Quantum Inf Process (2018) 17:156
https://doi.org/10.1007/s11128-018-1923-9

Constructing quantum Hash functions based on quantum walks on Johnson graphs

Wei-Feng Cao[1] · Yong-Ce Zhang[2] · Yu-Guang Yang[2] · Dan Li[3] · Yi-Hua Zhou[2] · Wei-Min Shi[2]

Received: 11 October 2017 / Accepted: 7 May 2018 / Published online: 15 May 2018

Abstract We present a quantum hash function in a quantum walk framework on Johnson graphs. In this quantum hash function, the message bit decides which coin operator, i.e., Grover operator or DFT operator, is applied on the coin at each step. Then a fixed conditional shift operator is applied to decide the movement of the walker. Compared with existing quantum-walk-based hash functions, the present hash function has a lower collision rate and quantum resource cost. It provides a clue for the construction of other cryptography protocols by introducing the quantum walk model into hash functions.

Keywords Quantum cryptography · Quantum walk · Hash function · Collision · Birthday attack

1 Introduction

Hash functions are aimed to map a message with arbitrary length to a fixed-size message digest or hash value [1]. As a quantum counterpart of conventional hash functions, quantum hash functions (QHFs) map a classical message into a quantum Hilbert space [2–6]. Such Hilbert space should be so small that eavesdroppers cannot retrieve so much information about classical message, which is guaranteed by Holevo

✉ Wei-Feng Cao
weifeng_cao@163.com

1 College of Electric and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

2 Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

3 College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

limit. Moreover, images of different messages should be as far away as possible in order to avoid collision. Buhrman et al. [2] first implicitly introduced QHFs as quantum fingerprinting (QF) based on binary error-correcting codes (BECC). Subsequently, Gavinsky and Ito [3] found that QF can be used as cryptographic primitive. However, binary QHFs are not very suitable for the situation where group operations are involved. Ablayev and Ablayev constructed a kind of non-binary QHF [4] and a balanced QHF [5]. Ziatdinov [6] also constructed two new QHFs: the one is based on expander graphs and the other based on extractor functions. Although existing QHFs have been proposed [2–6], the verification whether the hash values of two classical messages are equal depends on quantum SWAP test. However, the quantum SWAP test has a one-sided error probability up to 0.75. To reduce the one-sided error probability to arbitrary small $\varepsilon > 0$,$O(\log n \log(1/\varepsilon))$ qubits are required for encoding an n-bit message.

Quantum walks (QWs) [7] have received much attention for their possible applications such as element distinctness [8], triangle finding [9], graph isomorphism [10]. Li et al. presented a kind of QHF based on two-particle interacting QWs [11]. In their scheme, two 2×2 coin operators, i.e., I interaction and π-phase interaction, are used to represent the message bits '0' and '1.' Subsequently, Li et al. [12] and Yang et al. [13] proposed two kinds of QHFs by introducing the 4×4 coin operators [14–16], respectively. To reduce the quantum resource cost, Li et al. [17] presented a controlled alternate quantum walks (CAQW)-based QHF. To further improve the efficiency of the QHF, Yang et al. proposed an efficient QHF by dense coding of coin operators in discrete-time QW on cycles [18]. These observations offer a new perspective to construct QHFs in a QW framework. This QW framework plays an important role for the construction of QHFs in this paper. However, this does not imply anything about the specific QW models on specific graphs (graph, conditional shift operator and coin flip operator, coin state and position state) which need to be designed properly.

