Controlled alternate quantum walk-based block hash function

Dan Li¹ · Panpan Ding¹ · Yuqian Zhou¹ · Yuguang Yang²

Received: 19 April 2023 / Accepted: 14 September 2023 / Published online: 4 October 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
The hash function is an important branch of cryptology. The controlled quantum walk-based hash function, which is presented by Li, is a kind of novel hash function. It is safe, flexible, and compatible. All existing controlled quantum walk-based hash functions are controlled by one bit of message in each step. To process the message in batch amounts, in this paper, the controlled alternate quantum walk-based block hash function is presented by using the time-position-dependent controlled alternate quantum walk on the complete graph with self-loops. The presented hash function accelerates the hash processing dramatically, so it is more efficient. That could save many resources and execution time further.

Keywords Quantum walk · Block cipher · Hash function · Quantum hash function

1 Introduction

The hash function is an important branch of cryptology. The hash function includes all irreversible functions that can be used to map arbitrary-size data onto fixed-size data. The values returned by a hash function are called hash values. A hash function allows one to easily verify whether the input data map onto a given hash value, but if the input data is unknown it is deliberately difficult to reconstruct it by knowing the stored hash value. There are many theoretical studies about mature classical hash functions, such as MD5, SHA1, and SHA512. Furthermore, hash functions can be used in message authentication codes and public key infrastructure. These hash functions are generally constructed based on computationally hard problems, thus they are computationally secure.
The rapid development of quantum computation brings a great challenge to cryptology. Shor’s integer factorization algorithm collapses the security of many public-key cryptography protocols. Grover’s search algorithm threatens the security of all symmetric cryptography protocols. Classical hash functions may be no longer safe in the further. Therefore, how to develop secure cryptology is very important. Except for post-quantum cryptography, quantum cryptography [1, 2] is the principal method, for instance, quantum key distribution. In addition, quantum computation plays an unignorable act in the field of quantum cryptography.

Quantum walk, the quantum counterpart of classical random walk, is an underlying mathematical model in realizing quantum computation. Alternate quantum walks, quantum walks with memory, and quantum walks on different kinds of graphs are presented in Ref. [3–7] for different purposes. Quantum walk has wide applications in quantum computation and quantum communication, such as database searching, element distinctness, graph isomorphism testing, and quantum network communication.

In 2013, Li et al. [8, 9] presented the two-particle controlled interacting quantum walk (CIQW) and CIQW-based quantum hash function. A comprehensive analysis of CIQW proved that it is suitable for designing quantum hash functions. This quantum hash function guarantees the security of the hash function by the irreversibility of measurement rather than hard mathematic problems. This work opens the door to the controlled quantum walk (CQW)-based quantum hash function research.

In the next years, the CQW-based quantum hash function attracts much attention from researchers. Many kinds of CQW on different graphs are introduced for building different CQW-based quantum hash functions. Yang et al. [10] improved the CIQW-based quantum hash function and found its applications in the privacy amplification process of quantum key distribution, pseudo-random number generation, and image encryption. After that, Yang et al. [11] presented a quantum hash function based on the controlled quantum walk on Johnson graphs, which has a lower collision rate and quantum resource cost. The CQW-based quantum hash function with controlled quantum walk on cycles was also presented by Yang et al [12] in 2018. The CQW-based quantum hash function with a controlled quantum walk on cycles with two coins was presented by Yang et al. [13] in 2019. Besides, the CQW-based quantum hash function with a decoherent quantum walk on a cycle is shown by Yang et al. [14] in 2021.

In 2018, Li et al. [15] presented the controlled alternate quantum walk (CAQW)-based quantum hash function. The controlled alternate quantum walk saves quantum resource costs dramatically. Then, the controlled alternate quantum walk is used in constructing a pseudo-random number generator and quantum color image encryption [16–19]. Zhou presented the quantum hash function based on controlled alternate quantum walks with memory in 2021 [20]. They claim that the proposed quantum hash function has near-ideal statistical performance.

