Quantum computing on encrypted data with arbitrary rotation gates

Mohit Joshi,* Manoj Kumar Mishra, and S. Karthikeyan

Department of Computer Science,
Banaras Hindu University,
Varanasi, India - 221005

An efficient technique of computing on encrypted data allows a client with limited capability to perform complex operations on a powerful remote server without leaking anything about the input or output. Quantum computing provides information-theoretic security to solve such a problem, and many such techniques have been proposed under the premises of half-blind quantum computation. However, these approaches are based on some universal combination of non-parametric gates (H, S, T, CX, CZ, and CCX). Hybrid quantum-classical algorithms require parametric gates, which, when decomposed into non-parametric gates, inadvertently increase the depth of the circuit and hence the communication rounds. We propose an approach for recursive decryption of any parametric gate, $R_z(\theta)$, without prior decomposition; exactly when $\theta = \pm\pi/2^m$ for $m \in \mathbb{Z}^+$, and approximately with arbitrary precision ϵ for any given θ. We also show that a blind algorithm based on such a technique needs at most $O(\log_2^2(\pi/\epsilon))$ computation steps and communication rounds, while the techniques based on a non-parametric resource set require $O(\ln^{3.97}(1/\epsilon))$ steps. We use this approach to propose a half-blind quantum computation protocol to enable efficient computation on encrypted data in the NISQ era.

I. INTRODUCTION

The future quantum internet will be enabled in a distributed setting, much like the classical supercomputing framework. The client will most likely be a limited-resource device capable of communicating with a powerful remote quantum device capable of performing complex computation. This is evident with companies like IBM, Google, Intel, Amazon, and Microsoft providing quantum-as-a-service platforms [1–4].

This quantum internet will be vastly capable of providing security unachievable in the classical internet [5]. Childs proposed one such technique that enables a client to securely delegate their complex operations to a remote quantum server [6]. This type of security comes under the premises of blind quantum computation. It is broadly divided into two categories, namely, fully blind quantum computation (FBQC) and half-blind quantum computation (HBQC).

FBQC is performed over a universal resource set that assures the security of data and computation. Broadbent *et al.* proposed the first universal resource set, called the brickwork state, based on measurement-based quantum computation (MBQC) [7]. Due to their fault-tolerance and verifiability, several alternative resource sets have been proposed using MBQC over the years [8, 9]. However, these protocols require the server to implement highly entangled cluster states [10–13]. This limits the protocol's

applicability for near-future applications. This has been shown in experimental demonstrations with four-qubit cluster states implemented on optical setups [14–16].

HBQC, on the other hand, promises security to data only. This premise enables quantum computation on encrypted data for near-future implementation of secure quantum protocols. Tham *et al.* recently used such a protocol to experimentally demonstrate the application of quantum fully-homomorphic encryption [17, 18]. Other applications of such protocols include secure multiparty computation, interactive proofs, and quantum one-time programs [17–22].

These protocols encrypt the data using classically assisted Pauli's X and Z gates. The decryption is performed using some commutation property of the corresponding gates. Fisher *et al.* first demonstrated this protocol using an optical setup of polarized beam splitters, half-waveplates, and quarter-waveplates [23]. Various such protocols have been proposed using some universal combination of non-parametric universal resource set $\{H, S, T, CX, CZ, CCX\}$ [24–27].

However, modern hybrid quantum-classical algorithms are based on parametric circuits like R_z rotation gates [28–31]. This means the circuit needs to be decomposed into non-parametric sets to be implemented using existing HBQC protocols. This inadvertently increases the size of the blind circuit, which affects the depth and communication rounds of the protocols.

Recently, it has been shown that phase-shifted microwave pulses can implement the arbitrary R_z rotation gates [32, 33]. This can greatly reduce the size of

* joshimohit@bhu.ac.in

the blind circuit using protocols based on direct R_z gate decryption without the need for decomposition. Moreover, extensive circuit-based quantum simulation platforms enable rapid prototyping of such protocols [34, 35].

In this study, we propose the first technique of decryption for an arbitrary rotation gate without the need for prior decomposition, hence reducing the server resources in terms of depth, which in turn substantially reduces the communication overhead required for such schemes. Our main results are:

- We introduce a method of recursive decryption of $R_z(\theta)$; exactly when $\theta = \pm\pi/2^m$ for $m \in \mathbb{Z}^+$ and approximately with arbitrary precision ϵ for any given θ.

- We show that any blind approach based on this recursive decryption requires at most $O(\log_2^2(\pi/\epsilon))$ communication rounds, while approaches based on non-parametric gates using Solovey-Kitaev decomposition require $O(\ln^{3.97}(1/\epsilon))$ communication rounds.

- Based on the presented decryption technique of $R_z(\theta)$, we propose a half-blind quantum computing protocol for efficient computing on encrypted data in the NISQ era.

The rest of the study is organized as follows: Sec. II introduces the preliminary of the subject matter. In Sec. III, we present the technique of recursive decryption of arbitrary Z rotation gates (Sec. III A), and the protocol of half-blind quantum computation along with its proof of universality, correctness, and blindness of data (Sec. III B). In Sec. IV, a simple implementation example has been presented, and at last, Sec. V gives the concluding remarks and future implications of the presented results.

II. PRELIMINARY

The protocols of half-blind quantum computation work by encrypting the data using Pauli's X and Z rotation gates [6]. The general form of the Child's blindness process can be represented as:

$$2^n \mathbb{I} = \sum_{j_1, j_2, \ldots, j_{2n} \in \{0,1\}} \left(\bigotimes_{i=1}^{n} Z_i^{j_{2i}} X_i^{j_{2i-1}} \right) |\psi\rangle$$
$$\langle\psi| \left(\bigotimes_{i=1}^{n} X_i^{j_{2i-1}} Z_i^{j_{2i}} \right), \quad (1)$$

where $j_k \in_r \{0,1\}$ is the kth key out of total $2n$ classical keys used in the protocol for a n-qubit system $|\psi\rangle$. X and Z are Pauli's rotation matrices, and $\mathbb{I}$ is

the identity matrix of size n. This equation shows that nothing except the upper bound on the number of qubits is revealed about the data $|\psi\rangle$.