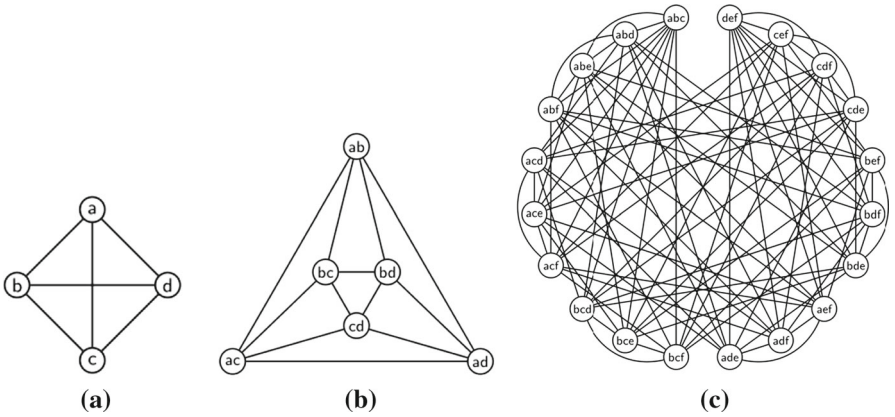
Recently, Johnson graphs $J(n,k)$ have made many hot topics in the computer science and quantum computation fields [19, 20] due to the quasi-polynomial algorithm for graph isomorphism [21]. In this paper, we investigate the applications of the Johnson graph $J(n,k)$ to the construction of QHFs and construct a new QHF based on discrete-time QW on Johnson graphs. Compared with existing QW-based hash functions, the proposed QHF has a lower collision rate and quantum resource cost. It provides a clue for the construction of other cryptography protocols by introducing the QW model into hash functions.

The rest of this paper is organized as follows. In Sect. 2, preliminaries are provided about the knowledge of discrete-time QW and the Johnson graph $J(n, k)$. In Sect. 3, a new QHF is presented, and Sect. 4 provides a performance comparison between our proposal and other QW-based QHFs. Finally, a conclusion is given in Sect. 5.

2 Preliminaries

Definition 1 Basic discrete-time quantum walks on a line.

The time evolution of the basic discrete-time QW on a line is controlled by the evolution operator $U = S\,(I \otimes C)$, where S is the conditional shift operator which affects both the coin space $\mathcal{H}_c$ and the position space $\mathcal{H}_p$. Here $\mathcal{H}_c$ and $\mathcal{H}_p$ are spanned by the



(a) **(b)** **(c)**

Fig. 1 **a** The Johnson graph $J(4, 1)$. **b** The Johnson graph $J(4, 2)$. **c** The Johnson graph $J(6, 3)$

coin orthogonal basis $\{|0\rangle, |1\rangle\}$ and the position states $\{|x\rangle, x \in \mathbb{Z}\}$, respectively. C is the coin operator which only affects the coin space $\mathcal{H}_c$, and I is the identity operator applied on the position space $\mathcal{H}_p$.

For one-dimensional discrete-time QW on a line, the most common coin operator can be written as

$$C = H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{1}$$

which is the unbiased Hadamard operator. The conditional shift operator S is given by

$$\hat{S} = \sum_x (|x+1, 0\rangle\langle x, 0| + |x-1, 1\rangle\langle x, 1|), \tag{2}$$

where the sum is taken over all discrete positions in the position space $\mathcal{H}_p$. x stands for the position. The conditional shift operator S decides the direction in which the walker moves.

Definition 2 The Johnson graph $J(n, k)$.

The Johnson graph $J(n, k)$ is defined by n symbols, where vertices are k-element subsets of the symbols, and vertices are adjacent if they differ in exactly one symbol, as shown in Fig. 1. In particular, $J(n, 1)$ is the complete graph K_n, $J(n, 2)$ is the strongly regular triangular graph T_n, and $J(n, 3)$ is the n-tetrahedral graph. They all support fast spatial search by continuous-time QW.

3 Quantum Hash function on the Johnson graph $J(n, 1)$

The efficient QHF can be constructed by subtly modifying discrete-time QW on the Johnson graph $J(n, k)$. For simplicity and without loss of generality, we assume $k=1$ and consider how to construct the QHF based on discrete-time QWs on the complete

graph $J(n, 1)$. Concretely, in a general discrete-time QW, the coin operator is invariable. Suppose the coin operator at each iteration depends on a binary string, i.e., the *message*, and accordingly a QHF is constructed, similar to that in Refs. [11–13, 17, 18]. The ith bit of the *message* controls the choice of the coin operators at the ith iteration. Here we introduce two coin operators, i.e., Grover operator and DFT operator which are controlled by the *message* bit '0' and '1,' respectively. They are shown as follows.

