STUDYING QUANTUM HASHING CRYPTOGRAPHIC STRENGTH

Victor O. Georgiev*
Ph.D. in technical sciences, senior lecturer, department of software technologies, Kazan Federal University (KFU), Kazan, Russia
Vladislav M. Gorbunov
undergraduate, department of software technologies, Kazan Federal University (KFU), Kazan, Russia
Nikolai A. Prokopyev
master of technical sciences, assistant lecturer, department of software technologies, Kazan Federal University (KFU), Kazan, Russia
Rustam A. Burnashev
master of technical sciences, assistant lecturer, department of software technologies, Kazan Federal University (KFU), Kazan, Russia, info@ores.su

Abstract. This research solves the problem of studying quantum hashing cryptographic strength. The most important criteria, that should be taken into consideration during cryptographic strength studying, is quantum hashing strength against collisions, and irreversibility of quantum hash-functions. Strength against collisions for selected quantum hash-function depends on many numeric parameters, and it is necessary to find a corresponding optimization solution. It is necessary to conduct comparative analysis of known methods in this research to achieve the goal and offer new methods to deliver the result. In the course of research different algorithms were used and modified to ensure cryptographic strength of quantum hash-functions, and an algorithm on the basis of linear codes is developed to find a decision in case of high dimensionalities.

Keywords: Quantum computing, quantum cryptography, quantum hashing.

1.Introduction . Preserving information confidentiality is one of the most important current topics. Every day a large quantity of various information is transmitted through various communication channels. Various encryption algorithms are used to protect information. Classical cryptography is based on one-way functions, such as the prime factorization task. Note that this task for a quantum computer is effectively solvable.

Quantum cryptography offers its own approaches to ensure information confidentiality. A particularly important approach is reflected in the quantum information theory, which proposes the technique of quantum hashing. The approach aims to ensure the cryptographic stability of protocols based on the laws of quantum mechanics. It offers various quantum hash functions, and analyzes properties the functions should have. To use cryptographic algorithms, it is necessary to convert the input information with the help of hash functions that map an input array of random length to an output bit string of a certain length. In the quantum case, these functions must have at least two important properties: one-wayness and strength against quantum collisions.

This paper solves the problem of researching into perfect cryptography of quantum hashing. The most important criteria that should be considered in the study of cryptographic stability are the strength of quantum hashing against collisions, and one-wayness of quantum hash functions. The collision resistance for the selected quantum hash function depends on the set of numeric parameters, to find which, it is necessary to solve the corresponding optimization problem.

2.Methods. Let us consider the basic concepts and terminology used in the article to determine the methods represented:

Qubit — a unit of information in quantum computation. In contrast to the classical bit, which at any time can be in one of two states - in the state 0 or 1 (or | 0> and |1> for the qubit), the qubit can be simultaneously in a superposition of these states.

We will introduce the main properties of cryptographic functions according to the book [1]:

1**) Irreversibility** For a given value of *v* of the hash function, it must be “computationally infeasible” to find some message *w*, for which *v = hash(w)*

2**) Effective computability** For a given value of *w*, it is easy to calculate *v=hash(w))*

These two properties together give ***the property of irreversibility***

3) Collision resistance For a given value of *w*, it must be “Computationally difficult” to find another value $w \neq w'$, for which *hash(w) = hash(w’)*.

The following algorithms have been singled out for the research in this paper:

1) Random search
2) Genetic algorithm
3) Algorithm of simulated annealing
4) Constructive algorithm based on linear codes

In those cases when we cannot be sure of the accuracy of the solution obtained or in the absence of a general solution, we turn to heuristic algorithms. They also cannot guarantee an exact solution, but they allow finding the best-enough solution in most variants.

We carried out an analysis of the object domain and derived the following function. Let's consider it:

$$\lceil < \psi_B(w) \lceil \psi_B(w') > | = \left| \frac{1}{|B|} * \sum_{j=1}^{|B|} (-1)^{(b_j; y)} \right|, \quad \text{где } y = w \oplus w'$$

Here B is the set of binary sets $B = \{b_1, b_2, \dots, b_j, \dots\}$

Let us write down:

$$\tau(B) = \max_{y \neq 0} \left| \frac{1}{|B|} * \sum_{j=1}^{|B|} (-1)^{(b_j; y)} \right|, \quad \text{где } y = w \oplus w' \qquad (2)$$

The problem consists in finding the set of binary sets B for which the function (2) comes up to a minimum.

