IEEE Transactions on Quantum Engineering

Received September 19, 2021; revised November 12, 2021; accepted November 18, 2021; date of publication November 24, 2021; date of current version January 5, 2022.

Digital Object Identifier 10.1109/TQE.2021.3130256

Hash Function Based on Controlled Alternate Quantum Walks With Memory (September 2021)

QING ZHOU[1] **AND SONGFENG LU**[1,2]

[1]School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[2]Shenzhen Huazhong University of Science and Technology Research Institute, Shenzhen 518057, China

(Corresponding author: Songfeng Lu.)

This work was supported in part by the National Natural Science Foundation of China under Grant 62101197, in part by the China Postdoctoral Science Foundation under Grant 2021M691148, in part by the Key Research and Development Plan of Hubei Province of China under Grant 2021BAA038, and in part by the Project of Science, Technology and Innovation Commission of Shenzhen Municipality of China under Grant JCYJ20210324120002006.

ABSTRACT We propose a Quantum inspired Hash Function using controlled alternate quantum walks with Memory on cycles (QHFM), where the jth message bit decides whether to run quantum walk with one-step memory or to run quantum walk with two-step memory at the jth time step, and the hash value is calculated from the resulting probability distribution of the walker. Numerical simulation shows that the proposed hash function has near-ideal statistical performance and is at least on a par with the state-of-the-art hash functions based on quantum walks in terms of sensitivity of hash value to message, diffusion and confusion properties, uniform distribution property, and collision resistance property; and theoretical analysis indicates that the time and space complexity of the new scheme are not greater than those of its peers. The good performance of QHFM suggests that quantum walks that differ not only in coin operators but also in memory lengths can be combined to build good hash functions, which, in turn, enriches the construction of controlled alternate quantum walks.

INDEX TERMS Controlled alternate quantum walks (CAQW), hash function, quantum walks with memory (QWM), statistical properties, time and space complexity.

I. INTRODUCTION

As one of the principal tools of information security, cryptographic hash functions not only act as essential components of identification, message authentication, digital signatures, and random number generation, but also play an important part in privacy amplification process of quantum key distribution [1]. Classical hash functions based on hard computational problems are, however, subject to an inherent security limitation: the existence of one-way functions is still an open conjecture that cannot be proved (a proof, with no assumptions, of existence would establish P $\neq$ NP [2]). As a result, they only satisfy computational security and are challenged by cryptanalysis equipped with quantum algorithms. Such a fact stimulates researchers to develop hash functions with a higher level of security, such as hash functions based on (or inspired by) quantum computing [3]–[16], whose preimage resistance property is ensured by quantum mechanics rather than hardness assumptions.

There are two kinds of quantum-computing-based hash functions: classical-quantum hash functions based on quantum one-way functions (QOWF) [3]–[8] (hereafter, simply QOWF-based hash functions) and classical-classical hash functions based on discrete quantum walks (QW) on cycles [9]–[16] (hereafter, simply QW-based hash functions). The former have balanced one-way resistance property and collision resistance property that are well-defined and strictly proved in the quantum setting, the latter take advantage of the chaotic characteristics of quantum walks and belong to dedicated hash functions, whose capabilities of collision resistance are difficult to prove and are mainly assessed by means of statistical analysis. On the other hand, for QOWF-based hash functions, the output length and the number of hashing parameters are both positively correlated with the input size, while QW-based hash functions map messages of arbitrary finite length to digests of fixed length. In addition, the output length of QW-based hash functions can be easily extended (to withstand brute-force attacks) by increasing the number of nodes of the cycle or the number of hash bits "contributed" by each node, and the hash result is calculated classically. Thus, QW-based hash functions are of greater practical utility before large-scale quantum computers are built, for they can currently be used to improve the security of hash-function-based schemes.

The essence of the design of QW-based hash functions is combining two or more different quantum walk procedures governed by evolution operators $\{U_0, U_1, \ldots\}$ to construct a controlled alternate quantum walk (CAQW) model, where the choice among $\{U_0, U_1, \ldots\}$ at the jth time step is determined by the jth bit of a binary string. Theoretically, a valid CAQW model could be constructed if the walker can "switch" freely among $\{U_0, U_1, \ldots\}$, and evolution operators that only differ in coin transform naturally satisfy this requirement. Therefore, various quantum walks, such as one-dimensional broken-line quantum walks [9], one-dimensional one-particle quantum walks [10]–[12], two-dimensional one-particle quantum walks [13], quantum walks on Johnson graphs [14], one-dimensional two-particle (interacting) quantum walks [15], [16], and one-dimensional quantum walks with memory (QWM) [17]–[22] can all be used to construct valid (but may not good) hash functions as long as they are modified to utilize coin operators controlled by input messages. Among these walks, the evolution of QWM is governed by the following three (rather than two) stages: flipping a coin, determining the next direction according to the coin state and the previous direction(s), and moving a step according to the new direction. Here, an extra operator—the direction-determine transform—could be taken into account when designing hash functions based on QWM: the alternately performed evolution operators can differ in coin transform or direction-determine transform, or both.

To examine the feasibility and utility of this idea, we combine two quantum walks with different memory lengths, i.e., QW1M [22] and QW2M [19], to achieve a valid CAQW process, for different quantum walks with unequal memory lengths typically have different direction-determine transforms, and they can also use different coin operators. Based on this walking process, we construct a new hash function (named QHFM) and then assess its performance. Simulation results show that the statistical properties of the proposed hash function are as good as those of the existing QW-based hash functions, and theoretical analysis indicates that the space and time complexity of QHFM are not greater than those of theirs peers.

II. ONE-DIMENSIONAL CONTROLLED QUANTUM WALK WITH ONE- AND TWO-STEP MEMORY (CQWM)

A one-dimensional CQWM takes place in the Hilbert space $\mathcal{H}_p \otimes \mathcal{H}_{\mathrm{dr}_2} \otimes \mathcal{H}_{\mathrm{dr}_1} \otimes \mathcal{H}_c$ spanned by vectors $|x, \mathrm{dr}_2, \mathrm{dr}_1, c\rangle$, where c (with $c \in \{0, 1\}$, or $c \in \mathbb{Z}_2$) is the coin state, dr_1 (with $\mathrm{dr}_1 \in \mathbb{Z}_2$) is the direction of the most recent step (0 stands for left and 1 stands for right), dr_2 (with $\mathrm{dr}_2 \in \mathbb{Z}_2$) is the direction of the penultimate step, and x is the current position. If the walker moves on a line, then $x \in \mathbb{Z}$ (all integers); and if the walker moves on a cycle with n nodes, then $x \in \{0, 1, 2, \ldots, n-1\}$ (or $x \in \mathbb{Z}_n$).

Formally, the evolution of CQWM controlled by a t-bit string $\mathrm{msg} = (m_1, m_2, \ldots, m_t) \in \{0, 1\}^t$ is the product of t unitary transforms

$$U_{\mathrm{msg}} = U^{(m_t)} U^{(m_{t-1})} \cdots U^{(m_2)} U^{(m_1)}. \tag{1}$$

Here, $U^{(m_j)}$ $(j = 1, 2, \ldots t)$ is the one-step transform controlled by the jth bit of msg, and it is defined as

$$U^{(m_j)} = S \cdot \left(I_n \otimes D^{(m_j)}\right) \cdot \left(I_{4n} \otimes C^{(m_j)}\right) \tag{2}$$

where $C^{(m_j)}$ is a 2×2 coin operator controlled by m_j, I_k $(k = 4n$ or $n)$ is a $k \times k$ identity operator, $D^{(m_j)}$ is an 8×8 direction-determine operator controlled by m_j, and S is the conditional shift operator controlled by the next direction. If $m_j = 0$, then $C^{(m_j)}$ is parameterized by an angle θ_0, i.e.,

$$C^{(0)} = \begin{pmatrix} \cos(\theta_0) & \sin(\theta_0) \\ \sin(\theta_0) & -\cos(\theta_0) \end{pmatrix} \tag{3}$$

and $D^{(m_j)}$ (becomes $D^{(0)}$) describes the direction-determine process of QW1M; if $m_j = 1$, then $C^{(m_j)}$ is parameterized by another angle θ_1, i.e.,

$$C^{(1)} = \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \sin(\theta_1) & -\cos(\theta_1) \end{pmatrix} \tag{4}$$

and $D^{(1)}$ describes the direction-determine process of QW2M.

