Chapter 2

On Quantum Fingerprinting and Quantum Cryptographic Hashing

Farid Ablayev and Marat Ablayev

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.70692

Abstract

Fingerprinting and cryptographic hashing have quite different usages in computer science, but have similar properties. Interpretation of their properties is determined by the area of their usage: fingerprinting methods are methods for constructing efficient randomized and quantum algorithms for computational problems, whereas hashing methods are one of the central cryptographical primitives. Fingerprinting and hashing methods are being developed from the mid of the previous century, whereas quantum fingerprinting and quantum hashing have a short history. In this chapter, we investigate quantum fingerprinting and quantum hashing. We present computational aspects of quantum fingerprinting and quantum hashing and discuss cryptographical properties of quantum hashing.

Keywords: quantum computations, quantum cryptography, fingerprinting, hashing

1. Introduction

Fingerprinting and hashing are well-known techniques. Fingerprinting is widely used in various meanings in different areas of computer science. We restrict ourselves to the area of computational complexity theory where the notion of fingerprinting is more or less formalized. Cryptographic hashing allows to securely present objects and mathematically is more formalized. Fingerprinting and cryptographic hashing have quite different usages in computer science, but have similar properties. Interpretation of their properties is determined by the area of their usage: fingerprinting methods are methods for constructing efficient randomized and quantum algorithms for computational problems, whereas hashing methods are one of the central cryptographical primitives.

Fingerprinting and hashing methods are being developed from the mid of the previous century, whereas quantum fingerprinting and quantum hashing have a short history.

In this chapter, we present computational aspects of quantum fingerprinting, discuss cryptographical properties of quantum hashing, and present the possible use of quantum hashing for quantum hash-based message authentication codes (QMAC).

1.1. Classical and quantum fingerprinting

Fingerprinting in complexity theory is a procedure that maps a large data item to a much shorter string, its fingerprint, that identifies the original data (with high probability). The key properties of classical fingerprinting methods are (i) they allow to build efficient randomized computational algorithms and (ii) the resulting algorithms have bounded error [1].

Rusins Freivalds was one of the first researchers who introduced methods (later called fingerprinting) for constructing efficient randomized algorithms (which are more efficient than any deterministic algorithm) [2, 3].

In quantum case, fingerprinting is a procedure that maps classical data to a quantum state that identifies the original data (with high probability). One of the first applications of the quantum fingerprinting method is due to Ambainis and Freivalds [4]: for a specific language, they have constructed a quantum finite automaton with an exponentially smaller size than any classical randomized automaton. An explicit definition of the quantum fingerprinting was introduced by Buhrman et al. [5] in (2001) for constructing efficient quantum communication protocol for equality testing. It is worth noting that the fingerprinting by Buhrman et al. has been used as a cryptographic hash function in [6, 7].

1.2. Cryptographic quantum hashing

Cryptographic hashing has a lot of fruitful applications in cryptography. Note that in cryptography functions satisfying (i) one-way property and (ii) collision resistance property (in different specific meanings) are called hash functions, and we propose to do so when we are considering cryptographical aspects of quantum functions with the above properties. So, we suggest to call a quantum function that satisfies properties (i) and (ii) (in the quantum setting), a cryptographic quantum hash function or just quantum hash function. Note, however, that there is only a thin line between the notions of quantum fingerprinting and quantum hashing. One of the first considerations of a quantum function (that maps classical words into quantum states) as a cryptographic primitive, having one-way property and collision resistance property is due to [6], where the quantum fingerprinting function from [5] was used. Another approach to constructing quantum hash functions from quantum walks was considered in [8, 9, 10], and it resulted in privacy amplification in quantum key distribution and other useful applications.

1.3. The chapter organization

In Section 3, we consider quantum fingerprinting as a mapping of classical inputs to quantum states, which allows to construct efficient quantum algorithms for computing Boolean functions. We consider the quantum fingerprinting function from [5] as well as the quantum

http://dx.doi.org/10.5772/intechopen.70692

fingerprinting technique from [11]. The latter was motivated by the paper [4] and its generalization [12].

We define a notion of quantum (δ, ε)-hash function that is quantumly one-way δ-resistant and quantumly collision ε-resistant.

We show that one-way property and collision resistance property are correlated for a quantum hash function. The more the function is one-way, the less it is collision resistant and vice versa. We show that such a correlation can be balanced.

We present an approach for quantum hash function constructions by establishing a connection with small-biased sets [13] and quantum hash function constructions: we prove that each ε-biased set allows to generate quantum collision ε-resistant function. Note that one-way property of this function depends on the size of such ε-biased set: the smaller ε-biased set allows to generate a quantum function with the better one-way characteristics. Such a connection adds to the long list of small-biased sets' applications.

In particular, it was observed in [13, 14] that the ε-bias property is closely related to the error-correcting properties of linear codes. In particular, for the binary case, a set S is ε-biased iff every pair of distinct code words of corresponding error correcting code C_S has relative Hamming distance $(1 \pm \varepsilon)/2$.

Note that the quantum fingerprinting function from [5] is based on a binary error-correcting code, and so it solves the problem of constructing quantum hash functions for the binary case. For the general (nonbinary) case, ε-bias does not correspond to Hamming distance. Thus, in contrast to the binary case, an arbitrary linear error correcting code cannot be used directly for quantum hash functions.

Note that one-way property of function means computational effectiveness of this function. We show that considered construction of quantum (δ, ε)-hash function is computed effectively in the model of quantum branching programs. We consider two complexity measures: a number $width(Q)$ of qubits that QBP Q uses for computation and a number time(Q) of computational steps of QBP Q. Such QBP Q is of $width(Q) = O(\log \log q)$ and $time(Q) = \log\ q$.

We prove that such QBP construction is optimal. That is, we prove lower bounds $\Omega(\log \log q)$ for QBP width and $\Omega(\log q)$ for QBP time for quantum (δ, ε)-hash function presentation.

2. Preliminaries

We recall that mathematically a qubit is described as a unit vector in the two-dimensional Hilbert complex space $\mathcal{H}^2$. Let $s \geq 1$. Let $\mathcal{H}^d$ be the $d = 2^s$-dimensional Hilbert space, describing the states of s qubits. Another notation for $\mathcal{H}^d$ is $(\mathcal{H}^2)^{\otimes s}$, i.e., $\mathcal{H}^d$ is made up of s copies of a single qubit space $\mathcal{H}^2$.

$$\left(\mathcal{H}^2\right)^{\otimes s} = \mathcal{H}^2 \otimes, \ldots, \otimes \mathcal{H}^2 = \mathcal{H}^{2^s}. \tag{1}$$

Conventionally, we use notation $|i\rangle$ for the vector from H^d, which has a 1 on the i-th position and 0 elsewhere. An orthonormal basis $|1\rangle, \ldots, |d\rangle$ is usually referred to as the *standard computational basis*.

