Frontiers of Physics
https://doi.org/10.1007/s11467-021-1141-2
Front. Phys.
17(4), 41501 (2022)

RESEARCH ARTICLE

A quantum circuit design of AES requiring fewer quantum qubits and gate operations

Ze-Guo Wang[1], **Shi-Jie Wei**[1,2,†], **Gui-Lu Long**[1,2,3,4,‡]

[1] *State Key Laboratory of Low-Dimensional Quantum Physics and Department of Physics, Tsinghua University, Beijing 100084, China*
[2] *Beijing Academy of Quantum Information Sciences, Beijing 100193, China*
[3] *Beijing National Research Center for Information Science and Technology and School of Information Tsinghua University, Beijing 100084, China*
[4] *Frontier Science Center for Quantum Information, Beijing 100084, China*
Corresponding authors. E-mail: [†] *weishijie@mail.tsinghua.edu.cn,* [‡] *gllong@tsinghua.edu.cn*
Received November 1, 2021; accepted December 13, 2021

Advanced Encryption Standard (AES) is one of the most widely used block ciphers nowadays, and has been established as an encryption standard in 2001. Here we design AES-128 and the sample-AES (S-AES) quantum circuits for deciphering. In the quantum circuit of AES-128, we perform an affine transformation for the SubBytes part to solve the problem that the initial state of the output qubits in SubBytes is not the $|0\rangle^{\otimes 8}$ state. After that, we are able to encode the new round sub-key on the qubits encoding the previous round sub-key, and this improvement reduces the number of qubits used by 224 compared with Langenberg *et al.*'s implementation. For S-AES, a complete quantum circuit is presented with only 48 qubits, which is already within the reach of existing noisy intermediate-scale quantum computers.

Keywords AES, S-AES, quantum circuit, quantum attack

1 Introduction

Classical cryptography are divided into asymmetric cryptography and symmetric cryptography [1]. RSA cryptography [2] is a typical asymmetric cipher, whose security relies on our inability to effectively factor large integers. However, RSA cryptography will be broken in the face of quantum computer. Shor's algorithm [3] can effectively solve the problem in $O((\log N)^3)$ time, where N is the large integer to be factorized. AES is a typical symmetric cryptography [4]. Grover's quantum algorithm [5, 6], finding a marked item from an unsorted database with square-root speedup, can attack the AES cryptography. Fortunately, the AES is not fully broken. By doubling the length of the key in AES, its security can be maintained. However, it is vital to study the level of influence of quantum computer on the AES.

Yamamura *et al.* [7] studied the impact of quantum algorithms on symmetric cryptography. Such as the AES cryptography, which can be decrypted in $O(2^{n/2})$ time using Grover's algorithm, where n is the key length. Kaplan proposed a quantum version [8] of the classical Meet-in-the-middle attack [9] using the Ambainis quantum algorithm [10]. The time complexity and qubits complexity of Kaplen's algorithm both are $O(2^{2n/3})$. In 2014, Roetteler *et al.* proposed the quantum related key attack [11] by using Simon's algorithm [12]. Grassl *et al.* [13] presented a quantum implementation of AES in 2015 and analyzed the consumption of quantum resources. This is the first time that the "zig-zag" method is used to reduce the number of qubits. In the "zig-zag" method, the qubits that are used to encode the cipher in the front rounds of encryption can be released and reused to encode later rounds of cipher. Later on, Kim *et al.* [14] made an improvement on SubBytes operation, one of the four operations in AES, in Grassl *et al.*'s work, which saves one multiplication operation. In 2018, Almazrooie *et al.* [15] proposed a quantum circuit that uses Grover's algorithm to attack S-AES cryptography. Owing to the small size, this is the most likely case to implement in the near-term quantum computing systems [16, 17]. Recently, Langenberg *et al.* designed a quantum circuit [18] for SubBytes based on Boyar *et al.*'s classical algorithm [19]. It reduced the number of Toffoli gates by 88% compared to Grassl *et al.*'s work.