The most appropriate solution is achieved when $B = Z_2^n$, whence it follows that $\delta = 0$

But with this approach, we will not be satisfied with the size of hash, since there will be no preimage resistance.

In this regard, it is required to carry out a comparative analysis of various algorithms to find the answer to the problem, as well as to develop new approaches to its solution.

3.Results.We will become familiar with the above algorithms in the research context in this work: Random search

In application to our problem, unlike a full enumeration, the search consists in generating a random set of parameters, that is determined depending on the input data. The algorithm generates a random set at each iteration, checks the fulfillment of condition $\tau < \delta$, and if the condition is met, we get the sought set B.

It is worth noting that the time span when using this algorithm is reduced at times.

Here we can already work with different sizes of sets and even with different sizes of input messages in each set.

1) Genetic algorithm

Heuristic algorithm, aimed at more effective search for solutions to ensure cryptographic strength and collision information to a minimum.

For more on the algorithm:

In our case, the genotype will be a randomly defined set. This is the set of the initial population.

The fitness function $\tau(B)$, is given, mutation is the changing the random bit in each set, crossing is the splitting the sets of parents into 2 parts and exchanging these parts.

At the start of the algorithm, the parameters of dimension n, d are also defined, after which a random set with such parameters is generated. After this, the mutation of one of the set elements is performed on the set. Next, the fitness function on the set is tested.

At the next stage, using the fitness function, the optimus half is determined. This half crosses with each other and "off-spring" is formed - new sets are determined from each pair of sets in this half (0-1, 2-3, etc.). There will be half of one and half of the other set. Then this "off-spring" is added to the original set, sorting according to the fitness function is performed and the worst sets are removed. As a result, the size of the set is sorted to its original size. We obtain the sought set. At the next stage, the algorithm starts with the new data.

2) Stipulated annealing algorithm

A stochastic heuristic algorithm, or an algorithm that can find by chance.

The physical process of crystallization rsed in metallurgy is under consideration. The metal has a certain crystal lattice in which the atoms are located. At the same time, it has energy, and the lower it is, the better the crystal lattice is. Atoms are aimed to go into the state with less energy. The metal is first heated, then gradually cooled.

In our problem, the energy is the maximum value of δ, which we fix as one of the parameters, neighboring sets are the sets that differ in value of one element. When atoms are in the crystal lattice, they can still move from one cell to another, if the energy decreases.

First, a random set is generated. This set is a crystal lattice.

At the next step, a transition to an adjacent variant of lattice takes place, with a certain probability (using the Gibbs distribution), which depends on the temperature (which in turn depends on the maximum δ on the set).

These iterations are repeated, and the temperature gradually decreases. The work of the algorithm is completed when the desired value of δ, specified as a parameter, is obtained.

The result - we get the desired set of parameters B at the output.

An example how the algorithm works:

annealing d = 8, n = 8

([74, 11, 67, 45, 151, 70, 183, 126], 0.75, 30.01941783550427)

annealing d = 16, n = 8

([130, 2, 49, 247, 204, 253, 119, 123, 20, 141, 91, 176, 9, 39, 152, 135], 0.5, 30.09218839747274)

3) Constructive algorithm based on linear codes

In this paper we will give consideration to the representation of classical information in the form of a quantum superposition of the following form:

$$\tau(\mathbf{B}) = \max_{y \neq 0} \left| \frac{1}{|B|} * \sum_{j=1}^{|B|} (-1)^{(b_j;y)} \right|, \qquad (3)$$

where the set $\boldsymbol{B \subseteq \{0,1\}^n, |B| = d}$ will be called a set with small deviation if for random set $y \subseteq \{0,1\}^n$, it is fulfilled:

$$\max_{y \neq 0} \left| \frac{1}{|B|} * \sum_{j=1}^{|B|} (-1)^{(b_j;y)} \right| \leq \varepsilon$$

where scalar product - $(b_j; y) - скалярное\ произведение, \boldsymbol{\varepsilon} -$ the parameter that enables the probability of the appearance of quantum collisions.

Consider a finite Galois Fq field of the order q, in which q is the number of elements of the field. The finite field is defined by the order $q = p^2$, $p = 2^l$, where p is a prime number, l is any integer.