We let $\mathbb{Z}_q$ to be a finite additive group of Z/qZ, the integers modulo q. Let Σ^k be a set of words of length k over a finite alphabet Σ. Let $\mathbb{X}$ be a finite set. In this paper, we let $\mathbb{X} = \Sigma^k$ or $\mathbb{X} = \mathbb{Z}_q$. For $K = |\mathbb{X}|$ and integer $s \geq 1$, we define a $(K; s)$ classical-quantum function (or just quantum function) to be mapping

$$\psi : \mathbb{X} \to \left(\mathcal{H}^2\right)^{\otimes s} \quad \text{or} \quad \psi : w \mapsto |\psi(w)\rangle. \tag{2}$$

In order to outline a computational aspect and present a procedure for quantum function ψ, we define ψ to be a unitary transformation (determined by an element $w \in \mathbb{X}$) of the initial state $|\psi_0\rangle \in (\mathcal{H}^2)^{\otimes s}$ to a quantum state $|\psi(w)\rangle \in (\mathcal{H}^2)^{\otimes s}$

$$\psi : \{\left|\psi_0\right\rangle\} \times \mathbb{X} \to \left(\mathcal{H}^2\right)^{\otimes s} \qquad \left|\psi(w)\right\rangle = U(w)\left|\psi_0\right\rangle, \tag{3}$$

where $U(w)$ is a unitary matrix.

Extracting information on w from $|\psi(w)\rangle$ is a result of measurements of quantum state $|\psi(w)\rangle$. In this chapter, we consider quantum transformations and measurements of quantum states with respect to computational basis.

3. Quantum fingerprinting

The ideas of the fingerprinting technique in the quantum setting for the first time appeared in [4]. The authors used a succinct presentation of the classical input by a quantum automata state, which resulted in an exponential improvement over classical algorithm. Later in the works of [12] the ideas were developed further to give an arbitrarily small probability of error. This was the basis for the general quantum fingerprinting framework proposed in [11].

However, the term "quantum fingerprinting" is mostly used in scientific literature to address a seminal paper [5], where this notion first appeared explicitly. To distinguish between different versions of the quantum fingerprinting techniques, the fingerprinting function from [5] is called as "binary" (since it uses some binary error-correcting code in its construction), whereas the fingerprinting from [11] is called "q-ary" for it uses presentation of the input in $\mathbb{Z}_q$.

3.1. Binary quantum fingerprinting

The quantum fingerprinting function was formally defined in [5], where it was used for quantum equality testing in a quantum communication model. It is based on the notion of a binary error-correcting code.

An (n, k, d) *error-correcting code* is a map $C: \Sigma^k \to \Sigma^n$ such that, for any two distinct words w, $w' \in \Sigma^k$, the Hamming distance $d(C(w), C(w'))$ between code words $C(w)$ and $C(w')$ is at least d. The code is binary if $\Sigma = \{0, 1\}$.

The construction of the quantum fingerprinting function is as follows.

- Let $c > 2$ and $\varepsilon < 1$. Let k be a positive integer and $n = ck$. Let $E: \{0,1\}^k \to \{0,1\}^n$ be a (n, k, d) binary error-correcting code with Hamming distance $d \geq (1 - \varepsilon)n$.
- Define a family of functions $F_E = \{E_1, \ldots, E_n\}$, where $E_i : \{0,1\}^k \to \mathbb{F}_2$ is defined by the rule: $E_i(w)$ is the i-th bit of the codeword $E(w)$.
- Let $s = \log\ n + 1$. Define the quantum function $\psi_{F_E}: \{0,1\}^k \to (\mathcal{H}^2)^{\otimes s}$, determined by a word w as

$$|\psi_{F_E}(w)\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle |E_i(w)\rangle. \tag{4}$$

Original paper of [5] used this function to construct a quantum communication protocol that tests equality in the simultaneous message passing (SMP) model with no shared resources. This protocol requires $O(\log\ n)$ qubits to compare n-bit binary strings, which is exponentially smaller than any classical deterministic or even randomized protocol in the SMP setting with no shared randomness. The proposed quantum protocol has one-sided error of $1/2(1 + \langle \psi_{F_E}(x) |\ \psi_{F_E}(y)\rangle^2)$, where $|\psi_{F_E}(x)\rangle$ and $|\psi_{F_E}(y)\rangle$ are two different quantum fingerprints. Their inner product $|\langle \psi_{F_E}(x) |\ \psi_{F_E}(y)\rangle|$ is bounded by ε, if the Hamming distance of the underlying code is $(1 - \varepsilon)n$. Thus, ε is determined by the chosen error-correcting code. For instance, Justesen codes mentioned in the paper give $\varepsilon < 9/10 + 1/(15c)$ for any chosen $c > 2$.

In the same paper, it was shown that this result can be improved by choosing an error-correcting code with Hamming distance between any two distinct code words $(1 - \varepsilon)n/2$ and $(1 + \varepsilon)n/2$ for any $\varepsilon > 0$ (however, the existence of such codes can only be proved nonconstructively via probabilistic argument).

Further research on this topic mostly used the following phase presentation version of quantum fingerprinting. We define the quantum fingerprinting function $\psi: \{0,1\}^k \to (\mathcal{H}^2)^{\otimes s}$ determined by a word w as

$$\psi_{F_E}(w) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (-1)^{E_i(w)} |i\rangle \tag{5}$$

This function gives the following bound for the fingerprints of distinct inputs

$$\left| \left\langle \psi_{F_E}(x) | \psi_{F_E}(y) \right\rangle \right| = \frac{1}{n} \sum_{i=1}^{n} (-1)^{E_i(w) \oplus E_i(w')} = \frac{n - d(E(w), E(w'))}{n} \leq \varepsilon \tag{6}$$

3.2. *q*-ary quantum fingerprinting

In this section, we demonstrate the generalization of binary fingerprinting function to the q-ary case. General technique is presented in [11, 15]. Here, we present the idea using specific Boolean function $g:\{0,1\}^n \rightarrow \{0,1\}$ where $g(\sigma)=1$ iff $\sigma = 0 \bmod sq$. We treat σ also as an integer encoded by binary string σ.

To test g, we rotate the initial state $|0\rangle$ of a single qubit by an angle $\theta = \pi\sigma/q$:

$$|0\rangle \rightarrow |\psi(\sigma)\rangle = \cos\theta|0\rangle + \sin\theta|1\rangle \tag{7}$$

Then, this state $|\psi(\sigma)\rangle$ is measured and the input σ is accepted iff the result of the measurement is $|0\rangle$.