The client sends this encrypted state to the server, which implements the desired unitary to complete the computation. The client then decrypts the result he obtained from the server using appropriate corrections. The correction is a unitary D that operates on the result such that encryption gets nullified as:

$$U = D \cdot U Z^b X^a, \quad (2)$$

where $a, b \in_r \{0, 1\}$.

For universal computation, the client should be able to delegate gates from a set of universal resources and correct the result. This implies the client should be able to decrypt gates from the set $\{H, S, T, CX, CZ, CCX\}$. Client is assumed to have the capability to perform X and Z gates, which are essential for encryption. The decryption of Clifford gates (H, S, CX, CZ) does not require any interaction, as given below [6]:

$$H_1(X_1^a Z_1^b |\psi\rangle_1) = X_1^b Z_1^a (H_1 |\psi\rangle_1), \quad (3)$$

$$P_1(X_1^a Z_1^b |\psi\rangle_1) = X_1^a Z_1^{a\oplus b} (P_1 |\psi\rangle_1), \quad (4)$$

$$CX_{12}(X_1^a Z_1^b X_2^c Z_2^d |\psi\rangle_{12}) = (X_1^a Z_1^{b\oplus d})$$
$$(X_2^{a\oplus c} Z_2^d)(CX_{12} |\psi\rangle_{12}), \quad (5)$$

$$CZ_{12}(X_1^a Z_1^b X_2^c Z_2^d |\psi\rangle_{12}) = (X_1^a Z_1^{b\oplus c})$$
$$(X_2^c Z_2^{a\oplus d})(CZ_{12} |\psi\rangle_{12}), \quad (6)$$

The decryption of the non-Clifford gate T gate needs an additional ancilla qubit and assistance of S correction as shown below [24]:

$$T(X_1^a Z_1^b |\psi\rangle_1 S_2^y Z_2^d |+\rangle_2) = S_2^{a\oplus y} X_2^{a\oplus m}$$
$$Z_2^{a(m\oplus y\oplus 1)\oplus b\oplus d\oplus y} T |\psi\rangle_2, \quad (7)$$

Here, m is the measurement result from the first qubit. Also, the CCX gate require the assistance of two CX and one CZ gate as given below [26]:

$$CCX_{123}(X_1^a Z_1^b X_2^c Z_2^d X_3^e Z_3^f |\psi\rangle_{123}) = (CX_{13}^c X_1^a Z_1^b)$$
$$(CX_{23}^a X_2^c Z_2^d)$$
$$(CZ_{12}^f X_3^e Z_3^f)$$
$$(CCX_{123} |\psi\rangle_{123}), \quad (8)$$

III. PROPOSED PROTOCOL

We assume the client has the capability to perform gates from the set $\{X, Z, Swap, Measure\}$, and

the server has the capability to perform $\{H, CZ, R_z\}$ for universal computation. We can contrast it with ancilla-driven approaches, which require the client to possess the capability of $R_z(\pi/m)$ rotation with $m \in \{0, ...7\}$, and MBQC approaches, which require the server to possess the capability to process highly entangled graph states [7, 11, 12].

We also assume the client is capable of generating random classical bits with either some classical pseudo-random number generator or using quantum random number generators, which is a standard BQC assumption [6]. These client-generated random keys do not need to be directly transmitted outside their private space and only need to help regulate the circuit's quantum gate implementation.

We now present the protocol, by firstly proposing an efficient technique for recursive decryption of an arbitrary R_z gate (Sec. III A). We then propose a universal scheme of half-blind quantum computation (Sec. III B) based on the presented recursive decryption scheme.

A. Decryption of arbitrary $R_z(\theta)$ gates

In this subsection, we explain our protocol that enables a client capable of performing X and Z gates to securely delegate the desired rotation of the $R_z(\theta)$ with arbitrary precision ϵ.

It is well established in BQC primitives that the decryption of the T gate requires an additional implementation of the S gate (as evident from Eq. (7)), which can be generalized to any $R_z(\theta)$ gate requiring $R_z(2\theta)$ for decryption [27]. We formally prove this generalization (Theorem A.1) and use it to present a technique of recursive decryption for arbitrary $R_z(\theta)$ gates as:

$$R_z(\theta)Z^b X^a = R_z^a(2\theta)X^a Z^b R_z(\theta). \qquad (9)$$

However, for the protocol to work, this recursion must stop. We observe that this recursion will stop exactly when $\theta = \pm\pi/2^m$ for $m \in \mathbb{Z}^+$ (Alogrithm 1) and for $\theta \neq \pm\pi/2^m$, we can decrypt with arbitrary precision ϵ (Alogirithm 2). We describe these scenarios in detail below:

Exact Decryption: If $\theta = \pm\pi/2^m$ for $m \in \mathbb{Z}^+$, then the recursive decryption stops exactly at $\theta = \pm\pi/2$, as the base condition of this recursion is an $S = R_z(\pi/2)$ gate, which can be decrypted using only the client available X and Z gates as:

$$R_z(\pm\pi/2)X^a Z^b = e^{\mp ia\pi/2}Z^a X^a Z^b R_z(\pm\pi/2). \qquad (10)$$

Futhermore, for higher powers of π, it can be noted that $R_z(m\pi)$, where $m \in \mathbb{Z}$, can be decrypted

at client side only as:

$$R_z(m\pi) = \begin{cases} I & \text{if } m \equiv 0 \ (\text{mod } 2), \\ Z & \text{if } m \equiv 1 \ (\text{mod } 2). \end{cases} \qquad (11)$$

Using Eq. (10) and Eq. (11), we can decrypt $R_z(\pm\pi/2^m)$ exactly with recursive application of R_z gates (Theorem A.2). However, this procedure inadvertently reveals the value of the encryption key a as the next step in the decryption is dependent on the previous step. To prevent this leakage of information to the server, we use an ancilla qubit and $Swap$ gate to flip the content of the working qubit and ancilla qubit as soon as the run_of_one (subsequent $a = 1$ in recursion) stops, as sketched in Algorithm 1. Fig. 1 shows the circuit of the recursive decryption process $R_z(\pm\pi/2^m)$ with assumption that run_of_one stops after k^{th} step when $\theta = \pm\pi/2^{k-1}$.