The direction-determine transforms of QW1M [22] and QW2M [19] can be, respectively, written as $\mathrm{DT}_0 : |\mathrm{dr}_1, c\rangle \to |\mathrm{dr}_1 \oplus \bar{c}, c\rangle$ and $\mathrm{DT}_1 : |\mathrm{dr}_2, \mathrm{dr}_1, c\rangle \to |\mathrm{dr}_1, \mathrm{dr}_2 \oplus \bar{c}, c\rangle$, where $\bar{c} \equiv 1 \oplus c$, $\mathrm{dr}_1 \oplus \bar{c}$ in the first expression specifies the next direction of the walker performing QW1M, and $\mathrm{dr}_2 \oplus \bar{c}$ in the second expression specifies the next direction of the walker performing QW2M. To enable QW1M and QW2M to be performed alternately, one may add a redundant state $|\mathrm{dr}_2\rangle$ into QW1M and let $D^{(0)} : |\mathrm{dr}_2, \mathrm{dr}_1, c\rangle \to |\mathrm{dr}_2, dr_1 \oplus \bar{c}, c\rangle$ determines the next direction when the controlling bit equals 0; otherwise, the next direction is determined by $D^{(1)} = \mathrm{DT}_1$.

According to [19], any 4-term basis state $|x, \mathrm{dr}_2, \mathrm{dr}_1, c\rangle$ in $\mathcal{H}_p \otimes \mathcal{H}_{\mathrm{dr}_2} \otimes \mathcal{H}_{\mathrm{dr}_1} \otimes \mathcal{H}_c$ can be rewritten as a 2-term basis state $|x, j\rangle = |x, 2^2\mathrm{dr}_1 + 2^1\mathrm{dr}_2 + 2^0 c\rangle$ in $\mathcal{H}_p \otimes \mathcal{H}^8$, where $\mathcal{H}^8$ is the 8-dimensional Hilbert space. Conversely, from any 2-term basis state $|x, j\rangle$ $(j \in \mathbb{Z}_8)$, one can deduce the coin value and the most recent two directions as follows:

$$\begin{cases} c = j \bmod 2 \\ \mathrm{dr}_2 = (j \bmod 4 - j \bmod 2)/2 \\ \mathrm{dr}_1 = (j - j \bmod 4)/4. \end{cases} \tag{5}$$

According to this correspondence, $D^{(0)}$ can be reformulated to $|2^2\mathrm{dr}_1 + 2^1\mathrm{dr}_2 + 2^0 c\rangle \to |2^2(\mathrm{dr}_1 \oplus \bar{c}) + 2^1\mathrm{dr}_2 +$

$2^0c\rangle$, or, under the 2-term states

$$D^{(0)}: |j\rangle \to |j_a + j_b\rangle$$
$$j_a = 4\left[(j - j \bmod 4)/4 \oplus (\overline{j \bmod 2})\right]$$
$$j_b = j \bmod 4. \tag{6}$$

Analogously, $D^{(1)}$ can be expressed as

$$D^{(1)}: |j\rangle \to |j'_a + j'_b\rangle$$
$$j'_a = 4\left[(j \bmod 4 - j \bmod 2)/2 \oplus (\overline{j \bmod 2})\right]$$
$$j'_b = (j - j \bmod 4)/2 + j \bmod 2. \tag{7}$$

With (6) and (7), one can verify that $D^{(0)}$ and $D^{(1)}$ are both unitary.

Once the next direction, the new dr_1, is determined, the walker then moves according to the shift operator controlled by dr_1. If the walk takes place on a line, then the action of S is expressed as $|x, \mathrm{dr}_2, \mathrm{dr}_1, c\rangle \to |x + 2\mathrm{dr}_1 - 1, \mathrm{dr}_2, \mathrm{dr}_1, c\rangle$; if the walk takes place on a cycle with n nodes, then S becomes $|x, \mathrm{dr}_2, \mathrm{dr}_1, c\rangle \to |x + 2\mathrm{dr}_1 - 1 \pmod n, \mathrm{dr}_2, \mathrm{dr}_1, c\rangle$, which can be reformulated (in 2-term states) to

$$S: |x, j\rangle \to |x + (j - j \bmod 4)/2 - 1 \pmod n, j\rangle. \tag{8}$$

In (8), the next position is calculated using modular arithmetic under modulus n.

III. HASH FUNCTION USING QUANTUM WALKS WITH ONE- AND TWO-STEP MEMORY ON CYCLES

The proposed hash function is constructed by running CQWM on a circle with n nodes under the control of the input message msg, where each node contributes m bits to the hash result $H(\mathrm{msg})$. The process of CQWM-based hash function is described as follows.

1) Select the values of parameters $(n, m, l, \theta_0, \theta_1, \alpha)$ satisfying the following constraints: n is odd; $n \times m$ equals the bit length of the hash value; $10^l \gg 2^m$; and $\theta_0, \theta_1, \alpha \in (0, \pi/2)$.
2) Initialize the walker in the state $|\psi_0\rangle = \cos\alpha|0, 1, 0, 0\rangle + \sin\alpha|0, 1, 0, 1\rangle$ (or, in the 2-term state $|\psi_0\rangle = \cos\alpha|0, 2\rangle + \sin\alpha|0, 3\rangle$).
3) Apply U_{msg} to $|\psi_0\rangle$ and generate the resulting probability distribution prob $= (p_0, p_1, \ldots, p_{n-1})$, where p_x $(x \in \mathbb{Z}_n)$ is the probability that the particle locates at node x when the walk is finished.
4) The hash value of msg is a sequence of n blocks $H(\mathrm{msg}) = B_0 \| B_1 \| \ldots \| B_{n-1}$, where each block B_x is the m-bit binary representation of $\lfloor p_x \cdot 10^l \rfloor \bmod 2^m$ ($\lfloor\cdot\rfloor$ denotes the floor of a number), and $B_x \| B_{x+1}$ denotes the concatenation of B_x and B_{x+1}.

IV. STATISTICAL PERFORMANCE ANALYSIS

QHFM, like other QW-based hash functions [9]–[16], belongs to dedicated hash functions, whose performances are mainly evaluated through statistical analysis. To make our statistical tests reusable and usable by anyone else, we perform these tests on a collection of items (i.e., input messages) randomly drawn from an open dataset, named "arXiv Dataset," of about 1.8 million records and upload the complete MATLAB code for hash tests to "GitHub." One can download the dataset from https://www.kaggle.com/Cornell-University/arxiv and get the test code for QHFM from https://github.com/ChloeZhouqing/Hash-functions-based-on-quantum-walks.

TABLE I **Values of Parameters Chosen for the Seven Instances of the Proposed Hash Scheme**

Hash Instances	n	m
QHFM-296	37	8
QHFM-264	33	8
QHFM-221	17	13
QHFM-200	25	8
QHFM-195	15	13
QHFM-136	17	8
QHFM-120	15	8

To make comparisons between the proposed scheme and the existing ones with (detailed) experimental results [9]–[15] in a fair and informative manner, we consider seven "instances" QHFM-L ($L = 296, 264, 221, 200, 195, 136, 120$) of QHFM, where QHFM-$L$ produces L-bit hash values and will be compared with the existing QW-based hash functions with L- or close-to-L-bit output length (QHFM-136 and QHFM-120 will be compared with the 128-bit scheme in [15]). Different instances of QHFM share the same l values, same θ_0 values, same θ_1 values, and the same α values, which are taken to be $8, \pi/4, \pi/3$, and $\pi/4$, respectively. Distinction between QHFM-L and QHFM-L' $(L \neq L')$ lies in the values of n and m, which are listed in Table I.

A. SENSITIVITY OF HASH VALUE TO MESSAGE

Let msg_0 be an original message and msg_j $(j \in \{1, 2, 3\})$ the slightly modified result of msg_0, which are obtained under the following four conditions.

- Condition 1: Randomly choose an original message msg_0.
- Condition 2: Flip a bit of msg_0 at a random position and then obtain the modified message msg_1.
- Condition 3: Insert a random bit into msg_0 at a random position and then obtain msg_2.
- Condition 4: Delete a bit from msg_0 at a random position and then obtain msg_3.

The sensitivity of hash value to message is assessed by comparing the hash values $H(\mathrm{msg}_j)$ of the modified messages with the hash value $H(\mathrm{msg}_0)$ of the original one. In our sensitivity test, a record is randomly picked out from the arXiv dataset, then the article abstract within this record serves as msg_0.