Obviously, this quantum state is $\pm|0\rangle$ iff $\sigma = 0 \bmod q$. In the worst case, this algorithm gives the one-sided error of $\cos^2\pi(q-1)/q$, which can be arbitrarily close to 1.

The above description can be presented as follows using $\log t + 1 = (\log\log q) + 1$ qubits:

$$\underbrace{|0\rangle \otimes \cdots \otimes |0\rangle}_{\log t} \otimes |0\rangle \rightarrow \frac{1}{\sqrt{t}} \sum_{i=1}^{t} |i\rangle(\cos\theta_i|0\rangle + \sin\theta_i|1\rangle), \tag{8}$$

where $\theta_i = \frac{2\pi s_i \sigma}{q}$ and the set $S = \{s_1, \ldots, s_t\} \subseteq \mathbb{Z}_q$ is chosen in order to guarantee the small probability of error [11, 15]. That is, the last qubit is simultaneously rotated in t different subspaces by corresponding angles θ_i.

The above q-ary quantum fingerprinting method can be presented in the following procedure:

1. The initial state of the quantum register is $|0\rangle^{\otimes \log t}|0\rangle$.
2. The Hadamard transform creates the uniform superposition $\frac{1}{\sqrt{t}}\sum_{j=1}^{t}|j\rangle|0\rangle$ of the basis states $\{|j\rangle|0\rangle : j \in \{1, \ldots, t\}\}$.
3. Based on the input σ, its fingerprint is created: $\frac{1}{\sqrt{t}}\sum_{j=1}^{t}|j\rangle\left(\cos\frac{2\pi s_j\sigma}{q}|0\rangle + \sin\frac{2\pi s_j\sigma}{q}|1\rangle\right)$.
4. The Hadamard transform turns the fingerprint into the state $|\psi\rangle = \left(\frac{1}{t}\sum_{l=1}^{t}\cos\frac{2\pi s_l\sigma}{q}\right)|0\rangle^{\otimes \log t}|0\rangle + \ldots$
5. The quantum state $|\psi\rangle$ is measured and the input is accepted iff the result is $|0\rangle^{\otimes \log t}|0\rangle$.

In [11, 15, 16], we have applied this technique to construct efficient quantum algorithms for a certain class of Boolean functions in the model of read-once quantum branching programs [17].

3.2.1. Quantum branching programs

Branching program is a well-known computational model in computer science, also known as a binary decision diagram in Applied Computer Science. Informally speaking, branching

http://dx.doi.org/10.5772/intechopen.70692

program is a circuit with ability to test in each of its computational step a needed bit of an input. Such circuit is a realization of a program that uses only "if then else" and "go to" primitives. We use the definition from [18]

Definition 1 ([18]) *A Quantum Branching Program Q over the Hilbert space $\mathcal{H}^d$ is defined as*

$$Q = \langle T, |\psi_0\rangle\rangle, \tag{9}$$

where T is a sequence of l instructions: $T_j = (x_{i_j}, U_j(0), U_j(1))$ *is determined by variable* x_{i_j} *tested on the step j, and* $U_j(0)$ and $U_j(1)$ *are unitary transformations in* $\mathcal{H}^d$.

Vectors $|\psi\rangle \in \mathcal{H}^d$ *are called states (state vectors) of Q,* $|\psi_0\rangle \in \mathcal{H}^d$ *is the initial state of Q.*

We define a computation of Q on an input $\sigma = \sigma_1, \ldots, \sigma_n \in \{0,1\}^n$ *as follows*:

1. *A computation of Q starts from the initial state* $|\psi_0\rangle$.
2. *The j-th instruction* of *Q reads the input symbol* σ_{i_j} *(the value of* x_{i_j}*) and applies the transition matrix* $U_j = U_j(\sigma_{i_j})$ *to the current state* $|\psi\rangle$ *to obtain the state* $|\psi'\rangle = U_j(\sigma_{i_j})|\psi\rangle$.
3. *The final state is*

$$|\psi(\sigma)\rangle = \left(\prod_{j=l}^{1} U_j\left(\sigma_{i_j}\right)\right)|\psi_0\rangle. \tag{10}$$

Accepting of an input sequence is a result of measuring of final state $|\psi(\sigma)\rangle$ in computational basis and is formalized as follows. Let A*ccept* $\subseteq \{1, 2, \ldots d\}$ be the set of indices of accepting basis states. After the *l*-th (last) step of quantum transformation, *Q* measures its configuration $|\psi_\sigma\rangle = (\alpha_1, \ldots, \alpha_d)^T$ and the input σ is accepted with probability

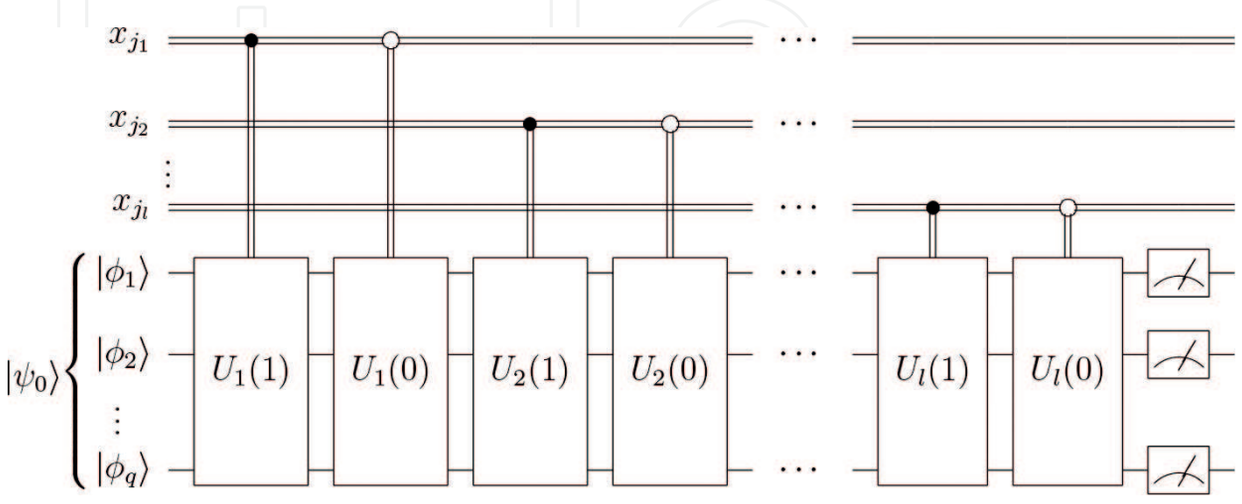

Figure 1. Branching program in the form of circuit. Variables $x_{i_1}, \ldots, x_{i_l}$ denoting classical control (input) bits. Single wires carry quantum information, and double wires denote classical information and control.

$$Pr_{\text{accept}}(\sigma) = \sum_{i \in \text{Accept}} |\alpha_i|^2. \tag{11}$$

3.2.2. Circuit representation

Quantum circuits are good formalism for quantum algorithms representation [19, 20]. A quantum branching programs can be viewed as a quantum circuit aided with an ability to read classical bits as control variables for unitary operations (see **Figure 1**).