Algorithm 1: Decryption of $R_z(\pm\pi/2^m)$ where $m \in \mathbb{Z}^+$

Input: $|\psi\rangle$ and $\theta = \pi/2^m$ where $m \in \mathbb{Z}^+$.
Result: Decrypted state $R_z(\theta)|\psi\rangle$.
Client generates $2m$ encryption keys
  $a_i, b_i \in_r \{0, 1\}$ for $i \in \{0, \ldots, m-1\}$ and assign $run\_of\_one \leftarrow 0$;
Client encrypts input $|\psi\rangle$ using $Z^{b_0}X^{a_0}$ and sends it to server;
Server computes $R_z(\theta)|\psi\rangle$ and sends result to client;
Client decrypts $|\psi\rangle$ using $X^{a_0}Z^{b_0}$ and updates $\theta \leftarrow 2\theta$;
if $a_0 = 1$ **then**
  | Client updates $\theta \leftarrow 2\theta$ and $run\_of\_one \leftarrow 1$;
for $k \leftarrow 1$ **to** $\log_2(\pi/2\theta)$ **do**
  | **if** $run\_of\_one = 1$ **and** $a_{k-1} = 0$ **then**
  | | Client applies $Swap$ gate on $|\psi\rangle$ between working and ancillary qubit;
  | | $run\_of\_one \leftarrow 2$;
  | Client encrypts $|\psi\rangle$ using $Z^{b_k}X^{a_k}$ and sends to server;
  | Server applies $R_z(\theta)|\psi\rangle$ and returns result;
  | Client decrypts $|\psi\rangle$ using $X^{a_k}Z^{b_k}$ and update $\theta \leftarrow 2\theta$;
if $run\_of\_one = 2$ **then**
  | Client applies $Swap$ gate;
return $|\psi\rangle$;

This recursion only requires m rotation gates, which implies m communication rounds between client and server (see proof Theorem A.2). Also note, the expected length for $runs\_of\_one$ (subsequent $a = 1$ in recursion) is 1. This implies that the asymptotic complexity of decryption is $O(1)$, which can be used for further optimization. However, this
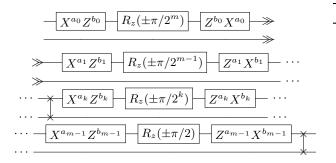
Figure 1: Delegation of $R_z(\pm\pi/2^m)$, where $m \in \mathbb{Z}^+$ with recursive decryption of $R_z(\pm\pi/2^{m-1})$ untill $m=1$. Here, the *Swap* gate is applied when *run_of_one* (subsequent $a=1$ in recursion) is exhausted. Note, $\{R_z(\theta)\}$ is implemented by server, while $\{X, Z, Swap\}$ are client implementable gates.

optimization is omitted here for the sake of simplicity.

Approximate Decryption: The exact decryption of $R_z(\pm\pi/2^m)$ for $m \in \mathbb{Z}^+$ can be used to approximate any arbitray θ using a simple observation that:

$$\theta \approx p\pi + \sum_{m=1}^{M} \frac{a_m\pi}{2^m}, \qquad (12)$$

with appropriate value of $p \in \mathbb{Z}$ and $a_m \in \{-1, 0, 1\}$ for $m \in \{1, ..., M\}$ where $M = \lceil \log_2(\pi/\epsilon) \rceil$.

Hence, $R_z(\theta)$ can be decrypted using:

$$R_z(\theta) \approx R_z(p\pi) \prod_{m=1}^{M} R_z(a_m\pi/2^m). \qquad (13)$$

The procedure of decryption of arbitary $R_z(\theta)$ is sketched in Algorithm 2, where $R_z(p\pi)$ is implemented using a simple application of the Z gate (Eq. (13)), and $R_z(a_m\pi/2^m)$ is recursively decrypted using Algorithm 1 as a subroutine for all values of $m \in \{1, ..., M\}$.

This recursive decryption requires atmost $O(\log_2^2(\pi/\epsilon))$ rotation gates, which implies atmost $\log_2^2(\pi/\epsilon)$ communication rounds (see proof Theorem A.3). Fig. 2 shows the increase in depth between the blind approach based on a parametric R_z gate is $O(\log_2^2(\pi/\epsilon))$, while non-parametric resource set-based blind decryption is $O(\ln^{3.97}(1/\epsilon))$ based on Solovay-Kitaev decomposition (see Corollary A.3.1). Also note, the asymptotic complexity $(O(\log_2(\pi/\epsilon)))$ makes the recursive decryption with optimization efficient even when modern decomposition algorithms like Ross-Sellinger algorithm (gridsynth) are used, which has asymptotic complexity of $3\log_2(1/\epsilon) + O(\log_2(\log_2(1/\epsilon)))$ [36].

Algorithm 2: Decryption of arbitrary $R_z(\theta)$

Input: $|\psi\rangle$ and θ.
Result: Decrypted state $R_z(\theta)|\psi\rangle$.
Find coefficients $p \in \mathbb{Z}$ and $a_m \in \{-1, 0, 1\}$
  $\forall m \in \{1, ..., M\}$, where $M \leftarrow \lceil \log_2(\pi/\epsilon) \rceil$, such that $\theta = p\pi + \sum_{m=1}^{M} \frac{a_m\pi}{2^m}$;
if $p \equiv 1 \pmod 2$ **then**
  | Client applies Z to $|\psi\rangle$;

for $m \leftarrow 1$ **to** $\log_2(\pi/\epsilon)$ **do**
  | **if** $a_m = 1$ *or* $a_m = -1$ **then**
  | | $|\psi\rangle \leftarrow$ Call Algorithm 1 with $|\psi\rangle$ and
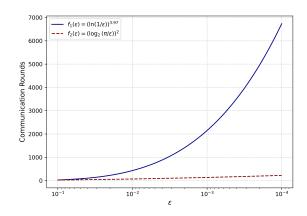 | | $\theta = a_m\pi/2^m$;

return $|\psi\rangle$;

Figure 2: Increase in depth of proposed blind decryption based on arbitrary rotation gate $R_z(\theta)$ $(O(log_2^2(\pi/\epsilon)))$ and non-parameteric gates $\{H, S, T\}$ $(O(\ln^{3.97}(1/\epsilon)))$ using Solovey-Kitaev decomposition.

