained bits, resulting in $f'(\vec{a}, \vec{b}) = 1 \oplus 0 \oplus 0 = 1$. This outcome is then transmitted to Alice and Bob, who calculate the final result by performing an XOR operation between the received classical bit and the shared random bit, yielding $f(\vec{a}, \vec{b}) = 1 \oplus 0 = 1$. The obtained result confirm the correctness of the protocol. The simulation results were performed on the *'qasm_simulator'* within Google Colab, Ubuntu 20.04.6 LTS, Python 3.10.12, and QisKit-0.43.2. Implementations are carried out on the ASUS Zenbook 14 UX425E laptop with 4 cores and an 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80 GHz processor, and 16 GB of RAM.

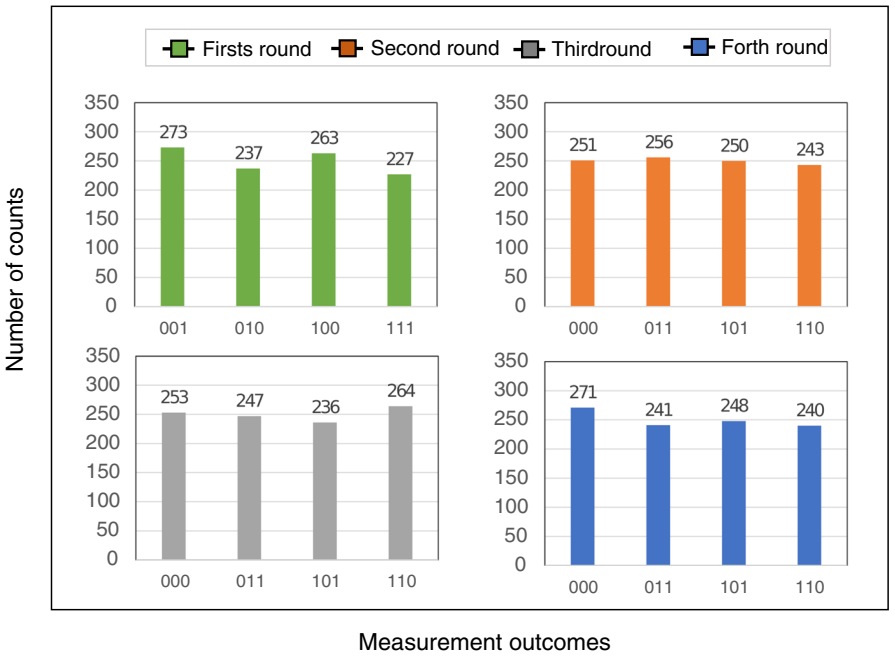

Fig. 2 Measurement results of the quantum circuit demonstrated in Fig. 1, considering a particular scenario involving a 2-bit $OR(\vec{a}, \vec{b})$ function. The simulation are performed on *'qasm_simulator'* simulator. The circuit is run over 4 rounds with 1000 shots

6 Result and discussion

In this section, we provide security, privacy, and efficiency analyses of the proposed protocol. Additionally, we provide a comparative analysis between quantum SMC protocols and the protocol proposed in this work.

6.1 Privacy analysis

To evaluate the privacy of the proposed protocol, we examine the data leakage in each step as follows:

1. Computation of $P(\vec{a})$ and $K(\vec{b})$: as the computation is conducted locally, there is no disclosure of any information regarding the inputs.
2. Qubit measurement by parties: no information is leaked.
3. Transmission of $(P_i \oplus r_i)$ and $(K_i \oplus r_i)$ to Charlie: resorting to the random bit r ensures that Charlie remains unaware of any details regarding the inputs. Despite using a random bit, Charlie gains knowledge about the parity of the inputs, at this stage.
4. Qubit rotation by parties: no information is leaked.

Once the value of the function is announced, participating parties remain uninformed regarding each other's inputs.