4. Quantum hashing

In this section, we present notion of quantum (δ, ε)-resistant hash function based on [21].

4.1. One-way δ resistance

We present the following definition of a quantum δ-resistant one-way function. Let "information extracting" mechanism M be a function $M : \left(\mathcal{H}^2\right)^{\otimes s} \to \mathbb{X}$. Informally speaking, mechanism M makes some measurements to state $|\psi\rangle \in (\mathcal{H}^2)^{\otimes s}$ and decodes the result of measurement to $\mathbb{X}$.

Definition 2 ([21]) *Let X be a random variable distributed over $\mathbb{X}$ $\{\Pr[X = w] : w \in \mathbb{X}\}$. Let $\psi : \mathbb{X} \to \left(\mathcal{H}^2\right)^{\otimes s}$ be a quantum function. Let Y be any random variable over $\mathbb{X}$ obtained by some mechanism* **M** *making measurement to the encoding ψ of X and decoding the result of the measurement to $\mathbb{X}$. Let $\delta > 0$. We call a quantum function ψ a one-way δ-resistant function if*

1. *it is easy to compute, i.e., a quantum state $|\psi(w)\rangle$ for a particular $w \in \mathbb{X}$ can be determined using a polynomial-time algorithm.*
2. *for any mechanism* **M**, *the probability* $\Pr[Y = X]$ *that* **M** *successfully decodes Y is bounded by δ*

$$\Pr[Y = X] \leq \delta. \tag{12}$$

For the cryptographic purposes, it is natural to expect (and we do this in the rest of the paper) that random variable X is uniformly distributed.

A quantum state of $s \geq 1$ qubits can theoretically record an infinite amount of information. On the other hand, the Holevo's theorem [22] states that by a quantum measurement, one can extract $O(s)$ bits of information about the state. Here, we use the result of [23] motivated by the Holevo's theorem.

Property 1 ([23]) *Let X be a random variable uniformly distributed over $\{0,1\}^k$. Let $\psi : \{0,1\}^k \to (\mathcal{H}^2)^{\otimes s}$ be a quantum function. Let Y be a random variable over $\{0,1\}^k$ obtained by some mechanism* **M** *making some measurement of the encoding ψ of X and decoding the result of measurement to $\{0,1\}^k$. Then, the probability of correct decoding is given by*

http://dx.doi.org/10.5772/intechopen.70692

$$\Pr[Y = X] \leq \frac{2^s}{2^k}. \tag{13}$$

So, extracting an information on input σ from state $|\psi(\sigma)\rangle$ in conditions of Property 1 is "hard." The effectiveness of computation $|\psi(\sigma)\rangle$ depends on construction of quantum hash function ψ. In Section 4.4, we consider quantum hash function construction based on small-biased sets and prove effectiveness of this construction.

4.2. Collision ε resistance

The following definition was presented in [24].

Definition 3 *Let $\varepsilon > 0$. We call a quantum function $\psi : \mathbb{X} \to \left(\mathcal{H}^2\right)^{\otimes s}$ a collision ε-resistant function if for any pair w, w' of different inputs, $|\langle\psi(w)|\ \psi(w')\rangle| \leq \varepsilon$.*

Informally speaking, we need two states $|\psi(w)\rangle$ and $|\psi(w')\rangle$ that is almost orthogonal in order to get small probability of collision, that is, if one tests states $|\psi(w)\rangle$ and $|\psi(w')\rangle$ for equality, then a testing procedure should give positive result with a small probability. We start with quantum testing procedures.

4.2.1. Testing equality

The crucial procedure for quantum hashing is an equality test for $|\psi(v)\rangle$ and $|\psi(w)\rangle$ that can be used to compare encoded classical messages v and w. This procedure can be a well-known SWAP test [5] or something that is adapted for specific hashing function, like REVERSE test, see for example [6].

The SWAP test is the known quantum test for the equality of two unknown quantum states $|\psi\rangle$ and $|\psi'\rangle$ (see [6, 25] for more information).

We denote $Pr_{\text{SWAP}}[v = w]$ a probability that the SWAP test having quantum hashes $|\psi(v)\rangle$ and $|\psi(w)\rangle$ outputs the result "$v = w$" (outputs the result "$|\psi(v)\rangle = |\psi(w)\rangle$").

Property 2 ([6]) *Let function $\psi : w \mapsto |\psi(w)\rangle$ satisfy the following condition. For any two different elements $v, w \in \mathbb{X}$, it is true that $|\langle\psi(v)|\ \psi(w)\rangle| \leq \varepsilon$. Then,*

$$Pr_{\text{swap}}[v = w] \leq \frac{1}{2}\left(1 + \varepsilon^2\right). \tag{14}$$

Proof. From the description of SWAP test, it follows that

$$Pr_{\text{swap}}[v = w] = \frac{1}{2}\left(1 + |\langle\psi(v)|\psi(w)\rangle|^2\right). \tag{15}$$

4.2.1.1. REVERSE test

The test for equality, which we are presenting here, was first mentioned in [6]. In our paper [25], we call this test a REVERSE test. This test checks if a quantum state $|\psi\rangle$ is a hash of an element v by applying the procedure that inverts the creation of a quantum quantum

hash. That is, the REVERSE test procedure transforms the quantum hash to the initial quantum state.

Formally, let the procedure of quantum hashing, given initial state $|0\rangle$, maps the input w by unitary transformation $U(w)$: i.e., quantum hashing produces quantum state $|\psi(w)\rangle = U(w)|0\rangle$. Then, the REVERSE test, given v and $|\psi(w)\rangle$, applies $U^{-1}(v)$ to the state $|\psi(w)\rangle$ and measures the resulting state with respect to initial state $|0\rangle$. The output of REVERSE *test* is "$v = w$" iff the measurement outcome is $|0\rangle$. The output of REVERSE test is "$v \not= w$" iff the measurement outcome is different from $|0\rangle$. The probability that the *REVERSE test* having quantum state $|\psi(w)\rangle$ and an element v outputs the result $v = w$ are denoted by $Pr_{\text{REVERSE}}[v = w]$.