Moreover, if $\theta = \pm\pi/2^m$ for $m \in \mathbb{Z}^+$, then the recursive decryption can be performed in $O(1)$ steps asymptotically. For instance, a setup implementing $R_z(\pi/128)$ using gridsynth will require 104 T gates asymptotically and $255,856$ T gates using Solovay-Kitaev decomposition for $\epsilon = 10^{-10}$. However, using recursive decryption without decomposition requires only one R_z gate asymptotically, and at most $m = log_2(128) = 7$ gates.

B. Universal Half-Blind Quantum Computation with arbitrary $R_z(\theta)$ gates

In this section, we propose a half-blind quantum computation protocol based on the presented decryption of arbitrary rotation gates in Section III A.

The protocol allows us to securely delegate an algorithm $\mathcal{A}$ given as a collection of unitaries U_q with depth D, and total number of qubits n, where q de-

---

Algorithm 3: Proposed Half-Blind Quantum Computation Protocol

---

Input: Algorithm $\mathcal{A}$ as a collection of unitaries U_q with depth D, and total number of qubits n, where q is the ordered set of qubits on which U acts.

Initialize input $|\psi\rangle_0$;

Create $\mathcal{J} \leftarrow \{U_{q,d} \mid d \in \{0, \ldots, D'-1\}\}$ from the given algorithm $\mathcal{A}$ with $n' > n$ total number of qubits, including ancilla qubits, and D' total depth of the circuit, including original and trap gates;

Client sends $\mathcal{J}$ to server over a classical channel;

for $j \leftarrow 0$ **to** $D'-1$ **do**

    Client generates a random key set $\mathcal{K}_j = \{k_i \mid k_i \in_r \{0,1\},\ i \in \{0,1,\ldots,2n'-1\}\}$;

    Client encrypts previous input: $|\psi\rangle_{j-1,enc} \leftarrow \left(\bigotimes_{q=0}^{n-1} Z_q^{\mathcal{K}_j[2q+1]} X_q^{\mathcal{K}_j[2q]}\right) |\psi\rangle_{j-1}$;

    Client sends $|\psi\rangle_{j-1,enc}$ to server via quantum channel;

    Server computes $|\psi\rangle_{j,enc} \leftarrow \mathcal{J}_j |\psi\rangle_{j-1,enc}$;

    Server sends $|\psi\rangle_{j,enc}$ to client;

    Client decrypts $|\psi\rangle_j \leftarrow dec(|\psi\rangle_{j,enc}, \mathcal{J}_j)$ using Eq. (17), Eq. (18), and Algorithm 2;

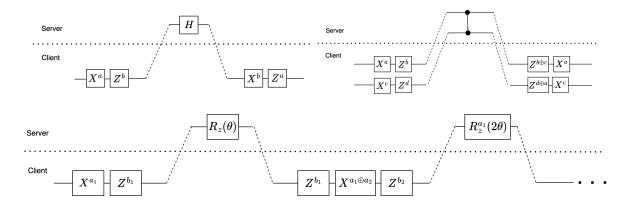**return** $|\psi\rangle_{D'-1}$;

Figure 3: The process of secure delegation of resources $\{H, CZ, R_z\}$ which is encryption and decrypted by client capable of performing $\{X, Z, Swap, Measure\}$ assited by random classical bit $a_i, b_i \in_r \{0,1\}$.

notes the ordered set of qubits on which the unitary U needs to be applied.

The client begins with an input state $|\psi\rangle_0$ defined on $n' > n$-dimensional Hilbert state with at least one ancillary qubit, Client uses $\mathcal{A}$ to define a computation set $\mathcal{J} = \{U_{q,d}\ \ d \in \{0, ..., D'-1\}\}$, where D' is the depth of the given circuit with some trap operations. For each jth step of D' computation step, which prepares a set of classical random keys $\mathcal{K}_j = \{k_i | k_i \in_r \{0,1\}, i = \{0,1,...,2n'-1\}\}$ of size $2n'$. The total encryption keys needed to complete the D' steps are $(2n'D')$ drawn from the possible binary key space of size $2^{2n'D'}$.

The protocol runs as an *encrypt-compute-decrypt* cycle for every unitary set at j^{th} depth in computation set $\mathcal{J}$, i.e., D' times, using a quantum channel comprising of working and ancillary qubits, which is a subset of $\{0,...,n'-1\}$.

In the jth cycle of *encrypt* phase, the client encrypts the previous state $|\psi\rangle_{j-1}$ by applying classical controlled X and Z gates on state using the generated encryption key $\mathcal{K}_j$ as shown in Eq. (14):

$$|\psi\rangle_{j-1,enc} = \left(\bigotimes_{q=0}^{n-1} X_q^{\mathcal{K}_j[2q]} Z_q^{\mathcal{K}_j[2q+1]}\right) |\psi\rangle_{j-1}. \quad (14)$$

The client then transmits this encrypted state $|\psi\rangle_{j,enc}$ to the server.

In the jth cycle of *compute* phase, the server applies the unitary $U_{q,j}$ on $|\psi\rangle_{j-1,enc}$ defined in the computation set $\mathcal{J}$ as shown in Eq. (15):

$$|\psi\rangle_{j,enc} = U_{q,j}|\psi\rangle_{j-1,enc}, \quad (15)$$

where q denotes the qubits on which the computation U_j has to be performed. The resultant state $U_j|\psi\rangle_{j,enc}$ is returned to the client.

In the jth cycle of *decrypt* phase, the client recovers the correct state by decrypting the server output using Eq. (16):

$$U_j|\psi\rangle_j = dec(U_j|\psi\rangle_{j,enc}), \quad (16)$$

where decryption map $dec(\cdot)$ is defined using the keys in encryption set $\mathcal{K}_j$ and the relations given in Eq. (17) to Eq. (19) using Algorithm 2. Fig. 3 shows the encryption and decryption process of the universal set $\{H, CZ, R_z\}$.

$$H_1 Z_1^b X_1^a = X_1^b Z_1^a H \tag{17}$$

$$CZ_{1,2} Z_2^d X_2^c Z_1^b X_1^a = Z_2^{d \oplus a} X_2^c Z_1^{b \oplus c} X_1^a CZ_{1,2} \tag{18}$$

$$R_z(\theta)_1 Z_1^b X_1^a = R_z^a(2\theta)_1 Z_1^b X_1^a R_z(\theta)_1 \tag{19}$$

Proof of Universality: As a set of Clifford and non-Clifford gates like $\{X, Z, H, S, T, CX\}$ form a universal set for quantum computation [37]. It is trivial to show that a server capable of performing only $\{H, CZ, R_z\}$ will be able to let a client with X and Z gates perform universal computation.