Corresponding to the conditions mentioned above, four hash values in hexadecimal format produced by QHFM-195 are obtained as follows.

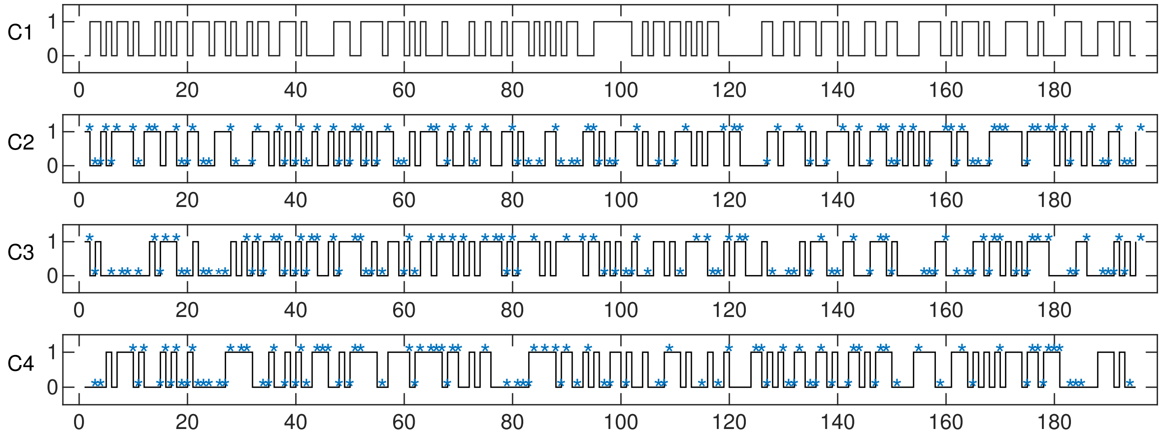

FIG. 1. (Color online) Plots of the hash values produced by QHFM-195 under the four conditions, where Cj stands for Condition j ($j = 1, 2, 3, 4$).

- Condition 1: $H(\text{msg}_0)$ ="3 5 A 2B 76 96 74 1 C F7 51 09 2E AB 1F CB 6 A C0 33 77 46 61 E5 D1 E4 38 EC".
- Condition 2: $H(\text{msg}_1)$ ="4 BC EC C7 0E A9 2B 5 C 5C 93 34 30 69 E9 3 A EC 1B D3 D3 95 7B 0F DF 5 A 31".
- Condition 3: $H(\text{msg}_2)$ ="5 00 5 C 40 AB AB 2F 26 9B AB D7 AF B5 23 4F 16 20 5 C 63 A0 30 6D 5E 0 C 15".
- Condition 4: $H(\text{msg}_3)$ ="0 5D 14 81 F1 29 CB E7 BE CB 01 F6 53 48 E8 90 D4 CD 35 C3 C7 55 DB 80 E8".

Notice that the first hexadecimal digit of the hash value under condition j only represents the first three (rather than four) bits of $H(\text{msg}_j)$, since the output length of QHFM-195 is not a multiple of four.

The plots of hash values $H(\text{msg}_0)$, $H(\text{msg}_1)$, $H(\text{msg}_2)$, and $H(\text{msg}_1)$ in binary format are shown in Fig. 1, where each asterisk (*) in the jth subgraph ($j > 1$) marks a different bit between $H(\text{msg}_{j-1})$ and $H(\text{msg}_0)$. Fig. 1 indicates that a tiny modification to the message could cause a significant change in the hash value, and the positions of those changed bits are evenly distributed over the entire interval [1,195] of position numbers. A similar result can be obtained using any other instance of QHFM; thus, the output digest of the proposed hash scheme is highly sensitive to its input message.

B. DIFFUSION AND CONFUSION PROPERTIES

The test data for the diffusion and confusion properties of QHFM-L are collected by making N random draws (with replacement) from the arXiv Dataset. On each draw, an original message msg$_0$ is selected, then a slightly modified result msg$_1$ of this message is obtained by inverting a bit of msg$_0$ at a random position. Let B_i be the Hamming distance between the hash values of the original and modified messages obtained on the ith draw and N is the number of draws, the diffusion and confusion properties (reflecting the avalanche effect) of the proposed hash instances are assessed based on the following four indicators:

1) mean changed bit number $\overline{B} = \sum_{i=1}^{N} B_i/N$;
2) mean changed probability $P = \overline{B}/(n \times m) \times 100\%$;
3) standard deviation of the changed bit number $\Delta B = \sqrt{\sum_{i=1}^{N}(B_i - \overline{B})^2/(N-1)}$;
4) standard deviation of the changed probability $\Delta P = \sqrt{\sum_{i=1}^{N}[B_i/(n \times m) - P]^2/(N-1)} \times 100\%$.

TABLE II Diffusion-and-Confusion-Test Results for the Proposed and Existing QW-Based Hash Functions

Hash Instances or Schemes	$\overline{B}$	$P(\%)$	ΔB	$\Delta P(\%)$	$I_{\text{DC}}(\%)$
QHFM-296	147.9101	49.9696	8.5997	2.9053	1.4679
QHFM-264	131.8667	49.9495	8.1378	3.0825	1.5665
QHFM-221	110.5313	50.0142	7.4455	3.3690	1.6916
QHFM-200	100.0205	50.0103	7.1654	3.5827	1.7965
QHFM-195	97.5591	50.0303	6.9844	3.5817	1.8060
QHFM-136	68.0530	50.0390	5.8782	4.3222	2.1806
QHFM-120	60.0914	50.0762	5.4699	4.5582	2.3172
Yang21-296 [9]	147.8640	49.9541	8.6141	2.9102	1.4781
Yang19-264 [10]	131.6803	49.8789	8.8877	3.3666	1.7439
Yang18-264 [11]	132.1108	50.0420	8.0405	3.0457	1.5439
Yang18-221 [12]	112.7791	51.0313	8.2029	3.7117	2.3715
Li18-200 [13]	99.9010	49.9505	7.1133	3.5567	1.8031
Cao18-195 [14]	124.7000	63.9600	6.4300	6.3000	10.1300
Yang16-128 [15]	64.2894	50.2261	5.6686	4.4286	2.3274

The ideal values of $\overline{B}$ and P are $(n \times m)/2$ and 50%, respectively; and smaller standard deviations (ΔB and ΔP) are more desirable. For a specific hash function with fixed output length, $\overline{B}$ and ΔB are directly proportional to P and ΔP, respectively; thus, only P and ΔP, or a combination of them, e.g., $I_{\text{DC}} = (\Delta P + |P - 50\%|)/2 \times 100\%$, would suffice to assess the confusion and diffusion properties of this hash function: the smaller I_{DC}, the better the avalanche effect achieved. The diffusion and confusion test on QHFM-L is performed with $N = 10\,000$, and the simulation results are presented in Table II. For comparison, the reported results (with $N \geq 10\,000$) of the corresponding variables for the existing QW-based hash schemes [9]–[15] are also listed in the same table, where the values for Yang21-296 and Yang18-221 (which have multiple instances) are the test results for the representative instances of them. Here, the representative instance of a hash scheme X is the one that has the P result closest to 50% (over all instances of X).

The values of I_{DC} suggest that the test results for QHFM-264 is better than that for Yang19-264 but slightly poorer than that for Yang18-264, and the results for other instances of the proposed hash scheme are better than those for their peers (QHFM-296 versus Yang21-296; QHFM-221 versus Yang18-221; QHFM-200 versus Li18-200; QHFM-195 versus Cao18-195; QHFM-136 and QHFM-120 versus Yang16-128). Thus, the diffusion and confusion properties of the proposed hash function outperform or are at least on a par with the existing QW-based hash schemes.

C. UNIFORM DISTRIBUTION ANALYSIS

Similar to the case of diffusion and confusion properties, the uniform distribution property (which reflects the strict avalanche effect) could also be assessed based on the following four indicators:

1) mean number of draws with flipped hash bit over $n \times m$ bit positions $\overline{T} = \sum_{j=1}^{n\times m} T_j/(n \times m)$;
2) mean percentage of draws with flipped hash bit (over $n \times m$ bit positions) $Q = \overline{T}/N \times 100\%$;
3) standard deviation of the number of draws with flipped hash bit $\Delta T = \sqrt{\sum_{j=1}^{n\times m}(T_j - \overline{T})^2 \big/ (n \times m - 1)}$;
4) standard deviation of the percentage of draws with flipped hash bit $\Delta Q = \sqrt{\sum_{j=1}^{n\times m}(T_j/N - Q)^2 \big/ (n \times m - 1)} \times 100\%$;

where T_j ($j = 1, 2, \ldots, n \times m$) is the number of draws on which a bit-flip occurs in the hash value at the jth bit position after a random message bit is inverted. The theoretical values of $\overline{T}$ and Q are $(n \times m)/2$ and 50%, respectively.