In this paper, we design an improved key expansion

* This article can also be found at http://journal.hep.com.cn/fop/EN/10.1007/s11467-021-1141-2.

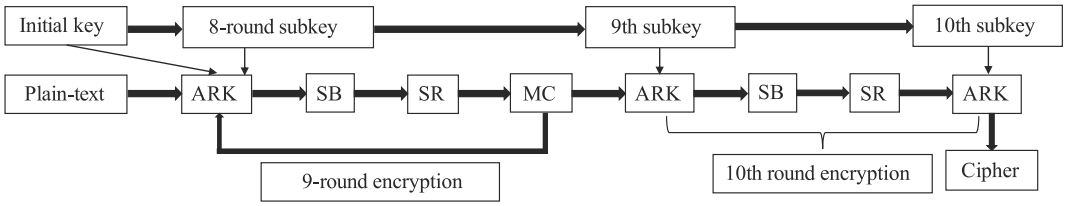

Fig. 1 AES-128: ARK represents AddRoundKey, SB represents SubBytes, SR represents ShiftRows, and MC represents MixColumns.

quantum circuit. The SubBytes quantum circuit in [18] is modified firstly, which eliminates the effect that the output qubits in SubBytes are not in $|0\rangle^{\otimes 8}$. Then, we construct the quantum circuit to encode the new round sub-key on the qubits encoding previous round sub-key. This can reduce the cost of qubits by 224 compared to Langenberg *et al.*'s work [18]. Employing the lowest possible number of qubits, we optimize the quantum circuit of S-AES using the methods used in AES-128. Compared with Almazrooie *et al.*'s work [15], it saves 71% of Toffoli gates, over 66% C-NOT gates and one third(24) for qubits.

This paper is organized as follows: the first part briefly reviews the structure of AES algorithm and the quantum circuit. Then an improved key expansion quantum circuit is proposed, and the SubBytes quantum circuit from Ref. [18] is modified. The quantum circuit of S-AES is optimized using the same methods as in AES. Next, we analyze the complexity of AES-128 and S-AES quantum circuits and compared with the results in Ref. [18] and Ref. [15] respectively. Finally, discussion and conclusion are presented.

2 Introduction to AES-128

As a symmetric block cipher, AES-128 uses the same 128-bit key each time to encrypt 128-bit plain-text. Both the key and the plain-text are arranged into a rectangular array of 4×4 bytes. The entire encryption process contains 4 encryption operations, namely AddRoundKey, SubBytes, ShiftRows and MixColumns. In the first nine rounds of encryption, the above 4 encryption operations are executed in order, and the tenth round of encryption contains two AddRoundKey operations but without MixColumns operation. Therefore, in addition to the 128-bit initial key, 10 rounds sub-key are needed to be generated for AddRoundKey. The specific flow chart is shown in Fig. 1. In the key expansion operation, transposition and SubBytes are mainly used. Below we briefly introduce the 4 kinds of encryption operations.

AddRoundKey implements the bit-wise XOR of the round keys, and it needs 128 C-NOT gates each time and can be executed in parallel.

ShiftRows is a particular permutation of the current AES state. For the first row in the plain-text array of 4×4 bytes, we do nothing; for the second row, rotate one unit (a byte) to the left; for the third row, rotate two units to the left; for the last row, rotate three units to the left. All of the above operations are equivalent to adjust the position of the bytes, so no additional operations are required.

MixColumns operates on an entire column of the plain-text array at a time as an affine transformation, so MixColumns is equivalent to a 32×32 matrix acting on every column. A MixColumns requires 277 C-NOT gates.

SubBytes is the most important and complex part of the entire algorithm which prevents linear correspondence between the plain-text and the cipher-text. The main function of SubBytes is to map one byte to another through the S-box. SubBytes can be achieved by performing the inverse operation for each byte in the 4×4 bytes array on the finite field, and then performing an affine transformation for every byte.

