SCIENCE CHINA
Physics, Mechanics & Astronomy

•Article•

March 2018 Vol. 61 No. 3: 030312
https://doi.org/10.1007/s11433-017-9132-y

Improving the efficiency of quantum hash function by dense coding of coin operators in discrete-time quantum walk

YuGuang Yang[1*], YuChen Zhang[1], Gang Xu[2], XiuBo Chen[2], Yi-Hua Zhou[1] & WeiMin Shi[1]

[1] *Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China;*
[2] *Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*

Received October 3, 2017; accepted November 8, 2017; published online January 18, 2018

Li et al. first proposed a quantum hash function (QHF) in a quantum-walk architecture. In their scheme, two two-particle interactions, i.e., *I* interaction and π-phase interaction are introduced and the choice of *I* or π-phase interactions at each iteration depends on a message bit. In this paper, we propose an efficient QHF by dense coding of coin operators in discrete-time quantum walk. Compared with existing QHFs, our protocol has the following advantages: the efficiency of the QHF can be doubled and even more; only one particle is enough and two-particle interactions are unnecessary so that quantum resources are saved. It is a clue to apply the dense coding technique to quantum cryptographic protocols, especially to the applications with restricted quantum resources.

quantum cryptography, quantum walk, discrete-time quantum walk, collision, birthday attack

PACS number(s): 03.67.Dd, 03.67.Hk, 03.67.Lx

Citation: Y. G. Yang, Y. C. Zhang, G. Xu, X. B. Chen, Y. H. Zhou, and W. M. Shi, Improving the efficiency of quantum hash function by dense coding of coin operators in discrete-time quantum walk, Sci. China-Phys. Mech. Astron. 61, 030312 (2018), https://doi.org/10.1007/s11433-017-9132-y

1 Introduction

In the 1950's Knuth [1] first introduced the concept of hash functions. Hash functions can map a message of arbitrary length to a fixed-size message digest or hash value. They have played an important role in many cryptography applications, such as message authentication and digital signature. Conventional hash functions such as MD4, MD5, SHA-1, SHA-2, HAVAL-128, and RIPEMD have been proposed. Unfortunately, some of them such as MD5, SHA1, and RIPEMD [2-4] were investigated to have many security loopholes in terms of the collision frequencies.

As we know, there exist many cryptographic algorithms, whose security is based on the unproven computational complexity assumption. For example, by exploiting the attribute-based cryptography, Shen et al. [5] proposed a novel framework for urban data sharing. As for search algorithm, many studies had been done. An innovative semantic search scheme in the encrypted datasets is proposed [6]. An efficient multi-keyword fuzzy ranked search scheme which can achieve high accuracy [7] and a content-aware search scheme which can make semantic search smarter [8] were proposed by Fu et al. [7,8]. But, these classical cryptographic algorithms might be susceptible to the powerful parallelism capability of quantum computation, such as Shor's algorithms for factoring and Grover's algorithm for database search [9,10]. Fortunately, this difficulty may be conquered by quantum cryptography [11,12], where the security is guaranteed by the quantum physical laws. As an important branch of quantum cryptography, quantum hash functions

*Corresponding author (email: yangyang7357@bjut.edu.cn)

(QHFs) have been attracting a great deal of attention [13-19].

QHFs map a classical message into a Hilbert space. Such Hilbert space should be as small as possible, so hackers cannot obtain too much information about the classical message (this is guaranteed by quantum physical laws such as Holevo-Nayak's theorem and Holevo limit). Furthermore, quantum images of different classical messages should be as far apart as possible to avoid collision. QHFs were first implicitly introduced in Buhrman et al. [13] as quantum fingerprinting (QF) based on binary error-correcting codes. Then Gavinsky et al. [14] found that QF can be used as cryptographic primitive. Ablayev et al. gave a definition and construction of non-binary QHFs [15], and also a balanced QHF [16]. However, the construction of the balanced QHF depends on quantum entanglement. Ziatdinov [17] demonstrated how to generalize quantum hashing to arbitrary finite groups and he also presented two new constructions of QHFs [18]. The first one is based on expander graphs and the second one is based on extractor functions. In existing QHFs with classical input and quantum hash value [13-19], the verification of equality of two quantum hash values corresponding to two classical messages relies on the quantum SWAP-test. However, the one-sided error probability of discriminating between two classical messages w and w' is $\frac{1}{2}(1+\delta^2)=0.75$. Here $\langle\psi(w)\,|\,\psi(w')\rangle\le\delta$. 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 classical message.