Since $\overline{T}$ and ΔT are directly proportional to Q and ΔQ, respectively, the uniform distribution property of a hash function could be evaluated using $|Q - 50\%|$ and ΔQ: the smaller they are, the better the strict avalanche effect achieved. Additionally, the experimental value of Q is always equivalent to the value of P if the test data (i.e., N pairs of original and modified messages) used in the diffusion and confusion test are reused in the uniform distribution test. Such a result can also be obtained through a simple reasoning: in $P = \sum_{i=1}^{N} B_i \big/ (n \times m \times N) \times 100\%$ and $Q = \sum_{j=1}^{n\times m} T_j \big/ (n \times m \times N) \times 100\%$, both $\sum_{i=1}^{N} B_i$ and $\sum_{j=1}^{n\times m} T_j$ count the total number of hash bits that are flipped over N draws. Thus, T or Q alone is insufficient for assessing the uniform distribution property of a hash function, it should be considered along with ΔQ.

The uniform distribution test on QHFM-L is conducted as follows.

1) Set $T_j = 0$ for every bit position j in the hash value.
2) Randomly draw an article record from arXiv Dataset, take the abstract of this article as the original message msg_0.
3) Randomly flip a bit of msg_0 and then generate the modified message msg_1.
4) Compute the hash values of the two messages and get the digest pair $(H(\mathrm{msg}_0), H(\mathrm{msg}_1))$; compare $H(\mathrm{msg}_0)$ with $H(\mathrm{msg}_1)$ bit by bit, if $H(\mathrm{msg}_0)$ differs from $H(\mathrm{msg}_1)$ at the jth bit position, then the value of T_j is incremented by one.
5) Repeat steps 2) to 4) N times.
6) Calculate $\overline{T}$, Q, ΔT, and ΔQ from the obtained data.

The data collected in step 2) are reused for different instances of the proposed hash scheme as well as for different hash properties (i.e., the diffusion and confusion properties, the uniform distribution property, and the collision resistant property). As a result, the experimental values of P and Q for each instances are equal, which gives $N \times P = \overline{T}$ and $|P - 50\%| = |Q - 50\%|$. On the other hand, for the existing schemes [9]–[15], the reported results of $\overline{T}$ are not equivalent to the corresponding outcomes of $N \times P$, this is probably because their input messages used in the uniform distribution test are not the same as that used in the diffusion and confusion test. Nevertheless, the reported values of $\overline{T}$ are generally close to the corresponding results of $N \times P$.

Since the test results of ΔQ (or ΔT) for the existing schemes are unavailable for comparison, we collect reported data related to the uniform distribution property from [9]–[15] as much as possible and list the corresponding results for the proposed and existing schemes in Table III, where "****.**, same" denotes a pair of identical values, and "N/A" means "not available." The values of $|Q - 50\%|$ (in %) for the existing schemes are deduced from the reported results of $\overline{T}$: $|Q - 50\%| = |\overline{T}/N \times 100\% - 50\%|$, where $N = 16\,383$ for Cao18-195 and $N = 10\,000$ for others. Similar to Table II, the values presented in the 8th and 11th rows are results for the representative instances of Yang21-296 and Yang18-221, respectively.

It can be seen from the last column of Table III that the experimental values of P and Q for QHFM-221, QHFM-195, and QHFM-136 together with QHFM-120 are closer to their theoretical values than those for Yang18-221, Cao18-195, and Yang16-128, respectively; and the values of $|Q - 50\%|$ for the remaining instances of QHFM are on a par with those for their peers. In addition, the results of ΔQ for all instances of QHFM are very small, indicating that the proposed hash scheme has a very good uniform distribution property.

To provide an intuitive description for this property of our scheme, we plot the number of draws with flipped hash bit on every bit position of QHFM-195 in Fig. 2, which suggests that the proposed scheme has a good resistance to statistical attacks.

D. COLLISION RESISTANCE

The test data for the diffusion and confusion properties or the uniform distribution property can also be used to analyzing the collision resistance property, which is generally assessed in terms of the following two groups of indicators: 1) the

TABLE III Uniform-Distribution-Test Results for the Proposed and Existing QW-Based Hash Functions

Hash Instances or Schemes	$N\times P, \overline{T}$	ΔT	$\Delta Q(\%)$	$\lvert P-50\%\rvert, \lvert Q-50\%\rvert(\%)$
QHFM-296	4996.96, same	48.4334	0.4843	0.0304, same
QHFM-264	4994.95, same	48.9253	0.4893	0.0505, same
QHFM-221	5001.42, same	51.6083	0.5161	0.0142, same
QHFM-200	5001.03, same	51.6897	0.5169	0.0103, same
QHFM-195	5003.03, same	50.5134	0.5051	0.0303, same
QHFM-136	5003.90, same	46.6002	0.4660	0.0390, same
QHFM-120	5007.62, same	48.6068	0.4861	0.0762, same
Yang21-296 [9]	4995.41, 4998.1	N/A	N/A	0.0459, 0.019
Yang19-264 [10]	4987.89, 4996.6	N/A	N/A	0.1211, 0.034
Yang18-264 [11]	5004.20, 5003.9	N/A	N/A	0.0420, 0.039
Yang18-221 [12]	5103.13, N/A	N/A	N/A	1.0313, N/A
Li18-200 [13]	4995.05, 4998.2	N/A	N/A	0.0495, 0.018
Cao18-195 [14]	10478.57, 6495.0	N/A	N/A	13.9600, 10.355
Yang16-128 [15]	5022.61, 4973.5	N/A	N/A	0.2261, 0.265

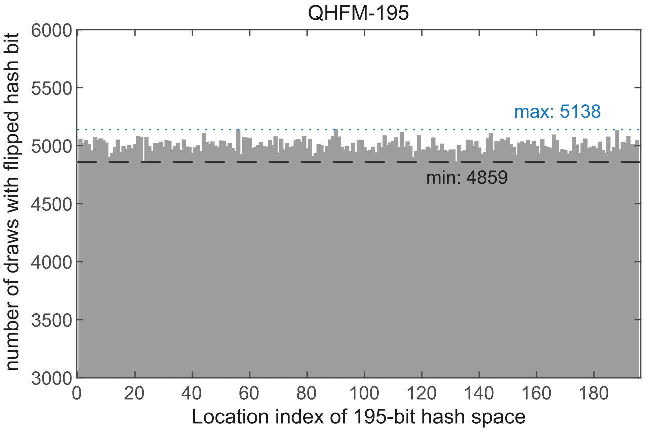

FIG. 2. Histogram of the 195-bit hash space, with $N = 10\,000$.

number of draws $W_N^e(\omega)$ (out of N random selections) on which the hash values of the original and modified messages (i.e., $H(\text{msg}_0)$ and $H(\text{msg}_1)$) contain ω bytes with the same value at the same location (here ω is also called the number of hits satisfying $0 \le \omega \le g \equiv \lceil (n\times m)/8 \rceil$, and $W_N^e(0)+W_N^e(1)+\cdots+W_N^e(g)=N$); and 2) the mean of the absolute difference per byte $\bar{d}_{\text{byte}}^e$ between $H(\text{msg}_0)$ and $H(\text{msg}_1)$ over N draws. If the results of $W_N^e(\omega)$ and $\bar{d}_{\text{byte}}^e$ (the experimental values) are very close to their theoretical values, then the related hash function could be regarded as having a good property of collision resistance.

The number of hits on each draw can be obtained as follows: first, divide both $H(\text{msg}_0)$ and $H(\text{msg}_1)$ into g bytes (if $n\times m$ is not divisible by 8, then add a prefix of $c = 8-(n\times m) \bmod 8$ zeros to the hash values), so that the two hash values can be expressed as $h = e_1\|e_2\|e_3\|\ldots\|e_g$ and $h' = e_1'\|e_2'\|e_3'\|\ldots\|e_g'$, respectively ($e_j$ and e_j', respectively, represents the jth byte of h and h'); second, compare h and h' byte by byte and compute ω according to

$$\omega = \sum_{j=1}^{g} \delta\left[t(e_j), t(e_j')\right] \tag{9}$$

where $t(e_j)$ is the decimal value of e_j and $\delta[\cdot]$ is the Kronecker delta function.

