Private Computation of Boolean Functions Using Single Qubits

Zeinab Rahmani[1,2,3], Armando N. Pinto[1,2], and Luis S. Barbosa[3,4,5]

[1] Department of Electronics, Telecommunications and Informatics, University of Aveiro, Aveiro, Portugal
{zeinab.rahmani,anp}@ua.pt
[2] Instituto de Telecomunicações, Aveiro, Portugal
[3] International Iberian Nanotechnology Laboratory, Braga, Portugal
[4] Department of Computer Science, University of Minho, Braga, Portugal
lsb@di.uminho.pt
[5] Institute of Systems and Computer Engineering, Technology and Science, Braga, Portugal

Abstract. Secure Multiparty Computation (SMC) facilitates secure collaboration among multiple parties while safeguarding the privacy of their confidential data. This paper introduces a two-party quantum SMC protocol designed for evaluating binary Boolean functions using single qubits. Complexity analyses demonstrate a reduction of 66.7% in required quantum resources, achieved by utilizing single qubits instead of multi-particle entangled states. However, the quantum communication cost increased by 40% due to the amplified exchange of qubits among participants. Furthermore, we bolster security by performing additional quantum operations along the y-axis of the Bloch sphere, effectively hiding the output from potential adversaries. We design the corresponding quantum circuit and implement the proposed protocol on the IBM Qiskit platform, yielding reliable outcomes.

Keywords: Secure multiparty computation · Boolean functions · IBM Qiskit.

1 Introduction

In today's data-driven age, information serves as a vital resource for scientific developments. However, the increasing flow of information also poses significant privacy challenges. Secure Multiparty Computation (SMC) has emerged as a promising tool that offers a robust solution for collaborative computation while ensuring data privacy. Within SMC a group of N parties $\{P_1, P_2, ..., P_N\}$, each having a secret input a_i $(1 \leq i \leq N)$, collaboratively calculates a function $f(a_1, a_2, ..., a_N)$, without leaking any information about their secret inputs to others. The significance of SMC extends across diverse domains such as machine learning [1], health care [2], and vehicular networks [3]. However, the majority of conventional SMC implementations rely on public-key cryptography, leading to substantial computational and communication costs, which challenges achieving the desired levels of security and efficiency. Moreover, classical SMC implementations are not secure in the rise of quantum computers that are considered a serious threat for current cryptographic protocols such as Rivest-Shamir-Adleman (RSA) [4], Diffie-Hellman Key Exchange (DHKE) [5], and Elliptic Curve Cryptography (ECC) [6].

Andrew Yao introduced the concept of SMC in his 1982 paper [7]. Subsequently, more sophisticated classical SMC protocols designed for engaging more than two parties were proposed [8–15]. Despite the huge development of classical SMC, its widespread adoption is being delayed due to inefficient algorithms. To tackle the problems of classical SMC, quantum-based approaches were implemented [16–19]. The first approach utilizes quantum communication technologies such as Quantum Key Distribution (QKD) [20], Quantum Oblivious Transfer (QOT) [21], and Quantum Random Number Generation (QRNG) [22] along with cryptographic primitives, to achive SMC. These technologies harness the principles of quantum mechanics, such as no-cloning, to establish secure communication channels and exchange information without the risk of interception or tampering. This quantum communication-based approach exhibits a high Technology Readiness Level (TRL), indicating a significant degree of maturity and readiness for practical application. For instance in [3], QKD and QOT technologies were integrated to classical Faster Malicious Arithmetic Secure Computation with Oblivious Transfer (MASCOT) [23] protocol to provide a lane change service in vehicular networks. In [2], authors computed phylogenetic trees of SARS-CoV-2 genomes by integrating QRNG, QKD and QOT with the Yao protocol. The second approach explores the implementation of SMC within the quantum computing framework. Within this framework, researchers explored different quantum resources such as entangled particles [16, 24–27] and single qubits [17,28,29], to implement more efficient SMC protocols. For instance, in [16, 27], multiple schemes for private computation of Boolean functions are proposed resorting to the entanglement of Greenberger-Horne-Zeilinger (GHZ) state through Measurement-Based Quantum Computing (MBQC) [30]. In [26], a two-party protocol for secure comparison is proposed, resorting to n sequences of three-particle entangled states. In [31], authors introduced a quantum summation protocol that utilizes the multi-particle GHZ state. In [28], the secure Manhattan distance between two points by performing a phase-shift operation on a sequence of qubits is computed. In [29], a protocol for Privacy Set Intersection Cardinality is proposed, in which a sequence of n qubits is used to compute the intersections between parties' private sets without disclosing any details about the content of their respective sets. In [17], authors suggested a new approach to compute pairwise AND function by employing single qubit measurements and linear classical computing.