Next, we briefly introduce the key expansion algorithm. The 16-byte initial key is arranged into a 4×4 array, which is divided into k_0, k_1, k_2, and k_3 according to the row, and each part has 4 bytes. First, for the 4 bytes of k_3, we loop them to the left by one unit (a byte) and do SubBytes operation for each of the byte, followed by performing XOR operations with a known 32-bit string. Finally, XOR operations between k_0 and the above results are performed to get k_4. k_5 can be obtained by making XOR operations between k_1 and k_4. Then, k_6 and k_7 can be gained. A round key expansion is finished. The last 9-round sub-key can be gained by repeating the above process. The 32 known bits that perform XOR operations with the 4-byte key are different in each round, a total of 16 Pauli-X gates are required for 10 rounds. In Langenberg *et al.*'s work, the key expansion quantum circuit needs 224 qubits to generate the 10-round sub-key.

In the last part of this section, the quantum circuit of AES-128 is shown in Fig. 2. Round n ($n = 1, 2, ..., 10$) represents the encryption round and Inverse n represents the inverse operation for the corresponding encryption round. The plain-text is encoded on the qubits that encodes the initial key, which can save 128 qubits used to encode plain-text. We denote the XOR operations between the plain-text and the initial key as P.

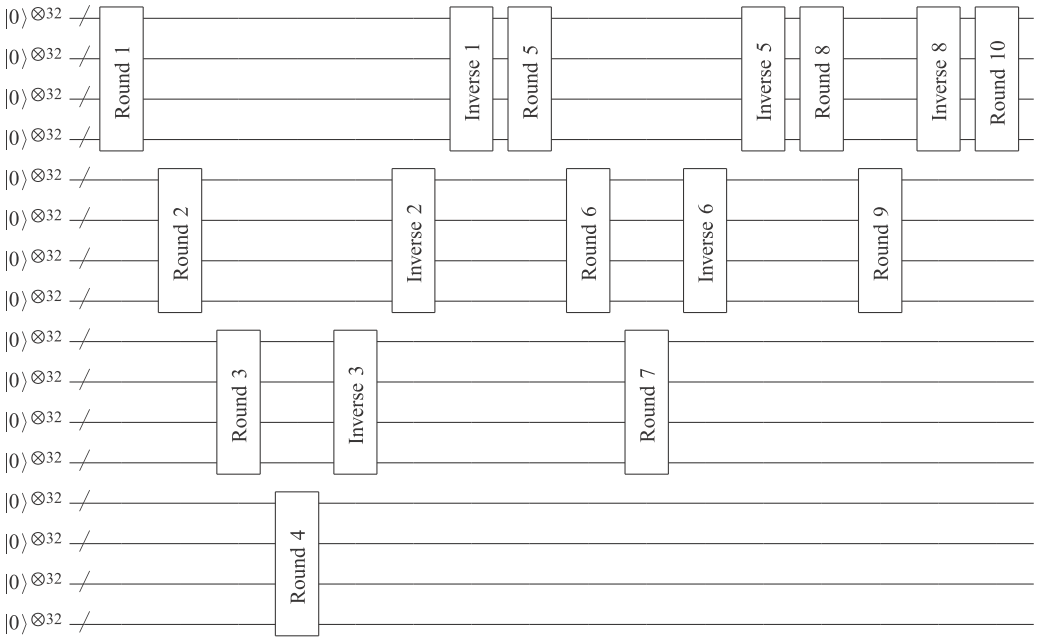

Fig. 2 Quantum circuit of AES-128: a line represents 32 qubits.

3 Improvements in quantum implementation of AES-128

We propose an improved design for key expansion, which does not need the extra qubits for saving the sub-key. It is shown in Fig. 3.