The theoretical value (denoted by $W_N^t(\omega)$) of $W_N^e(\omega)$ is given by the product of N and the theoretical probability $P^t(\omega)$ that ω hits occur in (h, h'). Specifically, $P^t(\omega)$ is given by the binomial distribution formula

$$P^t(\omega) = \frac{g!}{\omega!(g-\omega)!}\left(\frac{1}{2^8}\right)^{\omega}\left(1-\frac{1}{2^8}\right)^{g-\omega} \tag{10}$$

and the theoretical number of draws with ω hits is obtained by

$$W_N^t(\omega) = \text{int}\left[N\times P^t(\omega)\right] \tag{11}$$

where int$[\cdot]$ denotes rounding a real number to its nearest integer.

Since $P^t \equiv \{P^t(\omega)\,|\,\omega = 0, 1, \ldots g\}$ and $P^e \equiv \{W_N^e(\omega)/N\,|\,\omega = 0, 1, \ldots g\}$, respectively, describe the theoretical and experimental distributions of ω (or simply hit distributions), the similarity or difference between $\{W_N^e(\omega)\,|\,\omega = 0, 1, \ldots g\}$ (hereafter, simply $\{W_N^e(\omega)\}$) and $\{W_N^t(\omega)\,|\,\omega = 0, 1, \ldots g\}$ (hereafter, simply $\{W_N^t(\omega)\}$) could be measured by Kullback–Leibler divergence between P^t and P^e, i.e.,

$$\begin{aligned} D_{\text{KL}}\left(P^e\|P^t\right) &= \sum_{\omega=0}^{g} P^e(\omega)\log_2\left(\frac{P^e(\omega)}{P^t(\omega)}\right) \\ &= \sum_{\omega=0}^{g} \frac{W_N^e(\omega)}{N}\log_2\left(\frac{W_N^e(\omega)/N}{P^t(\omega)}\right) \end{aligned} \tag{12}$$

a smaller $D_{\text{KL}}(P^e\|P^t)$ indicates a closer similarity between $\{W_N^e(\omega)\}$ and $\{W_N^t(\omega)\}$.

The absolute difference per byte between h and h' is calculated by

$$d_{\text{byte}} = \frac{1}{g}\sum_{j=1}^{g}\left|t(e_j)-t(e_j')\right| \tag{13}$$

and the theoretical value (denoted by $\bar{d}_{\text{byte}}^t$) of the mean of d_{byte} (denoted by $\bar{d}_{\text{byte}}^e$) over N draws is $\bar{d}_{\text{byte}}^t = 85.33$ [9].

The collision resistant test on QHFM-L is performed with $N = 10\,000$, and the simulation results are shown in Table IV, where $W_N^e(4+)$ denotes the number of draws on which more than three hits occur in the hash values of the

TABLE IV Collision-Resistance-Test Results for the Proposed and Existing QW-Based Hash Functions

Hash Instances or Schemes	$\{W_N^{\mathrm{e}}(\omega)\vert\omega = 0, 1, 2, 3, 4+\}$	$\{W_N^{\mathrm{t}}(\omega)\vert\omega = 0, 1, 2, 3, 4+\}$	$D_{\mathrm{KL}}\ (P^{\mathrm{e}}\Vert P^{\mathrm{t}})$	$\bar{d}^{\mathrm{e}}_{\mathrm{byte}}$	$\vert\bar{d}^{\mathrm{e}}_{\mathrm{byte}} - \bar{d}^{\mathrm{t}}_{\mathrm{byte}}\vert$
QHFM-296	$8605, 1312, 81, 2, 0$	$8652, 1255, 89, 4, 0$	0.000361	85.36	0.03
QHFM-264	$8762, 1159, 74, 5, 0$	$8788, 1137, 71, 3, 0$	0.000146	85.27	0.06
QHFM-221	$8674, 1230, 93, 3, 0$	$8962, 984, 52, 2, 0$	0.006711	82.85	2.48
QHFM-200	$9071, 895, 34, 0, 0$	$9068, 889, 42, 1, 0$	0.000302	85.30	0.03
QHFM-195	$8066, 1796, 130, 8, 0$	$9068, 889, 42, 1, 0$	0.069364	82.03	3.30
QHFM-136	$9352, 626, 21, 1, 0$	$9356, 624, 20, 0, 0$	0.000058	85.32	0.01
QHFM-120	$9416, 570, 13, 1, 0$	$9430, 555, 15, 0, 0$	0.000145	85.41	0.08
Yang21-296 [9]	$8321, 1547, 110, 22, 0$	$8652, 1255, 89, 4, 0$	0.008616	85.22	0.11
Yang19-264 [10]	$9019, 923, 52, 2, 4$	$8788, 1137, 71, 3, 0$	0.005647	89.76	4.43
Yang18-264 [11]	$8904, 1026, 68, 2, 0$	$8788, 1137, 71, 3, 0$	0.000969	83.64	1.69
Yang18-221 [12]	$9854, 71, 0, 0, 75$	$8962, 984, 52, 2, 0$	0.188620	N/A	N/A
Li18-200 [13]	$8982, 989, 25, 4, 0$	$9068, 889, 42, 1, 0$	0.001689	N/A	N/A
Cao18-195 [14]	$16063, 314, 6, 0, 0$	$14856, 1456, 69, 2, 0$	0.066791	N/A	N/A
Yang16-128 [15]	$9367, 617, 16, 0, 0$	$9393, 589, 17, 0, 0$	0.000151	87.21	1.88

original and modified messages, that is, $W_N^e(4+) = N - [W_N^e(0) + W_N^e(1) + W_N^e(2) + W_N^e(3)]$. One may notice that the sums of $W_N^t(\omega)$ (over all ω) for 264- and 128-bit hash schemes (or instances) are not equivalent to N, this is due to the rounding operations performed on $N \times P^t(\omega)$. The values of $D_{\mathrm{KL}}(P^e\Vert P^t)$ and $|\bar{d}^e_{\mathrm{byte}} - \bar{d}^t_{\mathrm{byte}}|$ for the existing schemes are deduced from the reported results of $\{W_N^e(\omega)\}$ and $\bar{d}^e_{\mathrm{byte}}$ (or the mean of $d_{\mathrm{byte}} \times g$), respectively.

The values of $D_{\mathrm{KL}}(P^e\Vert P^t)$ indicate that the experimental result of hit distribution for QHFM-L with $L \geq 200$ has closer similarity to the theoretical distribution of ω than those for the existing ones with L-bit output length, and the Kullback–Leibler divergence between P^e and P^t for QHFM-L with $L < 200$ is on a par with that for its peer (Cao18-195 or Yang16-128). As for the average difference per byte in two hash values, the test results of $\bar{d}_{\mathrm{byte}}$ for QHFM-296, QHFM-264, and QHFM-136 (together with QHFM-128) are closer to the theoretical value 85.33 than those for Yang21-296, Yang19-264, and Yang18-128, and the differences between $\bar{d}^e_{\mathrm{byte}}$ and $\bar{d}^t_{\mathrm{byte}}$ for the remaining instances of QHFM are very small. Therefore, the proposed hash scheme has a good capability of collision resistance.

E. RESISTANCE TO BIRTHDAY ATTACKS

Since the proposed hash function has variable digest length, one can easily obtain a QHFM instance that withstands birthday attacks by assigning appropriate values (large enough) to the parameters m and n according to the (cryptanalytic) hardware and software capabilities considered.

V. TIME AND SPACE COMPLEXITY ANALYSIS

The hash value of an input message sent to a QW-based hash function can be calculated by cascading three stages:

1) initializing the state of the walker;
2) performing the underlying CAQW on a cycle according to the bit values of the message;
3) calculating the hash value from the resulting probability distribution of the walker.

The time and space complexity of the proposed scheme or an existing one can, thus, be obtained by analyzing the number of arithmetic operations taken by each stage of the related hashing process. Since QW-based hash functions can be considered as quantum inspired classical algorithms (quantum transforms in QW are simulated by matrix multiplications, and hash values are calculated classically from the resulting probability distribution of the walker), this section will concentrate on classical complexity.