6.2 Security analysis

The security of our protocol is derived directly from the principles of secure NAND computation outlined in [34] which rely on fundamental principles of quantum mechanics, such as the no-cloning theorem and the inability to measure certain quantum properties without disturbing the system. These features make it extremely difficult for an adversary to extract information from quantum systems without leaving detectable traces. As a result, entangled states and their quantum correlations offer unique opportunities for achieving secure communication. Consider the security against the attack from a party (for example Alice). If Alice wants to learn about Bob's input, she needs to intercept the bit value $K_i(\vec{b}) \oplus r_i$ that is transmitted between Bob and Charlie at the step 4 of Protocol 1. However, since the transmission occurs over a secure classical channel, Alice fails to acquire any information about the bit value. Furthermore, if Charlie aims to retrieve the function output, his attempt will be unsuccessful due to his lack of knowledge concerning the random classical bits r_i.

The protocol lacks security against a coalition attack because Charlie possesses knowledge of the parity of input bits at each stage. This implies that if Charlie forms a coalition with either Alice or Bob, they can acquire information about the input of the other party. Consequently, the protocol can only be considered secure with a threshold of $th = 1$. The protocol is passively secure, which means that while the adversary can attempt to gather information from others, they are not permitted to deviate from the specified protocol execution.

6.3 Efficiency analysis

Efficiency analysis involves three factors: quantum resources, communication complexity, and round complexity of the protocol. To compute Boolean functions as described in Eq. (6), two types of quantum resources are required: Bell and GHZ states. The application of Bell states can be substituted with a standard Quantum Key Distribution protocol to enable the sharing of random bits between Alice and Bob. The necessity for these quantum resources aligns with the requirements of the SMC protocol introduced in [31], which similarly emphasizes Boolean function evaluation. However, our scheme surpasses security level compared to the protocol outlined in [31] due to the use of an additional Z-phase rotation technique, concealing the output from Charlie. The communication cost, that is the number of bits transmitted among parties, is $2n$ bits, for each rounds of protocol execution. The round complexity of our protocol which refers to the number of rounds required for the execution of the protocol is 4.

Table 1 shows the functions to be computed, the required quantum resources, and the communication and the round complexity for various SMC protocols. While some SMC protocols listed in Table 1 utilize fewer quantum resources (single qubit), this research focuses on a different and more general type of functions. Furthermore, the use of single qubit resulted in increased communication costs when compared to the GHZ state. Although the use of quantum resources, such as GHZ state, can increase with the input size n, the communication cost remains minimal.

Table 1 Comparison of different quantum SMC protocols

QSMC protocols	Computed function	RoundCx	CommCx	Quantum resources
Ref. [31]	Boolean functions	4	$\mathcal{O}(2n)$	GHZ state + Bell state (Prot. A)
Ref. [15]	Pairwise AND	2	$\mathcal{O}(2n^2)$	Single qubit
Ref. [16]	N-tuple pairwise AND	2	$\mathcal{O}(2n^2)$	Single qubit
Ref. [17]	N-variable polynomials	3	$\mathcal{O}(\ell n^2)$	Single qudit (Prot. Γ_1)
				Entangled state (Prot. Γ_2)
Ref. [18]	Summation function	1	$\mathcal{O}(1)$	Entangled state
This work	Boolean functions	4	$\mathcal{O}(2n)$	GHZ state + Bell state

ℓ indicates the number of monomails. RoundCx and CommCx denote round complexity and communication complexity, respectively

7 Conclusions

We have addressed the challenges faced by classical Secure Multiparty Computation in terms of security and efficiency. By leveraging the entanglements of the GHZ state, we proposed a quantum-based two-party protocol to compute binary Boolean functions, with the help of a third party. Although this work primarily focuses on the scenarios involving three parties, the prospect of extending the protocol to accommodate an arbitrary number of n participants holds considerable promise, contributing to the development of SMC protocols in quantum computing. We conducted privacy, security, and efficiency analyses for the proposed protocol. Our results demonstrate the robust security offered by our quantum-based approach, while also highlighting its communication efficiency. Furthermore, we have implemented our protocol on the IBM QisKit platform and obtained experimental results confirming the feasibility and practicality of our approach.

