Front. Phys. 12(1), 120305 (2017)
DOI 10.1007/s11467-016-0643-9

RESEARCH ARTICLE

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], Wan-Su Bao[1,2,†]

[1] *Zhengzhou Information Science and Technology Institute, 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 230026, China*
[3] *Hefei National Laboratory for Physical Sciences at Microscale and Department of Modern Physics, University of Science and Technology of China, Hefei 230026, China*
Corresponding author. E-mail: †glhhl0773@126.com
Received November 7, 2016; accepted November 25, 2016

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

PACS numbers 03.67.Ac, 03.65.Ud, 03.67.Lx, 42.50.p

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 quantum 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 quantum 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

*arXiv: 1612.02886.

Frontiers of Physics

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 $\boldsymbol{b}$, we aim to solve the equations $A\boldsymbol{x} = \boldsymbol{b}$. To convert the problem to a quantum version, we rescale $\boldsymbol{x}$ and $\boldsymbol{b}$ to $\|\boldsymbol{x}\| = \|\boldsymbol{b}\| = 1$. Thus, we can encode the problem as

$$A|x\rangle = |b\rangle. \tag{1}$$

Denote $\{|u_i\rangle\}$ and $\{\lambda_i\}$ as the eigenbasis and eigenvalues of matrix A, respectively. The input state $|b\rangle$ can be expanded in the eigenbasis of A as $|b\rangle = \sum_{i=1}^{N} \beta_i|u_i\rangle$, where $\beta_i = \langle u_i|b\rangle$. To seek the solution $|x\rangle = A^{-1}|b\rangle/||A^{-1}|b\rangle||$, the algorithm can be decomposed into three main steps (see Fig. 1).

In the first step, phase estimation is applied to the transformation of $|b\rangle|0\rangle^{\otimes n}$ into $\sum_{i=1}^{N} \beta_i|u_i\rangle|\lambda_i\rangle$, where $|0\rangle^{\otimes n}$ is the eigenvalue register of n qubits, and eigenvalues $|\lambda_i\rangle$ are stored in the eigenvalue register after phase estimation.

In the second step, the map $|\lambda_i\rangle \to {\lambda_i}^{-1}|\lambda_i\rangle$ is implemented to extract the eigenvalues of A^{-1}. By implementing a controlled $R(\lambda^{-1})$ rotation on an additional ancilla qubit initially in the state $|0\rangle$, we can transform the system to

$$\sum_{i=1}^{N} \beta_i|u_i\rangle|\lambda_i\rangle \left(\sqrt{1 - \frac{C^2}{\lambda_i^2}}|0\rangle + \frac{C}{\lambda_i}|1\rangle\right). \tag{2}$$

The final step is to implement inverse phase estimation to disentangle the eigenvalue register to $|0\rangle^{\otimes n}$. Then, we have

$$\sum_{i=1}^{N} \beta_i|u_i\rangle \left(\sqrt{1 - \frac{C^2}{\lambda_i^2}}|0\rangle + \frac{C}{\lambda_i}|1\rangle\right). \tag{3}$$

After measuring and postselecting the ancillary qubit of $|1\rangle$, we will obtain an output state $\sum_{i=1}^{N} C(\beta_i/\lambda_i)|u_i\rangle$, which is proportional to our expected result state $|x\rangle$.

To delegate the task of solving linear equations to IBM's cloud quantum computing platform, one can directly encode the quantum circuit on the servers without considering security. This is equivalent to sending the cloud server the matrix A and vector $\boldsymbol{b}$ directly. However, when security is required, this approach is no longer feasible. Inspired by homomorphic encryption (see Fig. 2), we can compile a homomorphically encrypted version of the algorithm.