We introduce a generic approach for private computation of binary Boolean functions using single qubits within the context of quantum computing framework. Our approach presents a two-party secure multiparty computation protocol, augmented by the involvement of a third party. The efficiency and complexity analyses demonstrate a 66.7% increase in efficiency while maintaining the same computation overheads. However, our method requires 40% more communication resources due to the increased qubit exchange among participants. Additionally, a quantum operation V is employed to improve the security level by hiding the output from untrusted participants. We implement the proposed protocol on the IBM Qiskit platform and evaluate its correctness.

In the reminder of the paper, Section 2 overviews an approach in which the computation of the pairwise AND is done using a single qubit. In Section 3, a quantum-based SMC protocol to compute binary Boolean functions is proposed. In Section 4, we design the corresponding quantum circuit and implement the proposed protocol in the IBM Qiskit platform. In Section 5, we provide privacy, security, and efficiency analyses of the proposed scheme. Finally, Section 6 concludes the paper.

2 Pairwise AND Computation

This section reviews the private computation of pairwise AND initially proposed by [17] which serves as the basis for our protocol. This approach allows multiple participants to collectively compute the pairwise AND of their inputs without exposing any information about their inputs. Consider the pairwise AND function written as:

$$f(x_1, ..., x_n) = \bigoplus_{j=1}^{n-1} \Big(x_{j+1} \cdot (\bigoplus_{i=1}^{j} x_i) \Big), \tag{1}$$

where $\bigoplus$ is addition modulo 2 (XOR) and "." denotes the AND operation. This function computes the pairwise AND operation for each pair of values in the sequence, and then performs a bitwise XOR on the results.

Suppose that n parties with input bits $x_1, x_2, ..., x_n$ want to compute pairwise AND of their private inputs with the assistance of a server. In the initial step, a secret shared random bit $r = \overset{n}{\underset{i=1}{\oplus}} r_i$ is distributed among parties such that each party i holds r_i. Let us specify the $-\pi/2$ rotation and the π rotation around the y axis of the Bloch sphere as:

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

and

$$V = R_y(\pi) = e^{-i\pi\sigma_y/2}. \tag{3}$$

Initially, the server prepares a qubit $|0\rangle$ and sends it to the first party. Party P_1 performs the operations $V^{r_1}U^{x_1}$ on the qubit based on the input x_1 and random bit r_1. The rotations are carried out in such a way that if the bit value is 0, the operation U (V) is applied to the qubit, and if the bit value is 1, the operation U (V) is omitted and the qubit remains unchanged. The modified qubit is then forwarded to the subsequent party P_2, where the rotations $V^{r_2}U^{x_2}$

are applied. This sequence of actions iteratively repeats until all the parties have applied their rotations to the qubit. Employing an XOR routine detailed at [17], the parties compute the XOR of their peers' private inputs $\oplus_i x_i$. Subsequently, one of the parties performs the $(U^\dagger)^{\oplus_i x_i}$ operation on the qubit resulting in

$$|r \oplus f\rangle = (U^\dagger)^{\oplus_i x_i} \underbrace{V^{r_n} U^{x_n}}_{\mathcal{P}_n} ... \underbrace{V^{r_2} U^{x_2}}_{\mathcal{P}_2} \underbrace{V^{r_1} U^{x_1}}_{\mathcal{P}_1} |0\rangle \,. \tag{4}$$

Afterwards, the qubit is returned to the server who measures it in the computational basis, revealing the classical outcome $(r \oplus f)$. Exploiting the XOR routine, parties locally compute r by XORing all the random bits r_i of their peers, and retrieve the final output as follows:

$$f(x_1, ..., x_n) = r \oplus (r \oplus f). \tag{5}$$

In the next section, we use the result of Eq. (5) to compute binary Boolean functions in a secure multiparty manner.