Quantum walks (QWs) [20] have received much attention for their intrinsic interest and many possible applications and they have been experimentally demonstrated [21-23]. Li et al. [24] first put forward a kind of QHF based on two-particle interacting QWs. In their scheme, two two-particle interactions, i.e., I interaction and π-phase interaction are introduced and the choice of I or π-phase interaction as the coin operator at each iteration depends on a message bit. To further improve the efficiency and reduce implementation complexity of the QHF, in this paper, we propose an efficient QHF by dense coding of coin operators in discrete-time QW on cycles. Compared with existing QHFs, our protocol has the following advantages. (1) The efficiency of the QHF can be doubled and even more. (2) Only one particle is enough and two-particle interactions are unnecessary.

2 Preminaries

The time evolution of discrete-time QW on a graph is controlled by the unitary operator $U=S(I\otimes C)$, where S is the conditional shift operator which affects both the coin space $\mathcal{H}_{\rm c}$ and the position space $\mathcal{H}_{\rm p}$. Here $\mathcal{H}_{\rm c}$ and $\mathcal{H}_{\rm p}$ are spanned by the 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 $\mathcal{H}_{\rm c}$ and I is the identity operator applied on the position space $\mathcal{H}_{\rm p}$.

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

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

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

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

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

For one-dimensional discrete-time QW on a cycle with n nodes, the conditional shift operator $S_{\rm c}$ is expressed by

$$\begin{aligned}S_{\rm c}=&\sum_{x\in\{1,2,\cdots,N-1\}}(|\,x+1,0\rangle\langle x,0\,|+|\,1,0\rangle\langle N,0\,|)\\&+\sum_{x\in\{2,3,\cdots,N\}}(|\,x-1,1\rangle\langle x,1\,|+|\,N,1\rangle\langle 1,1\,|).\end{aligned} \tag{3}$$

3 Efficient QHF by dense coding of coin operators in discrete-time QW

3.1 Description of the efficient QHF

The efficient QHF can be constructed by subtly modifying discrete-time QW on a cycle. 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., message, and accordingly a QHF is constructed, similar to that in refs. [24-26]. The $(2i-1)$th and $(2i)$th bits of the message controls the choice of the coin operators at the ith iteration. Here we introduce four coin operators C_1, C_2, C_3, and C_4, which are controlled by the message bits "00", "01", "10", and "11", respectively. They are shown as follows:

$$C_1=\begin{bmatrix}\cos\theta_1 & \sin\theta_1\\ \sin\theta_1 & -\cos\theta_1\end{bmatrix},\ C_2=\begin{bmatrix}\cos\theta_2 & \sin\theta_2\\ \sin\theta_2 & -\cos\theta_2\end{bmatrix},$$

$$C_3=\begin{bmatrix}\cos\theta_3 & \sin\theta_3\\ \sin\theta_3 & -\cos\theta_3\end{bmatrix},\ C_4=\begin{bmatrix}\cos\theta_4 & \sin\theta_4\\ \sin\theta_4 & -\cos\theta_4\end{bmatrix}.$$

Assume the message is "01001110111001001011". The final state is given by

$$\begin{aligned}|\varphi_{\rm final}\rangle&=U(\text{message})|\varphi_0\rangle\\&=U_{11}U_{10}U_{00}U_{01}U_{10}U_{11}U_{10}U_{11}U_{00}U_{01}|\varphi_0\rangle,\end{aligned} \tag{4}$$

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

$$P(x,\text{message})=\sum_{i\in\{0,1\}}\left|\langle x,i\,|\,U(\text{message})|\varphi_0\rangle\right|^2, \tag{5}$$

where $U_{00}=S_{\mathrm{c}}(I\otimes C_1)$, $U_{01}=S_{\mathrm{c}}(I\otimes C_2)$, $U_{10}=S_{\mathrm{c}}(I\otimes C_3)$, $U_{11}=S_{\mathrm{c}}(I\otimes C_4)$. Here S_{c} is given in eq. (3). $|\varphi_0\rangle$ is the initial state of the total quantum system, i.e., $|\varphi_0\rangle=|x{=}0\rangle_{\mathrm{p}}\otimes(\cos\alpha\,|0\rangle+\sin\alpha\,|1\rangle)_{\mathrm{c}}$. The subscripts p and c represent the position and the coin, respectively.

Assume the message is $M=(m_1, m_2, m_3, m_4, \cdots, m_{2q-1}, m_{2q})$. Note that if the length of the message is not an even number, a bit "0" should be added at the end of the message.