Using Euler's Z-X-Z decomposition, we can represent any single qubit gate as:

$$U = e^{i\phi} R_z(\alpha) R_x(\beta) R_z(\gamma) \tag{20}$$

Also $R_x(\theta) = H R_z(\theta) H$. Hence, any single-qubit gate can be performed using $H, R_z(\theta)$.

Also, $CX_{1,2} = H_2 CZ_{1,2} H_2$, which can be used to reach any multi-qubit operation along with non-Clifford resource $T(= R_z(\pi/4))$ gate.

Proof of Correctness: Correctness of H and CZ directly follows from literature [24, 26], and can be verified for all possible $|\psi\rangle$ and $a, b, c, d \in_r \{0, 1\}$ with the Eq. (17) and Eq. (18). Theorem A.1, Theorem A.2, and Theorem A.3 proves the correctness of recursive decryption of $R_z(\theta)$ as given in Eq. (21).

$$R_z(\theta) |\psi\rangle \langle\psi| R_z(\theta) = R_z^a(2\theta) Z^b X^a R_z(\theta) |\psi\rangle$$
$$\langle\psi| X^a Z^b R_z(\theta) \tag{21}$$

Proof of Blindness of Data: For the blindness of data, we need to show that the encryption keys do not leak to the server. The blindness of Clifford's resources is straightforward, as it does not require any additional interaction between client and server. This can be proven using Child's equation:

$$2\mathbb{I} = \sum_{a,b \in \{0,1\}} Z^b X^a H |\psi\rangle \langle\psi| X^a Z^b H \tag{22}$$

$$4\mathbb{I} = \sum_{a,b,c,d \in \{0,1\}} Z_2^d X_2^c Z_1^b X_1^a CZ_{1,2} |\psi\rangle$$
$$\langle\psi| X_1^a Z_1^b X_2^c Z_2^d CZ_{1,2} \tag{23}$$
$$\tag{24}$$

The proof for the blindness of $R_z(\theta)$ is based on showing that the proposed protocol of recursive decryption, as presented in Section III A, can be converted to an equivalent delayed version, a technique of proof similar to Ref. [24].
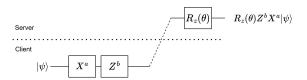

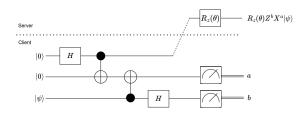Figure 4: The delegation of $R_z(\theta)$ gate in the proposed protocols.

Figure 5: The entanglement-based equivalent of proposed decryption of $R_z(\theta)$ with delayed correction.

We start by noting that the first delegation of the $R_z(\theta)$ gate decryption can be represented as shown in Fig. 4. This can be represented by an equivalent entanglement-based circuit as represented in Fig. 5. The delayed measurement of qubits shows that the first delegation of $R_z(\theta)$ can be implemented without revealing the encryption keys a, b.
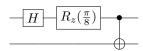
However, for exact decryption, successive delegation of $R_z(2\theta)$ reveals the previous encryption key (Eq. A10). Application of the *Swap* gate and ancillary qubit at the client side makes the recursion steps independent of encryption keys by swapping the content of ancilla and working qubit, as soon as *run_of_one* stops. At the server side, this will resemble as if the encryption key comprises all 1s, without revealing any additional information. This also ensured the privacy for any arbitrary R_z gate as the approximate decomposition of θ as given in Eq. 12 is independent of encryption keys.

This procedure prevents the leakage of the encryption key, hence preserving the blindness of data $|\psi\rangle$.

IV. EXAMPLE

In this section, we show the secure implementation of an example circuit containing H, $R_z(\pi/8)$, and CX gates using Algorithm 3.

Suppose we are given a two-qubit circuit $\mathcal{C}$ as shown below:

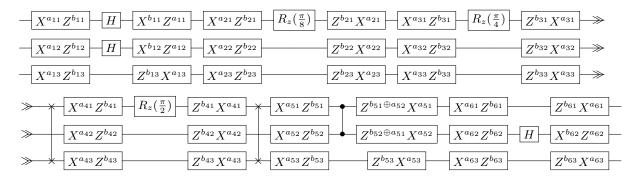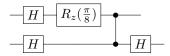The expected output for given input $|\psi\rangle_{1,2} =$

Figure 6: Complete blind implementing of circuit given by computation set $\mathcal{J}_1$ in Eq. (26), where $\{X, Z, Swap\}$ gate set is performed by client while $\{H, CZ, R_z\}$ gate set is delegated to remote server.

$\alpha_1 |00\rangle_{1,2} + \alpha_2 |10\rangle_{1,2} + \alpha_3 |01\rangle_{1,2} + \alpha_4 |11\rangle_{1,2}$ of the circuit $\mathcal{C}$ is:

$$CX_{1,2}R_z(\pi/8)_1 H_1 |\psi\rangle_{1,2} = (\alpha_1 + \alpha_3) |00\rangle_{1,2}$$
$$+ (\alpha_2 + \alpha_4) |10\rangle_{1,2}$$
$$+ (\alpha_2 - \alpha_4) e^{i\pi/8} |01\rangle_{1,2}$$
$$+ (\alpha_1 - \alpha_3) e^{i\pi/8} |11\rangle_{1,2}$$
$$(25)$$

The circuit is first transpiled to given universal resource set $\{H, CZ, R_z\}$ as:

Now, we start by creating the computation set $\mathcal{J}_1$ from this circuit which will be:

$$\mathcal{J}_1 = \{H_{(0,),0}, H_{(1,),0}, R_z(\pi/8)_{(0,),1}, R_z(\pi/4)_{(0,),2},$$
$$R_z(\pi/2)_{(0,),3}, CZ_{(0,1),4}, H_{(1,),5}\}. \quad (26)$$

Here, $D' = 6$, $n' = 3$, and the secure implementation requires the key of size $2n'D' = 36$.