3 A Two-Party SMC Protocol for Boolean Function Computation

In this section, we propose a quantum-based SMC protocol to compute binary Boolean functions using single qubits. As outlined in [32], a Boolean function can be computed using two vectors $P_i(a)$ and $K_i(b)$ as:

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

where $\boldsymbol{a} = (a_1, ..., a_n)$ and $\boldsymbol{b} = (b_1, ..., b_n)$ represent Alice's and Bob's input data; P_i represents polynomials depending on $\boldsymbol{a} \in \{0,1\}^n$ and K_i represents monomials depending on $\boldsymbol{b} \in \{0,1\}^n$. The right-hand side of Eq. (6) indicates that $f(\boldsymbol{a}, \boldsymbol{b})$ can be computed using the secure AND computation method explained in Section 2. Depending on the particular Boolean function under examination, the polynomials described by Eq. (6) are computed as follows: consider the Equivalence function $EQ(\boldsymbol{a}, \boldsymbol{b})$ in which the output of the computation is true if the two statements or conditions are equivalent. The polynomials needed for a 2-bit Equivalence function $EQ(\boldsymbol{a}, \boldsymbol{b})$ can be computed as follows [32]:

$$EQ(\boldsymbol{a}, \boldsymbol{b}) = 1 + \boldsymbol{a} + \boldsymbol{b} \,, \tag{7}$$

therefore,

$$\begin{aligned}
EQ(\boldsymbol{a},\boldsymbol{b}) =& EQ(a_1,b_1)\,.\,EQ(a_2,b_2)\\
=&(1+a_1+b_1)\,.\,(1+a_2+b_2)\\
=&1+1.a_2+1.b_2+a_1.1+a_1a_2+a_1b_2+b_1.1+b_1a_2+b_1b_2\\
=&\underbrace{(1+a_1+a_2+a_1a_2)}_{P_1}.\underbrace{1}_{K_1}+\underbrace{1}_{P_2}.\underbrace{b_1b_2}_{K_2}+\underbrace{(1+a_2)}_{P_3}.\underbrace{b_1}_{K_3}+\underbrace{(1+a_1)}_{P_4}.\underbrace{b_2}_{K_4}\\
=&\sum_{i=1}^{4}P_i(a_1,a_2).K_i(b_1,b_2). \qquad (8)
\end{aligned}$$

In Eqs. (7) and (8), addition and multiplication are the XOR, and the logical AND, respectively. Equation (8) indicates that a 2-bit Equivalence function $EQ(\boldsymbol{a},\boldsymbol{b})$ can be computed using the following vectors of polynomials:

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

Note that the size of P and K can grow with n implying a greater demand for quantum resources to compute the desired function.

The proposed SMC protocol progresses through the following steps. Initially, Alice and Bob decide on a Boolean function and independently calculate the necessary polynomial vectors P and K based on their respective inputs. The protocol is executed over m rounds, corresponding to the number of elements in the polynomial vectors P and K. Each round i $(1 \leq i \leq m)$ proceeds as follows. Initially, a Bell state $|\varphi_i\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ is distributed between Alice and Bob, with each party possessing a qubit. Utilizing this Bell state, the two parties share a secret random bit r_i known only to them. The use of Bell state can be replaced by a standard quantum key distribution protocol to distribute the random bit r_i between parties. Afterwards, Charlie prepares a qubit at state $|0\rangle$ and sends it to Alice. Note that although the proposed protocol is tailored for two-party computation with independent inputs, the involvement of a third party, referred to as Charlie, is essential. This is because unconditionally secure two-party computation is not achievable, as outlined in [33]. However, Charlie's input is not independent of the inputs from the other parties. It is determined by the parity of the two other inputs $(P \oplus K)$. Next, Alice receives the qubit and performs the operation $V^{r_i}U^{P_i(\boldsymbol{a})}$ on the qubit, considering the input $P_i(\boldsymbol{a})$ and the random bit r_i. Note that, the objective of the U rotation is to encrypt Alice's input, whereas the V rotation is employed to obscure the function's output from untrusted parties. Afterwards, Alice sends the altered qubit to Bob who performs $V^{r_i}U^{K_i(\boldsymbol{b})}$ on the received qubit. Bob then sends the qubit to Charlie who performs $U^{\dagger(P_i(\boldsymbol{a})\oplus K_i(\boldsymbol{b}))}$ on the qubit leading to