To implement homomorphic encryption, we must ensure that the operations of the server and the encryption scheme of the user meet the following conditions:

$$f(E(x)) = E(f(x)), \tag{4}$$

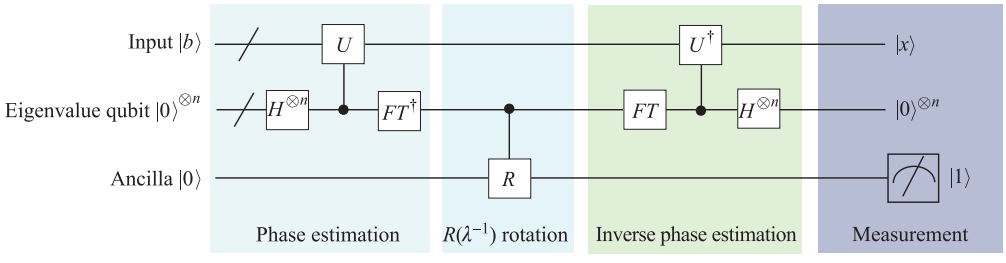

Fig. 1 Quantum circuit for solving systems of linear equations. The circuit of the original quantum algorithm. In the circuit, $U = \sum_{k=0}^{T-1} |k\rangle\langle k| \otimes e^{iAkt_0/T}$, H is the Hadamard gate. FT is the Fourier transformation and $FT^{\dagger}$ is the inverse Fourier transformation. The output $|x\rangle$ is obtained when the ancilla is detected as $|1\rangle$.

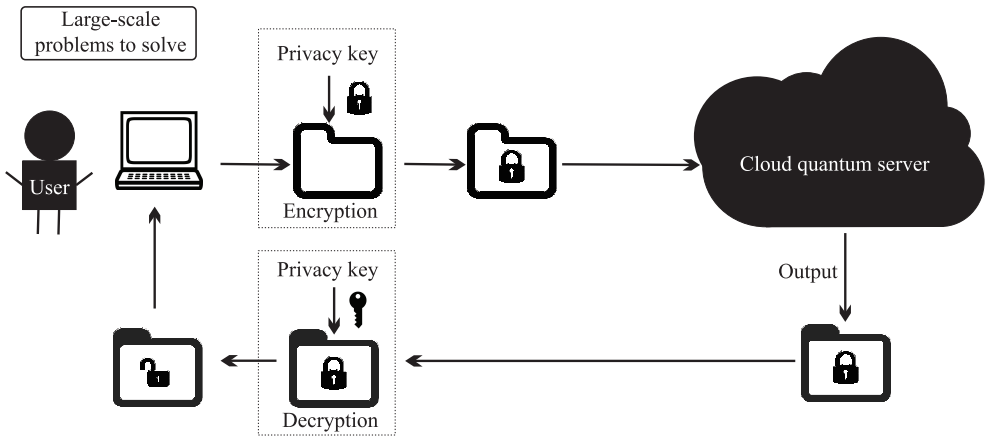

Fig. 2 The homomorphic encryption scheme. User encrypt the data using privacy key before sending it to the cloud quantum server. Server do not own the privacy key so that it cannot learn anything about the encrypted data. If the operations of server are homomorphic operations, then server can utilize arbitrary computations on the encrypted data without decrypting it. The output of the server remains in encrypted form, and can only be recovered by the user who have the privacy key.

where f denotes the operations of the server, and $E(x)$ denotes the encryption of the message x. According to the conditions of homomorphic encryption, we can design the protocol as follows:

Step 1. For a linear equation $A\boldsymbol{x} = \boldsymbol{b}$ with n variants, the user randomly chooses $a_i \in \{0, 1\}$ as private keys, letting $x_i = y_i + a_i$, and then substitutes it into the linear equations

$$A(\boldsymbol{y}_i + \boldsymbol{a}_i) = \boldsymbol{b}. \tag{5}$$

The user rewrites the equations as $A\boldsymbol{y}_i = \boldsymbol{b}'$, where $\boldsymbol{b}' = \boldsymbol{b} - A\boldsymbol{a}_i$, and then sends A and $\boldsymbol{b}'$ to the cloud server.

Step 2. The server implements a quantum algorithm to solve the equations received from the user and feeds back the results $\boldsymbol{r}$ to the user.