Although the research on CQW-based hash function thriving, how to improve the efficiency is an important research orientation. In each step of all existing CQW, the coin operator is controlled by a one-bit message. That means the number of steps is equal to the length of the message.
In this paper, to speed up the enforcement efficiency of the CQW-based hash function further, the time-position-dependent CAQW on the complete graph with self-loops is introduced to build the controlled alternate quantum walk-based block hash function (CAQWBH). The complete graph with self-loops is chosen for constructing block hash functions that satisfy the avalanche characteristic. The controlled alternate quantum walk is chosen for decreasing the dimension of the coin state. Therefore, CAQWBH could accelerate hash processing dramatically by processing a message in batch amounts.

The paper is structured as follows. In Sect. 2 and 3, the time-position-dependent CAQW on the complete graph with self-loops and CAQWBH are presented, respectively. The security analysis and statistical performance of CAQWBH are discussed in Sect. 4 and Sect. 5. Finally, a conclusion is given in Sect. 6.

2 The time-position-dependent controlled alternate quantum walk on the complete graph with self-loops

In this section, to speed up the enforcement efficiency of the CQW-based hash function further, the time-position-dependent CAQW on the complete graph with self-loops is introduced for the construction of CAQWBH.

To process a message in batch amounts, the complete graph with self-loops is selected. In this way, the variation of the coin operator in each position can lead to the variation of the final state. However, the quantum walks on the complete graph needs a large coin. Hence, the controlled alternate quantum walk is chosen for decreasing the dimension of the coin state.

Suppose \( G_N \) is the complete graph with \( N \) vertexes and \( N \) self-loops. Here \( N = 2^q \). The time-position-dependent CAQW on \( G_N \) takes place in the product space \( H_p \otimes H_c \). \( H_p \) is the position space, which is a Hilbert space spanned by \( \{ |x\rangle, x \in \{ 0, \ldots, N - 1 \} \} \). Furthermore, \( H_p \) can also be spanned by \( \{ |x_q, \ldots, x_1, x_k \rangle, x_k \in \{ 0, 1 \} \} \). \( H_c \) is the coin space, which is a Hilbert space spanned by \( \{ |c\rangle, c \in \{ 0, 1 \} \} \). Let \( |x, c\rangle \) or \( |x_q, \ldots, x_1, c\rangle \) be a basis state of the system of the time-position-dependent CAQW on \( G_N \), where \( x \) and \( c \) represent the position and the coin state of the walker, respectively.

The whole system begins with the initial state \( |\psi_0\rangle = |0\rangle \otimes |\phi_0\rangle = |0\rangle \otimes (\sum_i |\alpha_i\rangle \langle \alpha_i|) \), where \( \sum_i |\alpha_i|^2 = 1 \). The evolution of each step of the CAQW system can be described by the global unitary operator, denoted by \( U \),

\[
U(t) = S_q C(t) \cdots S_1 C(t).
\] (1)

If \( N = 32 \), \( U(t) = S_5 C(t)S_4 C(t)S_3 C(t)S_2 C(t)S_1 C(t) \). In this process, the coin operator is unaltered. By applying the unitary operator \( U \), the walker can arrive at any position of \( G_N \) in one step, i.e., walking on the complete graph with self-loops.

The coin operator \( C(t) = (\sum_x |x\rangle \langle x| \otimes C_2(x, t)) \) is a time-position-dependent unitary operator under the control of message, i.e., a binary string. The two-dimensional operator \( C_2(x, t) \) has two options \( C_{2,0} \) and \( C_{2,1} \) to select.
Two parameters $\theta_0, \theta_1$ are selected from \((0, \pi/4) \cup (\pi/4, \pi/2)\) to construct two unitary operators $C_2,0$ and $C_2,1$, respectively. If the bit is 0/1, the corresponding $C_2(x, t)$ should be $C_2,0$/$C_2,1$. Therefore, for each $t$, $C(t)$ is controlled by a $N$-bit message. As a result, the evolution of the CAQW is controlled by a $Nt$-bit message. If the length of the message is not a multiple of $N$, the Hadamard matrix is the default choice for $C_2(x, t)$.

The shift operators $S_i$, which act only on $|x_i, c\rangle$ effectively, are defined by

$$S_i = \sum_{j,c \in \{0,1\}} |j + c \mod 2\rangle \langle j| \otimes |c\rangle \langle c|.$$  

(3)

Here the expression of $S_i$ omits the identity operator on other $|x_k\rangle$s, where $k \neq i$.

