Math-Net.Ru

Общероссийский математический портал

A. V. Vasiliev, M. T. Ziatdinov, Минимизация коллизий при квантовом хешировании, *Матем. вопр. криптогр.*, 2016, том 7, выпуск 2, 47–52

DOI: <https://doi.org/10.4213/mvk182>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 35.173.137.237

26 апреля 2023 г., 17:43:27

Minimizing collisions for quantum hashing

A. V. Vasiliev, M. T. Ziatdinov

Kazan Federal University, Kazan, Russia

Получено 18.II.2015

Abstract. We present explicit algorithms for computation of quantum hashing parameters that minimize the probability of encountering quantum collisions.

Keywords: quantum computation, quantum hashing, hashing collisions, genetic algorithm, simulated annealing

Минимизация коллизий при квантовом хешировании

А. В. Васильев, М. Т. Зиятдинов

Казанский государственный университет, Казань

Аннотация. Предлагаются явные алгоритмы вычисления параметров квантового хеширования, минимизирующих вероятность возникновения квантовых коллизий.

Ключевые слова: квантовые вычисления, квантовое хеширование, коллизии хеширования, генетические алгоритмы, моделируемое остывание

1. Introduction

Hashing is a well-known technique, widely used in computer science. Following the ideas and properties of the cryptographic hashing [1] we have proposed its quantum analogue in [2]. Just like in classical case it may find applications in different communication scenarios including single-bit quantum digital signature protocol from [3] and quantum communication protocols (e.g. in one-way quantum communication model and simultaneous message passing model [4]).

The key property of both classical and quantum hashing is the collision resistance. In [2] we have analyzed the set of numeric parameters for quantum hashing that determine its collision resistance. In this paper we investigate the construction of that set in more detail. Although there was a general method of obtaining good hashing parameters, it makes sense for comparatively large inputs. That is why we construct different algorithms to complement the general one. In particular, we give two heuristic algorithms for this problem: a genetic approach and annealing simulation.

2. Preliminaries

In this section we recall a definition of quantum hash function from [2].

Let

$$q = 2^n \quad \text{and} \quad B = \{b_1, b_2, \dots, b_d\} \subset \mathbb{Z}_q.$$

We define a quantum hash function

$$\psi_{q,B} : \{0, 1\}^n \rightarrow (\mathcal{H}^2)^{\otimes (\log d + 1)}$$

as follows. For an input $x \in \{0, 1\}^n$ let

$$|\psi_{q,B}(x)\rangle = \frac{1}{\sqrt{d}} \sum_{i=1}^d |i\rangle \left(\cos \frac{2\pi b_i x}{q} |0\rangle + \sin \frac{2\pi b_i x}{q} |1\rangle \right). \quad (1)$$

It follows from this definition that the quantum hash $|\psi_{q,B}(x)\rangle$ of an n -bit string x consists of $\log d + 1$ qubits. We have shown that d may be of the order $O(n)$ without loosing the quality of hashing [2].

In [2] we have discussed the notion of *quantum collision*. The reason why we have defined it is the observation that in quantum hashing there might be no collisions in the classical sense: quantum hashes being the quantum states may store arbitrary amount of data and may be different for unequal messages. But the procedure of comparing those quantum states implies measurement, which may lead to collision-type errors.

So, a *quantum collision* is a situation when a procedure that tests an equality of quantum hashes outputs true, while hashes are different. This procedure may be a well-known SWAP-test (see for example [2] for more information and citations) or something that is adapted for specific hash function. Anyway, it deals with the notion of distinguishability of quantum states. And since non-orthogonal quantum states cannot be perfectly distinguished, we require them to be “nearly orthogonal”.

The set

$$B = \{b_1, b_2, \dots, b_d\}$$

of hashing parameters not only determines the size of the hash but also gives the function $\psi_{q,B}$ an ability to withstand collisions, i. e. to distinguish different hashes with bounded error probability. We have called this property *δ -resistance*.

Formally, for $\delta \in (0, 1)$ we say that a function

$$\psi : \mathbb{X} \rightarrow (\mathcal{H}^2)^{\otimes s}$$

is *δ -resistant* if for any pair w, w' of different inputs

$$|\langle \psi(w) | \psi(w') \rangle| \leq \delta. \quad (2)$$

The value of δ for the hash function $\psi_{q,B}$ entirely depends on q (which is fixed here by the size of the input) and on the set B , i. e.

$$\delta = \delta(q, B).$$

In [2] we have shown a construction for the set of polylogarithmic size (in q) based on [5].

We have also proved the following result.

