Homomorphic Encryption Experiments on IBM’s Cloud Quantum Computing Platform

He-Liang Huang,1,2 You-Wei Zhao,2,3 Tan Li,1,2 Feng-Guang Li,1,2 Yu-Tao Du,1,2 Xiang-Qun Fu,1,2 Shuo Zhang,1,2 Xiang Wang,1,2 and Wan-Su Bao1,2,3

1 Zhengzhou Information Science and Technology Institute, Henan, Zhengzhou 450000, China
2 CAS Centre for Excellence and Synergetic Innovation Centre in Quantum Information and Quantum Physics, University of Science and Technology of China, Hefei, Anhui 230026, China
3 Hefei National Laboratory for Physical Sciences at Microscale and Department of Modern Physics, University of Science and Technology of China, Hefei, Anhui 230026, China

Quantum computing has undergone rapid development in recent years. Owing to limitations on scalability, personal quantum computers still seem slightly unrealistic in the near future. The first practical quantum computer for ordinary users is likely to be on the cloud. However, the adoption of cloud computing is possible only if security is ensured. Homomorphic encryption is a cryptographic protocol that allows computation to be performed on encrypted data without decrypting them, so it is well suited to cloud computing. Here, we first applied homomorphic encryption on IBM’s cloud quantum computer platform. In our experiments, we successfully implemented a quantum algorithm for linear equations while protecting our privacy. This demonstration opens a feasible path to the next stage of development of cloud quantum information technology.

Keywords Quantum Computing, Homomorphic Encryption, Cloud Computing, IBM Quantum Experience, Linear Equations

1. Introduction

In recent years, much progress has been made in developing quantum computing technologies [1–5]. Because of the quantum superposition principle, quantum computers can outperform their classical counterparts when performing certain tasks, for example, Shor’s algorithm [6–10], quantum simulation [11–14], solving linear systems of equations [15–17], and quantum machine learning [18, 19]. Therefore, the emergence of quantum computers will change the world. Owing to high construction and maintenance costs, the first quantum computers are likely to be owned only by a small number of organizations. Fortunately, however, with cloud service, ordinary users are also expected to be able to apply so as to use these quantum computers.

As expected, IBM recently made a five-qubit quantum computer publicly available over the cloud [20]. Based on a five-qubit superconducting chip in a star geometry and a full Clifford algebra, the system can be reprogrammed and allows for circuit design and simulation. Through the classical internet, users can easily test and execute algorithms on an interactive platform called Quantum Experience. Several experiments have already been reported [21–23].

Future cloud quantum computing is likely to be available to users through an interface similar to IBM’s cloud computing platform, where users interact with the platform through a website. In this case, the quantum circuit, input data, and output data of users are completely accessible to the server. While sharing cloud-based computational resources for quantum computing, we also need to consider privacy. Although a number of protocols and experiments have been proposed to develop secure cloud quantum computing [24, 27], these encryption methods are not suited to the current level of technology, because input or output data cannot be accessible to the servers on the website in the previous protocol.

In classical cryptography, homomorphic encryption [28–30] is a scheme that allows certain operations to be performed on encrypted data without decryption. Thus, users can provide encrypted data to a remote server for processing without having to reveal the plaintext. Although the data are open to the server, the server cannot reveal the real data because the data are encrypted. After the server outputs the results to a user, the user can recover the actual output data through his privacy key. Therefore, homomorphic encryption has become a practical encryption technique for cloud computing.

In this study, we designed a homomorphic encryption protocol for cloud quantum computing, which is suitable for IBM’s cloud server. On the basis of the basic quantum gates provided by the server, we developed a series of construction methods for various operations. Finally, we successfully implemented a quantum algorithm for linear equations on IBM’s cloud server while protecting our privacy. This work will hopefully motivate more people to get involved in this field, because this study is the first to consider the security of users’ data on IBM’s cloud server and can provide guidance for future large-scale cloud quantum computing.

2. Methods

To solve linear equations on a quantum computer, we employ the quantum algorithm proposed by Harrow et al. [15], which can provide an exponential speedup over existing classical algorithms. Given a matrix $A$ and a vector $\vec{b}$, we aim to solve the equations $A\vec{x} = \vec{b}$. To convert the problem to a quantum version, we rescale $\vec{x}$ and $\vec{b}$ to $\|\vec{x}\| = \|\vec{b}\| = 1$. Thus, we can encode the problem as...
Considero la implementación de un algoritmo para resolver ecuaciones lineales utilizando el esquema de cifrado homomórfico. En este contexto, el problema consiste en enviar un conjunto de ecuaciones lineales a un servidor cuántico a través de una red de red privada. El servidor debe realizar operaciones matemáticas sobre los datos cifrados y enviar los resultados cifrados a la parte privada. El algoritmo que utilizamos es el siguiente:

1. **Preparación del modelo:** La matriz de ecuaciones lineales $A\vec{x} = \vec{b}$ se cifra utilizando el esquema de cifrado homomórfico. La matemática de esta etapa se muestra en la ecuación (1).