The process of the QHF is described as follows: (1) select the parameters $(n, \theta_1, \theta_2, \theta_3, \theta_4, \alpha)$ under the constraints: n is the number of nodes of a cycle which is an odd number and $0<\theta_1, \theta_2, \theta_3, \theta_4, \alpha<\frac{\pi}{2}$; (2) run the one-coin one-walker discrete-time QW on a cycle with n nodes under the control of the message M (Figure 1 for the circuit representation of the unitary operation at the ith iteration); (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 is the biggest size of the binary string generated by each element in the resulting probability distribution.

3.2 Performance analyses

It is 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 $n=17$ so that the hash value we consider here is 17×13=221 bits. Let $\theta_1=\frac{\pi}{3}$, $\theta_2=\frac{\pi}{4}$, $\theta_3=\frac{\pi}{7}$, $\theta_4=\frac{\pi}{9}$, $\alpha=\frac{\pi}{4}$. The following results show that the proposed QHF has excellent statistical performance.

3.2.1 *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 five conditions.

Condition 1: The original message;

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

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

Condition 4: Delete the first bit of the message;

Condition 5: Insert a bit into the original message randomly.

The corresponding 221-bit hash values in the hexadecimal format are given as follows:

Condition 1: ′C8′ ′99′ ′B02′ ′1EF′ ′19′ ′3FE′ ′1FE′ ′201′ ′59′ ′12′ ′30F′ ′1DB′ ′2EC′ ′31A′ ′417′ ′118′ ′17′

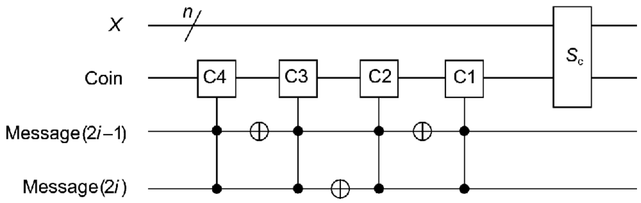

Figure 1 Circuit representation of the unitary operation at the ith iteration in the first QHF. The top n lines are the quantum register for storing the information about the position. The two lower lines are the two-bit message $m_{2i-1}m_{2i}$, and the other line is the quantum register for storing the information about the coin.

Condition 2: ′CC′ ′91′ ′A76′ ′287′ ′13′ ′4B1′ ′209′ ′1E8′ ′78′ ′6′ ′320′ ′1C3′ ′29F′ ′336′ ′3A5′ ′10D′ ′12′

Condition 3: ′AB′ ′18′ ′958′ ′14B′ ′4D′ ′3EA′ ′118′ ′1F3′ ′81′ ′4C′ ′376′ ′2CE′ ′378′ ′2A1′ ′531′ ′199′ ′6B′

Condition 4: ′2DD′ ′2′ ′5DB′ ′2BC′ ′16′ ′120′ ′609′ ′C8′ ′431′ ′D1′ ′245′ ′134′ ′32C′ ′16E′ ′5E4′ ′64′ ′2D′

Condition 5: ′13F′ ′1FF′ ′1D′ ′BE5′ ′37′ ′EC′ ′E4′ ′56C′ ′4F′ ′685′ ′DC′ ′BB′ ′23B′ ′D8′ ′3A1′ ′8D′ ′A9′.

Figure 2 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.

3.2.2 *Statistical analysis of diffusion and confusion*

The following definitions are given:

Mean normally changed bit number:

$$\overline{B}=\sum_{i=1}^{N}B_i/N;$$

Mean added bit number of zero:

$$\overline{A}=\sum_{i=1}^{N}A_i/N;$$

Mean sum of the changed bit number:

$$\overline{T}=\overline{A}+\overline{B};$$

Mean changed probability:

$$P=(\overline{T}/221)\times 100\%;$$

Standard variance of the total changed bit number:

$$\Delta T=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(T_i-\overline{T})^2};$$

Standard variance of the changed probability:

$$\Delta P=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(T_i/221-P)^2}\times 100\%.$$

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 and count the added zero called A_i.

(5) Repeat steps (1) to (4) N times. Here N is the number of tests.

The diffusion and confusion tests are performed with N=1024, 2048, 10000, respectively, as shown in Table 1. We concluded from the tests that the sum of mean changed bit number $\overline{T}$ is close to a half of the length of the hash value, i.e., 110.5, and the mean changed probability P close to 50%. ΔT and ΔP are very little. The excellent statistical feature ensures that it is infeasible to forge valid plaintext-ciphertext pairs given known plaintext-ciphertext pairs.