$$C_0 = G = 2|\phi\rangle\langle\phi| - I = \begin{bmatrix} \frac{2}{d}-1 & \frac{2}{d} & \cdots & \frac{2}{d} \\ \frac{2}{d} & \frac{2}{d}-1 & \cdots & \frac{2}{d} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{2}{d} & \frac{2}{d} & \cdots & \frac{2}{d}-1 \end{bmatrix}, \tag{3}$$

$$C_1 = \mathrm{DFT} = \frac{1}{\sqrt{d}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{d-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(d-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{d-1} & \omega^{2(d-1)} & \cdots & \omega^{(d-1)^2} \end{bmatrix}, \tag{4}$$

where $|\phi\rangle = \frac{1}{\sqrt{d}} \sum_{i=0}^{d-1} |i\rangle$, and $\omega = e^{\frac{2\pi i}{d}}$. Here, $d{=}n-1$.

The conditional shift operator S_J is given by

$$S_J = \sum_{x=0}^{n-1} \sum_{i=1}^{n-1} |x + i \bmod n, i\rangle\langle x, i|, \tag{5}$$

Assume the *message* is '0100111.' The final state is given by

$$|\varphi_{final}\rangle = U(message)|\varphi_0\rangle = U_1 U_1 U_1 U_0 U_0 U_1 U_0 |\varphi_0\rangle, \tag{6}$$

and the final probability of locating the walker at position x is

$$P(x, message) = \sum_{i \in \{0,1\}} |\langle x, i|U(message)|\varphi_0\rangle|^2, \tag{7}$$

where $U_0 = S_J\ (I \otimes C_0)$,$U_1 = S_J\ (I \otimes C_1)$.$|\varphi_0\rangle$ is the initial state of the total quantum system, i.e., $|\varphi_0\rangle{=}\ |x{=}\ 0\rangle_{\mathrm{p}} \otimes \frac{1}{\sqrt{n-1}} \sum_{i=1}^{n-1} |i\rangle_{\mathrm{c}}$. The subscripts p and c represent the position and the coin, respectively.

Assume the message is $M = (m_1, m_2, m_3, m_4, \cdots, m_q)$.

The process of the QHF is described as follows:

(1) Select the parameters n.
(2) Run the discrete-time QW on the Johnson graph $J(n, 1)$ under the control of the *message* M (see Fig. 2 for the circuit representation of the evolution operation at the ith iteration).

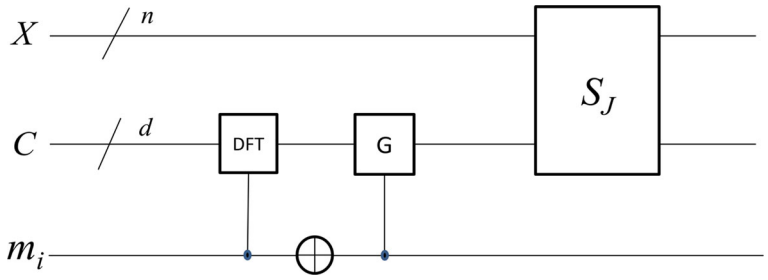


Fig. 2 Circuit representation of the evolution operation at the ith iteration in the QHF. The top n lines are the quantum register for storing the information about the position. The lowest line is the one-bit message m_i, and the other line is the quantum register for storing the information about the coin

(3) Multiply all elements in the resulting probability distribution by 10^4 times to generate a binary string as the hash value. The length of the obtained hash value is $L=n\times 13$, where 13 bits are the biggest size of the binary string generated by each element in the resulting probability distribution.

3.1 Performance analyses

It's difficult to perform strict mathematical proof of the security of hash functions. So, in this section, we performed several hash tests and theoretical analysis to evaluate the performance of the proposed QHF. We choose $n=15$ so that the hash value we consider here is $15\times 13=195$ bits. The following results show that the proposed QHF has excellent statistical performance.

3.1.1 Sensitivity of hash value to message

We constructed several different messages by subtly modifying the original message. Then, we calculated and compared the hash values of all resulting messages under the four conditions.

Condition 1: The original message;

Condition 2: Change a bit of the original message from '0' to '1' at the second position;

Condition 3: Change a bit of the original message from '0' to '1' at the third position;

Condition 4: Change a bit of the original message from '0' to '1' at the fourth position.