Then, the final state can be expressed by

$$|\psi_t\rangle = U(t) \times \cdots \times U(1)|\psi_0\rangle,$$  

(4)

where $|\psi_0\rangle$ is the initial state of the total quantum system. Hence the probability of finding the walker at position $x$ after $t$ steps is

$$P(x, t) = \sum_c \langle x, c|U(t) \times \cdots \times U(1)|\psi_0\rangle^2.$$  

(5)

By walking on the complete graph with self-loops, the variation of the coin operator in each position can lead to the variation of the probability distribution. Then, by adopting the time-position-dependent coin operator, the variation of each bit of the message can lead to the variation of the probability in each position. In this way, the corresponding CAQWBH satisfies the avalanche characteristic, which is the basic requirement of a hash function.

Furthermore, due to the operators $C(t)$ and $S_i$ being sparse matrices, executing the evolution operators $U(t)$ of the CAQW on a quantum computer or simulating them on a classical computer are both efficient and undemanding.

It is worth emphasizing that it is easy to get the final state from the known initial state, but it is hard to get any desired final state by selecting the message. That is because all possible final states are distributed in the evolution space discretely as the initial state change. There is no better way than an exhaustive search till now to verify if a state can be the final state.

3 The controlled alternate quantum walk-based block quantum hash function

CAQWBH is constructed by running the time-position-dependent CAQW on $G_N$. CAQW is chosen for decreasing the dimension of the coin state. The complete graph
with self-loops is chosen for constructing hash functions that satisfy the avalanche characteristic.

The process of CAQWBH is described as follows. The flowchart is shown in Fig. 1.

1. Select the parameters \((N, k, (\theta_1, \theta_2), (\alpha_i))\). \(\theta_1, \theta_2 \in (0, \pi/4) \cup (\pi/4, \pi/2)\), \(i \in \{0, 1\}\) and \(\sum_i |\alpha_i|^2 = 1\). \(\theta_1, \theta_2\) are the parameters of the two coin operators, respectively.

2. Initialize the quantum system. The initial state is \(|0\rangle \otimes (\sum_i \alpha_i|i\rangle)\). Then run the time-position-dependent CAQW on \(G_N\) one step under the control of a binary string which is consisted of \(N\) 0s. The purpose of this step is to make sure the quantum system begins with a superposition state where all probabilities in each position are nonzero.

3. Run the time-position-dependent CAQW on \(G_N\) to get the final state under the control of the message. The evolution of CAQW in each step is controlled by \(N\) bits. The default option for \(C_2\) is the Hadamard matrix, if the length of the message is not a multiple of \(2N\). So if the length the message is \(L_M\), the step number of CAQW is \(2 \lceil \frac{L_M}{2N} \rceil \approx \frac{L_M}{N}\).

4. Post-processing of the probability distribution to get the hash value. Multiply all values in the resulting probability distribution by \(10^k\). Then retain the remainders of modulo \(2^k\) to form a binary string as the hash value. The bit length of the hash value is \(N \times k\).

Because CAQWBH does an exceedingly good job of compressing the evolution space, the probability distribution is not extremely sensitive to the message and the initial state. Therefore, to make up for the shortfall, in step 3 of CAQWBH, the length of the message is replenished to a multiple of \(2N\) rather than \(N\). The relation between the parameters and the length of the hash value is shown in Table 1.

As a hash function, apart from satisfying the basic requirements, such as a fixed length of the hash value and avalanche characteristic, CAQWBH is flexible, highly-efficient, and compatible. The concrete analysis is as follows.