3.2.3 *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) Insert a bit into 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

$$\omega = \sum_{i=1}^{N} \delta(t(e_i) - t(e'_i)), \tag{6}$$

where $\delta(x)$ =1 for x=0 and 0 for $x\neq1$. e_i and e'_i represent the ith elements of the original and new hash value in ASCII format, respectively. $t(e_i)$ and $t(e'_i)$ represent e_i and e'_i in the decimal form, respectively. We performed the tests N=10000 times, and concluded that ω=0 occurs for 9914 times, ω=1 for 86 times, and the others for zero time.

3.2.4 *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 here is 17×13=221 bits. Therefore, it needs $2^{221/2}$

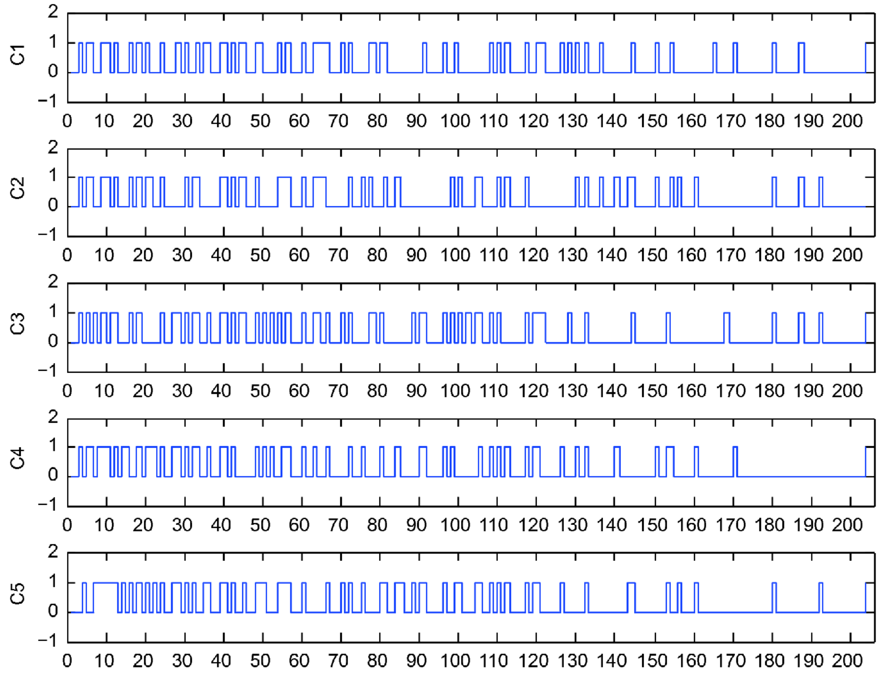

Figure 2 (Color online) Hash values of C1, C2, C3, C4, C5.

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

	N=1024	N=2048	N=10000	Mean
$\overline{B}$	38.8545	38.5542	38.5839	38.6642
$\overline{A}$	69.2021	69.2085	69.1016	69.1707
$\overline{T}$	108.0566	107.7627	107.6855	107.8349
P (%)	48.8944	48.7614	48.7265	48.7941
ΔT	6.9318	6.4671	6.4822	6.6270
ΔP	3.1366	2.9263	2.9331	2.9987

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 over 2^{64}, the hash function can be considered to be of less collision. In our proposed QHF, the length of hash value is 221 bits. Furthermore, the proposed QHF can be easily extended to the case with a larger number of nodes. Therefore, the experimental results and the length of the hash value suggest that the proposed QHF be resistant against the birthday attack.

4 A generalization to *m* (*m*>2) message bits at each iteration

4.1 Description of the proposed QHF

For simplicity and without loss of generality, we take the case with m=3 as an example. the coin operator at the ith iteration depends on the ($3i$−2)th, ($3i$−1)th and ($3i$)th bits of the message. Eight coin operators C_1, C_2, C_3, C_4, C_5, C_6, C_7, and C_8 are introduced, which are controlled by the message bits “000”, “001”, “010”, “011”, “100”, “101”, “110”, and “111”, respectively. The eight coin operators are given by

$$C_1=\begin{bmatrix}\cos\theta_1 & \sin\theta_1\\ \sin\theta_1 & -\cos\theta_1\end{bmatrix},\ C_2=\begin{bmatrix}\cos\theta_2 & \sin\theta_2\\ \sin\theta_2 & -\cos\theta_2\end{bmatrix},$$