$$\left|f_i'\right\rangle = \overbrace{U^{\dagger(P_i(\boldsymbol{a})\oplus K_i(\boldsymbol{b}))}}^{Charlie}\;\overbrace{V^{r_i}U^{K_i(\boldsymbol{b})}}^{Bob}\;\overbrace{V^{r_i}U^{P_i(\boldsymbol{a})}}^{Alice}\,|0\rangle\,. \qquad (10)$$

Charlie then measures the qubit in computation basis and stores the classical result. Once protocol is executed over m rounds, Charlie performs an XOR among all the measurement outcomes to obtain $f' = \bigoplus_{i=1}^{m} f'_i$. Charlie then sends the result to Alice and Bob, who retrieve the final output by XORing the received classical bit f' and the random bit $r = \bigoplus_{i=1}^{m} r_i$ as follows:

$$f(\boldsymbol{a}, \boldsymbol{b}) = r \oplus f'. \tag{11}$$

Protocol 1 provides an overview of the procedural steps. Utilizing the proposed protocol, in the next section, we compute the 2-bit Equivalence function using the IBM Qiskit platform.

Protocol 1 Quantum SMC Protocol

Inputs: Inputs $\boldsymbol{a} = (a_1, a_2, ..., a_n)$ for Alice, and $\boldsymbol{b} = (b_1, b_2, ..., b_n)$ for Bob.
Outputs: $f(\boldsymbol{a}, \boldsymbol{b})$ for Alice and Bob.

1. For $1 \leq i \leq m$, repeat steps 2-9 for each term.
2. Alice and Bob compute the associated polynomials $P_i(\boldsymbol{a})$ and $K_i(\boldsymbol{b})$ based on the specific function being calculated.
3. Alice and Bob are provided with two qubits, constituting a Bell state $|\varphi_i\rangle = (|00\rangle + |11\rangle)\big/\sqrt{2}$. Subsequently, each of them measures a qubit of the Bell state and records the outcome as r_i.
4. Using a secure classical channel, Alice and Bob send to Charlie the bits $P_i(\boldsymbol{a}) \oplus r_i$ and $K_i(\boldsymbol{b}) \oplus r_i$, respectively.
5. Charlie obtains the parity of parties' inputs by computing $(P_{i(\boldsymbol{a})} \oplus r_i) \oplus (K_{i(\boldsymbol{b})} \oplus r_i) = P_i(\boldsymbol{a}) \oplus K_i(\boldsymbol{b})$. The resulting value corresponds to Charlie's input.
6. Charlie provides a qubit in state $|0\rangle$ and sends it to Alice.
7. Alice performs $V^{r_i} U^{P_i(\boldsymbol{a})}$ on the qubit, considering the values of P_i and r_i, and sends the qubit to Bob.
8. Bob performs $V^{r_i} U^{K_i(\boldsymbol{b})}$ on the qubit and then sends it to Charlie.
9. Charlie performs $U^{\dagger(P_i(\boldsymbol{a}) \oplus K_i(\boldsymbol{b}))}$ on the qubit and measure it in computational basis.
10. After these steps are repeated over m rounds (where $i = m$), Charlie performs an XOR among all the measurement outcomes and sends the result to Alice and Bob.
11. Alice and Bob retrieve the final output as $f(\boldsymbol{a}, \boldsymbol{b}) = r \oplus f'(\boldsymbol{a}, \boldsymbol{b})$, with $r = \overset{i}{\underset{i=1}{\oplus}} r_i$.

4 Qiskit Implementation

In this section, we design a quantum circuit for the proposed protocol and explain its implementation in IBM Qiskit under both ideal and noisy conditions. We compute a special case of the 2-bit EQ$(\boldsymbol{a}, \boldsymbol{b})$ function. Our code is accessible in the GitHub repository https://github.com/Quantum-SMC.