(a) **Flexible**: CAQWBH with the hash value of different lengths are easy to construct by changing the parameters \(N\) and \(k\) rather than the structure of CAQWBH. The
relation between the parameters and the length of the hash value is shown in Table 1.

b) Highly-efficient: As a CQW-based hash function, the hash process from the initial state to the final state is linear, i.e., the Kronecker product of sparse matrices. That makes sure CAQWBH is highly-efficient. Furthermore, in each step of the time-position-dependent CAQW on $G_N$, the coin operator is controlled by a $N$-bit message. Therefore, CAQWBH can process messages in batch amounts, which could accelerate the hash processing dramatically compare to other CQW-based quantum hash functions. For CAQWBH-256, it can process 32 bits in one step, so the walker only needs to walk about $\frac{L_M}{N}$ steps instead of $L_M$, if the length of the message is $L_M$. Related data are shown in Table 1.

c) Compatible: CQW-based hash function is compatible to be executed on a quantum computer or a classical computer. So is CAQWBH. To get the accurate probability distribution, the time-position-dependent CAQW needs to be executed many times on a quantum computer. Taking the difficulty of the commercialization of the quantum computer into consideration, it is not practical yet. At the same time, CAQWBH is safe and even more highly efficient on a classical computer at present. CAQWBH is practical even in current hardware technology.

4 Security analysis

The first shield to protect the message from an unauthorized party is the modulo arithmetic. By using the modulo operator, the probability distribution is transformed into the hash value,

$$\text{HashValue} = \text{mod}(10^kP, 2^k).$$

This process is irreversible because it is a many-to-one relationship. An unauthorized party can only guess the probability distribution based on the hash value. Nevertheless, the probability of transforming the hash value back to the right probability distribution is approximately 0.

The second shield of CAQWBH is the irreversibility of measurement. The final state of the time-position-dependent CAQW is in the following form

$$|\psi_{t}\rangle = \sum_{x,c} \lambda_{x,c} |x, c \rangle.$$  

The probability distribution is the sum of squares of the probability amplitudes

$$P(x) = \sum_{c} |\lambda_{x,c}|^2.$$  

Because of the infinity of decomposing a number as the sum of squares, the probability of deducing the probability distribution into the right linear composition of squares of
amplitudes is 0, even the unauthorized party gets the probability distribution.

\[ P(x) = \sum_c |\lambda_{x,c}|^2 = \sum_c |\lambda'_{x,c}|^2. \]  

(9)

So the measurement processing is nonlinear. An unauthorized party can only guess the final state based on the probability distribution, let alone the probability distribution comes from speculation.

Above two irreversible computational processes guarantee that it is impossible to backtrack the right final state from the hash value. In addition, even if an unauthorized party has a final state of speculation, it is still hard to get the message. The security analysis of this stage can be classified into two types: the initial state is private or public.

a) *The initial state is public.* All possible final states are distributed in the evolution space discretely as the initial state change. If the initial state is public, all possible final states can be calculated by anyone. And the final state is linear with the initial state. Even so, there is no better way than an exhaustive search till now to find the message according to the final state or get any desired final state by selecting the message.

b) *The initial state is private.* If the initial state is private, even an exhaustive search can not succeed. Because the number of the possible initial states is infinite, i.e., aleph-one. Therefore, it is impossible to get the initial state or the message by the hash value even an unauthorized party has a quantum computer.

To sum up, CAQWBH can resist the attraction from a quantum computer if the initial state is private. Even though the initial state is public, CAQWBH is safe enough that only an exhaustive search could work.

**5 Statistical performance analysis**

In this section, several hash tests are performed to evaluate the performance of CAQWBH-256. \( N = 32, \ k = 8, \ \cos(\theta_1) = 3/5, \ \cos(\theta_2) = 8/17 \) are chosen in this part. Statistical performance analysis proves that CAQWBH-256 is good enough that the statistical performance of the hash value is similar to that of a random number string.

**5.1 Statistical performance analysis for initial state**

The message we considered here is 43652 bits. The results show that CAQWBH has outstanding statistical performance for different initial states.