$$C_3=\begin{bmatrix}\cos\theta_3 & \sin\theta_3\\ \sin\theta_3 & -\cos\theta_3\end{bmatrix},\ C_4=\begin{bmatrix}\cos\theta_4 & \sin\theta_4\\ \sin\theta_4 & -\cos\theta_4\end{bmatrix},$$

$$C_5=\begin{bmatrix}\cos\theta_5 & \sin\theta_5\\ \sin\theta_5 & -\cos\theta_5\end{bmatrix},\ C_6=\begin{bmatrix}\cos\theta_6 & \sin\theta_6\\ \sin\theta_6 & -\cos\theta_6\end{bmatrix},$$

$$C_7=\begin{bmatrix}\cos\theta_7 & \sin\theta_7\\ \sin\theta_7 & -\cos\theta_7\end{bmatrix},\ C_8=\begin{bmatrix}\cos\theta_8 & \sin\theta_8\\ \sin\theta_8 & -\cos\theta_8\end{bmatrix}.$$

Assume the original message is M=(m_1, m_2, m_3, $\cdots$, m_{3n-2}, m_{3n-1}, m_{3n}). Note that if the length of the message is not the times of three, necessary zeros can be added at the end of the message in order to satisfy the requirement.

The process of the proposed QHF is described as follows:

(1) Select the parameters (n, θ_1, θ_2, θ_3, θ_4, θ_5, θ_6, θ_7, θ_8, α) under the constraints: n is limited to odd numbers and $0<\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8, \alpha<\frac{\pi}{2}$.

(2) Run the one-coin one-walker discrete-time QW on a cycle with n nodes under the control of the message M (Figure 3 for the circuit representation of the unitary operation at the ith iteration), respectively.

(3) Multiply all values in the resulting probability distribution by 10^4 times to generate a binary string as the hash value.

4.2 Performance analysis

We set the initial parameters $\theta_1=\frac{\pi}{3}$, $\theta_2=\frac{\pi}{5}$, $\theta_3=\frac{\pi}{7}$, $\theta_4=\frac{\pi}{9}$, $\theta_5=\frac{\pi}{11}$, $\theta_6=\frac{\pi}{13}$, $\theta_7=\frac{\pi}{15}$, $\theta_8=\frac{\pi}{17}$, $\alpha=\frac{\pi}{4}$ for test.

4.2.1 Sensitivity of hash value to message

Similar to the method in sect. 3.2.1, we constructed five different messages by making a tiny modification of the original message. Then, we calculated and compared the hash values of all resulting messages.

The corresponding 221-bit hash values in the hexadecimal format are given by

Condition 1: ′13′ ′7EC′ ′172′ ′3E6′ ′404′ ′FA′ ′195′ ′146′ ′184′ ′18F′ ′D8′ ′394′ ′427′ ′10D′ ′6E′ ′401′ ′B7′

Condition 2: ′10′ ′7CD′ ′156′ ′36A′ ′535′ ′F1′ ′1CB′ ′170′ ′160′ ′13B′ ′F3′ ′3EB′ ′405′ ′E7′ ′59′ ′38B′ ′C0′

Condition 3: ′83′ ′642′ ′43′ ′405′ ′3C2′ ′E9′ ′164′ ′65′ ′1BE′ ′2E7′ ′F7′ ′332′ ′4B8′ ′133′ ′103′ ′51D′ ′AB′

Condition 4: ′2DD′ ′2′ ′5DB′ ′2BC′ ′16′ ′120′ ′609′ ′C8′ ′431′ ′D1′ ′245′ ′134′ ′32C′ ′16E′ ′5E4′ ′64′ ′2D′

Condition 5: ′1E8′ ′91′ ′DE′ ′751′ ′146′ ′137′ ′14′ ′9F8′ ′4F′ ′1F9′ ′1AB′ ′71′ ′12C′ ′56C′ ′2E0′ ′1FB′ ′0′.

The plots of the corresponding hash values are shown respectively in Figure 4 and it is clearly indicated that any tiny modification to the original message will cause a huge