In the Fig. 6, we show the complete circuit of client and server side to perform computation on the encrypted state $|\psi\rangle$. The gate set $\{X, Z, Swap\}$ is performed by the client while the gate set $\{H, CZ, R_z\}$ is server implemented.

The protocol starts with the client encrypting the input state $|\psi\rangle$ using $Z_1^{b_{11}} X_1^{a_{11}}$ on first qubit, $Z_2^{b_{12}} X_2^{a_{12}}$, and $Z_3^{b_{13}} X_3^{a_{13}}$ on third qubit. The client then sent these qubits to the server, who then applies $H_1 H_2$, which will be decrypted by the client with the appropriate key as given in Eq. (17).

Client then again encrypts the resultant state using next round keys $a_{21} a_{22}, a_{23}$ for X gate and

b_{21}, b_{22}, b_{23} for Z gate. The $R_z(\pi/8)$ needs two additional recursive call to $R_z(\pi/4)$ and $R_z(\pi/2)$ for decryption. In the chosen example, we have assumed that run_of_one stops at a_{31} i.e. $a_{11} = 1, a_{21} = 1, a_{31} = 0$. Hence, the $Swap$ gate is used after $R_z(\pi/4)$ rotation. Another $Swap$ gate is applied after the recursive decryption of $R_z(\pi/8)$ has been completed at the client's side to restore the state of the working qubit. After this, $CZ_{1,2}$ and H_2 gates are decrypted using Eq. (18) and Eq. (17) to complete the given blind circuit.

V. CONCLUSION

In this paper, we present a protocol for recursive decryption of R_z gates with arbitrary rotations. We showed that this recursion requires at most $O(\log_2^2(\pi/\epsilon))$ server resources and communication rounds. This is in contrast to protocols based on the decryption of non-parametric resources $\{H, S, T, CX, CZ, CCX\}$, which will require $O(\ln^{3.97}(1/\epsilon))$ steps if decomposition is performed using the Solovay-Kitaev theorem. Using the recursive decryption of arbitrary rotation gates, we have presented an efficient technique to perform half-blind quantum computation. This enables computing on encrypted data with the least amount of entangled resources and communication rounds. This implies that larger proof-of-principle experiments can be implemented using phase-shifted microwave pulses [32, 33], enabling secure implementation of hybrid quantum-classical algorithms.

Our study considers only a blind approach, while an efficient full blind approach, where θ is blind, will enhance the practicality of the solution. Moreover, in our approach client still needs the ability to prepare the state $|\psi\rangle$, which can be improved using the assumption of non-communication server [38] and entanglement-swapping-based triple server

protocols [39]. Moreover, verification of the server's honesty, that is, the server performed the operations correctly, is an important question to answer. Although a simple trap-based mechanism can be implemented for verification of the server in our protocol, still more elegant and cost-effective solution can be explored [40, 41].

Appendix A: Decryption of arbitrary Z gate

Theorem A.1. *The decryption of arbitrary $R_z(\theta)$ is dependent on $R_z(2\theta)$, i.e., $R_z(\theta)Z^bX^a = R_z^a(2\theta)X^aZ^bR_z(\theta)$.*

Proof. For decryption of R_z gate encrypted by X and Z gate, we need to find a unitary D such that:

$$R_z(\theta) = D \cdot R_z(\theta)Z^bX^a, \tag{A1}$$

where $a, b \in_r \{0, 1\}$. Here, solving for D we get:

$$D = R_z(\theta)X^aZ^bR_z(-\theta). \tag{A2}$$

Note that $R_z^\dagger(\theta) = R_z(-\theta)$. Also,

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}. \tag{A3}$$

Given X and Z are standard Pauli rotation gates, we can represent X^a and Z^b as:

$$X^a = \begin{pmatrix} 1 - 1_a & 1_a \\ 1_a & 1 - 1_a \end{pmatrix}, Z^b = \begin{pmatrix} 1 & 0 \\ 0 & (-1)^{1_b} \end{pmatrix}, \tag{A4}$$

where

$$1_x = \begin{cases} 1, & \text{if } x = 1, \\ 0, & \text{if } x = 0. \end{cases} \tag{A5}$$

This indicator variable representation of X and Z gate allows us to algebraically manipulate the matrix form of D, which can be represented as:

$$D = \begin{pmatrix} 1 - 1_a & (-1)^{1_b}1_a e^{-i\theta} \\ 1_a e^{i\theta} & (-1)^{1_b}(1 - 1_a) \end{pmatrix}. \tag{A6}$$

This matrix representation of D can be further decomposed into simpler gate combinations as:

$$D = \begin{pmatrix} e^{i\theta 1_a} & 0 \\ 0 & e^{i\theta 1_a} \end{pmatrix} \begin{pmatrix} 1 - 1_a & 1_a \\ 1_a & 1 - 1_a \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & (-1)^{1_b} \end{pmatrix},$$
$$= R_z^a(2\theta)X^aZ^b. \tag{A7}$$

Here, we have used the following identities associated with indicator variables 1_a and 1_b, where $a, b \in_r \{0, 1\}$:

$$(-1)^{2 \cdot 1_b} = 1, \qquad (1 - 1_a) \cdot 1_a = 0,$$
$$(1 - 1_a)^2 = 1 - 1_a, \quad (1 - 1_a) + 1_a e^x = e^{x \cdot 1_a}. \tag{A8}$$

Hence,

$$R_z(\theta)Z^bX^a = R_z^a(2\theta)X^aZ^bR_z(\theta). \tag{A9}$$

This shows that the decryption of the rotation gate $R_z(\theta)$ is dependent on the $R_z(2\theta)$ gate. $\square$

Theorem A.2. *Any rotation gate $R_z(\pm\frac{\pi}{2^m})$ for $m \in \mathbb{Z}^+$ can be exactly decrypted using atmost m rotation gates.*

Proof. We start by showing that the decryption of the $R_z(\pm\pi/2)$ gate is trivial and can be performed using the client's Pauli rotations only. Using theorem A.1, we can verify this fact as:

$$R_z(\pm\pi/2)X^aZ^b = R_z^a(\pm\pi)X^aZ^bR_z(\pm\pi/2),$$
$$= e^{\mp ia\pi/2}Z^aX^aZ^bR_z(\pm\pi/2). \tag{A10}$$