An important component in this paper is the following property:

Suppose $\boldsymbol{irreducible \in Fq, (q = 2)}$ – prime polynomial of power 2^l, then the F_2^{2l} field will contain any root of an irreducible polynomial. That is, F_2^{2l} is the splitting field of the polynomial over the F2 field.

In the Fq field, we are interested in the operations of addition, multiplication, and the exponential operation to solve an equation, which we will consider below. For this we need the Boolean function XOR:

0 xor 0 = 1 xor 1= 0

0 xor 1 = 1 xor 0 =1

In this case, the linear code C acts from the finite Fq field (since we have operations in this field) to the set B, where d is the size of the set.

Present the following algorithm:

1. Input data: n, ε
2. Choose $p = 2^l$ – power of two so, that $\frac{1}{2}\left(\frac{n}{\varepsilon^2}\right)^{1/4} < p < \left(\frac{n}{\varepsilon^2}\right)^{1/4}$
3. Let $q = p^2, r = \varepsilon p^3$
4. Solve the equation $y^p + y = x^{p+1}; x, y \in GF(q)$. Let A— the set of all pairs of roots of the equation.

To realize it, we assume that $p \leq 2^6 = 64$ (otherwise we do not have enough memory for the set D of size p^5), so $q = p^2 \leq 2^{12} = 4096$

Since we have x, y – the roots of the equation, which can take q of different values, then we need to search q^2 of the pairs, which is not more than $2^{24} = 16777216$

Therefore, a complete search can be considered acceptable in this solution.

5. Put down the set

$$D = \left\{ \left\langle bin\left(a^i b^j\right), bin(c) \right\rangle_2, (a,b) \in A, c \in GF(q), i + j \leqslant \frac{r}{p+1} \right\},$$

where bin — the binary representation of the field element, the angle brackets — the scalar product modulo 2.

4.Discussion.

4.1. Methods of comparing algorithms . Each of the above algorithms has the same parameters to obtain the results, the following criteria were selected for comparing the algorithms:

n – the size of input message;
d – the size of the set;
δ – the parameter which the probability of collisions depends on;
t – the time of algorithm working.

maximum of search 2^{n-1}

Let's give a graphic interpretation to our algorithms. For illustrative comparison, we will consider several algorithms that can cope with different sizes of input data. Depending on the input parameters, the results may vary.

4.2. Approximation . Let us try to take a large dimension n for the linear algorithm n >> 8. Let us try to approximate the construction for large sizes n. We assume that log (D) increases in proportion to n.

$$D = O\left(n^{\frac{5}{4}}\right) - \text{полиномиальный рост размера множества}$$

Polynominal growth of the size of set

It is worth noting that for large dimensions n the objective function $\tau\,(B)$ is calculated exponentially long. With n=21, the calculation will be 32 times longer. Therefore, we will take the sets 2^l to construct a set of parameters.

Next, we will use the **randsearch, genetic, anneling** algorithms of exponential complexity considered earlier.

Recall that:

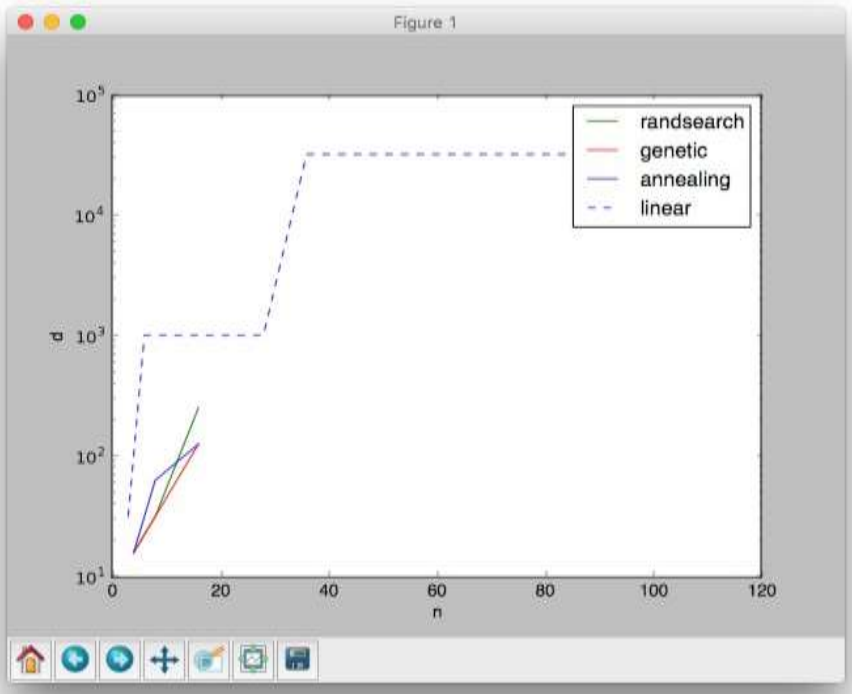
1) Random search **(randsearch)** is the search that consists in generating a random set of parameters, determined depending on the input data. The algorithm generates a random set at each iteration, checks the conditions $\tau < \varepsilon$, and when the condition is reached, we get the desired set.