Step 3. The user decrypts the results as

$$x_i = r_i + a_i. \tag{6}$$

Here, we analyze the entire process of the protocol. Step 1 can be regarded as the encryption process. That is, for $\forall \boldsymbol{x}$, $E(\boldsymbol{x}) = \boldsymbol{x}'$, where $\boldsymbol{x}'$ is the encryption of message $\boldsymbol{x}$. Let $g(\boldsymbol{x}')$ and F be linear equations and the algorithm for solving linear equations, respectively. Then step 2 can be represented as $F(g(\boldsymbol{x}')) \to \boldsymbol{x}'$. Finally, step 3 is used to decrypt the results as $D(\boldsymbol{x}') \to \boldsymbol{x}$, where D 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 2×2 linear equations. The quantum circuit of the algorithm for 2×2 linear equations can be compiled into the circuit shown in Fig. 3(a) [13]. A unitary R is introduced to diagonalize matrix A as $A = R^{\dagger}\left(\begin{smallmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{smallmatrix}\right)R$, where λ_i is the eigenvalue of A. $R(\lambda^{-1})$ rotation can be realized by using a controlled $R_y(\theta)$, where $R_y(\theta) = \exp(-\mathrm{i}\theta\sigma_y/2)$, σ_y is the usual Pauli matrix, and θ 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 $|1\rangle$. In our implementation, we choose the following two systems of linear equations:

$$\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \cdot \boldsymbol{x} = \begin{pmatrix} 1/\sqrt{2} + 0.7 \\ 1/\sqrt{2} + 0.3 \end{pmatrix}, \tag{7}$$

$$\begin{pmatrix} 1.75 & 0.75 \\ 0.75 & 1.75 \end{pmatrix} \cdot \boldsymbol{x} = \begin{pmatrix} 1/\sqrt{2} + 1.75 \\ -1/\sqrt{2} + 0.75 \end{pmatrix}. \tag{8}$$

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

$$\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \cdot \boldsymbol{y} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} / \sqrt{2}, \tag{9}$$

$$\begin{pmatrix} 1.75 & 0.75 \\ 0.75 & 1.75 \end{pmatrix} \cdot \boldsymbol{y} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} / \sqrt{2}. \tag{10}$$

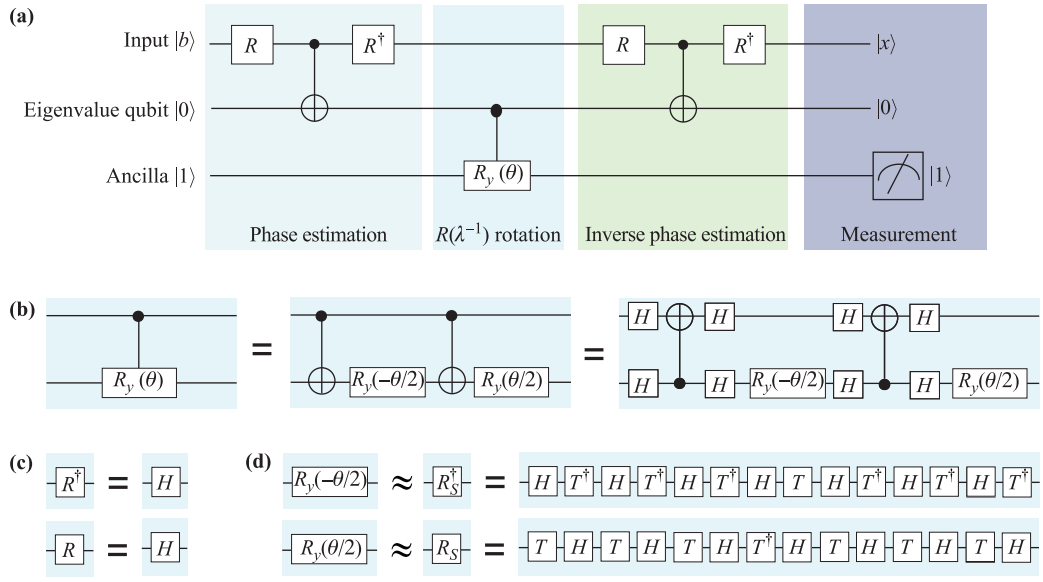

Fig. 3 The construction of the gates for solving Eqs. (9) and (10). **(a)** The optimised circuit for 2×2 system of linear equations. R is a unitary operation that can diagonalizes the matrix A as $A = R^{\dagger}\left(\begin{smallmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{smallmatrix}\right)R$, where λ_i is the eigenvalue of A. **(b)** The construction of the controlled $R_y(\theta)$. **(c)** The R gate in (a) can be compiled to Hadamard gate for solving the equations (9) and (10). **(d)** The construction of the $R_y(\theta/2)$ and $R_y(-\theta/2)$ in the (b).