Using Eq. (A10), we can recursively decrypt any $R_z(\pm\pi/2^k)$ with sucessive application of $R_z(\pm\pi/2^{k-1})$ gates untill the $k = 1$ i.e. $\pm\pi/2^k = \pm\pi/2$. This shows that the recursive decryption of $R_z(\pm\pi/2^m)$ where $m \in \mathbb{Z}^+$ requires at most m applications of the R_z gate and at most m rounds of communication between client and server. $\square$

Theorem A.3. *Any R_z rotation gate with given θ can be approximately decrypted with arbitrary precision ϵ using at most $\log_2^2(\pi/\epsilon)$ rotation gates.*

Proof. We start by showing that any θ can be approximated by a finite series S with arbitrary precision ϵ using apprropriate number of elements ($M + 1$):

$$\mathcal{S} = \left\{ p\pi + \sum_{m=1}^M \frac{a_m \pi}{2^m} \Big| a_m \in \{-1, 0, 1\}, p \in \mathbb{Z} \right\}, \tag{A11}$$

Given an arbitrary θ, an integral multiple of π can be directly reached using an appropriate value of $p = \theta \pmod \pi$. Now, $\theta_1 = \theta - p\pi < \pi$.

Using a modified series, similar to dyadic expression $\sum_{m=1}^\infty \frac{a_m}{2^m} \in [0, 1], \forall a_m \in \{0, 1\}$, we can say that $\sum_{m=1}^\infty \frac{a_m \pi}{2^m} \in [-\pi, \pi], \forall a_m \in \{-1, 0, 1\}$. However, for finite approximation of θ_1, we need to find $\theta' \in \mathcal{S}$ such that:

$$|\theta_1 - \theta'| < \epsilon. \tag{A12}$$

The precision of the series will be the smallest value achievable in the set $\mathcal{S}$ of $M+1$ elements, i.e., $\pi/2^M$, which implies $|\theta_1 - \theta'| = \pi/2^M$. Hence, the appropriate value of M for approximation with arbitrary precision ϵ will be:

$$M > log_2(\pi/\epsilon). \tag{A13}$$

Using this limit on the number of elements for finite representation of θ, we now find the upper limit on decryption of arbitrary $R_z(\theta)$ using the following property,

$$R_z(x+y) = R_z(x)R_z(y). \tag{A14}$$

Any arbitrary $R_z(\theta)$ can be perfomed using series S using appropriate values of $p \in \mathbb{Z}, a_m \in \{-1, 0, 1\}$ for $m \in \{1, \cdots, M\}$ with $M = \lceil log_2(\pi/\epsilon) \rceil$ as:

$$R_z(\theta) \approx R_z(p\pi + \sum_{m=1}^{M} a_m\pi/2^m),$$

$$\approx R_z(p\pi) \prod_{m=1}^{M} R_z(a_m\pi/2^m). \tag{A15}$$

Here, $R_z(p\pi)$ can be implemented using at most one Z gate as evident from Eq. (11), while $R_z(a_m\pi/2^m)$ gates with $m \in \mathbb{Z}^+$ requires m rotation gates each as proven in Theorem A.2. This shows that any arbitrary rotation gate $R_z(\theta)$ can be implemented using at most M integral $R_z(\pm\pi/2^m)$ gates (Eq. (A15)), which in turn requires m rotation gates each for decryption (Theorem A.2).

Hence, the total rotation gates required for recursive decryption of $R_z(\theta)$ is at most $M^2 = log_2^2(\pi/\epsilon)$, which is also equal to the number of communication rounds needed between client and server to perform the secure computation.

$\square$

Corollary A.3.1. *The decyrption based on an arbitrary R_z rotation is efficient than using a non-parametric gate set whose decomposition is based on the Solovay-Kitaev theorem.*

Proof. According to the Solovay-Kitaev theorem, the transformation of arbitrary $R_z(\theta)$ rotation gates to the H, S, T sequence needs a length of $O(\ln^c(1/\epsilon))$ for ϵ precision. This requires any blind quantum technique based on such non-parametric gates to consume communication rounds of size $O(\ln^c(1/\epsilon))$, where $c = 3.97$ [42].

However, Theorem A.3 shows that an arbitrary rotation gate-based blind approach has an upper bound on communication rounds as $O(\log_2^2(\pi/\epsilon))$.

Hence, showing that a parametric resource set based on $R_z(\theta)$ is more efficient than a non-parametric resource set based on $\{H, S, T\}$ gates for given ϵ precision as:

$$O(\log_2^2(\pi/\epsilon)) < O(\ln^{3.97}(1/\epsilon)), \tag{A16}$$

$\square$

---

[1] K. Azuma, S. E. Economou, D. Elkouss, P. Hilaire, L. Jiang, H.-K. Lo, and I. Tzitrin, Quantum repeaters: From quantum networks to the quantum internet, Reviews of Modern Physics **95**, 045006 (2023).

[2] J. F. Fitzsimons, Private quantum computation: an introduction to blind quantum computing and related protocols, npj Quantum Information **3**, 23 (2017).

[3] F. Schmidt, D. Miller, and P. Van Loock, Error-corrected quantum repeaters with Gottesman-Kitaev-Preskill qudits, Physical Review A **109**, 042427 (2024).

[4] S. Kumar, N. Lauk, and C. Simon, Towards long-distance quantum networks with superconducting processors and optical links, Quantum Science and Technology **4**, 045003 (2019).

[5] S. Wehner, D. Elkouss, and R. Hanson, Quantum internet: A vision for the road ahead, Science **362**, eaam9288 (2018).

[6] A. Childs, Secure assisted quantum computation, Quantum Information and Computation **5**, 456 (2005).

[7] A. Broadbent, J. Fitzsimons, and E. Kashefi, Universal blind quantum computation, in *2009 50th annual IEEE symposium on foundations of computer science* (IEEE, 2009) pp. 517–526.

[8] V. Giovannetti, L. Maccone, T. Morimae, and T. G. Rudolph, Efficient universal blind quantum computation, Physical review letters **111**, 230501 (2013).