**5.1.1 Sensitivity of hash value**

Let \( I_s1 = \{|0\rangle_p|0\rangle_c \}, \ I_s2 = \{|0\rangle_p|0.001\rangle_c \}, \ I_s3 = \{|0\rangle_p|0.002\rangle_c \}, \) and \( I_s4 = \{|0\rangle_p|0.003\rangle_c \}. \)
Here the hash values corresponding to the above four different initial states are shown to demonstrate the high sensitivity of the hash value to the initial state. And the corresponding 256-bit hash values in the hexadecimal format are given in the following:

H1: 15C16F398540C24A254532805CBBE4F1
    7C8F3B1DE86B08204099CF60C509CE15
H2: FE39409C88455AB4A1C22A9600E1ABDD
    9BAF47B0109D39C5640D53B6C908E956
H3: E8B213008F40F61C1E394FA5A60273C7
    BBCC5A423D987068897DE20AD0060697
H4: D42AE664983195839BABABA3AB4D1D3EB2
    DBE874D36F5EAD0BAEEA7C5DDA0225D8

The plots of the hash values in the binary format are shown in Fig. 2. It clearly indicates that any tiny modification to the initial state will cause a substantial change in the final hash value. A similar result can be obtained using any other instances of CAQWBH. Therefore, the output hash value of CAQWBH is highly sensitive to its initial state.

5.1.2 Statistical analysis of diffusion and confusion

The diffusion and confusion tests are performed as follows:

(1) The original initial state is $|0\rangle_p |0\rangle_c$. Generate the hash value corresponding to the initial state;
(2) 1000 initial states $\{|0\rangle_p |j/1000\rangle_c, j=1,...,1000\}$ are chosen for comparison.
Generate the hash values corresponding to the initial states;
(3) Compare the two hash values generated by $|0\rangle_p|0\rangle_c$ and $|0\rangle_p |j/1000\rangle_c$. And count the changed bits called $B_i$.

The diffusion and confusion properties of CAQWBH to different initial states are assessed based on the following indicators:
- Minimum changed bit number $B_{min} = \min(\{B_i\})$;
- Maximum changed bit number $B_{max} = \max(\{B_i\})$;
- Mean changed bit number $\overline{B} = \sum_{i=1}^{T} B_i / T$;
- Mean changed probability $P = (\overline{B}/256) \times 100\%$;
- Standard deviation of the changed bit number
  $\Delta B = \sqrt{\frac{1}{T-1} \sum_{i=1}^{T} (B_i - \overline{B})^2}$;
- Standard deviation of the changed probability
  $\Delta P = \sqrt{\frac{1}{T-1} \sum_{i=1}^{T} (B_i / 256 - P)^2} \times 100\%$.

The diffusion and confusion tests to different initial states are shown in Table 2. We concluded from the tests that the mean changed bit number $\overline{B}$ 127.40 and the mean changed probability $P$ 49.76 are close to the ideal value 128 and 50%, respectively. $\Delta B$ and $\Delta P$ are little enough. $B_{min} = 104$ and $B_{max} = 151$ are around 128. That demonstrates the stability of diffusion and confusion.

### 5.1.3 Uniform distribution on hash space

In order to check the distribution capacity of a hash value to different initial states in hash space, 1000 hash values are generated according to 1000 initial states $\{|0\rangle_p |j/1000\rangle_c, j=1,...,1000\}$. Then count the number of the bits at each location. The statistical results are shown in Fig. 3. The mean of the bit number 498.77 is very close to 500, which accounts for half of the test times. That means the hash value is distributed uniformly in the hash space and the statistical property of the hash value is similar to that of a random number string.

### 5.2 Statistical performance analysis for message

The initial state is $|0\rangle_p|0\rangle_c$ in this part. The original message is 43652 bits. The results show that CAQWBH has outstanding statistical performance for different messages.
5.2.1 Sensitivity of hash value

Let $Mes_1$ be the original message. $Mes_2$, $Mes_3$, $Mes_4$ represent the messages with tiny modifications of $Mes_1$. The results listed below show the high sensitivity to the message and its tiny changes.