The new design saves 224 qubits compared with Langenberg *et al.*'s work because we generate the sub-key in the qubits that encode the previous key, instead of using another 224 qubits in $|0\rangle$ state in the first 7 rounds totally. The quantum circuit for SubBytes in Ref. [18], requiring

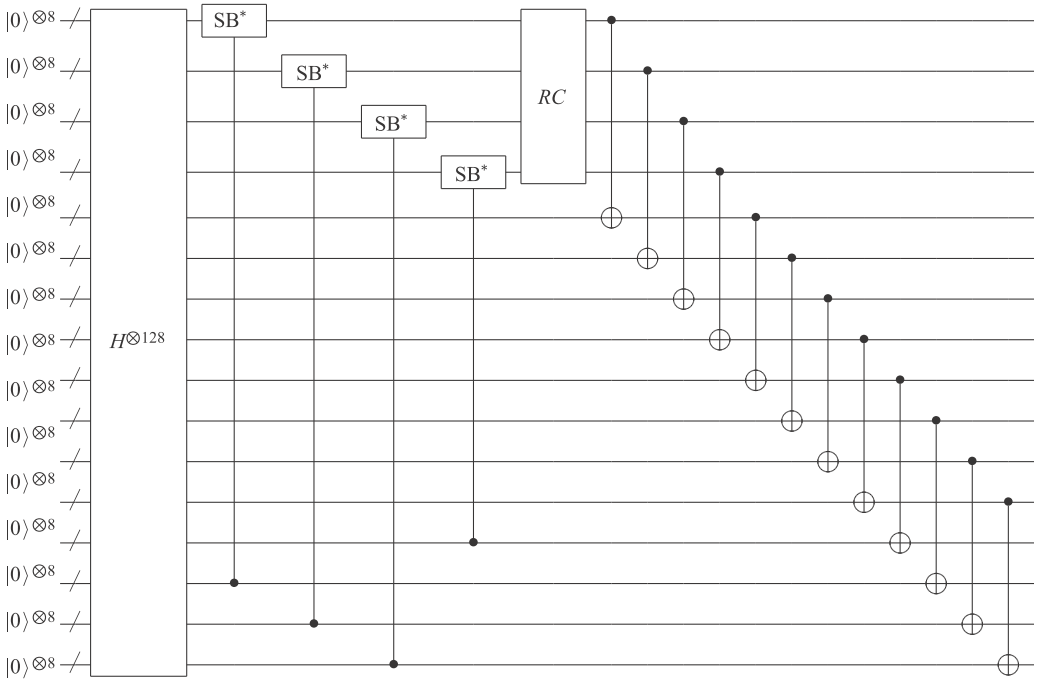

Fig. 3 Key expansion: SB* represents the modified SubBytes, RC represents making XOR operations with the known 32-bits string.

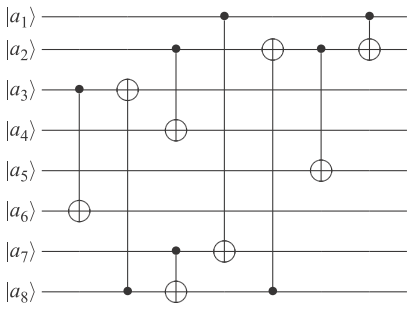

Fig. 4 The affine transformation on the output qubits.

16 auxiliary qubits, 55 Toffoli gates, 314 C-NOT gates and 4 NOT gates, can be used in the 10-round encryption but cannot be applied in the key expansion since the initial state of the output qubits in SubBytes are not $|0\rangle^{\otimes 8}$. In order to use this quantum circuit of SubBytes, an affine transformation shown in Fig. 4 operates on the output qubits which eliminates the effect that the output qubits are not in $|0\rangle^{\otimes 8}$ state, and the final result is the same as the output of SubBytes and the initial state of the output qubits to do XOR operations. It is denoted as SB*. Suppose the input is X and the initial state of output qubits is Y in this step, this process can be written as a function:

$$\mathrm{SB}^*(X) = Y \oplus \mathrm{SB}(X). \tag{1}$$

This affine transformation contains 8 C-NOT gates but does not increase the depth of the entire quantum circuit of SubBytes. So the overall cost of SubBytes in key expansion is 16 auxiliary qubits, 55 Toffoli gates, 322 C-NOT gates and 4 NOT gates.