The corresponding 195-bit hash values in the hexadecimal format are given as follows:

Condition 1: 'A82' '1DE' '1DE' '11F' '1DE' '48D' '11F' '1DE' '1DE' '11F' '48D' '1DE' '11F' '1DE' '1DE'

Condition 2: '2C7' '1A9' '2D6' '223' '1DE' '241' '2E0' '2D8' '342' '255' '313' '358' '263' '2FD' '266'

Condition 3: '355' '235' '357' '2BA' '289' '2A7' '288' '281' '19C' '3B4' '18B' '3FB' '16F' '2BA' '235'

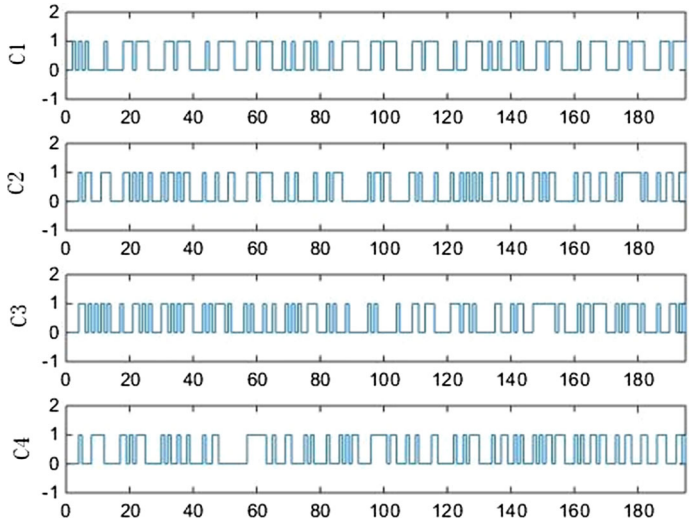


Fig. 3 Hash values of C1, C2, C3, C4

Condition 4: '23C' '35C' '292' '260' '1F9' '30A' '22B' '1F6' '486' '12E' '265' '2B7' '29B' '263' '38D'

Figure 3 exhibits the plots of the corresponding hash values, respectively. It is clearly shown that any tiny modification to the original message will cause a huge change in the new hash value.

3.1.2 Statistical analysis of diffusion and confusion

The following definitions are given:

Mean normally changed bit number: $\bar{B} = \sum_{i=1}^{N} B_i/N$;

Mean added bit number of zero: $\bar{A} = \sum_{i=1}^{N} A_i/N$;

Mean sum of the changed bit number: $\bar{T} = \bar{A} + \bar{B}$;

Mean changed probability $P = (\bar{T}/(n \times 13)) \times 100\%$;

Standard variance of the total changed bit number $\Delta T = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(T_i - \bar{T})^2}$;

Standard variance of the changed probability $\Delta P = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(T_i/(n \times 13) - P)^2} \times 100\%$.

The diffusion and confusion tests are performed as follows:

(1) Select a message randomly and calculate the corresponding hash value.
(2) Change a bit of the original message randomly and calculate the corresponding hash value.
(3) Compare the two hash values and count the changed bits called B_i.
(4) Add zero to make the size of all n binary strings corresponding to the new probability distribution be 13 bits and count the added zeros called A_i.
(5) Repeat steps (1) to (4) N times. Here N is the number of tests.

Table 1 Results for diffusion and confusion tests in the first QHF

	$N=4095$	$N=8191$	$N=16{,}383$	Mean
$\bar{B}$	84.95	70.71	77.30	77.65
$\bar{A}$	47.45	47.43	47.43	47.44
$\bar{T}$	132.4	118.14	124.7	125.08
P (%)	67.9	60.59	63.96	64.15
ΔT	6.02	7.29	6.43	6.58
ΔP	13.8	14.71	6.3	11.6

The diffusion and confusion tests are performed with $N=4095$, 8191, 16,383, respectively, as shown in Table 1. We concluded from the tests that the mean total changed bit number $\bar{T}$ is close to 125 or so, and the mean changed probability P close to 64%. ΔT and ΔP are very small. The excellent statistical feature ensures that it is infeasible to forge valid plaintext–cipher pairs given known plaintext–cipher pairs.