Acknowledgements This work is funded by Fundação para a Ciência e a Tecnologia (FCT)/MCTES through national funds and when applicable co-funded EU funds. The work of Zeinab Rahmani was supported by the FCT through Fundo Social Europeu and through national funds, by the European Regional Development Fund (FEDER), through the Competitiveness and Internationalization Operational Programme (COMPETE 2020) of the Portugal 2020 framework under the International Iberian Nanotechnology Laboratory (INL) Quantum Portugal Initiative PhD Grant with Ref. SFRH/BD/151111/2021. The work of Armando N. Pinto was supported by QuantaGenomics project funded within the QuantERA II Programme that has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 101017733, and with funding organisations, The Foundation for Science and Technology—FCT (QuantERA/0001/2021), Agence Nationale de la Recherche—ANR, and State Research Agency—AEI. The work of Luis S. Barbosa, was supported by IBEX project which was funded by National Funds through FCT and I.P. (Portuguese Foundation for Science and Technology) with Ref. https://doi.org/10.54499/PTDC/CCI-COM/4280/2021.

Author Contributions Zeinab Rahmani: Conceptualization, Investigation, Methodology, Software, Writing (original draft), Validation. Armando Humberto Moreira Nolasco Pinto: Supervision, Funding acquisition, Project administration, Resources, Writing (review and editing). Luis Manuel Dias Coelho Soares Barbosa: Supervision, Funding acquisition, Project administration, Resources, Writing (review and editing).

Funding Open access funding provided by FCT|FCCN (b-on).

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest to disclose.