4 An improved quantum implementation of S-AES

Here, the same method is used in S-AES. It saves 16 qubits without increasing the gate complexity. Figure 5 shows the quantum circuit of S-AES.

For S-AES, we design a SubBytes quantum circuit by using the Boolean functions corresponding to the inverse in finite field $GF(2^4)$. The inverse representation in $GF(2^4)$ based on the normal basis [19] is as follows:

$$\begin{aligned} y_1 &= x_2x_3x_4 + x_1x_3 + x_2x_3 + x_3 + x_4, \\ y_2 &= x_1x_3x_4 + x_1x_3 + x_2x_3 + x_2x_4 + x_4, \\ y_3 &= x_1x_2x_4 + x_1x_3 + x_1x_4 + x_1 + x_2, \\ y_4 &= x_1x_2x_3 + x_1x_3 + x_1x_4 + x_2x_4 + x_2, \end{aligned} \tag{2}$$

where x_i represents the input qubit and y_i represents the output qubit ($i = 1, 2, 3, 4$). The derivation of the equation is based on the representation in the finite field, in which an element of $GF(2^4)$ can be written as $\Delta = (x_1W + x_2W^2)Z^2 + (x_3W + x_4W^2)Z^8$, where W is a root of $x^2 + x + 1$ over $GF(2)$ and Z is a root of $x^2 + x + W^2$ over $GF(2^2)$. The inverse of this element can be represented as $\Delta' = (y_1W + y_2W^2)Z^2 + (y_3W + y_4W^2)Z^8$.

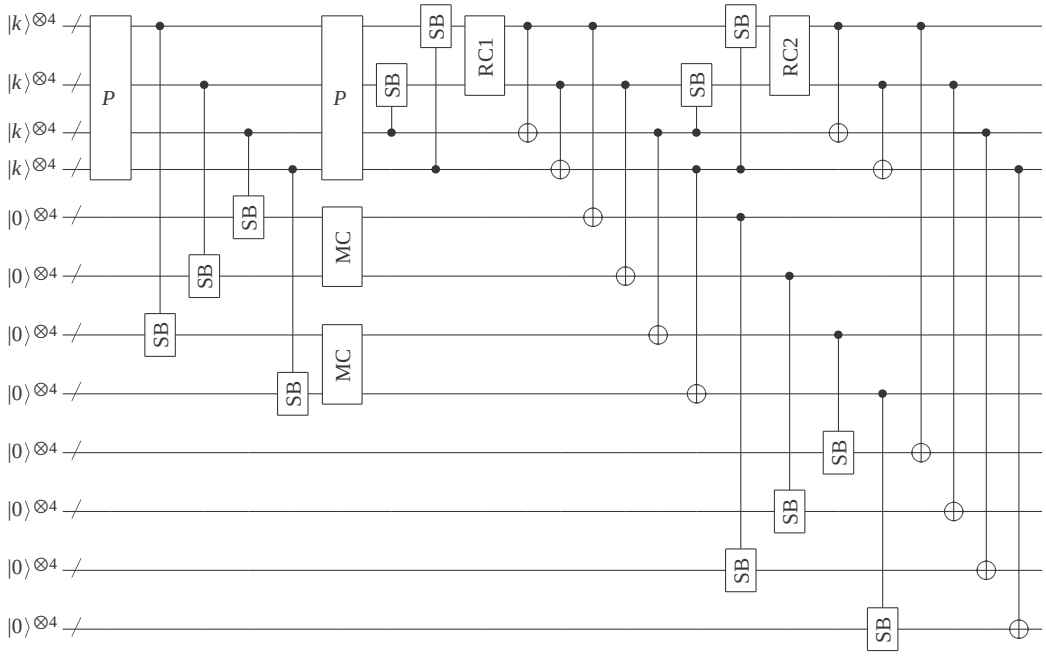

Fig. 5 A more economic quantum circuit for S-AES: P represents the XOR operations between the key and the plain-text; SB represents the SubBytes; MC represents the MixColumns; RC represents XOR operations with the known 8-bit string; the last 16-qubit represents the cipher.