2. **Implementación del algoritmo:** Para resolver la matriz de ecuaciones, se implementa un circuito cuántico que utiliza el punto de partida $\{u_i\}$ y los valores propios $\lambda_i$ de $A$. Los valores $\lambda_i$ son almacenados en el registro de valores propios $|\lambda_i\rangle$, donde $|\lambda_i\rangle$ es el registro de valores propios después de la estimación de fase inversa. La ecuación (2) ilustra este proceso.

3. **Transformación de fase inversa:** La transformación de fase inversa $\lambda_i^{-1}$ se realiza para desentrenar el registro de valores propios después de la estimación de fase inversa. La ecuación (3) muestra este proceso.

La secuencia se completará cuando se reciba el resultado final de la parte privada. Este enfoque permite resolver ecuaciones lineales de forma segura y eficiente, sin exponer los datos originales a la parte pública.
The construction of the Hadamard gate for solving the equations (9) and (10). (d) controlled protocol to solve systems of 2 value qubit, and an ancilla qubit, we can use the protocol on IBM’s cloud quantum computing platform. Using one state qubit as the two-vector as

\[ A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \]

denotes the decryption operation; then the user can obtain the actual results. Note that the homomorphic encryption process perfectly hides the input and output of the user, and the server cannot obtain any of the private data, because it deals only with encrypted data.

3. Experimental Realization

Here we present a proof-of-principle experiment of this protocol on IBM’s cloud quantum computing platform. Using one state qubit as the two-vector \(|b\rangle\), one eigenvalue qubit, and an ancilla qubit, we can use the protocol to solve systems of 2 \(\times\) 2 linear equations. The quantum circuit of the algorithm for 2 \(\times\) 2 linear equations can be compiled into the circuit shown in Fig. 3(a) [13]. A unitary \(R\) is introduced to diagonalize matrix \(A\) as \(A = R^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} R\), where \(\lambda_i\) is the eigenvalue of \(A\). \(R(\lambda^{-1})\) rotation can be realized by using a controlled \(R_y(\theta)\), where \(R_y(\theta) = \exp(-i \theta \sigma_y / 2)\), \(\sigma_y\) is the usual Pauli matrix, and \(\theta\) is controlled by the eigenvalue qubit with the function \(\theta_i = -2 \arccos(\lambda_1 / \lambda_2)\). The algorithm succeeds with probability when the ancilla qubit is measured in the state [1]. In our implementation, we choose the following two systems of linear equations:

\[
\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \cdot \bar{x} = \begin{pmatrix} 1 / \sqrt{2} + 0.7 \\ 1 / \sqrt{2} + 0.3 \end{pmatrix}
\]

(7)

\[
\begin{pmatrix} 1.75 & 0.75 \\ 0.75 & 1.75 \end{pmatrix} \cdot \bar{x} = \begin{pmatrix} 1 / \sqrt{2} + 1.75 \\ -1 / \sqrt{2} + 0.75 \end{pmatrix}
\]

(8)

Without loss of generality, we set the private key of the user to \(a_1 = 1, a_2 = 0\). By substituting \(x_i = y_i + a_i\) into the linear equations, the user can perfectly hide his input data, and the equations can be rewritten as

\[
\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \cdot \bar{y} = \begin{pmatrix} 1 \\ 1 / \sqrt{2} \end{pmatrix}
\]

(9)

\[
\begin{pmatrix} 1.75 & 0.75 \\ 0.75 & 1.75 \end{pmatrix} \cdot \bar{y} = \begin{pmatrix} 1 \\ -1 / \sqrt{2} \end{pmatrix}
\]

(10)