A. TIME AND SPACE COMPLEXITY OF THE PROPOSED SCHEME

The quantum state of the walker after t steps of CQWM ($t \geq 0$) can be expressed as

$$|\psi_t\rangle = \sum_{x,j} A_t^{x,j}|x, j\rangle \tag{14}$$

where $A_t^{x,j}(x \in \mathbb{Z}_n, j \in \mathbb{Z}_8)$ is the amplitude of the 2-term basis state $|x, j\rangle$ at time t, and the correspondence between 2-term and 4-term basis states is described by equation group (5). Before ($t = 0$) and during ($t > 0$) the walk, the state of the particle is identified with these $8n$ amplitudes.

When $t = 0$, the particle is in the state $|\psi_0\rangle = \cos\alpha|0, 2\rangle + \sin\alpha|0, 3\rangle$, which gives $A_0^{0,2} = \cos\alpha$, $A_0^{0,3} = \sin\alpha$, $A_0^{0,j} = 0$ for $j \neq 2$ and $j \neq 3$, and $A_0^{x,j} = 0$ for $x \neq 0$. Thus, the classical representation of the initial state $|\psi_0\rangle$ can be specified using $8n$ assignments.

When $t > 0$, if the tth message bit equals 0, the values of $\{A_t^{x,j}|x \in \mathbb{Z}_n, j \in \mathbb{Z}_8\}$ (hereafter, simply $\{A_t^{x,j}\}$) are determined by $U^{(0)} = S(I_n \otimes D^{(0)})(I_{4n} \otimes C^{(0)})$ and $\{A_{t-1}^{x,j}|x \in \mathbb{Z}_n, j \in \mathbb{Z}_8\}$ (hereafter, simply $\{A_{t-1}^{x,j}\}$). For the sake of simplicity of notation, we denote $C^{(0)}$ by $\binom{a_0\ b_0}{c_0\ d_0}$, then the action of this coin operator on $|x, \mathrm{dr}_2, \mathrm{dr}_1, c\rangle$ can be formulated as

$$\begin{aligned} C^{(0)} : |x, \mathrm{dr}_2, \mathrm{dr}_1, c\rangle \rightarrow\ & \bar{c} \cdot a_0|x, \mathrm{dr}_2, \mathrm{dr}_1, 0\rangle \\ & + \bar{c} \cdot c_0|x, \mathrm{dr}_2, \mathrm{dr}_1, 1\rangle \\ & + c \cdot b_0|x, \mathrm{dr}_2, \mathrm{dr}_1, 0\rangle \\ & + c \cdot d_0|x, \mathrm{dr}_2, \mathrm{dr}_1, 1\rangle. \end{aligned} \tag{15}$$

Converting the 4-term states in (15) into 2-term states gives

$$\begin{aligned} C^{(0)}: |x,j\rangle \to\ & (\overline{j \bmod 2})(a_0|x,j\rangle + c_0|x,j+1\rangle) \\ & + (j \bmod 2)(b_0|x,j-1\rangle + d_0|x,j\rangle) \end{aligned} \quad (16)$$

where $j+1$ and $j-1$ are both calculated using modular arithmetic under modulus 8. Combining (16), (6), and (8), one can obtain the action of $U^{(0)}$ on each 2-term basis state as well as on $|\psi_{t-1}\rangle$ and then deduce the relation between $\{A_t^{x,j}\}$ and $\{A_{t-1}^{x,j}\}$. Specifically, the actions of $U^{(0)}$ on the components $A_{t-1}^{x,j}|x,j\rangle$ of $|\psi_{t-1}\rangle$ are

$$\begin{aligned}
A_{t-1}^{x,0}|x,0\rangle &\to a_0A_{t-1}^{x,0}|x+1,4\rangle + c_0A_{t-1}^{x,0}|x-1,1\rangle \\
A_{t-1}^{x,1}|x,1\rangle &\to b_0A_{t-1}^{x,1}|x+1,4\rangle + d_0A_{t-1}^{x,1}|x-1,1\rangle \\
A_{t-1}^{x,2}|x,2\rangle &\to a_0A_{t-1}^{x,2}|x+1,6\rangle + c_0A_{t-1}^{x,2}|x-1,3\rangle \\
A_{t-1}^{x,3}|x,3\rangle &\to b_0A_{t-1}^{x,3}|x+1,6\rangle + d_0A_{t-1}^{x,3}|x-1,3\rangle \\
A_{t-1}^{x,4}|x,4\rangle &\to a_0A_{t-1}^{x,4}|x-1,0\rangle + c_0A_{t-1}^{x,4}|x+1,5\rangle \\
A_{t-1}^{x,5}|x,5\rangle &\to b_0A_{t-1}^{x,5}|x-1,0\rangle + d_0A_{t-1}^{x,5}|x+1,5\rangle \\
A_{t-1}^{x,6}|x,6\rangle &\to a_0A_{t-1}^{x,6}|x-1,2\rangle + c_0A_{t-1}^{x,6}|x+1,7\rangle \\
A_{t-1}^{x,7}|x,7\rangle &\to b_0A_{t-1}^{x,7}|x-1,2\rangle + d_0A_{t-1}^{x,7}|x+1,7\rangle
\end{aligned} \quad (17)$$

where $x \pm 1$ are calculated using modular arithmetic under modulus n. Summing up the transformed results on the right side, one can observe that the amplitudes of the walker being at position x at time $t-1$ contribute $a_0A_{t-1}^{x,0} + b_0A_{t-1}^{x,1}$ to $A_t^{x+1,4}$, $c_0A_{t-1}^{x,0} + d_0A_{t-1}^{x,1}$ to $A_t^{x-1,1}$, and $a_0A_{t-1}^{x,2} + b_0A_{t-1}^{x,3}$ to $A_t^{x+1,6}$, etc.; here $A_t^{x\pm1,j}$ is the amplitude of $|x \pm 1 \pmod n, j\rangle$ at time t. Moreover, the amplitudes of being at position x at time $t-1$ only contribute to $A_t^{x-1,j}$ with $j \le 3$ and to $A_t^{x+1,j}$ with $j \ge 4$; conversely, the former 4 amplitudes (with $j \le 3$) at an arbitrary position x are contributed by the amplitudes at position $x+1 \pmod n$, while the latter 4 amplitudes (with $j \ge 4$) at position x are contributed by those at position $x-1 \pmod n$. As a result, each amplitude of being at position x at time t, denoted by $A_t^{x,j}$, is only contributed by the amplitudes of being at a single position ($x+1 \pmod n$ or $x-1 \pmod n$) at time $t-1$. Thus, the relation between $\{A_t^{x,j}\}$ and $\{A_{t-1}^{x,j}\}$ after a step of QW1M can be expressed as follows:

$$\begin{aligned}
A_t^{x,0} &= a_0A_{t-1}^{x+1,4} + b_0A_{t-1}^{x+1,5} \\
A_t^{x,1} &= c_0A_{t-1}^{x+1,0} + d_0A_{t-1}^{x+1,1} \\
A_t^{x,2} &= a_0A_{t-1}^{x+1,6} + b_0A_{t-1}^{x+1,7} \\
A_t^{x,3} &= c_0A_{t-1}^{x+1,2} + d_0A_{t-1}^{x+1,3} \\
A_t^{x,4} &= a_0A_{t-1}^{x-1,0} + b_0A_{t-1}^{x-1,1} \\
A_t^{x,5} &= c_0A_{t-1}^{x-1,4} + d_0A_{t-1}^{x-1,5} \\
A_t^{x,6} &= a_0A_{t-1}^{x-1,2} + b_0A_{t-1}^{x-1,3} \\
A_t^{x,7} &= c_0A_{t-1}^{x-1,6} + d_0A_{t-1}^{x-1,7}.
\end{aligned} \quad (18)$$

Similarly, if the tth message bit equals 1, the values of $\{A_t^{x,j}\}$ are determined by $U^{(1)} = S(I_n \otimes D^{(1)})(I_{4n} \otimes C^{(1)})$ and $\{A_{t-1}^{x,j}\}$. We denote $C^{(1)}$ by $\begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}$, then the action of $C^{(1)}$ on $|x,j\rangle$ is

$$\begin{aligned} C^{(1)}: |x,j\rangle \to\ & (\overline{j \bmod 2})(a_1|x,j\rangle + c_1|x,j+1\rangle) \\ & + (j \bmod 2)(b_1|x,j-1\rangle + d_1|x,j\rangle). \end{aligned} \quad (19)$$