Property 3 ([23]) *Let hash function $\psi : w \mapsto |\psi(w)\rangle$ satisfies the following condition. For any two different elements, v and $w \in \mathbb{X}$, it is true that $|\langle\psi(v)|\,\psi(w)\rangle| \le \varepsilon$. Then,*

$$Pr_{\text{REVERSE}}[v = w] \le \varepsilon^2. \tag{16}$$

$$Pr_{\text{REVERSE}}[v = w] = |\langle 0|\,U^{-1}(v)\psi(w)\rangle|^2 = |\langle U^{-1}(v)\psi(v)|\,U^{-1}(v)\psi(w)\rangle|^2$$

$$= |\Big\langle \psi(v)|\psi(w)\rangle|^2 \le \varepsilon^2. \tag{17}$$

4.3. Balanced quantum (δ, ε) resistance

The combination of one-way and collision-resistant function definitions gives the definition of quantum cryptographic function.

Definition 4 ([21]) *Let $K = |\mathbb{X}|$ and $s \ge 1$. Let $\delta > 0$ and $\varepsilon > 0$. We call a function $\psi : \mathbb{X} \to \left(\mathcal{H}^2\right)^{\otimes s}$ a quantum (δ, ε)-hash function iff ψ is one-way δ-resistant and is collision ε-resistant function.*

We present below the following two examples to demonstrate how one-way δ resistance and collision ε resistance are correlated. The first example was presented in [4] in terms of quantum automata.

Example 1 *Let $v \in \{0, \ldots, 2^k - 1\}$. Number v is encoded by a single qubit as follows:*

$$\psi : v \mapsto \cos\left(\frac{2\pi v}{2^k}\right)|0\rangle + \sin\left(\frac{2\pi v}{2^k}\right)|1\rangle. \tag{18}$$

Extracting information from $|\psi\rangle$ by measuring $|\psi\rangle$ with respect to the basis $\{|0\rangle, |1\rangle\}$ gives the following result. The function ψ is one-way $\frac{2}{2^k}$ resistant (see Property 1) and collision $\cos(\pi/2^{k-1})$ resistant. Thus, the function ψ has a good one-way property but has a bad collision resistance property for large k.

Clearly, that one can store (*to hash*) in this way an arbitrary large amount of classical information, that is, for arbitrary large k one can store all numbers from $\{0, \ldots, 2^k - 1\}$ in a single qubit. Holevo bound [22] proves that given $s \ge 1$ qubits, the amount of classical information that can

be retrieved, i.e., accessed, can be only up to s classical bits. This is a quantum mechanical approach for the one-way property.

The map ψ is one to one. So, there is no collision in a "quantum level." But extracting the result from quantum state is a probabilistic procedure. This means that one can get the situation when some procedure that tests the equality of different quantum hashes $|\psi(v)\rangle$, $|\psi(w)\rangle$ outputs "the hashes are the same" (equivalently "the numbers v, w are the same"), while the numbers v and w are different. For example, two numbers 0 and 2^{k-2} generate orthogonal states $|\psi(0)\rangle = |1\rangle$ and $|\psi(2^{k-2})\rangle = |0\rangle$. So, numbers 0 and 2^{k-2} are distinguishably reliable in respect of the above encoding. But two numbers 0 and 1 cannot be reliably distinguished by encoding ψ.

Example 2 *Binary word* $v = \sigma_1, \ldots, \sigma_k \in \{0,1\}^k$ *encoded by k qubits (each bit encoded by a qubit):* ψ: $v \mapsto |v\rangle = |\sigma_1\rangle, \cdots, |\sigma_k\rangle$.

Clearly, we have that such encoding is collision one-way, 1-resistant, and 0-resistant. So, in contrast to Example 1, the encoding ψ from Example 2 for different words v and w, their images (quantum states) $|\psi(v)\rangle$ and $|\psi(v)\rangle$ are orthogonal and therefore reliably distinguished; but ψ is easily invertible: the function ψ is not one-way resistant.

The following result [24] proves that a quantum collision ε-resistant function needs at least $\log\log K - c(\varepsilon)$ qubits.

Property 4 ([24]) *Let* $s \geq 1$ *and* $K = |\mathbb{X}| \geq 4$. *Let* $\psi : \mathbb{X} \to \left(\mathcal{H}^2\right)^{\otimes s}$ *be a collision* ε*-resistant quantum hash function. Then,*

$$s \geq \log\ \log\ K - \log\ \log\left(1 + \sqrt{2/(1-\varepsilon)}\right) - 1. \tag{19}$$

Proof. First, we observe that from the definition $\||\psi\rangle\| = \sqrt{\langle\psi|\psi\rangle}$ of the norm, it follows that

$$\||\psi\rangle - |\psi'\rangle\|^2 = \||\psi\rangle\|^2 + \||\psi'\rangle\|^2 - 2\langle\psi|\psi'\rangle. \tag{20}$$

Hence, for an arbitrary pair w, w' of different elements from $\mathbb{X}$, we have that

$$\||\psi(w)\rangle - |\psi(w')\rangle\| \geq \sqrt{2(1-\varepsilon)}. \tag{21}$$

We let $\Delta = \sqrt{2(1-\varepsilon)}$. For short, we let $(\mathcal{H}^2)^{\otimes s} = V$ in this proof. Consider a set $\Phi = \{|\psi(w)\rangle : w \in \mathbb{X}\}$. If we draw spheres of radius $\Delta/2$ with centers $|\psi\rangle \in \Phi$, then spheres do not pairwise intersect. All these K spheres are in a large sphere of radius $1 + \Delta/2$. The volume of a sphere of radius r in V is $cr^{2^{s+1}}$ for the complex space V. The constant c depends on the metric of V. From this, we have that the number K is bonded by the number of "small spheres" in the "large sphere"

$$K \leq \frac{c(1+\Delta/2)^{2^{s+1}}}{c(\Delta/2)^{2^{s+1}}}. \tag{22}$$

Hence,

$$s \geq \log \log K - \log \log \left(1 + \sqrt{2/(1-\varepsilon)}\right) - 1. \tag{23}$$

Properties 1 and 4 provide a basis for building a "balanced" one-way δ-resistance and collision ε-resistance properties. That is, roughly speaking, if we need to hash elements w from the domain $\mathbb{X}$ with $|\mathbb{X}| = K$ and if one can build for an $\varepsilon > 0$ a collision ε-resistant $(K; s)$ hash function ψ with $s \approx \log\log K - c(\varepsilon)$ qubits, then the function f is one-way δ resistant with $\delta \approx (\log K/K)$. Such a function is balanced with respect to Property 4.

To summarize the above considerations, we can state the following. A quantum (δ, ε)-hash function is a function that satisfies all of the properties that a "classical" hash function should satisfy. Preimage resistance follows from Property 1. Second preimage resistance and collision resistance follow, because all inputs are mapped to states that are nearly orthogonal. Therefore, we see that quantum hash functions can satisfy the three properties of a classical cryptographic hash function.