Then the user can encode the circuit on IBM’s cloud quantum computing platform. For both of these linear equations, the \(R\) gate in Fig. 3(a) can be compiled into a Hadamard gate [see Fig. 3(c)]. Note that IBM provides only the CNOT gate as a two-qubit gate. To realize the controlled \(R_y(\theta)\) operation (\(\theta\) is equal to \(-57.34^\circ\) for both of the linear equations), we decomposed the controlled \(R_y(\theta)\) gate into two CNOT gates, one \(R_y(\theta / 2)\) gate and one \(R_y(-\theta / 2)\) gate. In our implementation, we set the eigenvalue register as the central qubit of IBM’s superconductor quantum chip. As the chip allows operation of CNOT gates only with the central qubit as the target qubit in their star geometry, if we want to operate CNOT gates with the central qubit as a control qubit, we need to combine a CNOT gate and four Hadamard gates. Then the controlled \(R_y(\theta)\) gate can be compiled to a combination of several Hadamard gates, CNOT gates, the \(R_y(\theta / 2)\) gate, and the \(R_y(-\theta / 2)\) gate [see Fig. 3(b)]. Now the question becomes how to construct an \(R\) gate, because only Clifford gates \((X, Y, Z, H, S, S^\dagger\) and CNOT) and two non-Clifford gates \((T\) and \(T^\dagger\)) are available on the platform. Adding almost any non-Clifford gate to the Clifford gates is universal [31]. Therefore, by adding the \(T\) gate to the Clifford gates, it is possible to reach all the points of the Bloch sphere. A Monte Carlo simulation indicates that the more \(T\) gates our circuit has, the more densely we can cover the Bloch sphere with states we can reach. Figure 4 depicts the states attainable by adding at most 1, 3, 5, and 7 \(T\) gates to the Clifford gates.

In Fig. 4, the red dot is the \(R_y(\theta / 2)\) operation we desired. On the basis of the results of the numerical simulation, we can approximate the \(R_y(\theta / 2)\) gate by the gate \(R_S\), which is a combination of seven \(T\) gates and seven Hadamard gates [see Fig. 3(d)]. To characterize its accuracy, we compute the similarity \(F = 1 / 2 \cdot \text{Tr}(U_{\text{ideal}}U_{\text{simu}})\) as 0.998, where \(U_{\text{ideal}}\) is the ideal unitary operation \(R_y(\theta / 2)\), and \(U_{\text{simu}}\) is the simulated unitary operation \(R_S\), indicating that our simulated unitary operation is very similar to the ideal operation \(R_y(\theta / 2)\). Through the red dot \((R_y(\theta / 2)\) operation) and the purple dot (simulated operation \(R_S\)) in Fig. 4(d), it is more intuitive that these two dots are very close. At this point, we can compile the full circuit for solving the two equations on the IBM servers.

4. Results

Measuring the first qubit of the circuit in Fig. 3(a) in the Pauli \(Z, X, Y\) basis, we can obtain the solutions of
FIG. 4. (color online). (a), (b), (c) and (d) are the Bloch sphere with the dots are the attainable states of $U|0\rangle$, where $U$ is the operation by adding at most 1, 3, 5 and 7 $T$ gates to the Clifford gates respectively. The red dot in (a), (b), (c) and (d) is the $R_y(\theta/2)$ operation we desired for solving the equations (9) and (10). The purple dot in (d) is the simulated operation of $R_y(\theta/2)$.

FIG. 5. (color online). Experimental results. (a) and (b) are the measurement results of the output state of equations (9) and (10). For each equations, the ideal (red bar) and experimentally obtained (blue bar) expectation values of the Pauli Z, X, and Y are presented. The error bars denote one standard deviation, deduced from propagated Poissonian counting statistics of the raw detection events.

the equations. Figure 5(a) and 5(b) show both the ideal (red bar) and experimentally obtained (blue bar) expectation values for each Pauli operator when the algorithm is implemented to solve equations (9) and (10), respectively. We compute the fidelity of the output state as $F = \langle x|\rho_{\text{exp}}|x\rangle$, where $|x\rangle$ is the ideal output state, and $\rho_{\text{exp}}$ is the experimentally obtained output state from the measurement results of the Pauli $Z$, $X$, and $Y$ basis. The output states have fidelities of 0.992(1) and 0.920(7) for equations (9) and (10), respectively, indicating that our experiments yielded highly reliable results.

By postprocessing the results using a classical computer, the user can easily decrypt the secret results to obtain the actual results as $\{x_1 = 1.7173, x_2 = 0.6967\}$ and $\{x_1 = 1.7227, x_2 = 0.6911\}$ for equations (7) and (8), respectively. Theoretical analysis shows that the error is within 2% of the actual solution. Thus, the homomorphic encryption protocol is found to be successful.

5. Conclusion

In summary, we presented the first experimental demonstration of a homomorphic encryption protocol for solving linear equations on IBM’s cloud quantum computer platform. The protocol is very suitable for current technology, which enables users to delegate the task of computation by encoding the circuit on the website of quantum servers while protecting their data. Even though the current quantum computations on IBM’s server are proof-of-principle demonstrations, the process can be scaled to larger systems in the future. Ideally, this work will provide a workable solution for future cloud quantum computation.