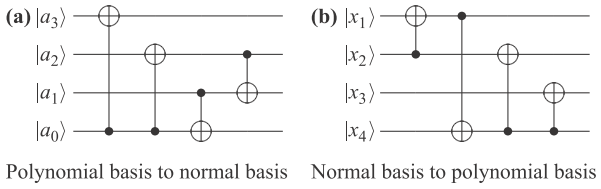

Fig. 6 Basis transformation.

There are 4 mapping matrices between this normal basis and polynomial basis, the simplest of which is as follow:

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \tag{3}$$

This matrix transforms the polynomial basis to the normal basis, and the inverse matrix transforms the normal basis to the polynomial basis. Figure 6 is the corresponding quantum circuit. It requires 4 C-NOT gates to implement the basis transformation.

We design a quantum circuit in Fig. 7 based on the Eq. (2), which does not require auxiliary qubits or the initial state of the output qubits in $|0\rangle^{\otimes 4}$. If the initial state of the output qubits are not in $|0\rangle^{\otimes 4}$, the results are equal to performing XOR operations with the initial state of the output qubits. This process requires 14 Toffoli gates and 11 C-NOT gates.

In addition, the affine transformation and the mapping matrix can be merged. The quantum circuit is shown in Fig. 8, which only requires 8 C-NOT gates and 2 NOT gates. So the SubBytes in S-AES requires 14 Toffoli gates, 23 C-NOT gates and 2 NOT gates.

5 Complexity analysis

In AES-128, there are 256 SubBytes in 10-round encryption and 72 SubBytes in key expansion, where the first, second, third and sixth rounds sub-key need to be generated twice and elimination once. So performing the XOR operations with the known 32-bit string in key expansion requires 24 C-NOT gates. In addition, the operation P, its gate complexity at most 128 X gates, is performed four times. The number of times each operation needs to be performed and the gates required per operation are shown in Table 1. In S-AES, there are 12 SubBytes. The number of times each operation needs to be performed and the gates required per operation are shown in Table 2.

Table 1 AES-128: IK represents the initialization of key; P represents the XOR operations between the plain-text and the initial key; ARK represents the AddRoundKey; SB represents the SubBytes in the round encryption; SB* represents the modified SubBytes in the key expansion; MC represents the MixColumns; RC represents the XOR operations with the known 32-bits string which needs 24 single qubit gates in the entire process (the gate used is different in each round); Others represent the remaining XOR operation in each key expansion.

	Toffoli	C-NOT	X gate	Times
IK			128	1
P			≤128	4
ARK		128		16
SB	55	314	4	256
SB*	55	322	4	72
MC		277		60
RC			24	
Others		96		18

Table 2 S-AES: SB represents the SubBytes; RC needs 3 single qubit gates in the entire process; Others represent the remaining XOR operation in each key expansion, and the other abbreviations are same as Table 1.

	Toffoli	C-NOT	X gate	Times
IK			16	1
P			≤16	2
ARK		16		2
SB	14	23	2	12
MC		20		2
RC			3	
Others		8		2

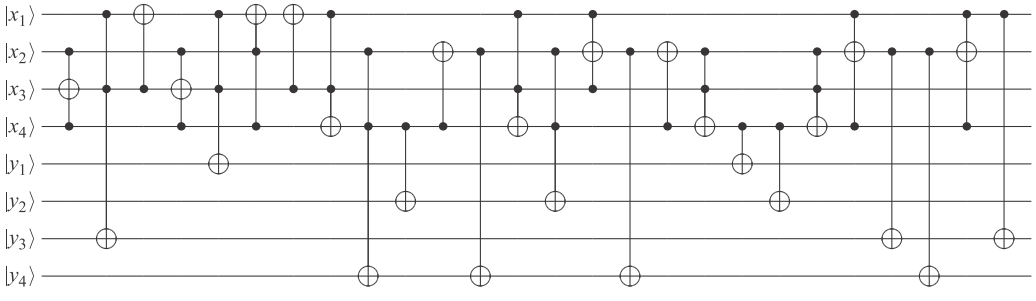

Fig. 7 The quantum circuit for finding the inverse in $GF(2^4)$.