4.4. Quantum hash functions construction via small-biased sets

This section is based on the paper [26]. We first present a brief background on ε-biased sets. For more information, see [27]. Note that ε-biased sets are generally defined for arbitrary finite groups, but here we restrict ourselves to $\mathbb{Z}_q$.

For an $a \in \mathbb{Z}_q$, a character χ_a of $\mathbb{Z}_q$ is a homomorphism $\chi_a : \mathbb{Z}_q \to \mu_q$, where μ_q is the (multiplicative) group of complex q-th roots of unity. That is, $\chi_a(x) = \omega^{ax}$, where $\omega = e^{\frac{2\pi i}{q}}$ is a primitive q-th root of unity. The character $\chi_0 \equiv 1$ is called a trivial character.

Definition 5 *A set $S \subseteq \mathbb{Z}_q$ is called ε biased, if for any nontrivial character $\chi \in \{\chi_a : a \in \mathbb{Z}_q\}$*

$$\frac{1}{|S|} \left| \sum_{x \in S} \chi(x) \right| \leq \varepsilon. \tag{24}$$

These sets are interesting when $|S| \ll |\mathbb{Z}_q|$ (as $S = \mathbb{Z}_q$ is 0 biased). In their seminal paper, Naor and Naor [13] defined these small-biased sets, gave the first explicit constructions of such sets, and demonstrated the power of small-biased sets for several applications.

Remark 1 *Note that a set S of $O(\log q/\varepsilon^2)$ elements selected uniformly at random from $\mathbb{Z}_q$ is ε biased with positive probability* [28].

Many other constructions of small-biased sets followed during the last decades.

Vasiliev [26] showed that ε-biased sets generate (δ,ε)-resistant hash functions. We present the result of [26] in the following form.

Theorem 1 *Let $S \subseteq \mathbb{Z}_q$ be an ε-biased set. Let $H_S = \{h_a(x) = ax (\mathrm{mod}\ q), \quad a \in S, h_a : \mathbb{Z}_q \to \mathbb{Z}_q\}$ be a set of functions determined by S. Then, a quantum function $\psi_{H_S} : \mathbb{Z}_q \to (\mathcal{H}^2)^{\otimes \log |S|}$*

$$|\psi_{H_S}(x)\rangle = \frac{1}{\sqrt{|S|}} \sum_{a \in S} \omega^{h_a(x)} |a\rangle \tag{25}$$

is a (δ, ε)-resistant quantum hash function, where $\delta \leq |S|/q$.

http://dx.doi.org/10.5772/intechopen.70692

Proof. One-way δ-resistance property of ψ_{H_s} follows from Property 1: a probability of correct decoding an x from a quantum state $|\psi_{H_s}(x)\rangle$ is bounded by $|S|/q$.

Collision ε-resistance property of ψ_{H_s} follows directly from the corresponding property of [26]. Note that

$$|\psi_{H_s}(x)\rangle = \frac{1}{\sqrt{|S|}}\sum_{a\in S}\omega^{h_a(x)}|a\rangle = \frac{1}{\sqrt{|S|}}\sum_{a\in S}\chi_x(a)|a\rangle. \tag{26}$$

We will prove that for arbitrary different elements $v, v^{'} \in \mathbb{Z}_q$, it is true that

$$\left|\left\langle \psi_{H_s}(v)|\psi_{H_s}(v')\right\rangle\right| = \frac{1}{|S|}|\sum_{a\in S}\chi_v^*(a)\chi_{v'}(a)| \le \varepsilon. \tag{27}$$

Let $\chi_v(x)$ and $\chi_{v'}(x)$ be characters of group $\mathbb{Z}_q$. Then, $\chi_v^*(x)$ is also a character of $\mathbb{Z}_q$ and so the following function is $\chi(x) = \chi_v^*(x)\chi_{v'}(x)$. $\chi(x)$ is nontrivial character of $\mathbb{Z}_q$, since $\chi_v(x) \not\equiv \chi_{v'}(x)$ and $\chi(x) = \chi_v^*(x)\chi_{v'}(x) \not\equiv \chi_v^*(x)\chi_v(x) \equiv 1$, where 1 is a trivial character of $\mathbb{Z}_q$. Thus, the statement of Theorem 1 follows from the definition of an ε-biased set.

$$|\langle\psi_{H_s}(v)|\psi_{H_s}(v')\rangle| = \frac{1}{|S|}|\sum_{a\in S}\chi_v^*(a)\chi_{v'}(a)| = \frac{1}{|S|}|\sum_{a\in S}\chi(a)| \le \varepsilon. \tag{28}$$

4.5. Quantum fingerprinting functions as hash functions

In this section, we give two explicit examples of the quantum hashing for specific finite abelian groups, which turn out to be the known quantum fingerprinting schemas.

4.5.1. Hashing the elements of the Boolean cube

For $G = \mathbb{Z}_2^n$, its characters can be written in the form $\chi_a(x) = (-1)^{(a,\,x)}$, and the corresponding quantum hash function is the following

$$|\psi_S(a)\rangle = \frac{1}{\sqrt{|S|}}\sum_{j=1}^{|S|}(-1)^{(a,s_j)}|j\rangle. \tag{29}$$

The resulting hash function is exactly the quantum fingerprinting by Buhrman et al. [5], once we consider an error-correcting code, whose matrix is built from the elements of S. Indeed, as stated in [29] an ε-balanced error-correcting code can be constructed out of an ε-biased set. Thus, the inner product (a, x) in the exponent is equivalent to the corresponding bit of the code word, and altogether, this gives the quantum fingerprinting function that stores information in the phase of quantum states de Wolf [30].

4.5.2. Hashing the elements of the cyclic group

For group $G = \mathbb{Z}_q$, the corresponding quantum hash function is given by

$$|\psi_S(a)\rangle = \frac{1}{\sqrt{|S|}} \sum_{j=1}^{|S|} \omega^{as_j} |j\rangle. \tag{30}$$

The above quantum hash function is essentially equivalent to the one we have defined earlier in [25], which is in turn based on the quantum fingerprinting function from [11].

- In the content of the definition of quantum hash generator [24] and the above consideration, it is natural to call the set H_S of functions (formed from ε-biased set S) a *uniform quantum* (δ, ε)*-hash generator* for $\delta = O(|S|/(q \log q))$.

As a corollary from Theorem 1 and the above consideration, we can state the following.