Condition 1: Choose an original message $Mes_1$;
Condition 2: Flip a bit of $Mes_1$ at a random position and then obtain the modified message $Mes_2$;
Condition 3: Delete a bit from $Mes_1$ at a random position and then obtain the modified message $Mes_3$;
Condition 4: Insert a random bit into $Mes_1$ at a random position and then obtain the modified message $Mes_4$;

The sensitivity of the hash value to the message is assessed by comparing the hash values of the modified messages with that of the original one. And the corresponding 256-bit hash values in the hexadecimal format are given by:

$C_1$: FD*A24DD10C2F861F6504B60DFCF88B27
2CAEE03DF95A1C226FF8BBC4BC5F8A6F
$C_2$: B5117570E60DDC769DC647BA12F74EA0
3F0D32A549E242A746FF2F79F0C72C9F
$C_3$: 4DC603BCA1229BC4228C410551AA4CF6
11D040AB29E4F924989551FE51C891AE
$C_4$: 1FE7E556C31352253CF30C2A47845F82
FAF21EA27A09D32BE0410B322102C5DB

The plots of the hash values in the binary format are shown in Fig. 4. It clearly indicates that any tiny modification to the message will cause a substantial change in the final hash value. A similar result can be obtained using any other instances of
CAQWBH. Therefore, the output hash value of CAQWBH is highly sensitive to its input message.

5.2.2 Statistical analysis of diffusion and confusion

The diffusion and confusion tests are performed as follows:

1. Choose an original message $Mes_1$ and generate the corresponding hash value;
2. Flip a bit of $Mes_1$ at a random position to obtain $Mes_2$ and generate a new hash value;
3. Compare the two hash values and count the number of changed bits at the same location called $B_i$;
4. Repeat steps (1) to (3) $T$ times.

The diffusion and confusion properties of CAQWBH are assessed based on the following indicators:

- Minimum changed bit number $B_{\text{min}} = \min\{\{B_i\}_i^T\}$;
- Maximum changed bit number $B_{\text{max}} = \max\{\{B_i\}_i^T\}$;
- Mean changed bit number $\overline{B} = \frac{\sum_{i=1}^{T} B_i}{T}$;
- Mean changed probability $P = (\overline{B}/256) \times 100\%$;
- Standard deviation of the changed bit number $\Delta B = \sqrt{\frac{1}{T-1} \sum_{i=1}^{T} (B_i - \overline{B})^2}$;
- Standard deviation of the changed probability $\Delta P = \sqrt{\frac{1}{T-1} \sum_{i=1}^{T} (B_i/256 - P)^2} \times 100\%$.

The diffusion and confusion tests are performed $T = 10000$ times as shown in Table 3. We concluded from the tests that the mean changed bit number $\overline{B}$ 128.11 and the mean changed probability $P$ 50.04 are close to the ideal value 128 and 50\%, respectively. $\Delta B$ and $\Delta P$ are very little. $B_{\text{min}} = 102$ and $B_{\text{max}} = 160$ are around 128.
Table 3  Static Number of Changed Bit $B$

| $T=10000$ | $\bar{B}$ | $P\,(\%)$ | $\Delta B$ | $\Delta P$ | $B_{\text{min}}$ | $B_{\text{max}}$ |
|-----------|-----------|-----------|-----------|-----------|----------------|----------------|
| CAQWBH-256 | 128.11 | 50.04 | 8.01 | 3.13 | 102 | 160 |

Fig. 5  Uniform Distribution on Hash Space

That demonstrates the stability of diffusion and confusion. The excellent statistical effect ensures that it is impossible to forge plaintext-cipher pairs given known plaintext-cipher pairs.

5.2.3 Uniform distribution on hash space

In order to check the distribution capacity in hash space, we generated two hash values according to the method described in the previous subsection and then counted the number of the changed bits at each location. The statistical results for $T = 10,000$ are shown in Fig. 5. The mean of the changed bit number 4995.40 is very close to 5000, which accounts for half of the test times. That means the hash value is distributed uniformly in the hash space and the statistical property of the hash value is similar to that of a random number string. Obviously, this demonstrates the resistance against statistical attacks.

5.2.4 Collision analysis

It is hard to provide a mathematical proof of the capability of collision resistance of chaotic hash functions. Thus, we performed tests for analyzing collision resistance. Let $W_T(\omega)$ be the number of draws on which the hash values of the original and modified message contain $\omega$ bytes with the same value at the same location. It is a common indicator of the collision resistance property. If the experimental result of
Table 4 Comparison of Experimental Values and Theoretical Values of $W_T (\omega)$

| $\omega$ | Experimental Values of $W_T (\omega)$ | Theoretical Values of $W_T (\omega)$ |
|---------|--------------------------------------|--------------------------------------|
| 0       | 8856                                 | 8822.81                              |
| 1       | 1077                                 | 1107.20                              |
| 2       | 63                                   | 67.30                                |
| 3       | 2                                    | 2.64                                 |
| 4       | 0                                    | 0.08                                 |
| 5⋅⋅⋅32  | 0                                    | 0                                    |

$W_T (\omega)$ is very close to the theoretical value, then the hash function could be regarded as having a good property of collision resistance.