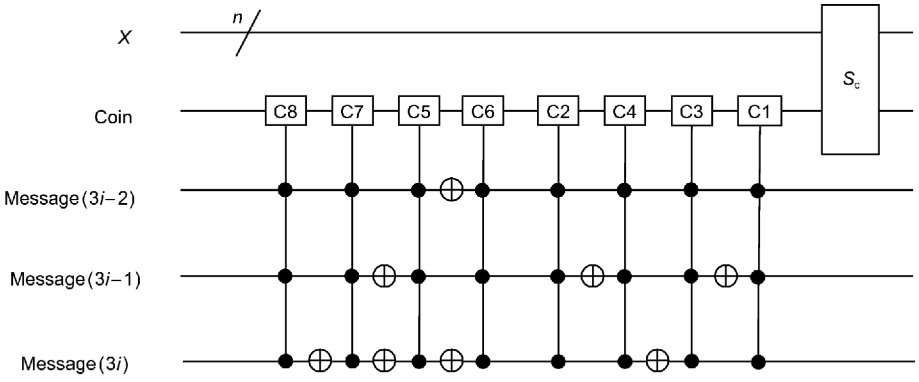

Figure 3 Circuit representation of the unitary operation at the ith iteration in the second QHF. The top n lines are the quantum register for storing the information about the position. The three lower lines are the three-bit message, and the other line is the quantum register for storing the information about the coin.

change in the new hash value.

4.2.2 Statistical analysis of diffusion and confusion

The diffusion and confusion tests are performed with N=1024, 2048, 10000, respectively, as shown in Table 2. We concluded from the tests that the sum of mean changed bit number $\overline{T}$ is close to a half of the length of the hash value, i.e., 110.5, and the mean changed probability P close to 50%, respectively. ΔT and ΔP are very little. The excellent statistical feature ensures that it is infeasible to forge valid plaintext-ciphertext pairs given known plaintext-ciphertext pairs.

4.2.3 Collision analysis

Similar to the method in sect. 3.2.3, we performed the tests N=10000 times for collision resistance and concluded that ω=0 occurs for 9854 times, ω=1 for 71 times, ω=17 for 75 times, and the others for zero times.

4.2.4 Resistance to birthday attack

The length of the hash value of the second QHF here is also 221 bits. Therefore, it can also be resistant against the birthday attack.

5 Comparison with other QHFs

Resistance against collision is an important indicator of evaluating a QHF protocol. Firstly, we compared our schemes with existing QHFs in terms of collision resistance, as shown in Table 3. Compared with refs. [24,25], our QHFs demonstrate better capability of collision resistance. For example, the number of no collision is 9914 in our first QHF and 9854 in our second QHF while it is 7688 in Li et al.'s QHF [24], 9367 in Yang et al.'s QHF [25] and 9068 in Li et al.'s QHF [26].

Resource consumption is another important indicator of evaluating a QHF protocol. We compared our schemes with existing QHF protocols in terms of quantum operation, the number of the particles and the number of message bits used per iteration, as shown in Table 4.

In ref. [24], Li et al. put forward a kind of QHF scheme

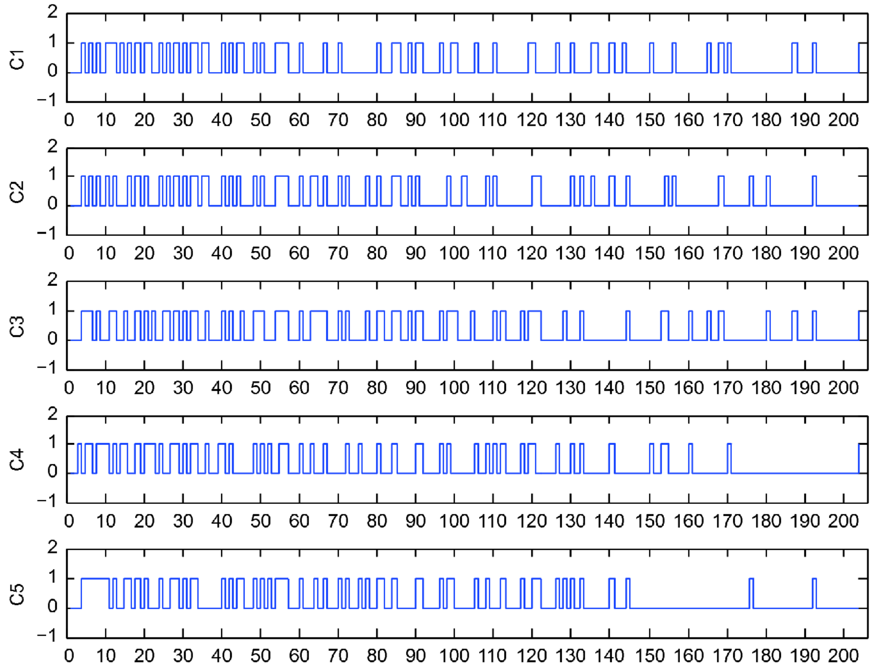

Figure 4 (Color online) Hash values of C1, C2, C3, C4, C5.