A Combination of (19), (7), and (8) gives the actions of $U^{(1)}$ on the components of $|\psi_{t-1}\rangle$

$$\begin{aligned}
A_{t-1}^{x,0}|x,0\rangle &\to a_1A_{t-1}^{x,0}|x+1,4\rangle + c_1A_{t-1}^{x,0}|x-1,1\rangle \\
A_{t-1}^{x,1}|x,1\rangle &\to b_1A_{t-1}^{x,1}|x+1,4\rangle + d_1A_{t-1}^{x,1}|x-1,1\rangle \\
A_{t-1}^{x,2}|x,2\rangle &\to a_1A_{t-1}^{x,2}|x-1,0\rangle + c_1A_{t-1}^{x,2}|x+1,5\rangle \\
A_{t-1}^{x,3}|x,3\rangle &\to b_1A_{t-1}^{x,3}|x-1,0\rangle + d_1A_{t-1}^{x,3}|x+1,5\rangle \\
A_{t-1}^{x,4}|x,4\rangle &\to a_1A_{t-1}^{x,4}|x+1,6\rangle + c_1A_{t-1}^{x,4}|x-1,3\rangle \\
A_{t-1}^{x,5}|x,5\rangle &\to b_1A_{t-1}^{x,5}|x+1,6\rangle + d_1A_{t-1}^{x,5}|x-1,3\rangle \\
A_{t-1}^{x,6}|x,6\rangle &\to a_1A_{t-1}^{x,6}|x-1,2\rangle + c_1A_{t-1}^{x,6}|x+1,7\rangle \\
A_{t-1}^{x,7}|x,7\rangle &\to b_1A_{t-1}^{x,7}|x-1,2\rangle + d_1A_{t-1}^{x,7}|x+1,7\rangle.
\end{aligned} \quad (20)$$

Thus, the relation between $\{A_t^{x,j}\}$ and $\{A_{t-1}^{x,j}\}$ after a step of QW2M can be expressed as

$$\begin{aligned}
A_t^{x,0} &= a_1A_{t-1}^{x+1,2} + b_1A_{t-1}^{x+1,3} \\
A_t^{x,1} &= c_1A_{t-1}^{x+1,0} + d_1A_{t-1}^{x+1,1} \\
A_t^{x,2} &= a_1A_{t-1}^{x+1,6} + b_1A_{t-1}^{x+1,7} \\
A_t^{x,3} &= c_1A_{t-1}^{x+1,4} + d_1A_{t-1}^{x+1,5} \\
A_t^{x,4} &= a_1A_{t-1}^{x-1,0} + b_1A_{t-1}^{x-1,1} \\
A_t^{x,5} &= c_1A_{t-1}^{x-1,2} + d_1A_{t-1}^{x-1,3} \\
A_t^{x,6} &= a_1A_{t-1}^{x-1,4} + b_1A_{t-1}^{x-1,5} \\
A_t^{x,7} &= c_1A_{t-1}^{x-1,6} + d_1A_{t-1}^{x-1,7}.
\end{aligned} \quad (21)$$

Relations (18) and (21) show that, given the amplitudes $\{A_{t-1}^{x_0,j} | j \in \mathbb{Z}_8\}$ of being at a fixed position x_0 at time $t-1$, the values of the amplitudes $\{A_t^{x_0,j} | j \in \mathbb{Z}_8\}$ of being at x_0 at time t can be calculated using 16 multiplications and 8 additions, which means all amplitudes at each time step of CQWM on a cycle with n nodes can be obtained using $O(n)$ basic arithmetic operations. To perform these operations, one needs to store the old (or the initial) $8n$ amplitudes and their 8 possible coefficients ($a_1, a_2, \ldots, d_1, d_2$) to calculate the new $8n$ amplitudes, and the values of both old and new amplitudes are refreshed at each time step. If the input message msg is a binary string of M bits, then the values of $\{A_M^{x,j}\}$ can be obtained using $O(Mn)$ basic operations with $O(n)$ memory space. Finally, the hash value is computed from $\{A_M^{x,j}\}$ using $O(n)$ multiplications and $O(n)$ modulo operations with $O(n)$ space. Thus, the time and space complexity of QHFM with input length M are $O(Mn)$ and $O(n)$, respectively.

In particular, if one wants to obtain an L-bit hash value (L is a multiple of m) of msg using QHFM-L, then the cycle utilized by QHFM-L has $L/m = O(L)$ nodes (here m is constant with respect to the input length M); in this case, the hash value is produced with $O(ML)$ time and $O(L)$ space.

B. TIME AND SPACE COMPLEXITY COMPARISON OF QW-BASED HASH SCHEMES

In a similar way, one can deduce the time and space complexity of the existing QW-based hash functions [9]–[16] with respect to the input and output length. To facilitate discussion, we divide the existing schemes into the following four groups:

1) the hash functions based on one-dimensional one-particle quantum walks [9]–[12];
2) the hash function based on two-dimensional one-particle quantum walks [13];
3) the hash function based on quantum walks on Johnson graphs [14];
4) the hash functions based on one-dimensional two-particle quantum walks [15], [16].

Again, suppose all schemes produce hash values of length L. In this case, the schemes in group 1) utilize a cycle with $O(L)$ nodes, and the amplitudes of the particle being at each node at time t can be calculated from the amplitudes of being at the two neighbors of this node at time $t-1$ (possibly calculated from the amplitudes of being at a single neighbor or remain unchanged during broken-line quantum walks [9]) using constant number of (basic arithmetic) operations. In group 2), Li18-200 utilizes cycles of length $O(\sqrt{L})$ in two-dimensional space, which lead to $O(L)$ positions for the walker, and the amplitudes of being at position (x, y) at time t can be calculated from the amplitudes of being at $(x \pm 1, y \pm 1)$ at time $t-1$ using constant number of operations. In 3), Cao18-195 utilizes a Johnson graph $J(n, 1)$ with $n = O(L)$ nodes, and the amplitudes of the particle being at each node at time t can be calculated from the amplitudes of being at the remaining $n-1$ nodes at time $t-1$ using $O(L)$ operations. Similar to group 2), schemes in group 4) also utilize a cycle with $O(\sqrt{N})$ nodes, which leads to $O(L)$ position pairs for the two particles, and the amplitudes of the first and second particles being, respectively, at nodes x and y at time t can be calculated from the amplitudes of the two particles being, respectively, at $x \pm 1$ and $y \pm 1$ at time $t-1$ using constant number of operations.

Thus, except for Cao18-195 [14], which performs $O(L^2)$ operations at each time step, the existing schemes [9]–[13], [15], [16] take $O(L)$ time to calculate the amplitudes at time t from the amplitudes at time $t-1$. If the input message is of bit-length M, then the resulting amplitudes at time M can be obtained with $O(ML^2)$ and $O(ML)$ operations in Cao18-195 and the remaining schemes, respectively. After that, for all these schemes [9]–[16], the hash value is computed from the resulting $O(L)$ amplitudes with $O(L)$ operations. Therefore, the time complexities of Cao18-195 and the other QW-based hash schemes are $O(ML^2)$ and $O(ML)$, respectively. Since each amplitude at time t is a linear combination of the amplitudes of $O(1)$ or $O(L)$ positions at time $t-1$ in all schemes, and the resulting probability distribution takes $O(L)$ space as well, the space complexities of the existing schemes all equals $O(L)$.

As a result, the proposed scheme has the same time and space complexity as the existing QW-based hash schemes except Cao18-195, whose time complexity is slightly greater than that of the other schemes, including the proposed one.

VI. CONCLUSION

In this article, a new hash function QHFM based on quantum walks with one- and two-step memory on circles is constructed, whose statistical properties as well as time and space complexity are evaluated and compared with the existing QW-based hash functions.

Unlike the existing analyses of hash schemes based on quantum walks without memory, where a single indicator $\overline{T}$ is used to evaluate the uniform distribution property, we adopted an additional indicator ΔQ to assess this property, since $\overline{T}$ alone is closely related to P, implying that it also suggests the diffusion and confusion properties. In the collision resistance analysis, we use Kullback–Leibler divergence to evaluate the similarity between the experimental and theoretical distributions of ω, so that the difference between $\{W_N^e(\omega)\}$ and $\{W_N^t(\omega)\}$ can be indicated by a single number.