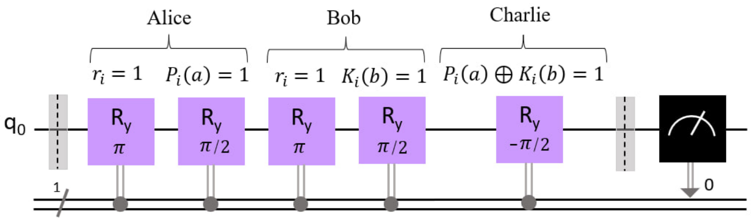

Fig. 1. Quantum circuit for the proposed protocol for round i. The qubit labeled as q_0 is the initial qubit with state $|0\rangle$. R_y indicates the rotation operation of the qubit along the y axis of the Bloch sphere, with respect to the classical bits r, P_i, K_i, and $P_i \oplus K_i$. Note that the rotation gates are exclusively applied when the bit values are equal to 1; otherwise, for bit values equal to 0, they are omitted from the circuit.

Figure 1 illustrates the corresponding quantum circuit where a qubit in the initial state $|0\rangle$ is prepared. To encrypt the inputs and output, we apply $R_y(\pi)$, $R_y(\pi/2)$, and $R_y(-\pi/2)$ operations to the qubit, corresponding to V, U, and $U^\dagger$, respectively. Afterwards, the qubit is measured in the computational basis, and the classical outcome is stored in the classical register of the Qiskit environment. In this circuit, the classical register C_0 is used to store the measurement result of q_0. Let us consider an example where the three parties, Alice, Bob, and Charlie, with input bits $\boldsymbol{a} = \{1, 0\}$, $\boldsymbol{b} = \{1, 0\}$, and $\boldsymbol{a} \oplus \boldsymbol{b} = \{0, 0\}$ aim to compute function $\mathrm{EQ}(\boldsymbol{a}, \boldsymbol{b}) = (1\,\mathrm{EQ}\,1)\,\mathrm{EQ}\,(0\,\mathrm{EQ}\,0))$, which should yield the output 1. Alice and Bob share random bits $\mathbf{r} = (0, 1, 1, 0)$ leading to $r = \bigoplus_{i=1}^{4} r_i = 0$. Considering Eq. (8), Alice and Bob compute $P = (1, 0, 1, 0)$ and $K = (0, 1, 1, 0)$, which correspond to the EQ function. Afterwards, parties execute the circuit for four rounds. Figure 2 illustrates the measurement outcomes for four rounds of circuit execution under (a) ideal noiseless setting and (b) noisy setting. The noise model includes bit-flip, phase-flip, amplitude damping, phase damping, and depolarizing errors, each with a probability of 0.1. This model was applied to both quantum gates and measurement operations. Our results indicate that the probability of obtaining the correct answer is, on average, 80.25%. This reduction in accuracy is due to quantum errors, which can be mitigated with error correction techniques. The most frequent measurement outcomes for rounds 1 to 4 are 0, 1, 0, and 0, respectively. Subsequently, Charlie's outcome is derived by XORing the measurement results from each round, yielding $0 \oplus 1 \oplus 0 \oplus 0 = 1$. This outcome is then transmitted to Alice and Bob, who retrieve the actual output of the EQ function by XORing the received classical bit and the private random bit (i.e., $1 \oplus 0 = 1$). The simulation results were obtained using the *'AerSimulator'* with 100 shots per round, conducted in Google Colab on Ubuntu 20.04.6 LTS, with Python 3.10.12 and Qiskit 0.43.2. We carried out the implementations on an ASUS Zenbook 14 UX425E laptop with 4 cores, an 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80 GHz processor, and 16 GB of RAM.