Property 5 *For an* ε*-biased set* $S = \{a_1, \ldots, a_T\} \subset \mathbb{F}_q$ *with* $T = O(\log q/\varepsilon^2)$*, for* $s = \log T$*, for* $\delta = O(1/(q\varepsilon^2))$*, a quantum uniform* (δ, ε)*-hash generator* H_S *generates quantum* (δ, ε)*-hash function*

$$\psi_{H_S} : \mathbb{F}_q \to \left(\mathcal{H}^2\right)^{\otimes s} \tag{31}$$

$$|\psi_{H_S}(x)\rangle = \frac{1}{\sqrt{T}} \sum_{j=0}^{T-1} \omega^{a_j x}, \tag{32}$$

5. Computing a quantum hash $|\psi_{H_S}(x)\rangle$ by QBP

Theorem 2 *Quantum* (δ, ε)*-hash function (6)*

$$\psi_{H_S} : \mathbb{F}_q \to \left(\mathcal{H}^2\right)^{\otimes s} \tag{33}$$

can be computed by quantum branching program Q *composed from* $s = O(\log \log q)$ *qubits in* $\log q$ *steps.*

Proof. Quantum function ψ_{H_S} (6) for an input $x \in \mathbb{F}_q$ determines quantum states (7)

$$|\psi_{H_S}(x)\rangle = \frac{1}{\sqrt{T}} \sum_{j=0}^{T-1} \omega^{a_j x} |j\rangle, \tag{34}$$

which is a result of quantum Fourier transformation (QFT) of the initial state

$$|\psi_0\rangle = \frac{1}{\sqrt{T}} \sum_{j=0}^{T-1} |j\rangle. \tag{35}$$

Such a QFT is controlled by the input x. QBP Q for computing quantum hash $|\psi_{H_S}(x)\rangle$ determined as follows. We represent an integer $x \in \{0, \ldots, q-1\}$ as the bit-string $x = x_0 \ldots x_{\log q - 1}$ that is, $x = x_0 + 2^1 x_1 + \ldots + 2^{\log q - 1} x_{\log q - 1}$. For a binary string $x = x_0 \ldots x_{\log q - 1}$ a quantum branching program Q over the space $(\mathcal{H}^2)^{\otimes s}$ for computing $|\psi_{H_S}(x)\rangle$ (composed of $s = \log T$ qubits) is defined as

http://dx.doi.org/10.5772/intechopen.70692

$$Q = \langle \mathbb{T}, |\psi_0\rangle \rangle, \tag{36}$$

where $|\psi_0\rangle$ is the initial state and $\mathbb{T}$ is a sequence of log q instructions:

$$\mathbb{T}_j = \left(x_j, U_j(0), U_j(1)\right) \tag{37}$$

is determined by the variable x_j tested on the step j, and $U_j(0)$ and $U_j(1)$ are unitary transformations in $(\mathcal{H}^2)^{\otimes s}$. More precise $U_j(0)$ is $T \times T$ identity matrix. $U_j(1)$ is the $T \times T$ diagonal matrix whose diagonal entries are $\omega^{a_0 2^j}$, $\omega^{a_1 2^j}$, …, $\omega^{a_{T-1} 2^j}$ and the off-diagonal elements are all zero. That is,

$$U_j(1) = \begin{bmatrix} \omega^{a_0 2^j} & & & \\ & \omega^{a_1 2^j} & & \\ & & \ddots & \\ & & & \omega^{a_{T-1} 2^j} \end{bmatrix}. \tag{38}$$

We define a computation of Q on an input $x = x_0, \ldots, x_{\log q - 1} \in \{0,1\}^{\log q}$ as follows:

1. A computation of Q starts from the initial state $|\psi_0\rangle$.

2. The j-th instruction of Q reads the input symbol x_j (the value of x) and applies the transition matrix $U_j(x_j)$ to the current state $|\psi\rangle$ to obtain the state $|\psi'\rangle = U_j(x_j)|\psi\rangle$.

3. The final state is

$$|\psi_{H_s}(x)\rangle = \left(\prod_{j=0}^{\log q - 1} U_j(x_j)\right)|\psi_0\rangle. \tag{39}$$

5.1. Complexity measures

We consider the following notations. For the QBP Q from Theorem 2, we let $width(Q) = s$ and $time(Q) = |\mathbb{T}|$. Next for quantum hash function ψ_{H_s} (6), we let

$$width(\psi_{H_s}) = \min width(Q), \qquad time(\psi_{H_s}) = \min time(Q) \tag{40}$$

where minimum is taken over all QBPs that compute ψ_{H_s}.

5.1.1. Upper bounds

Then from Theorem 2, we have the following corollary

Theorem 3

$$width(\psi_{H_s}) = O(\log \log q), \tag{41}$$

$$time(\psi_{H_s}) = O(\log q). \tag{42}$$

5.1.2. Lower bounds

Here, we show that the quantum branching program from Theorem 2 is optimal for function ψ_{H_s}

Theorem 4

$$width(\psi_{H_s}) = \Omega(\log \log q), \tag{43}$$

$$time(\psi_{H_s}) = \Omega(\log q). \tag{44}$$

Proof. Let Q be a QBP for the function ψ_{H_s} computation. ψ_{H_s} presented by Q as follows:

$$\psi_{H_s} : \{|\psi_0\rangle\} \times \{0,1\}^{\log q} \to \left(\mathcal{H}^2\right)^{\otimes s}. \tag{45}$$

The lower bound (10) for $width(\psi_{H_s})$ follows immediately from Property 4

$$s \geq \log \log q - \log \log \left(1 + \sqrt{2/(1-\varepsilon)}\right). \tag{46}$$

The lower bound (11) for $time(\psi_{H_s})$ follows from the fact that ψ_{H_s} is collision ε-resistant function. Indeed, the assumption that QBP Q for ψ_{H_s} can test less than log$q$ (that is, not all log$q$) variables of inputs $x \in \mathbb{F}_q$ means existence of (at least) two different inputs $w, w' \in \mathbb{F}_q$ such that Q produces the same quantum hashes $|\psi(w)\rangle$ and $|\psi(w')\rangle$ for w and w', that is, $|\psi(w)\rangle = |\psi(w')\rangle = |\psi\rangle$. The last contradicts to the fact that states $|\psi(w)\rangle$ and $|\psi(w')\rangle$ are ε orthogonal.

$$|\langle\psi(w)|\psi(w')\rangle| \leq \varepsilon. \tag{47}$$

6. Concluding remarks

To conclude, we first like to mention the results of the paper [31], which presents further development of quantum hash functions construction.

Recall that any ε-biased set gives rise to a Cayley expander graph [28]. We show how such graphs generate balanced quantum hash functions. Every expander graph can be converted to a bipartite expander graph. The generalization of these bipartite expander graphs is the notion of extractor graphs. Such point of view gives a method for constructing quantum hash functions based on extractors. This construction of quantum hash functions is applied to define the notion of keyed quantum hash functions. The latter is used for constructing quantum hash-based message authentication codes (QMAC). The security proof of QMAC is based on using strong extractors against quantum storage developed by Ta-Shma [32].