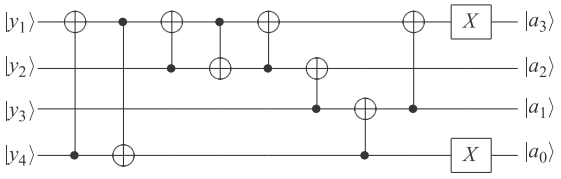

Fig. 8 The merged quantum circuit between the affine transformation and the mapping matrix.

The cost of quantum resources can be calculated by the data in Table 1 and Table 2. Table 3 compares the resource usage between Langenberg *et al.* 's work and this work in quantum implementation of AES-128. Table 4 compares the resource usage between Almazrooie *et al.*'s work and this work in quantum implementation of S-AES.

A comparison in Table 3 shows that the gate complexity between Langenberg *et al.*'work and our work is basically the same, but 224 qubits are saved. Table 4 shows that the quantum circuits we proposed for S-AES can save 71% Toffoli gates, over 66% C-NOT gates, and one third of the qubits usage compared to Almazrooie *et al.*'s work.

Table 3 AES-128: The cost comparison between Langenberg *et al.*'s work and this paper.

	Toffoli	C-NOT	X gate	Qubits
Langenberg *et al.*	16940	107960	1507	880
This work	18040	101174	1976	656

Table 4 S-AES: The cost comparison between Almazrooie *et al.*'s work and this paper.

	Toffoli	C-NOT	X gate	Qubits
Almazrooie *et al.*	576	1080	26	72
This work	168	364	75	48

6 Discussion and conclusion

In this work, we proposed an improved key expansion algorithm and modified the SubBytes circuit in Langenberg *et al.*'s work. The design for the key expansion algorithm was structured to utilize the lowest possible number of qubits, which cuts the number of qubits by 224 for AES-128. All of which advance the quantum implementation of AES. We applied the same methods to S-AES and got a similar result. The reduction of the number of qubits makes S-AES algorithm possible to be implemented on quantum computers in the near future, while the cost of 48 qubits also allows classical computers to do the full amplitude quantum simulation.

At present, the quantum attack on AES algorithm is mainly based on Grover's algorithm, while the quantum implementation of AES algorithm, as an Oracle, is the most important part in Grover attack. We have constructed a simpler Oracle that will accelerate the process of quantum Grover attack.

After our work was done, we noted that a similar paper was published [20]. There are also two main improvements compared with Langenberg *et al.*'s work. One improvement is in sub-key expansion. They solved the problem that the initial state of the output qubits in SubBytes is not in $|0\rangle^{\otimes 8}$ state by introducing an auxiliary qubit, while ours is making an affine transformation. The other improvement is proposing a new zig-zag method which can reduce 256 qubits in the encryption process compared with Grassl *et al.*'s work. If the new zig-zag method was used in our work, the quantum resource usage can be further reduced. Table 5 shows the number of times each operation needs to be performed and the gates required per operation using the new zig-zag method. The resource usage in using new zig-zag method is shown in Table 6.

Table 5 AES-128 with new zig-zag method: IK represents the initialization of key; P represents the XOR operations between the plain-text and the initial key; ARK represents the AddRoundKey; SB(r) represents the SubBytes in the round encryption; SB(k) represents the SubBytes in the key expansion; SB^{-1} represents the inverse process of SubBytes in Ref. [20]. MC represents the MixColumns; RC represents the XOR operations with the known 32-bits string which needs 16 single qubit gates in the entire process (the gate complexity is different in each round); Others represent the remaining XOR operation in each key expansion.