2) The genetic algorithm **(genetic)** is a heuristic algorithm, a key idea: a mutation in the set of the initial population, the crossing and breeding of a new offspring that satisfies the fitness functions. The aim of the algorithm is in more effective search for a solution to ensure cryptographic strength and collision reduction to a minimum.

3) Simulated annealing algorithm **(annealing)** is a stochastic heuristic algorithm that considers the physical process of annealing metals. To obtain a set of parameters, the Gibbs distribution is used.

We will calculate the objective function to n = 16 for all algorithms, then for **linear** algorithm we start the approximation for the sizes n >> 16. Fix ε and show the size of the sets d related to the size of the input message n:

$$\varepsilon = 0.5, \qquad n = 4, \ldots, 120$$



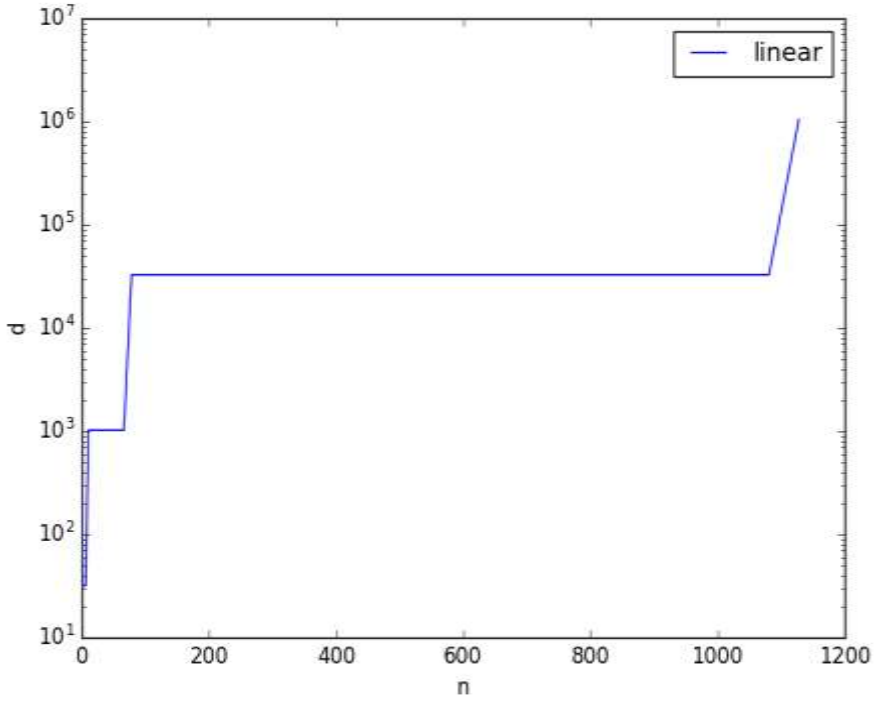
The size of sets d related to the size of the input n, for n = 4, ..., 120 in the algorithms of random search (randsearch), genetic algorithm, simulated annealing algorithm, linear algorithm in finite fields (linear) at $\varepsilon = 0.5$

We restricted the computation of the **randsearch, genetic, annealing** algorithms for n = 4,8,16, since calculations for larger sizes of n require a very long elapsed time.

linear algorithm is able to work on large dimensions n, but with small sizes n = 4,8 heuristic approaches and random search still justify themselves more. It is more efficient to use them for these dimensions.

In terms of speed, the **linear** algorithm works much faster even on large sizes n compared to the studied heuristic algorithms. It found a solution within a given interval ε by searching n = 120 d = 32768 in just 10 minutes, when the other algorithms considered would require polynomial time for this solution.