3.1.3 Collision analysis

Collision means different messages are mapped to the same hash value. It is difficult to prove the capability of collision resistance of hash functions by means of mathematical theory. Thus, we performed the following tests:

(1) Select a message randomly and generate the corresponding hash value in ASCII format.
(2) Change a bit of the message randomly and generate the corresponding hash value in ASCII format.
(3) Compare these two hash values and count the number of ASCII characters with the same value at the same location.
(4) Repeat steps (1) to (3) N times.

Moreover, the number of ASCII characters with the same value at the same location, i.e., ω, is given by

$$\omega = \sum_{i=1}^{N} \delta\left(t\left(e_i\right) - t\left(e_i'\right)\right), \tag{8}$$

where $\delta(x) = 1$ for $x=0$ and 0 for $x \neq 0$. e_i and e_i' represent the ith elements of the original and new hash value in ASCII format, respectively. $t(e_i)$ and $t\left(e_i'\right)$ represent e_i and e_i' in the decimal form, respectively. We performed the tests $N=16{,}383$ times and concluded that $\omega=0$ occurs for 16,063 times,$\omega=1$ for 314 times, $\omega=2$ for 6 times and the others for zero times. Therefore, the collision rate is about 1.95%.

3.1.4 Uniform distribution analysis

If the hash value is very uniform, then the hash value does not contain any statistical information of the original message after the diffusion and confusion processes. To analyze the uniform distribution property of the hash value, similar to Sects. 3.1.2 and 3.1.3, we did the following tests.

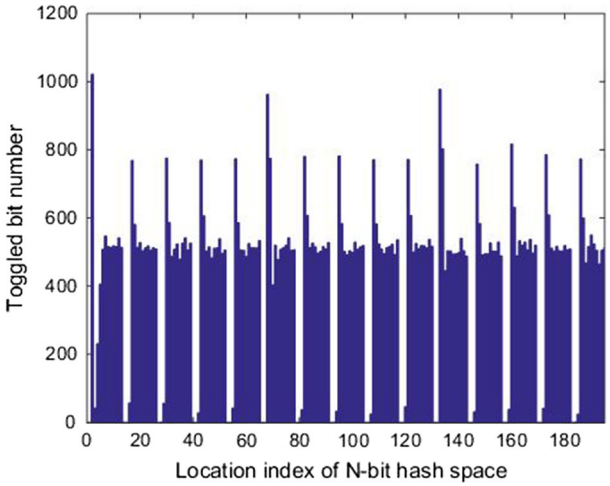


Fig. 4 Histogram of the 195-bit hash space, where $N = 16{,}383$

(1) Select a message randomly and generate the corresponding hash value in ASCII format.
(2) Change a bit of the message randomly and generate the corresponding hash value in ASCII format.
(3) Compare these two hash values and count the number of ASCII characters with the same value at the same location.
(4) Repeat steps (1) to (3) N times.

We performed the tests $N = 16{,}383$ times. The mean number of changed bits is 6495. From Fig. 4 we can see that our hash values have a very uniform dispersion state, which proves that the proposed QHF has a good ability to resist statistical attacks.

3.1.5 Resistance to birthday attack

Birthday attack implies a lower bound of the length of a secure hash value. The length of the hash value of the proposed QHF is $n \times 13$ bits. (n is the node number of the cycle graph.) Therefore, it needs $O\left(2^{(n\times 13)/2}\right)$ trials to find two messages with identical hash values with a probability of 1/2. Generally speaking, if the computational complexity of the birthday attack is $O\left(2^{128}\right)$, the hash function can be considered to be cryptographically secure. To be considered cryptographically secure, the length of the hash value must be not less than 256 bits. Although we chose $n = 15$ and obtained the hash value with $n \times 13 = 15 \times 13 = 195$ bits in the performance analyses, the parameter n is variable and can be adjusted according to the security requirement of hash functions. In fact, the proposed QHF can be easily extended to the case with a larger node number. That is, to be considered cryptographically secure, n can be set not less than 20 in the proposed QHF. Therefore, the proposed QHF can be resistant against the birthday attack.