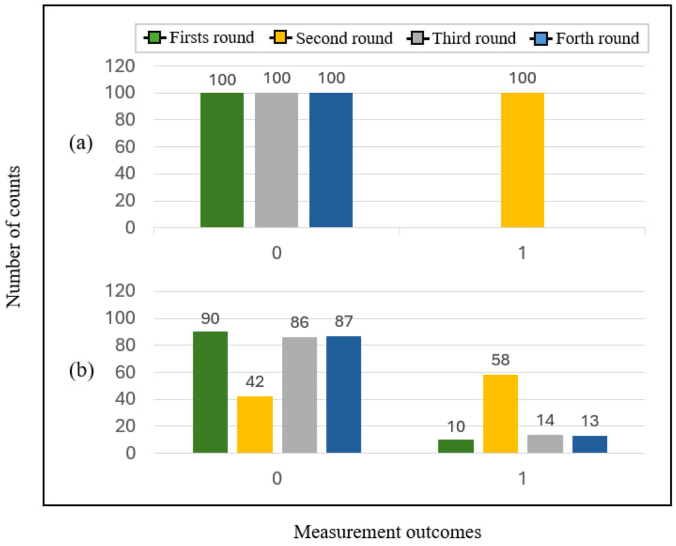

Fig. 2. The measurement outcomes of the corresponding quantum circuit with panel (a) illustrating the ideal noiseless results and panel (b) showing the results affected by quantum noise. Both simulations utilized the *'AerSimulator'* backend and were conducted over four rounds, each with 100 shots.

5 Result and Discussion

This section provides security, privacy, and complexity analyses of the proposed protocol. Additionally, we compare our work with another SMC protocol specifically designed for Boolean function computation.

5.1 Privacy Analysis

To validate the privacy of the proposed scheme, we assess the data leakage in each step as follows. In step 2 of the protocol, where the computation of $P(\boldsymbol{a})$ and $K(\boldsymbol{b})$ occurs locally, no information is disclosed regarding the inputs. During the transmission of $(P_i \oplus r_i)$ and $(K_i \oplus r_i)$ Charlie remains uninformed about parties' private inputs due to the use of a random bit. However, Charlie gains knowledge about the parity of the inputs at this stage. In the qubit transmission among parties, no information is revealed. Even if an eavesdropper successfully intercepts the particle transferred from one party to another, they are unable to measure it in the appropriate measurement basis. Qubit measurement and qubit rotation are done without the leakage of information.

5.2 Security Analysis

The security of this scheme is derived from the fundamental principles of quantum mechanics, which makes it difficult for an adversary to extract informa-

Table 1. Comparison of different quantum SMC protocols for Boolean function computation. m indicates the number of rounds. CompCx and CommCx denote computation and communication complexity, respectively. U and V specify the $-\pi/2$ rotation and the π rotation around the y axis of the Bloch sphere as defined in the paper. M represent the qubit measurement.

QSMC Prot.	CompCx	CommCx	Quantum resources	Quantum operations
Ref. [16]	$(6m+12)$ XOR	$\mathcal{O}(2m)$	GHZ + Bell	$(5m)M$
Ref. [27]	$(6m+14)$XOR+$(3m)$NOT	$\mathcal{O}(2m)$	GHZ + Bell	$(m)\sigma_z$+ $(3m)U$+$(3m)M$
This work	$(6m+12)$ XOR	$\mathcal{O}(5m)$	Single qubit+Bell	$(2m)V + (3m)U + (3m)M$

tion from quantum systems without leaving detectable traces. Consider security against potential attacks from Charlie. If Charlie aims to obtain any information about a party's private input, for instance, Alice, he needs to intercept the particle transmitted from Alice to Bob, and measure it in the right measurement basis ($|0\rangle$, $|1\rangle$). Nevertheless, Charlie cannot determine the correct measurement basis because he lacks information about the unitary operation $V^{r_i}U^{P(\boldsymbol{a})}$ and Alice's input bit. Secondly, if Charlie wants to extract the output of the function, he fails because he knows nothing about random bit r_i.

The protocol's vulnerability to a coalition attack arises from Charlie's awareness of the parity of input bits at every stage. This signifies that if Charlie collaborates with either Alice or Bob, they can gain insights into the input of the other party. As a result, the protocol's security can only be assured with a threshold of $th = 1$. The protocol maintains passive security, indicating that although the adversary can try to extract information from others, any deviation from protocol execution is prohibited.