The collision tests are performed as follows:

1. Choose an original message $Mes_1$ and generate the corresponding hash value in byte format;
2. Flip a bit of $Mes_1$ at a random position to obtain $Mes_2$ and generate a new hash value in byte format;
3. Compare the two hash values and count the number of the same bytes at the same location;
4. Repeat steps (1) to (3) $T$ times.

Let $\omega$ be the number of byte formats with the same value at the same location.

$$\omega = \sum_{i=1}^{T} \delta(e_i - e'_i)$$  \hspace{1cm} (10)

where $\delta(x)$ is the Dirac delta function. $e_i$ and $e'_i$ are the $i$th entries of the original and the new hash value in byte format, respectively.

And the theoretical number of $\omega$ same values through $T$ independent tests $W_T (\omega)$ can be computed according to the following formulas:

$$W_T (\omega) = T \times Prob[\omega] = \frac{n!}{\omega!(n-\omega)!} \left( \frac{1}{2^8} \right)^{\omega} \left( 1 - \frac{1}{2^8} \right)^{n-\omega}$$  \hspace{1cm} (11)

where $\omega = 0, 1, \cdots, n$. And $n = 256/8 = 32$ in CAQWBH-256. After running the above tests, the experimental values and the experimental values of $W_T (\omega)$ in the proposed hash function are shown in Table 4. The experimental values of $W_T (\omega)$ are similar to the theoretical values.

5.2.5 Resistance to birthday attack

If the initial state is private, any unauthorized party cannot generate the hash value without the initial state. Therefore, the birthday attack can only threaten the situation that the initial state is public.

The birthday attack implies a lower bound of the length of hash value. The length of the hash value we considered here is $256 = 32 \times 8$ bits. Therefore, it needs $2^{n/2} = 2^{128} \approx 3.4028 \times 10^{38}$ trials ($n$ is the size of hash value) to find two messages with identical hash values with a probability of 1/2 with a classical computer. Even
with a quantum computer, it needs $2^{n/3} \approx 2^{85.33} \approx 4.8741 \times 10^{25}$ trials. Furthermore, CAQWBH-256 can be easily extended to be 512 bits or more. Therefore, the results of the tests, the size of the hash value, and the collision resistance of the proposed CAQWBH suggest that the birthday attack is almost impossible and that the proposed algorithm is resistant against to this type of attack.

In conclusion, CAQWBH has outstanding statistical performance. Compare with other CQW-based quantum hash functions, there is no apparent distinction between them. If the differences of statistical performance have to be taken into account, the most important factor to the statistical performance is the length of hash value rather than the varieties of CQW-based quantum hash function.

6 Conclusion

The CQW-based quantum hash function is a kind of novel hash function, which is based on a general quantum computation model, i.e., quantum walk. It is safe, flexible, and highly efficient. Furthermore, the CQW-based quantum hash function is compatible to be executed on a quantum computer or a classical computer.

In this paper, we focus on how to improve the efficiency of the CQW-based quantum hash function further. All existing CQW-based quantum hash functions are controlled by one bit of message in each step. CAQWBH is presented by using the time-position-dependent CAQW on the complete graph with self-loops to process the message in batch amounts. CAQW is chosen for decreasing the dimension of the coin state. The complete graph with self-loops is chosen for constructing hash functions that satisfy the avalanche characteristic. It is shown in table 1 that CAQWBH accelerates the hash processing dramatically by processing $N$ bits in each step. For example, CAQWBH-256 can process 32 bits in each step.