Table 2 Comparison in terms of collision resistance

	Ref. [11]	Ref. [13]	Ref. [17]	Ref. [18]	Our scheme
Collision rate (%)	23.12	6.33	9.32	Average 1.16	1.95

Table 3 Comparison in terms of quantum resource consumption

	The number of the particles	Quantum operation
Ref. [11]	2	$S_{xy}\ (I \otimes C_2 \otimes C_2)$
Ref. [13]	2	$S_{xy}\ (I \otimes C_4)$
Ref. [17]	1	$S_y C_2 S_x C_2$
Ref. [18]	1	$S_c\ (I \otimes C_v)$
Our scheme	1	$S_J\ (I \otimes C_{d\times d})$

Here, C_2 and C_v are 2×2 coin operators, and C_4 is a 4×4 coin operator. $C_{d\times d}$ denotes $d \times d$ Grover operator and DFT operator

4 Comparison with other QW-based QHFs

Resistance against collision is an important indicator of evaluating a QHF protocol. We compared our schemes with existing QW-based QHFs in terms of collision resistance, as shown in Table 2. Compared with Refs [11, 13, 17, 18], our QHF demonstrates a better capability of collision resistance.

Resource consumption is another important indicator of evaluating a QHF protocol. We compared our schemes with existing QHF protocols in terms of quantum operation and the number of the particles, as shown in Table 3.

5 Conclusion

In this paper, we have presented a new QHF based on a QW on the Johnson graph $J(n, 1)$. At each step, the message bit decides which coin operator, i.e., Grover operator or DFT operator, to be applied on the coin state. Then a fixed conditional shift operator is applied to decide the movement of the walker on the Johnson graph $J(n,1)$. Compared with existing QW-based hash functions, the collision rate and the quantum resource consumption are lower in the proposed QHF. It provides a clue for the construction of other cryptography protocols by introducing the QW model into hash functions.

For future work, there are several open questions. First, here we only proved the security of the proposed QHF by performing several hash tests and theoretical analysis, and its security has not been studied by strict mathematical proof. Moreover, the difference of the messages has a significant effect on the test results. Therefore, it would be interesting to study a strict mathematical proof of the security of the proposed QHF. Second, the methods of the security analysis of hash functions include mainly the birthday attack [22], meeting-in-the-middle collision attack [23], fixed point attack [24], differential collision attack [25], preimage attack [26]. Studying whether these

methods can be applied for the security analysis of QW-based QHFs [11, 13, 17, 18] is also an open problem. Finally, we gave a new QHF based on a QW on the Johnson graph $J(n, 1)$. However, Johnson graphs $J(n, k)$ have different graph topologies with different k. Concretely, $J(n, 1)$ is the complete graph K_n and $J(n, 2)$ is the strongly regular triangular graph T_n, whereas $J(n, 3)$ is the n-tetrahedral graph, etc. The QHFs based on a QW on the Johnson graph $J(n, k)$ with $k \geq 2$ have not been studied. It would be interesting to construct novel QHFs based on a QW on the Johnson graphs $J(n, k)$ with $k \geq 2$.

Acknowledgements We thank Dr. Xiu-Bo Chen and Dr. Zheng Yuan for reviewing the original manuscript. This work was supported by the National Natural Science Foundation of China (Grant Nos. 61572053, 61671087, U1636106, 61602019, 61571226, 61701229, 61702367), Beijing Natural Science Foundation (Grant Nos. 4162005, 4182006), Natural Science Foundation of Jiangsu Province, China (Grant No. BK20170802), Jiangsu Postdoctoral Science Foundation.