Let's give an assessment from above to the linear algorithm to the threshold $\varepsilon = 0.5$



The size of the sets d related to the size of the input, for n = 4, ..., 1128

in a constructive algorithm in finite fields (linear) with $\varepsilon = 0.3$

Thanks to the upper assessment, the algorithm is able to find a solution even on such large dimensions d, n.

The time for each iteration does not exceed 5 seconds. Here, we refer to the rationale (4), thanks to which it was possible to obtain such results.

Thus, the constructive algorithm is capable of working at large values of n, and the work of the algorithm ends in an excellent time.

5. Conclusions. As a result of this work, the methods for ensuring the strength of a binary quantum hash function against collisions have been developed and analyzed. Using the proposed algorithms, it is possible to obtain quantum hashing parameters that provide a given level of collision resistance.

An experimental analysis of the operation of various algorithms has been carried out. Based on the results of experiments in most cases, the genetic algorithm and random search obtain a better solution, while the genetic algorithm based on deducing the "new generation" exceeds random search on the average. The simulated annealing algorithm is inferior to randsearch. It is also necessary to take into account the time spent searching for solutions for large sizes of input messages n: the larger the size of the input is, the more sets will need to be sorted out to find the desired solution.

It has been experimentally established, in which cases it makes sense to apply heuristic algorithms, and in which a constructive algorithm which is capable of working with large dimensions n in an acceptable time

6. Acknoledgements. The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

References

1. Aida Gainutdinova. The fundamentals of quantum computing, 2010
2. F.M. Ablayev and A.V. Vasiliev. Cryptographic quantum hashing. Laser Physics Letters, 11(2):025202, 2014.
3. Farid Ablayev and Marat Ablayev. Laser Physics Letters, Volume 12, Number 12 // Laser Physics Letters, Volume 12, Number 12, 2015 September 2015
4. F. Ablayev, A. Vasiliev : Algorithms for quantum branching programs based on fingerprinting, Electronic Proceedings in Theoretical Computer Science, 9:1-11, 2009
5. Harry Buhrman, Richard Cleve, John Watrous, and Ronald de Wolf. Quantum ngerprinting. Phys. Rev. Lett., 87(16):167902, Sep 2001. Michael A. Nielsen and Isaac L. Chuang. Quantum Computation and
6. Michael A. Nielsen and Isaac L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, 1 edition, October 2000.
7. Stinson, D.R. On the connections between universal hashing, combinatorial designs and error-correcting codes
8. In: Proc. Congressus Numerantium 114, 7–27 (1996).
9. Kholevo A.S. Some Assessments for Information Quantity Transmitted by Quantum Communication Channel,
10. Transferring Information Problem. 9 (3), 3–11 (1973).
11. Ablayev F. On the concept of cryptographic quantum hashing, Laser Physics Letters 12 (12), 125204 (2015),
12. http://stacks.iop.org/1612-202X/12/i=12/a=125204.
13. Naor J. Small-bias probability spaces: Effcient constructions and applications, Proceedings of the twenty
14. Second annual ACM Symposium on Theory of Computing STOC'90 (New York, NY, USA, ACM, 1990),
15. http://doi.acm.org/10.1145/100216.100244.
16. Alon N., Goldreich O., Hastad J., Peralta R. Simple constructions of almost k-wise
17. independent random variables, Random Structures & Algorithms 3 (3), 289–304 (1992),
18. Kurbanov, R. A., oglu Gurbanov, R. A., Belyalova, A. M., Maksimova, E. V., Leonteva, I. A., & Sharonov, I. A. (2016). Practical Advice for Teaching of University Students the Mechanisms of Self-Government of Safe Behavior. International Electronic Journal of Mathematics Education, 12(1), 35-42.
19. de Paulo Lobato, Christiane Batista, Samanta Borges Pereira, and Flávia Luciana Naves Mafra. "Resistências àsuperexploração das águas minerais em São Lourenço (MG)." Opción 34.86 (2018): 1043-1076.
20. Foundations of Computer Science FOCS'09 (Oct. 2009), 191–197.
21. Ablayev F. On computational power of quantum branching programs, Fundamentals of computation theory
22. Riga, 2001, 59–70, Lecture Notes in Comput. Sci. (Springer, Berlin, 2001), Vol. 2138.