References

1. Rahmani, Z., Barbosa, L.S., Pinto, A.N.: Quantum privacy-preserving service for secure lane change in vehicular networks. IET Quantum Commun. **4**(3), 103–111 (2023). https://doi.org/10.1049/qtc2.12059
2. Rahmani, Z., Barbosa, L., Pinto, A.N.: Collision warning in vehicular networks based on quantum secure multiparty computation. In: II Workshop de Comunicação e Computação Quântica WQuantum, pp. 1–6 (2022). https://doi.org/10.5753/wquantum.2022.223569
3. Santos, M.B., Gomes, A.C., Pinto, A.N., Mateus, P.: Private computation of phylogenetic trees based on quantum technologies. IEEE Access **10**, 38065–38088 (2022). https://doi.org/10.1109/ACCESS.2022.3158416
4. Jingzhong, W., Yue, Z., Haibin, L.: Electronic voting protocol based on ring signature and secure multi-party computing. In: 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 50–55 (2020). IEEE. https://doi.org/10.1109/CyberC49757.2020.00018
5. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. **41**(2), 303–332 (1999). https://doi.org/10.1137/S0036144598347011
6. Yao, A.C.: Protocols for secure computations. In: 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982), pp. 160–164 (1982). IEEE. https://doi.org/10.1109/SFCS.1982.38
7. Beaver, D., Micali, S., Rogaway, P.: The round complexity of secure protocols. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing. STOC '90, pp. 503–513. Association for Computing Machinery, New York (1990). https://doi.org/10.1145/100216.100287
8. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation, pp. 351–371. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3335741.3335756
9. Bendlin, R., Damgård, I., Orlandi, C., Zakarias, S.: Semi-homomorphic encryption and multiparty computation. In: Paterson, K.G. (ed.) Advances in cryptology—EUROCRYPT 2011. Lecture Notes in Computer Science, vol. 6632, pp. 169–188. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_11
10. Damgård, I., Pastro, V., Smart, N., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology—CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 643–662. Springer, Berlin, Heidelberg (2012).https://doi.org/10.1007/978-3-642-32009-5_38
11. Damgård, I., Keller, M., Larraia, E., Pastro, V., Scholl, P., Smart, N.P.: Practical covertly secure MPC for dishonest majority—or: breaking the spdz limits. In: Crampton, J., Jajodia, S., Mayes, K. (eds.) Computer Security—ESORICS 2013. Lecture Notes in Computer Science, vol. 8134, pp. 1–18. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40203-6_1
12. Keller, M., Orsini, E., Scholl, P.: Mascot: Faster malicious arithmetic secure computation with oblivious transfer. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. CCS '16, pp. 830–842. Association for Computing Machinery, New York (2016).https://doi.org/10.1145/2976749.2978357
13. Bennett, C.H., Brassard, G.: Quantum cryptography: public key distribution and coin tossing. Theoret. Comput. Sci. **560**, 7–11 (2014). https://doi.org/10.1016/j.tcs.2014.05.025
14. Attema, T., Bosman, J.W., Neumann, N.M.: Optimizing the decoy-state bb84 qkd protocol parameters. Quantum Inf. Process. **20**(4), 154 (2021). https://doi.org/10.1007/s11128-021-03078-0
15. Clementi, M., Pappa, A., Eckstein, A., Walmsley, I.A., Kashefi, E., Barz, S.: Classical multiparty computation using quantum resources. Phys. Rev. A **96**, 062317 (2017). https://doi.org/10.1103/PhysRevA.96.062317
16. Cao, H., Ma, W., Liu, G., Lü, L., Xue, Z.-Y.: Quantum secure multiparty computation with symmetric boolean functions*. Chin. Phys. Lett. **37**(5), 050303 (2020). https://doi.org/10.1088/0256-307X/37/5/050303
17. Lu, C., Miao, F., Hou, J., Su, Z., Xiong, Y.: Secure multi-party computation with a quantum manner. J. Phys. A: Math. Theor. **54**(8), 085301 (2021). https://doi.org/10.1088/1751-8121/ab9aea
18. Yang, H.-Y., Ye, T.-Y.: Secure multi-party quantum summation based on quantum Fourier transform. Quantum Inf. Process. **17**(6), 129 (2018). https://doi.org/10.1007/s11128-018-1890-1
19. Shi, R.-H., Mu, Y., Zhong, H., Cui, J., Zhang, S.: Secure multiparty quantum computation for summation and multiplication. Sci. Rep. **6**(1), 19655 (2016). https://doi.org/10.1038/srep19655