	Toffoli	C-NOT	X gate	Times
IK			128	1
P			$\leq$128	4
ARK		128		10
SB(r)	55	314	4	160
SB(k)	55	322	4	40
SB^{-1}	63	341	24	128
MC		277		36
RC			16	
Others		96		10

Table 6 shows that our work is slightly prior to Zou *et al.*'s work in gate complexity and reduces the qubits used by 112.

Table 6 AES-128: the quantum resource usage using new zig-zag method.

	Toffili	C-NOT	X gate	Qubits	Auxiliary qubits
Zou *et al.*	19788	128517	4528	512	0
This work	19064	118980	4528	384	16

Acknowledgements We gratefully acknowledges support from the National Natural Science Foundation of China under Grant Nos. 11974205 and 11774197, the National Key Research and

Development Program of China (No. 2017YFA0303700), the Key Research and Development Program of Guangdong province (No. 2018B030325002), and Beijing Advanced Innovation Center for Future Chip (ICFC). S.W. also acknowledges the China Postdoctoral Science Foundation (No. 2020M670172) and the National Natural Science Foundation of China under Grant No. 12005015.

References

1. M. Bellare and P. Rogaway, Introduction to modern cryptography, *Ucsd Cse* 207, 207 (2005)
2. R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public key cryptosystems, *Comm. ACM* 21(2), 120 (1978)
3. P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.* 26(5), 1484 (1997)
4. D. Joan and R. Vincent, The design of rijndael: AES — The advanced encryption standard, *Inf. Secur. Cryptogr* (2002)
5. L. K. Grover, A fast quantum mechanical algorithm for database search, in: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, 1996, pp 212–219
6. G. L. Long, Grover algorithm with zero theoretical failure rate, *Phys. Rev. A* 64(2), 022307 (2001)
7. A. Yamamura and H. Ishizuka, Quantum cryptanalysis of block ciphers (Algebraic Systems, Formal Languages and Computations), *RIMS Kokyuroku* 1166, 235 (2000)
8. M. Kaplan, Quantum attacks against iterated block ciphers, arXiv: 1410.1434 (2014)
9. R. J. Li and C. H. Jin, Meet-in-the-middle attacks on 10-round AES-256, *Des. Codes Cryptogr.* 80(3), 459 (2016)
10. A. Ambainis, Quantum walk algorithm for element distinctness, *SIAM J. Comput.* 37(1), 210 (2007)
11. M. Roetteler and R. Steinwandt, A note on quantum related-key attacks, *Inf. Process. Lett.* 115(1), 40 (2015)
12. D. R. Simon, On the power of quantum computation, in: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, 1994, pp 116–123
13. M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, Applying Grover's algorithm to AES: Quantum resource estimates, in: Post-Quantum Cryptography, Springer, 2016, pp 29–43
14. P. Kim, D. Han, and K. C. Jeong, Time–space complexity of quantum search algorithms in symmetric cryptanalysis: Applying to AES and SHA-2, *Quantum Inform. Process.* 17(12), 339 (2018)
15. M. Almazrooie, R. Abdullah, A. Samsudin, and K. N. Mutter, Quantum Grover attack on the simplified-AES, in: Proceedings of the 7th International Conference on Software and Computer Applications, 2018, pp 204–211
16. F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, et al., Quantum supremacy using a programmable superconducting processor, *Nature* 574(7779), 505 (2019)
17. J. Xu, S. Li, T. Chen, and Z. Y. Xue, Nonadiabatic geometric quantum computation with optimal control on superconducting circuits, *Front. Phys.* 15(4), 41503 (2020)
18. B. Langenberg, H. Pham, and R. Steinwandt, Reducing the cost of implementing the advanced encryption standard as a quantum circuit, *IEEE Trans. Quantum Eng.* 1, 1 (2020)
19. J. Boyar and R. Peralta, A new combinational logic minimization technique with applications to cryptology, in: International Symposium on Experimental Algorithms, Springer, 2010, pp 178–189
20. J. Zou, Z. H. Wei, S. W. Sun, X. M. Liu, and W. L. Wu, Quantum circuit implementations of AES with fewer qubits, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2020, pp 697–726