[9] C. A. Pérez-Delgado and J. F. Fitzsimons, Iterated Gate Teleportation and Blind Quantum Computation, Physical Review Letters **114**, 220502 (2015).

[10] A. Mantri, C. A. Pérez-Delgado, and J. F. Fitzsimons, Optimal blind quantum computation, Physical review letters **111**, 230502 (2013).

[11] X. Zhang, W. Luo, G. Zeng, J. Weng, Y. Yang, M. Chen, and X. Tan, A hybrid universal blind quantum computation, Information Sciences **498**, 135 (2019).

[12] S. Ma, C. Zhu, X. Liu, H. Li, and S. Li, Universal blind quantum computation with improved brick-

work states, Physical Review A **109** (2024), publisher: American Physical Society (APS).

[13] J. van Dam, G. Avis, T. B. Propp, F. F. da Silva, J. A. Slater, T. E. Northup, and S. Wehner, Hardware requirements for trapped-ion-based verifiable blind quantum computing with a measurement-only client, Quantum Science and Technology **9**, 045031 (2024).

[14] S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther, Demonstration of blind quantum computing, science **335**, 303 (2012).

[15] C. Greganti, M.-C. Roehsner, S. Barz, T. Morimae, and P. Walther, Demonstration of measurement-only blind quantum computing, New Journal of Physics **18**, 013020 (2016).

[16] H.-L. Huang, Q. Zhao, X. Ma, C. Liu, Z.-E. Su, X.-L. Wang, L. Li, N.-L. Liu, B. C. Sanders, C.-Y. Lu, *et al.*, Experimental blind quantum computing for a classical client, Physical review letters **119**, 050503 (2017).

[17] W. K. Tham, H. Ferretti, K. Bonsma-Fisher, A. Brodutch, B. C. Sanders, A. M. Steinberg, and S. Jeffery, Experimental demonstration of quantum fully homomorphic encryption with application in a two-party secure protocol, Physical Review X **10**, 011038 (2020).

[18] Y. Li, L. Cao, W. Luo, H. Zhang, H. Cai, M. F. Karim, F. Gao, J. Fitzsimons, Q. Song, and A.-Q. Liu, Experimental quantum homomorphic encryption using a quantum photonic chip, Physical review letters **132**, 200801 (2024).

[19] A. Broadbent, G. Gutoski, and D. Stebila, Quantum one-time programs, in *Annual Cryptology Conference* (Springer, 2013) pp. 344–360.

[20] C. Gustiani and D. P. DiVincenzo, Blind oracular quantum computation, Quantum science and technology **6**, 045022 (2021).

[21] A. Das and B. C. Sanders, Blind quantum factorization of 21, Physical Review A **106**, 012421 (2022).

[22] T. Kapourniotis, E. Kashefi, D. Leichtle, L. Music, and H. Ollivier, Asymmetric secure multi-party quantum computation with weak clients against dishonest majority, Quantum Science and Technology **10**, 025015 (2025).

[23] K. A. G. Fisher, A. Broadbent, L. K. Shalm, Z. Yan, J. Lavoie, R. Prevedel, T. Jennewein, and K. J. Resch, Quantum computing on encrypted data, Nature Communications **5**, 3074 (2014).

[24] A. Broadbent, Delegating private quantum computations, Canadian Journal of Physics **93**, 941 (2015).

[25] K. Marshall, C. S. Jacobsen, C. Schäfermeier, T. Gehring, C. Weedbrook, and U. L. Andersen, Continuous-variable quantum computing on encrypted data, Nature Communications **7**, 13795 (2016).

[26] X. Tan and X. Zhou, Universal half-blind quantum computation, Annals of Telecommunications **72**, 589 (2017).

[27] Y. Sano, Blind quantum computation using a circuit-based quantum computer, Journal of the Physical Society of Japan **90**, 124001 (2021).

[28] A. Callison and N. Chancellor, Hybrid quantum-classical algorithms in the noisy intermediate-scale quantum era and beyond, Physical Review A **106**, 010101 (2022).

[29] I. Khait, E. Tham, D. Segal, and A. Brodutch, Variational quantum eigensolvers in the era of distributed quantum computers, Physical Review A **108**, L050401 (2023).

[30] Y. Shingu, Y. Takeuchi, S. Endo, S. Kawabata, S. Watabe, T. Nikuni, H. Hakoshima, and Y. Matsuzaki, Variational secure cloud quantum computing, Physical Review A **105**, 022603 (2022).

[31] C. Li, B. Li, O. Amer, R. Shaydulin, S. Chakrabarti, G. Wang, H. Xu, H. Tang, I. Schoch, N. Kumar, *et al.*, Blind quantum machine learning with quantum bipartite correlator, Physical Review Letters **133**, 120602 (2024).

[32] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, Efficient z gates for quantum computing, Physical Review A **96**, 022330 (2017).

[33] J. Chen, D. Ding, C. Huang, and Q. Ye, Compiling arbitrary single-qubit gates via the phase shifts of microwave pulses, Physical Review Research **5** (2023), publisher: American Physical Society (APS).

[34] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, *et al.*, Quantum computing with qiskit, arXiv preprint arXiv:2405.08810 (2024).

[35] V. B. et al., Pennylane: Automatic differentiation of hybrid quantum-classical computations (2022), arXiv:1811.04968 [quant-ph].

[36] N. J. Ross and P. Selinger, Optimal ancilla-free clifford+ t approximation of z-rotations, arXiv preprint arXiv:1403.2975 (2014).

[37] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*, Vol. 2 (Cambridge university press Cambridge, 2001).

[38] T. Morimae and K. Fujii, Secure Entanglement Distillation for Double-Server Blind Quantum Computation, Physical Review Letters **111**, 020502 (2013).

[39] Q. Li, W. H. Chan, C. Wu, and Z. Wen, Triple-server blind quantum computation using entanglement swapping, Physical Review A **89**, 040302 (2014).

[40] T. Morimae, Measurement-only verifiable blind quantum computing with quantum input verification, Physical Review A **94**, 042301 (2016).

[41] J. F. Fitzsimons and E. Kashefi, Unconditionally verifiable blind quantum computation, Physical Review A **96**, 012303 (2017).

[42] C. Dawson and M. Nielsen, The solovay-kitaev algorithm, Quantum Information and Computation **6**, 81 (2006).