Table 2 Results for diffusion and confusion tests in the second QHF

	N=1024	N=2048	N=10000	Mean
$\overline{B}$	41.4688	41.3569	42.5407	41.7888
$\overline{A}$	70.2764	70.1060	70.2384	70.2069
$\overline{T}$	111.7452	111.4629	112.7791	111.9957
P (%)	50.5634	50.4357	51.0313	50.6768
ΔT	7.8213	8.5069	8.2029	8.1770
ΔP	3.5391	3.8493	3.7117	3.7000

Table 3 Comparison between our schemes and other QW-based QHFs in terms of collision resistance

	No collision	One collision	Two collisions	More collisions
ref. [24]	7688	375	1662	275
ref. [25]	9367	617	16	0
ref. [26]	9068	889	42	1
Our first scheme	9914	86	0	0
Our second scheme	9854	71	0	75

Table 4 Comparison between our schemes and other QW-based QHFs in terms of resource consumption. Here, C_2, C_v and C_w are 2×2 coin operators, and C_4 is a 4×4 coin operator

	The number of the particles	Quantum operation	The number of message bits
ref. [24]	2	$S_{xy}(I\otimes C_2\otimes C_2)$	1
ref. [25]	2	$S_{xy}(I\otimes C_4)$	1
ref. [26]	1	$S_yC_2S_xC_2$	1
Our first scheme	1	$S_c(I\otimes C_v)$	2
Our second scheme	1	$S_c(I\otimes C_w)$	3

based on two-particle interacting quantum walks, where two 2×2 coin operators, i.e., I interaction and π-phase interaction are introduced. The use of I interaction or π-phase interaction as the coin operator at each step depends on a bit of a message. To reduce the occurance of the collisions, ref. [25] introduced two 4×4 coin operators, i.e., Grover operator, a swap operator [27] and the coin operator E [28]. But it is more difficult to implement these 4×4 coin operators than a coin operator spanned by two 2×2 unitary operators. Furthermore, the coin operator to be used at each step depends on a message bit. To reduce the implementation difficulty, and improve the efficiency of implementation, we considered the technique of dense coding of coin operators in the QHF. Alternate coin operators are used which are controlled by two, three or even more message bits at each iteration.

6 Discussions and conclusion

Because QHFs have unique merits compared with their classical counterparts, they may take place of classical hash functions in some quantum cryptography protocols, e.g., quantum key distribution [29], quantum signature [30-33], quantum private comparison [34-36], quantum key agreement [37,38], quantum private query [39-42], etc., to improve the security of these protocols.

In this paper, we have proposed an efficient QHF by using the technique of dense coding of coin operators in discrete-time QW on cycles. The presented technique for dense coding of coin operators can be generalized to the case with controlling the coin operator by m (m>2) message bits at each iteration. Compared with existing QHFs, our protocol has the following advantages. (1) The efficiency of the QHF can be doubled and even more. (2) Only one particle is enough and two-particle interactions are unnecessary. However, it should be noted that m should be valued properly to improve the efficiency of implementing QHF. That means it may be not more operable with a bigger m when implemented experimentally. A bigger m may also have an effect on the collision performance of the proposed QHF. Next, we will focus on these problems in the future work.

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61572053, 61671087, U1636106, and 61602019), and Beijing Natural Science Foundation (Grant No. 4162005).

1 D. Knuth, *The Art of Computer Programming, Sorting and Searching* (Addison-Wesley, New Jersey, 1998).
2 X. Wang, D. Feng, X. Lai, and H. Yu, in *Rump Session of Crypto'04 E-print*, Santa Barbara, 2004.
3 X. Wang, X. Lai, D. Feng, X. Yu, and X. Yu, in *Proceedings of Eurocrypt'05*, Aarhus, 2005. pp. 1-18.
4 X. Wang, and H. Yu, in *Proceedings of Eurocrypt'05*, Aarhus, 2005. p. 19-35. ; A. Menezes, P. Van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography* (CRC Press, 1996).
5 J. Shen, D. Liu, J. Shen, Q. Liu, and X. Sun, Pervasive Mobile Computing **41**, 219 (2017).
6 Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, IEEE Trans. Parallel. Distrib. Syst. **27**, 2546 (2016).
7 Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, IEEE Trans. Inform. Foren. Secur. **11**, 2706 (2016).
8 Z. Fu, F. Huang, K. Ren, J. Weng, and C. Wang, IEEE Trans. Inform. Foren. Secur. **12**, 1874 (2017).
9 P. W. Shor, in *Proceedings of 35th Annual Symposium on the Foundations of Computer Science*, Santa Fe, 1994, pp. 124-134.
10 L. K. Grover, in *Proceedings of 28th Annual ACM Symposium on Theory of Computing*, New York, 1996, pp. 212-218.