Besides, CAQWBH inherits all advantages of the CQW-based quantum hash function. CAQWBH is flexible so that the hash value of different lengths is easy to attain by changing the parameters of the graph and the post-processing, i.e., $N$ and $k$. Furthermore, CAQWBH is a compatible hash function that can be executed on a quantum or classical computer. Considering the difficulty of the commercialization of quantum computers, CAQWBH is practical even in current hardware technology. Furthermore, the irreversible measurement and the modular arithmetic make sure that CAQWBH is extremely safe. Statistical analysis proves the claim too.

In summary, CAQWBH retains the superiorities of the CQW-based quantum hash function in security, flexibility, and compatibility, while speeding up its efficiency.

Acknowledgements This work is supported by the Fundamental Research Funds for the Central Universities (Grant No. NS2023047), the National Key Research and Development Program of China (Grant No. 2020YFB1005500), NSFC (Grant Nos. 62101270, 61901218, 61802025, 62071015), National Key Research and Development Program of China (Grant No.2020YFB1005504).

Data Availability The datasets generated during and/or analyzed during the current study are available from https://pan.baidu.com/s/1LrWGpPokJozcD-9OEubseQ?pwd=0gm8.
Declarations

Conflict of interest We declare that we have no known competing interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Yang, Y.G., Lv, X.L., Gao, S., et al.: Detector-device-independent quantum key agreement based on single-photon Bell state measurement. Int. J. Theor. Phys. 61, 50 (2022)
2. Keshavarzian, P., Ramu, K., Tang, D., et al.: A 3.3-Gb/s SPAD-based quantum random number generator. IEEE J. Solid-state Circuits (2023). https://doi.org/10.1109/JSSC.2023.3274692
3. Li, D., et al.: Generic quantum walks with memory on regular graphs. Phys. Rev. A 93, 042323 (2016)
4. Li, D., et al.: Szegedy quantum walks with memory on regular graphs. Quantum Inf. Process. 19, 32 (2020)
5. Li, D., et al.: Quantum walks with memory provided by parity of memory. Int. J. Theor. Phys. 59, 1934–1943 (2020)
6. Dai, W.J., et al.: Discrete-time quantum walk on the Cayley graph of the dihedral group. Quantum Inf. Process. 17, 330 (2018)
7. Liu, Y., et al.: Quantum walks on regular uniform hypergraphs. Sci. Rep. 8, 9548 (2018)
8. Li, D., et al.: Discrete-time interacting quantum walks and quantum Hash schemes. Quantum Inf. Process. 12, 1501–1513 (2013)
9. Li, D., et al.: Analysis of the two-particle controlled interacting quantum walks. Quantum Inf. Process. 12, 2167–2176 (2013)
10. Yang, Y.G., et al.: 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)
11. Cao, W.F., et al.: Constructing quantum Hash functions based on quantum walks on Johnson graphs. Quantum Inf. Process. 17, 156 (2018)
12. Yang, Y.G., et al.: Simple hash function using discrete-time quantum walks. Quantum Inf. Process. 17, 189 (2018)
13. Yang, Y.G., et al.: Hash function based on quantum walks. Int. J. Theor. Phys. 58, 1861–1873 (2019)
14. Yang, Y.G., et al.: Usefulness of decoherence in quantum-walk-based hash function. Int. J. Theor. Phys. 60, 1025–1037 (2021)
15. Li, D., et al.: Controlled alternate quantum walks based quantum hash function. Sci. Rep. 8, 225 (2018)
16. AbdEL-Latif, A.A., et al.: Controlled alternate quantum walk-based pseudo-random number generator and its application to quantum color image encryption. Physica A 547, 123869 (2020)
17. Yan, T., Li, D.: A novel color image encryption scheme based on controlled alternate quantum walks and DNA sequence operations. In: Proceedings of the Machine Learning for cyber security, pp. 297–306 (2020)
18. Yan, T., Li, D.: A novel quantum color image encryption scheme based on controlled alternate quantum walks. In: Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, pp. 519–530 (2020)
19. Gao, Y.J., et al.: A novel quantum image encryption technique based on improved controlled alternated quantum walks and hyperchaotic system. Physica A Stat. Mech. Appl. 598, 127334 (2022)
20. Zhou, Q., Lu, S.F.: Hash function based on controlled alternate quantum walks with memory. IEEE Trans. Quantum Eng. 3, 1–10 (2022)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.