Theorem. *For arbitrary $\delta \in (0, 1)$ there exists a set*

$$B = \{b_1, b_2, \dots, b_d\}$$

of size

$$d = \lceil (2/\delta^2) \ln(2q) \rceil$$

such that the quantum hash function $\psi_{q,B}$ is δ -resistant.

In other words, for arbitrary $\delta \in (0, 1)$ it is possible to construct a δ -resistant quantum hash function $\psi_{q,B}$ that would produce a qubit hash of size

$$\log d + 1 = O(\log \log q) = O(\log n)$$

out of n -bit input.

3. Optimization problem

It is easy to see that for the function $\psi_{q,B}(x)$ we have

$$|\langle \psi_{q,B}(w) | \psi_{q,B}(w') \rangle| = \left| \frac{1}{d} \sum_{i=1}^d \cos \frac{2\pi b_i(w - w')}{q} \right|,$$

and we want this function to be smaller than some δ for any value of $(w - w')$ except for 0. Thus, the optimization problem that aroused here is the following.

For a fixed q minimize the target function

$$\delta(q, B) = \max_{x \neq 0} \left| \frac{1}{d} \sum_{i=1}^d \cos \frac{2\pi b_i x}{q} \right|$$

over all $B = \{b_1, \dots, b_d\} \subset \mathbb{Z}_q$.

The best possible solution exists for $B = \mathbb{Z}_q$, since $\delta(q, \mathbb{Z}_q) = 0$. However, this would mean that the size of the hash is

$$\log q + 1 = n + 1,$$

i. e. even larger than the input, and hashing loses one of its important properties. So, we require that $d \ll q$, and we actually solve the above optimization problem several times for increasing d until it gives us the set B with desired value of $\delta(q, B)$.

4. Genetic algorithm

The idea of genetic algorithms is described e. g. in [6]. Research in this area has started in 1954 and became widely spread in 1970s-1980s.

When applied to our optimization problem:

- a phenotype is the set B sorted in ascending order,
- a fitness function is given by $\delta(q, B)$,
- a mutation is an increment or decrement of a random element of B ,
- a crossover is performed by splitting sets in two parts and exchanging them.

To start the algorithm, we randomly generate a family of sets (a population). Then the population is evolved as follows.

First of all, the population undergoes sudden mutations: we randomly pick several individuals and randomly “mutate” them, i. e. change the random element of a set by one.

Then for all individuals the value of the fitness function is evaluated. The half of all individuals with the best results give the next generation: we pick random pairs of phenotypes, their genotypes are split in two parts and exchanged in such a way that the values of the first parts are less or equal to the values of the second parts.

Finally, we remove the individuals with the worst fitness until the population has the initial size.

The evolution process repeats the given number of iterations or until a good enough solution is found. Thus, we need some value of δ as an input parameter.

5. Simulated annealing

We also have developed a simulated annealing algorithm to compute the set B . This algorithm is a heuristic search algorithm and it is described in [7]. We used `concurrent-sa` library for Haskell language for general procedure of simulated annealing.

Simulated Annealing is inspired by a physical process of melting some substance and then lowering the temperature slowly. This process allows the substance to get to optimal state (i. e. the state with the lowest energy).

So we generate a population of random sets and allow them to evolve into the other (neighbour) states according to the current temperature. This temperature slowly decreases. After sufficient time population will have sets with low δ . To change a set we randomly change one element of the set.

We have run simulated annealing for fixed time (1 sec) with population of 1000 random sets.

6. Acknowledgements

The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University. Work was in part supported by the Russian Foundation for Basic Research (under the grant 14-07-00878).

References

- [1] Rogaway P., Shrimpton T., “Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance”. In: *“Fast Software Encryption”*, FSE 2004, Lect. Notes Comput. Sci., **3017**, 2004, 371–388.
- [2] F. Ablyayev, A. Vasiliev, “Cryptographic quantum hashing”, *Laser Physics Letters*, **11**:2 (2014), 025202.
- [3] Gottesman D., Chuang I., “Quantum digital signatures”, arXiv:quant-ph/0105032.
- [4] Vasiliev A., “Quantum communications based on quantum hashing”, arXiv:1312.1661.
- [5] Razborov A. A., Szemerédi E., Wigderson A., “Constructing small sets that are uniform in arithmetic progressions”, *Comb. Probab. Comput.*, **2**:4 (1993), 513–518.
- [6] Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed., rev. extend., Heidelberg etc.: Springer, 1996, xx+387 pp.
- [7] Kirkpatrick S., Gelatt C. D., Jr., Vecchi M. P., “Optimization by simulated annealing”, *Science*, **220**:4598, May 27 (1983), 671–680.