Secondly, in [24], we offered a design that allows to build a large amount of different quantum hash functions. The construction is based on composition of classical δ-universal hash family and a given family $H_{\delta,q}$, a quantum hash generator. A resulting family of functions is a new quantum hash generator. In particular, we present a quantum hash generator G_{RS} based on Reed-Solomon code.

Author details

Farid Ablayev* and Marat Ablayev

*Address all correspondence to: fablayev@gmail.com

Kazan Federal University, Kazan, Russia

References

[1] Motwani R, Raghavan P. Randomized Algorithms. New York, USA: Cambridge University Press; 1995

[2] Rusins Freivalds. Probabilistic machines can use less running time. In: IFIP Congress. Vol. 839; 1977. pp. 842

[3] Freivalds R. Fast probabilistic algorithms. In: Becvar J, editor. Mathematical Foundations of Computer Science — Lecture Notes in Computer Science. Vol. 74. Heidelberg: Springer Berlin; 1979. p. 57-69

[4] Andris Ambainis, Rusins Freivalds. 1-way quantum finite automata: Strengths, weaknesses and generalizations. In: Proceeding of the 39th IEEE Conference on Foundation of Computer Science, FOCS '98, Washington DC USA: IEEE Computer Society; 1998. pp. 332-342

[5] Buhrman H, Cleve R, Watrous J, Wolf R d. Quantum fingerprinting. Physical Review Letters. 2001;**87**(16):167902

[6] Gottesman D, IC. Quantum digital signatures technical report arXiv:Quant-ph/0105032

[7] Gavinsky D, Ito T. Quantum fingerprints that keep secrets. Quantum Information & Computation. 2013;**13**(7–8):583-606

[8] Li D, Zhang J, Guo F-Z, Huang W, Wen Q-Y, Chen H. Discrete-time interacting quantum walks and quantum hash schemes. Quantum Information Processing. 2013;**12**(3):1501-1513

[9] Li D, Zhang J, Ma X-W, Zhang W-W, Wen Q-Y. Analysis of the two-particle controlled interacting quantum walks. Quantum Information Processing. 2013;**12**(6):2167-2176

[10] Yang Y-G, Peng X, Yang R, Zhou Y-H, Shi W-M. Quantum hash function and its application to privacy amplification in quantum key distribution, pseudo-random number generation and image encryption. Scientific Reports. 2016;**6**:19788

[11] Ablayev F, Vasiliev A. Algorithms for quantum branching programs based on fingerprinting. Electronic Proceedings in Theoretical Computer Science. 2009;**9**:1-11

[12] Ambainis A, Nahimovs N. Improved constructions of quantum automata. In: Kawano Y, Mosca M, editors. Theory of Quantum Computation, Communication, and Cryptography — Lecture Notes in Computer Science. Vol. 5106. Berlin/Heidelberg: Springer; 2008. p. 47-56

[13] Joseph Naor, Moni Naor. Small-bias probability spaces: Efficient constructions and applications. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, STOC '90, NY, USA, New York: ACM; 1990. 213-223

[14] Eli Ben-Sasson, Madhu Sudan, Salil Vadhan, Avi Wigderson. Randomness-efficient low degree tests and short pcps via epsilon-biased sets. In: Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, STOC '03, NY, USA, New York: ACM; 2003. pp. 612-621

[15] Ablayev F, Vasiliev A. On computational power of quantum read-once branching programs. Electronic Proceedings in Theoretical Computer Science. 2011;**52**:1-12

[16] Farid Ablayev, Alexander Vasiliev. Classical and quantum parallelism in the quantum fingerprinting method. In: Victor Malyshkin, editor. 11th International Conference PaCT 2011 Proceedings — Lecture Notes in Computer Science. Vol. 6873. Springer; 2011. pp. 1-13

[17] Ablayev F, Gainutdinova A, Karpinski M. On computational power of quantum branching programs. FCT. 2001;59-70

[18] Ablayev F, Gainutdinova A, Karpinski M, Moore C, Pollett C. On the computational power of probabilistic and quantum branching programs of constant width. Information and Computation. 2005;**203**:145-162

[19] Deutsch D. Quantum computational networks. Royal Society of London Proceedings Series A. 1989;**425**:73-90

[20] Andrew Chi-Chih Yao. Quantum circuit complexity. In: Proceedings of Thirty-fourth IEEE Symposium on Foundations of Computer Science. Palo Alto California USA: IEEE Computer Society; 1993. pp. 352-361

[21] Ablayev F, Ablayev M. On the concept of cryptographic quantum hashing. Laser Physics Letters. 2015;**12**(12):125204

[22] Alexander SH. Some estimates of the information transmitted by quantum communication channel (Russian). Probl. Pered. Inform Problems of Information Transmission. 1973;**9**(3):3-11

[23] Ashwin Nayak. Optimal lower bounds for quantum automata and random access codes. In: Foundations of Computer Science. 40th Annual Symposium on; 1999. pp. 369-376

[24] Ablayev F, Ablayev M. Quantum hashing via ε-universal hashing constructions and classical fingerprinting. Lobachevskii Journal of Mathematics. 2015;**36**(2):89-96

[25] Ablayev FM, Vasiliev AV. Cryptographic quantum hashing. Laser Physics Letters. 2014;**11**(2):025202

[26] Vasiliev A. Quantum hashing for finite abelian groups. Lobachevskii Journal of Mathematics. 2016;**37**(6):751-754

[27] Chen S, Moore C, Russell A. Small-bias sets for nonabelian groups. In: Raghavendra P, Raskhodnikova S, Jansen K, Rolim JDP, editors. Approximation, Randomization, and

Combinatorial Optimization. Algorithms and Techniques — Lecture Notes in Computer Science. Vol. 8096. Berlin Heidelberg: Springer; 2013. p. 436-451

[28] Alon N, Roichman Y. Random cayley graphs and expanders. Random Structures & Algorithms. 1994;**5**(2):271-284

[29] Ben-Aroya A., Ta-Shma A. Constructing small-bias sets from algebraic-geometric codes. In: Foundations of Computer Science, FOCS '09. 50th Annual IEEE Symposium on; 2009. pp. 191-197

[30] R d W. Quantum Computing Communication Complexity [Ph.D. Thesis]. Amsterdam: University of Amsterdam; 2001

[31] Ziatdinov M. From graphs to keyed quantum hash functions. Lobachevskii Journal of Mathematics. 2016;**37**(6):704-711

[32] Ta-Shma A. Short seed extractors against quantum storage. Proceedings of the ACM STOC. 2009;401-408

IntechOpen