The analysis results show that QHFM has near-ideal statistical performance and takes no more time and space than its peers based on CAQWs. Also, it can be shown that the proposed scheme is on a par, in terms of diffusion and confusion properties and collision resistance property, with the advanced hash algorithms beyond QW-based ones, such as hash functions based on chaotic systems [23]–[27]. The good statistical performance of QHFM suggests that alternately running two quantum walks differing in more than one respects, including coin operator and memory length, can also yield good hash functions. Thus, it is unnecessary to restrict the component parts of a controlled alternate quantum walk to a single kind of walk (equipped with controlled coins). In future work, we will explore the possibility of combining two quantum walks with more differences and investigate the effect of those differences on the performance of the resulting hash function.

REFERENCES

[1] C. H. Bennett, G. Brassard, and C. Crépeau, “Generalized privacy amplification,” *IEEE Trans. Inf. Theory*, vol. 41, no. 6, pp. 1915–1923, Nov. 1995, doi: 10.1109/18.476316.

[2] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, “Hash functions and data integrity,” in *Handbook of Applied Cryptography*, 1st ed. Boca Raton, FL, USA: CRC Press, 1996, ch. 9, p. 328.

[3] F. Ablayev, M. Ablayev, A. Vasiliev, and M. Ziatdinov, “Quantum fingerprinting and quantum hashing. Computational and cryptographical aspects,” *Baltic J. Mod. Comput.*, vol. 4, no. 4, pp. 860–875, Nov. 2016, doi: 10.22364/bjmc.2016.4.4.17.

[4] F. Ablayev, M. Ablayev, and A. Vasiliev, “On the balanced quantum hashing,” *J. Phys.: Conf. Ser.*, vol. 681, no. 1, 2016, Art. no. 012019, doi: 10.1088/1742-6596/681/1/012019.

[5] A. Vasiliev, “Quantum hashing for finite abelian groups,” *Lobachevskii J. Math.*, vol. 37, no. 6, pp. 753–757, Nov. 2016, doi: 10.1134/S1995080216060184.

[6] M. Ziatdinov, “From graphs to keyed quantum hash functions,” *Lobachevskii J. Math.*, vol. 37, no. 6, pp. 705–712, Nov. 2016, doi: 10.1134/S1995080216060202.

[7] F. M. Ablayev and A. V. Vasiliev, “Cryptographic quantum hashing,” *Laser Phys. Lett.*, vol. 11, no. 2, Dec. 2013, Art. no. 025202, doi: 10.1088/1612-2011/11/2/025202.

[8] F. M. Ablayev and M. T. Ziatdinov, “Universal hash functions from quantum procedures,” *Uchenye Zapiski Kazanskogo Universiteta. Seriya Fiziko-Matematicheskie Nauki*, vol. 162, no. 3, pp. 259–268, 2020, doi: 10.1007/s10773-021-04724-0.

[9] Y. G. Yang, J. R. Dong, Y. L. Yang, Y. H. Zhou, and W. M. Shi, “Usefulness of decoherence in quantum-walk-based hash function,” *Int. J. Theor. Phys.*, vol. 60, pp. 1025–1037, Jan. 2021, doi: 10.1007/s10773-021-04724-0.

[10] Y. G. Yang, J. L. Bi, D. Li, Y. H. Zhou, and W. M. Shi, “Hash function based on quantum walks,” *Int. J. Theor. Phys.*, vol. 58, no. 6, pp. 1861–1873, Mar. 2019, doi: 10.1007/s10773-019-04081-z.

[11] Y. G. Yang, J. L. Bi, X. B. Chen, Z. Yuan, Y. H. Zhou, and W. M. Shi, “Simple hash function using discrete-time quantum walks,” *Quantum Inf. Process.*, vol. 17, no. 8, pp. 189, Jun. 2018, doi: 10.1007/s11128-018-1954-2.

[12] 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.*, vol. 61, no. 3, Jan. 2018, Art. no. 030312, doi: 10.1007/s11433-017-9132-y.

[13] D. Li, Y. G. Yang, J. L. Bi, J. B. Yuan, and J. Xu, “Controlled alternate quantum walks based quantum hash function,” *Sci. Rep.*, vol. 8, no. 1, p. 225, Jan. 2018, doi: 10.1038/s41598-017-18566-6.

[14] W. F. Cao, Y. C. Zhang, Y. G. Yang, D. Li, Y. H. Zhou, and W. M. Shi, “Constructing quantum hash functions based on quantum walks on johnson graphs,” *Quantum Inf. Process.*, vol. 17, no. 7, p. 156, May 2018, doi: 10.1007/s11128-018-1923-9.

[15] Y. G. Yang, P. Xu, R. Yang, Y. H. Zhou, and W. M. Shi, “Quantum hash function and its application to privacy amplification in quantum key distribution, pseudo-random number generation and image encryption,” *Sci. Rep.*, vol. 6, Jan. 2016, Art. no. 19788, doi: 10.1038/srep19788.

[16] D. Li, J. Zhang, F. Z. Guo, W. Huang, Q. Y. Wen, and H. Chen, “Discrete-time interacting quantum walks and quantum hash schemes,” *Quantum Inf. Process.*, vol. 12, no. 3, pp. 1501–1513, Mar. 2013, doi: 10.1007/s11128-012-0421-8.

[17] D. Li, M. Mc Gettrick, Y. G. Yang, J. Xu, and Y. Wang, “Quantum walks with memory provided by parity of memory,” *Int. J. Theor. Phys.*, vol. 59, no. 6, pp. 1934–1943, Jun. 2020, doi: 10.1007/s10773-020-04466-5.

[18] W. Dai, J. Yuan, and D. Li, “Discrete-time quantum walk with memory on the cayley graph of the dihedral group,” *Int. J. Theor. Phys.*, vol. 59, no. 1, pp. 10–28, Jan. 2020, doi: 10.1007/s10773-019-04257-7.

[19] Q. Zhou and S. F. Lu, “One-dimensional quantum walks with two-step memory,” *Quantum Inf. Process.*, vol. 18, no. 12, p. 359, Oct. 2019, doi: 10.1007/s11128-019-2475-3.

[20] D. Li, M. Mc Gettrick, F. Gao, J. Xu, and Q. Y. Wen, “Generic quantum walks with memory on regular graphs,” *Phys. Rev. A.*, vol. 93, no. 4, Apr. 2016, Art. no. 042323, doi: 10.1103/PhysRevA.93.042323.

[21] M. Mc Gettrick and J. A. Miszczak, “Quantum walks with memory on cycles,” *Phys. A*, vol. 399, pp. 163–170, Apr. 2014, doi: 10.1016/j.physa.2014.01.002.

[22] M. Mc Gettrick, “One dimensional quantum walks with memory,” *Quantum Inf. Comput.*, vol. 10, no. 5, pp. 509–524, May 2010, doi: 10.5555/2011362.2011371.

[23] H. J. Liu, X. Y. Wang, and A. Kadir, “Constructing chaos-based hash function via parallel impulse perturbation,” *Soft Comput.*, vol. 25, pp. 11077–11086, May 2021, doi: 10.1007/s00500-021-05849-4.

[24] J. Wang, G. Liu, Y. Q. Chen, and S. Wang, “Construction and analysis of SHA-256 compression function based on chaos S-box,” *IEEE Access*, vol. 9, pp. 61768–61777, 2021, doi: 10.1109/ACCESS.2021.3071501.

[25] Y. Wang, L. Q. Chen, X. Y. Wang, G. Wu, K. L. Yu, and T. Y. Lu, “The design of keyed hash function based on CNN-MD structure,” *Chaos, Solitons Fractals*, vol. 152, Nov. 2021, Art. no. 111443, doi: 10.1016/j.chaos.2021.111443.

[26] R. I. Abdelfatah, E. A. Baka, and M. E. Nasr, “Keyed parallel hash algorithm based on multiple chaotic maps (KPHA-MCM),” *IEEE Access*, vol. 9, pp. 130399–130409, 2021, doi: 10.1109/ACCESS.2021.3113855.

[27] M. Alawida, A. Samsudin, N. Alajarmeh, J. S. Teh, and M. Ahmad, “A novel hash function based on a chaotic sponge and DNA sequence,” *IEEE Access*, vol. 9, pp. 17882–17897, 2021, doi: 10.1109/ACCESS.2021.3049881.