5.3 Efficiency Analysis

The efficiency of SMC protocols is crucial for practical applications. Various factors influence the efficiency of these protocols, such as computation and communication overheads, as well as the amount of required quantum resources. To obtain the computation complexity, we consider the required operations at each step: polynomials (10 XOR), parity of inputs ($3m$ XOR), quantum operations ($(2m)VU$, mU, and $(3m)M$, with M representing qubit measurement), Charlie's outcome (m XOR), and the final output by parties ($2(m+1)$ XOR). Overall, the computational cost of our protocol is $(6m+12)$ XOR, $(2m)$ instances of VU operation, m instances of U operation, and $(3m)$ qubit measurement M. The communication complexity of our protocol is $\mathcal{O}(5m)$, reflecting the number of bits exchanged during each round m.

To evaluate our scheme, in Table 1, we compare the complexity of the proposed protocol with other SMC protocols that emphasize on secure Boolean function computation. As shown in Table 1, two types of quantum resources are required to compute Boolean functions within our approach: Bell state and single qubit. The necessity for these quantum resources is reduced to one-third (66.7%) compared to other approaches, in which three-qubit GHZ states are employed via MBQC approach. Although the computation complexity remains consistent compared to [16], the communication overhead in our protocol scales as $\mathcal{O}(5m)$ which is 40% higher than that of other approaches. This outcome is anticipated since the exchange of single qubits among parties inherently elevates communication requirements. In contrast, the MBQC approach avoids qubit exchanges by using distributed entangled particles, though this method results in a higher demand for quantum resources and costly process of entangling particles. Furthermore, while our approach involves more quantum operations, this is justified by the enhanced security it provides.

6 Conclusion

This paper discusses the problems encountered in classical SMC concerning both security and efficiency. Using single qubits, we proposed a quantum-based SMC protocol capable of computing binary Boolean functions. We achieved a 66.7% enhancement in efficiency by using fewer quantum resources. Our method requires 40% more communication resources, due to the increased qubit exchange among parties. Furthermore, the implementation of a random quantum rotation around the y-axis of the Bloch sphere improved the security level, effectively concealing the output from potential adversaries. We designed the corresponding quantum circuit and implemented our protocol on the IBM Qiskit platform, obtaining consistent results that confirm the feasibility and correctness of our approach.

Acknowledgments. 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 Ph.D. 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 reference 10.54499/PTDC/CCI-COM/4280/2021.

References

1. Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten. Crypten: Secure multi-party computation meets machine learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S.

Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4961–4973. Curran Associates, Inc., 2021.

2. Manuel B. Santos, Ana C. Gomes, Armando N. Pinto, and Paulo Mateus. Quantum secure multiparty computation of phylogenetic trees of sars-cov-2 genome. In *2021 Telecoms Conference (ConfTELE)*, pages 1–5, 2021.
3. Zeinab Rahmani, Luis S. Barbosa, and Armando N. Pinto. Quantum privacy-preserving service for secure lane change in vehicular networks. *IET Quantum Communication*, 4(3):103–111, 2023.
4. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, feb 1978.
5. Whitfield Diffie and Martin E. Hellman. *New Directions in Cryptography*, page 365–390. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022.
6. René Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Mathematics of computation*, 44(170):483–494, 1985.
7. Andrew C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 160–164, 1982.
8. D. Beaver, S. Micali, and P. Rogaway. The round complexity of secure protocols. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, page 503–513, New York, NY, USA, 1990. Association for Computing Machinery.
9. Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. *Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation*, page 351–371. Association for Computing Machinery, New York, NY, USA, 2019.
10. Jesper Buus Nielsen, Peter Sebastian Nordholt, Claudio Orlandi, and Sai Sheshank Burra. A new approach to practical active-secure two-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, pages 681–700, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
11. Enrique Larraia, Emmanuela Orsini, and Nigel P. Smart. Dishonest majority multi-party computation for binary circuits. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, pages 495–512, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
12. Sai Sheshank Burra, Enrique Larraia, Jesper Buus Nielsen, Peter Sebastian Nordholt, Claudio Orlandi, Emmanuela Orsini, Peter Scholl, and Nigel P. Smart. High-performance multi-party computation for binary circuits based on oblivious transfer. *Journal of Cryptology*, 34(3):472, June 2021.
13. Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 643–662, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
14. Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel P. Smart. Practical covertly secure mpc for dishonest majority – or: Breaking the spdz limits. In Jason Crampton, Sushil Jajodia, and Keith Mayes, editors, *Computer Security – ESORICS 2013*, volume 8134 of *Lecture Notes in Computer Science*, pages 1–18, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
15. Marcel Keller, Emmanuela Orsini, and Peter Scholl. Mascot: Faster malicious arithmetic secure computation with oblivious transfer. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 830–842, New York, NY, USA, 2016. Association for Computing Machinery.