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 (θ is equal to $-57.34°$ 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 $F = 1/2 \cdot Tr(U_{ideal}U_{simu})$ as 0.998, where U_{ideal} is the ideal unitary operation $R_y(\theta/2)$, and U_{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 the equations. Figures 5(a) and (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 Eqs. (9) and (10), respectively. We compute the fidelity of the output state as $F = \langle x|\rho_{\exp}|x\rangle$, where $|x\rangle$ is the ideal output state, and $\rho_{\exp}$ is the experimentally output state from the measurement results of the Pauli Z, X, Y basis. The output states have fidelities of 0.992(1) and 0.920(7) for Eqs. (9)

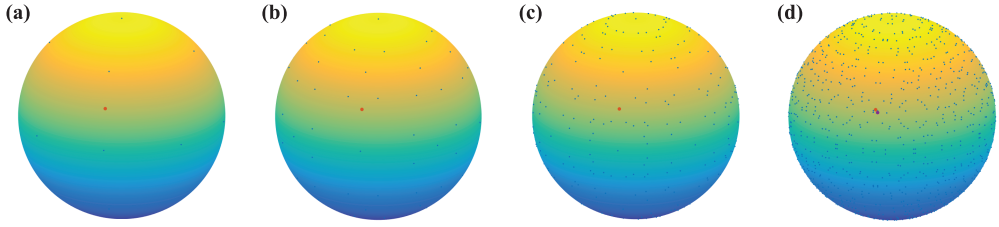

Fig. 4 **(a)**, **(b)**, **(c)** and **(d)** are the Bloch sphere with the dots are the attainable states of $U|0\rangle$, where U is the operation by adding at most 1, 3, 5, and 7 T gates to the Clifford gates respectively. The red dot in (a), (b), (c), and (d) is the $R_y(\theta/2)$ operation we desired for solving Eqs. (9) and (10). The purple dot in (d) is the simulated operation of $R_y(\theta/2)$.

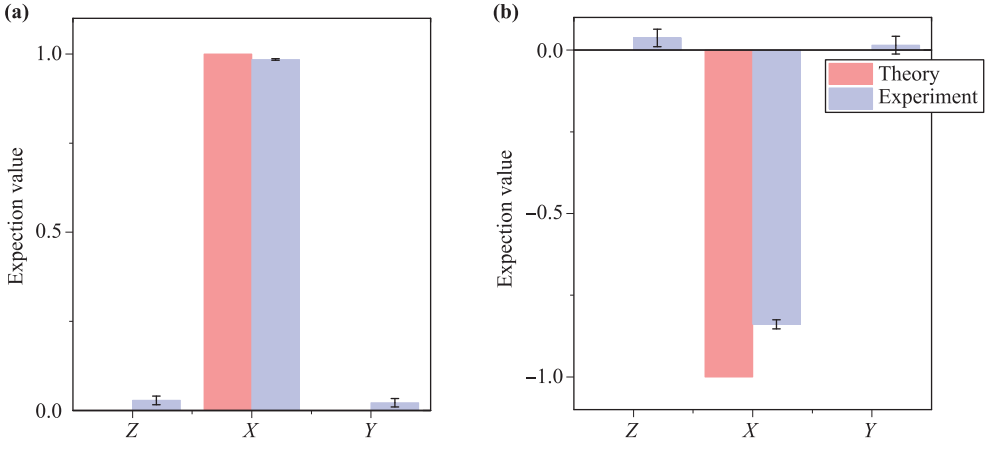

Fig. 5 Experimental results. **(a)** and **(b)** are the measurement results of the output state of Eqs. (9) and (10). For each equations, the ideal (red bar) and experimentally obtained (blue bar) expectation values of the Pauli Z, X, and Y are presented. The error bars denote one standard deviation,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

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 Eqs. (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.

Acknowledgements The authors acknowledge 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 was supported by the National Basic Research Program of China (Grant No. 2013CB338002), National Natural Science Foundation of China (Grant Nos. 11504430 and 61502526).

References and notes

1. T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien, Quantum computers, *Nature* 464(7285), 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, Quantum teleportation of multiple degrees of freedom of a single photon, *Nature* 518(7540), 516 (2015)
3. J. Kelly, R. Barends, A. Fowler, A. Megrant, E. Jeffrey, T. White, D. Sank, J. Mutus, B. Campbell, Y.