Acknowledgement

The author acknowledges the use of IBM’s Quantum Experience for this work. The views expressed are those of the author and do not reflect the official policy or position of IBM or the IBM Quantum Experience team. This project is supported by the National Basic Research Program of China (Grant No. 2013CB338002), National Natural Science Foundation of China (Grants No. 11504430 and No. 61502526).

* glhhl0773@126.com

[1] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O’Brien, Nature 464, 45 (2010).
[2] X.-L. Wang, X.-D. Cai, Z.-E. Su, M.-C. Chen, D. Wu, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, Nature 518, 516 (2015).
[3] J. Kelly, R. Barends, A. Fowler, A. Megrant, E. Jeffrey, T. White, D. Sank, J. Mutus, B. Campbell, Y. Chen, et al., Nature 519, 66 (2015).
[4] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. White, J. Mutus, A. Fowler, B. Campbell, et al., Nature 508, 500 (2014).
[5] R. Barends, L. Lamata, J. Kelly, L. García-Álvarez, A. Fowler, A. Megrant, E. Jeffrey, T. White, D. Sank, J. Mutus, et al., Nature communications 6 (2015).
[6] P. W. Shor, SIAM review 41, 303 (1999).
[7] L. M. Vandersypen, M. Steffen, G. Breyla, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Nature 414, 883 (2001).
[8] C.-Y. Lu, D. E. Browne, T. Yang, and J.-W. Pan, Physical Review Letters 99, 250504 (2007).
[9] E. Lucero, R. Barends, Y. Chen, J. Kelly, M. Mariantoni, A. Megrant, P. O’Malley, D. Sank, A. Vainsencher, J. Wenner, et al., Nature Physics 8, 719 (2012).
[10] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and...
R. Blatt, Science 351, 1068 (2016).
[11] R. P. Feynman, International journal of theoretical physics 21, 467 (1982).
[12] R. Blatt and C. Roos, Nature Physics 8, 277 (2012).
[13] A. A. Houck, H. E. Türeci, and J. Koch, Nature Physics 8, 292 (2012).
[14] A. Aspuru-Guzik and P. Walther, Nature Physics 8, 285 (2012).
[15] A. W. Harrow, A. Hassidim, and S. Lloyd, Physical review letters 103, 150502 (2009).
[16] S. Barz, I. Kassal, M. Ringbauer, Y. O. Lipp, B. Dakić, A. Aspuru-Guzik, and P. Walther, Scientific reports 4 (2014).
[17] X.-D. Cai, C. Weedbrook, Z.-E. Su, M.-C. Chen, M. Gu, M.-J. Zhu, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, Physical review letters 110, 230501 (2013).
[18] P. Rebentrost, M. Mohseni, and S. Lloyd, Physical review letters 113, 130503 (2014).
[19] X.-D. Cai, D. Wu, Z.-E. Su, M.-C. Chen, X.-L. Wang, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, Physical review letters 114, 110504 (2015).
[20] IBM, “The quantum experience,” http://www.research.ibm.com/quantum/ (2016).
[21] S. J. Devitt, arXiv preprint arXiv:1605.05709 (2016).
[22] D. Alsina and J. I. Latorre, arXiv preprint arXiv:1605.04220 (2016).
[23] R. Rundle, T. Tilma, J. Samson, and M. Everitt, arXiv preprint arXiv:1605.08922 (2016).
[24] A. Broadbent, J. Fitzsimons, and E. Kashefi, in Foundations of Computer Science, 2009. FOCS’09. 50th Annual IEEE Symposium on (IEEE, 2009) pp. 517–526.
[25] J. F. Fitzsimons and E. Kashefi, arXiv preprint arXiv:1203.5217 (2012).
[26] S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther, Science 335, 303 (2012).
[27] S. Barz, J. F. Fitzsimons, E. Kashefi, and P. Walther, Nature physics 9, 727 (2013).
[28] R. L. Rivest, L. Adleman, and M. L. Dertouzos, Foundations of secure computation 4, 169 (1978).
[29] C. Gentry, A fully homomorphic encryption scheme, Ph.D. thesis, Stanford University (2009).
[30] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, in Annual International Conference on the Theory and Applications of Cryptographic Techniques (Springer, 2010) pp. 24–43.
[31] M. A. Nielsen and I. L. Chuang, Quantum computation and quantum information (Cambridge university press, 2010).