References

1. Knuth, D.: The Art of Computer Programming, Sorting and Searching, vol. 3, 2nd edn. Addison-Wesley, Boston (1998)
2. Buhrman, H., Cleve, R., Watrous, J., de Wolf, R.: Quantum fingerprinting. Phys. Rev. Lett. **87**, 167902 (2001)
3. D. Gavinsky, T. Ito: Quantum fingerprints that keep secrets. Technical Report Cornell University Library. arXiv:1010.5342 (2010)
4. Ablayev, F., Vasiliev, A.: Cryptographic quantum hashing. Laser Phys. Lett. **11**(2), 025202 (2014)
5. Ablayev, F., Ablayev, M., Vasiliev, A.: On the balanced quantum hashing. J. Phys: Conf. Ser. **681**, 012019 (2016)
6. M. Ziatdinov: From graphs to keyed quantum hash functions. arXiv:1606.00256v1 (2016)
7. D. Aharonov, A. Ambainis, J. Kempe, et al.: Quantum walks on graphs. In: Proceedings of the 33rd ACM Symposium on Theory of Computing, pp. 50–59 (2001)
8. Ambainis, A.: Quantum walk algorithm for element distinctness. SIAM J. Comput. **37**(1), 210–239 (2007)
9. Magniez, F., Santha, M., Szegedy, M.: Quantum algorithms for the triangle problem. SIAM J. Comput. **37**(2), 413–424 (2007)
10. Tamascelli, D., Zanetti, L.: A quantum-walk-inspired adiabatic algorithm for solving graph isomorphism problems. J. Phys. A: Math. Theor. **47**(32), 325302 (2014)
11. Li, D., Zhang, J., Guo, F.-Z., Huang, W., Wen, Q.-Y., Chen, H.: Discrete-time interacting quantum walks and quantum Hash schemes. Quantum Inf. Process. **12**(3), 1501–1513 (2013)
12. Li, D., Zhang, J., Ma, X.W., Zhang, W.W., Wen, Q.Y.: Analysis of the two-particle controlled interacting quantum walks. Quantum Inf. Process. **12**(6), 2167–2176 (2013)
13. Yang, Y.-G., Xu, P., Yang, R., Zhou, Y.H., Shi, W.M.: Quantum Hash function and its application to privacy amplification in quantum key distribution, pseudo-random number generation and image encryption. Sci. Rep. **6**, 19788 (2016)
14. Xue, P., Sanders, B.C.: Two quantum walkers sharing coins. Phys. Rev. A **85**, 022307 (2012)
15. Shenvi, N., Kempe, J., Whaley, K.B.: Quantum random-walk search algorithm. Phys. Rev. A **67**, 052307 (2003)
16. Stefaňák, M., Barnett, S.M., Kollár, B., Kiss, T., Jex, I.: Directional correlations in quantum walks with two particles. New J. Phys. **13**, 033029 (2011)
17. Li, D., Yang, Y.-G., Bi, J.-L., Yuan, J.-B., Xu, J.: Controlled alternate quantum walks based quantum Hash function. Sci. Rep. **8**, 225 (2018)
18. Yang, Y.-G., Zhang, Y.-C., Xu, G., Chen, X.-B., Zhou, Y.-H., Shi, W.-M.: Improving the efficiency of quantum Hash function by dense coding of coin operators in discrete-time quantum walk. Sci. China-Phys. Mech. Astron. **61**(3), 030312 (2018)
19. S. Aaronson: G. Phi. Fo. Fum. http://scottaaronson.com/blog/?p=2521. Accessed 13 May 2018

20. J.A. Kun: Quasi-polynomial time algorithm for graph isomorphism: the details. http://jeremykun.com/2015/11/12/a-quasipolynomial-time-algorithm-for-graph-isomorphism-the-details/. Accessed 13 May 2018
21. L. Babai: Graph isomorphism in quasi-polynomial time. arXiv:1512.03547
22. M. Bellare, T. Kohno: Hash function balance and its impact on birthday attacks. In: Eurocrypt 04, LNCS, vol. 3027, pp. 401–418 (2004)
23. M.J. Saarinen: A meeting-in-the-middle collision attack against the new FORK-256. In: Indocrypt 2007, LNCS, vol. 4859, pp. 10–17 (2007)
24. Dobbertin, H.: Cryptanalysis of MD4. J. Cryptol. **11**(4), 253–271 (1998)
25. F. Chabaud, A. Joux: Differential collisions in SHA-0. In: Crypto'98, LNCS, vol. 1462, pp. 56–71 (1998)
26. Y. Sasaki, K. Aoki: Finding preimages in full MD5 faster than exhaustive search. In: Eurocrypt 2009, LNCS, vol. 5479, pp 134–152 (2009)