Chen, Z. Chen, B. Chiaro, A. Dunsworth, I. C. Hoi, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, A. Vainsencher, J. Wenner, A. N. Cleland, and J. M. Martinis, State preservation by repetitive error detection in a superconducting quantum circuit, *Nature* 519(7541), 66 (2015)

4. R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. White, J. Mutus, A. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O'Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and J. M. Martinis, Superconducting quantum circuits at the surface code threshold for fault tolerance, *Nature* 508(7497), 500 (2014)
5. R. Barends, L. Lamata, J. Kelly, L. García-Álvarez, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Yu Chen, Z. Chen, B. Chiaro, A. Dunsworth, I.-C. Hoi, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, A. Vainsencher, J. Wenner, E. Solano, and J. M. Martinis, Digital quantum simulation of fermionic models with a superconducting circuitm, *Nature Communications* 6, 7654 (2015)
6. P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Rev.* 41(2), 303 (1999)
7. L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance, *Nature* 414(6866), 883 (2001)
8. C. Y. Lu, D. E. Browne, T. Yang, and J. W. Pan, Demonstration of a compiled version of Shor's quantum factoring algorithm using photonic qubits, *Phys. Rev. Lett.* 99(25), 250504 (2007)
9. E. Lucero, R. Barends, Y. Chen, J. Kelly, M. Mariantoni, A. Megrant, P. O'Malley, D. Sank, A. Vainsencher, J. Wenner, T. White, Y. Yin, A. N. Cleland, and J. M. Martinis, Computing prime factors with a Josephson phase qubit quantum processor, *Nat. Phys.* 8(10), 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, Realization of a scalable Shor algorithm, *Science* 351(6277), 1068 (2016)
11. R. P. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* 21(6–7), 467 (1982)
12. R. Blatt and C. Roos, Quantum simulations with trapped ions, *Nat. Phys.* 8(4), 277 (2012)
13. A. A. Houck, H. E. Türeci, and J. Koch, On-chip quantum simulation with superconducting circuits, *Nat. Phys.* 8(4), 292 (2012)
14. A. Aspuru-Guzik and P. Walther, Photonic quantum simulators, *Nat. Phys.* 8(4), 285 (2012)
15. A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* 103(15), 150502 (2009)
16. S. Barz, I. Kassal, M. Ringbauer, Y. O. Lipp, B. Dakić A. Aspuru-Guzik, and P. Walther, A two-qubit photonic quantum processor and its application to solving systems of linear equations, *Sci. Rep.* 4, 6115 (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, Experimental quantum computing to solve systems of linear equations, *Phys. Rev. Lett.* 110(23), 230501 (2013)
18. P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification, *Phys. Rev. Lett.* 113(13), 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, Entanglement-based machine learning on a quantum computer, *Phys. Rev. Lett.* 114(11), 110504 (2015)
20. IBM, The quantum experience, http://www. research.ibm.com/quantum/, 2016
21. S. J. Devitt, Performing quantum computing experiments in the cloud, *Phys. Rev. A* 94, 032329 (2016)
22. D. Alsina and J. I. Latorre, Experimental test of Mermin inequalities on a 5-qubit quantum computer, *Phys. Rev. A* 94, 012314 (2016)
23. R. Rundle, T. Tilma, J. Samson, and M. Everitt, Quantum state reconstruction made easy: A direct method for tomography, arXiv: 1605.08922 (2016)
24. A. Broadbent, J. Fitzsimons, and E. Kashefi, Universal blind quantum computation, in: 2009 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009), IEEE, 2009, pp 517–526
25. J. F. Fitzsimons and E. Kashefi, Unconditionally verifiable blind computation, arXiv: 1203.5217 (2012)
26. S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther, Demonstration of blind quantum computing, *Science* 335(6066), 303 (2012)
27. S. Barz, J. F. Fitzsimons, E. Kashefi, and P. Walther, Experimental verification of quantum computation, *Nat. Phys.* 9(11), 727 (2013)
28. R. L. Rivest, L. Adleman, and M. L. Dertouzos, On data banks and privacy homomorphisms, *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: Cambridge University Press, 2010