16. Klearchos Loukopoulos and Daniel E. Browne. Secure multiparty computation with a dishonest majority via quantum means. *Phys. Rev. A*, 81:062336, Jun 2010.
17. Marco Clementi, Anna Pappa, Andreas Eckstein, Ian A. Walmsley, Elham Kashefi, and Stefanie Barz. Classical multiparty computation using quantum resources. *Phys. Rev. A*, 96:062317, Dec 2017.
18. Changbin Lu, Fuyou Miao, Junpeng Hou, Zhaofeng Su, and Yan Xiong. Secure multi-party computation with a quantum manner. *Journal of Physics A: Mathematical and Theoretical*, 54(8):085301, 2021.
19. Stefanie Barz, Vedran Dunjko, Florian Schlederer, Merritt Moore, Elham Kashefi, and Ian A. Walmsley. Enhanced delegated computing using coherence. *Phys. Rev. A*, 93:032339, Mar 2016.
20. Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7–11, 2014.
21. Mariano Lemus, Mariana F. Ramos, Preeti Yadav, Nuno A. Silva, Nelson J. Muga, André Souto, Nikola Paunković, Paulo Mateus, and Armando N. Pinto. Generation and distribution of quantum oblivious keys for secure multiparty computation. *Applied Sciences*, 10(12), 2020.
22. Maurício J. Ferreira, Nuno A. Silva, Armando N. Pinto, and Nelson J. Muga. Statistical validation of a physical prime random number generator based on quantum noise. *Applied Sciences*, 13(23), 2023.
23. Marcel Keller, Emmanuela Orsini, and Peter Scholl. Mascot: Faster malicious arithmetic secure computation with oblivious transfer. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 830–842, New York, NY, USA, 2016. Association for Computing Machinery.
24. Hans J Briegel, David E Browne, Wolfgang Dür, Robert Raussendorf, and Maarten Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, 2009.
25. Michael A. Nielsen. Cluster-state quantum computation. *Reports on Mathematical Physics*, 57(1):147–161, 2006.
26. Heng-Yue Jia, Qiao-Yan Wen, Ting-Ting Song, and Fei Gao. Quantum protocol for millionaire problem. *Optics Communications*, 284(1):545–549, 2011.
27. Zeinab Rahmani, Armando Humberto Moreira Nolasco Pinto, and Luis Manuel Dias Coelho Soares Barbosa. Secure two-party computation via measurement-based quantum computing. *Quantum Information Processing*, 23(6):221, Jun 2024.
28. Wen Liu and Wei Zhang. A quantum protocol for secure manhattan distance computation. *IEEE Access*, 8:16456–16461, 2020.
29. Run-Hua Shi. Quantum multiparty privacy set intersection cardinality. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(4):1203–1207, 2021.
30. Robert Raussendorf, Daniel E. Browne, and Hans J. Briegel. Measurement-based quantum computation on cluster states. *Phys. Rev. A*, 68:022312, Aug 2003.
31. Xiu-Bo Chen, Gang Xu, Yi-Xian Yang, and Qiao-Yan Wen. An efficient protocol for the secure multi-party quantum summation. *International Journal of Theoretical Physics*, 49:2793–2804, 2010.
32. Wim van Dam. Implausible consequences of superstrong nonlocality. *Natural Computing*, 12(1):9–12, November 2012.
33. Hoi-Kwong Lo. Insecurity of quantum secure computations. *Phys. Rev. A*, 56:1154–1162, Aug 1997.