11 C. H. Bennett, and G. Brassard, in *Proceedings of IEEE International Conference on Computers, Systems and Signal*, Bangalore, 1984, pp.175-179.

12 N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, Rev. Mod. Phys. **74**, 145 (2002).

13 H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Phys. Rev. Lett. **87**, 167902 (2001).

14 D. Gavinsky, and T. Ito, *Quantum Fingerprints that Keep Secrets*. Technical Report (Cornell University Library, 2010).

15 F. M. Ablayev, and A. V. Vasiliev, Laser Phys. Lett. **11**, 025202 (2014).

16 F. Ablayev, M. Ablayev, and A. Vasiliev, J. Phys.-Conf. Ser. **681**, 012019 (2016).

17 M. Ziatdinov. arXiv: 1412. 5135

18 M. Ziatdinov, Lobachev. J. Math. **37**, 705 (2016).

19 A. Vasiliev, Lobachev. J. Math. **37**, 753 (2016).

20 D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, in *Proceedings of the 33rd ACM Symposium on Theory of Computing*, Crete, 2001, pp. 50-59.

21 A. Ambainis, SIAM J. Comput. **37**, 210 (2007).

22 F. Magniez, M. Santha, and M. Szegedy, SIAM J. Comput. **37**, 413 (2007).

23 D. Tamascelli, and L. Zanetti, J. Phys. A-Math. Theor. **47**, 325302 (2014), arXiv: 1401.1278

24 D. Li, J. Zhang, F. Z. Guo, W. Huang, Q. Y. Wen, and H. Chen, Quantum Inf. Process. **12**, 1501 (2013).

25 Y. G. Yang, P. Xu, R. Yang, Y. H. Zhou, and W. M. Shi, Sci. Rep. **6**, 19788 (2016).

26 D. Li, Y.-G. Yang, J.-L. Bi, J.-B. Yuan, and J. Xu. arXiv: 1707.07389

27 P. Xue, and B. C. Sanders, Phys. Rev. A **85**, 022307 (2012), arXiv: 1112.1487

28 M. Štefaňák, S. M. Barnett, B. Kollár, T. Kiss, and I. Jex, New J. Phys. **13**, 033029 (2011), arXiv: 1102.4445

29 H. K. Lo, and H. F. Chau, Science **283**, 2050 (1999).

30 Y. G. Yang, Z. C. Liu, J. Li, X. B. Chen, H. J. Zuo, Y. H. Zhou, and W. M. Shi, Quantum Inf. Process. **16**, 12 (2017).

31 Y. G. Yang, H. Lei, Z. C. Liu, Y. H. Zhou, and W. M. Shi, Quantum Inf. Process. **15**, 2487 (2016).

32 T. Y. Wang, and Z. L. Wei, Quantum Inf. Process. **11**, 455 (2012).

33 T. Y. Wang, X. Q. Cai, Y. L. Ren, and R. L. Zhang, Sci. Rep. **5**, 9231 (2015).

34 Y. G. Yang, and Q. Y. Wen, J. Phys. A-Math. Theor. **42**, 055305 (2009).

35 Y. G. Yang, W. F. Cao, and Q. Y. Wen, Phys. Scr. **80**, 065002 (2009).

36 X. B. Chen, G. Xu, X. X. Niu, Q. Y. Wen, and Y. X. Yang, Opt. Commun. **283**, 1561 (2010).

37 Y. F. He, and W. P. Ma, Quantum Inf. Process. **15**, 5023 (2016).

38 B. Liu, F. Gao, W. Huang, and Q. Wen, Quantum Inf. Process. **12**, 1797 (2013).

39 F. Gao, B. Liu, W. Huang, and Q. Y. Wen, IEEE J. Sel. Top. Quantum Electron. **21**, 98 (2015).

40 C. Y. Wei, T. Y. Wang, and F. Gao, Phys. Rev. A **93**, 042318 (2016).

41 Y. G. Yang, Z. C. Liu, X. B. Chen, Y. H. Zhou, and W. M. Shi, Sci. China-Phys. Mech. Astron. **60**, 120311 (2017).

42 Y. G. Yang, Z. C. Liu, J. Li, X. B. Chen, H. J. Zuo, Y. H. Zhou, and W. M. Shi, Phys. Lett. A **380**, 4033 (2016).