20. Ji, Z., Zhang, H., Wang, H., Wu, F., Jia, J., Wu, W.: Quantum protocols for secure multi-party summation. Quantum Inf. Process. **18**, 1–19 (2019). https://doi.org/10.1007/s11128-018-2141-1
21. Cai, X.-Q., Wang, T.-Y., Wei, C.-Y., Gao, F.: Cryptanalysis of secure multiparty quantum summation. Quantum Inf. Process. **21**(8), 285 (2022). https://doi.org/10.1007/s11128-022-03638-y
22. Jozsa, R.: An introduction to measurement based quantum computation. NATO Sci. Ser., III: Comput. Syst. Sci. Quantum Inf. Process.-From Theory Exp. **199**, 137–158 (2006). https://doi.org/10.48550/arXiv.quant-ph/0508124 https://doi.org/10.48550/arXiv.quant-ph/0508124 https://doi.org/10.48550/arXiv.quant-ph/0508124 https://doi.org/10.48550/arXiv.quant-ph/0508124 https://doi.org/10.48550/arXiv.quant-ph/0508124
23. Briegel, H.J., Browne, D.E., Dür, W., Raussendorf, R., Nest, M.V.: Measurement-based quantum computation. Nat. Phys. **5**(1), 19–26 (2009). https://doi.org/10.1038/nphys1157
24. Li, W., Ma, X., Lee, Y.-H., Zhang, Y., Gu, Y.: Finding new multipartite entangled resources for measurement-based quantum computation. Quantum Inf. Process. **22**(3), 130 (2023). https://doi.org/10.1007/s11128-023-03870-0
25. Raussendorf, R., Briegel, H.J.: A one-way quantum computer. Phys. Rev. Lett. **86**, 5188–5191 (2001). https://doi.org/10.1103/PhysRevLett.86.5188
26. Nielsen, M.A., Dawson, C.M.: Fault-tolerant quantum computation with cluster states. Phys. Rev. A **71**, 042323 (2005). https://doi.org/10.1103/PhysRevA.71.042323
27. Briegel, H.J., Raussendorf, R.: Persistent entanglement in arrays of interacting particles. Phys. Rev. Lett. **86**, 910–913 (2001). https://doi.org/10.1103/PhysRevLett.86.910
28. Nielsen, M.A.: Quantum computation by measurement and quantum memory. Phys. Lett. A **308**(2–3), 96–100 (2003). https://doi.org/10.1016/S0375-9601(02)01803-0
29. Broadbent, A., Fitzsimons, J., Kashefi, E.: Universal blind quantum computation. In: 2009 50th Annual IEEE Symposium on Foundations of Computer Science, pp. 517–526 (2009). https://doi.org/10.1109/FOCS.2009.36
30. Raussendorf, R., Browne, D.E., Briegel, H.J.: Measurement-based quantum computation on cluster states. Phys. Rev. A **68**, 022312 (2003). https://doi.org/10.1103/PhysRevA.68.022312
31. Loukopoulos, K., Browne, D.E.: Secure multiparty computation with a dishonest majority via quantum means. Phys. Rev. A **81**, 062336 (2010). https://doi.org/10.1103/PhysRevA.81.062336
32. Li, Y.-B., Wen, Q.-Y., Qin, S.-j.: Comment on "secure multiparty computation with a dishonest majority via quantum means". Phys. Rev. A **84**, 016301 (2011). https://doi.org/10.1103/PhysRevA.84.016301
33. Li, Y.-B., Wen, Q.-Y., Qin, S.-J.: Improved secure multiparty computation with a dishonest majority via quantum means. Int. J. Theor. Phys. **52**, 199–205 (2013). https://doi.org/10.1007/s10773-012-1319-z
34. Anders, J., Browne, D.E.: Computational power of correlations. Phys. Rev. Lett. **102**, 050502 (2009). https://doi.org/10.1103/PhysRevLett.102.050502
35. Mermin, N.D.: Quantum mysteries revisited. Am. J. Phys. **58**(8), 731–734 (1990). https://doi.org/10.1119/1.16503
36. Dunjko, V., Kapourniotis, T., Kashefi, E.: Quantum-enhanced secure delegated classical computing. Quantum Inf. Comput. **16**(1–2), 61–86 (2016). https://doi.org/10.26421/QIC16.1-2-5
37. Dam, W.: Implausible consequences of superstrong nonlocality. Nat. Comput. **12**(1), 9–12 (2012). https://doi.org/10.1007/s11047-012-9353-6
38. Lo, H.-K.: Insecurity of quantum secure computations. Phys. Rev. A **56**, 1154–1162 (1997). https://doi.org/10.1103/PhysRevA.56.1154
39. Anis, M.S., Abraham, H., AduOffei, R.A., Agliardi, G., Aharoni, M., Akhalwaya, I.Y., Aleksandrowicz, G., Alexander, T., Amy, M., Anagolum, S.: Qiskit: an open-source framework for quantum computing. Qiskit/qiskit (2021). https://doi.org/10.1088/1742-6596/2438/1/012148
40. Paddle Quantum (2020). https://github.com/PaddlePaddle/Quantum
41. Kashif, M., Al-Kuwari, S.: Qiskit as a simulation platform for measurement-based quantum computation. In: 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C), pp. 152–159 (2022). https://doi.org/10.1109/ICSA-C54293.2022.00037