2-2020

How Quantum Cryptography and Quantum Computing Can Make Cyber-Physical Systems More Secure

Deepak Tosh
The University of Texas at El Paso, dktosh@utep.edu

Oscar Galindo
The University of Texas at El Paso, ogalindomo@miners.utep.edu

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Olga Kosheleva
The University of Texas at El Paso, olgak@utep.edu

Comments:

Technical Report: UTEP-CS-20-13

How Quantum Cryptography and Quantum Computing Can Make Cyber-Physical Systems More Secure

1st Deepak Tosh
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
dktosh@utep.edu

2nd Oscar Galindo
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
ogalindomo@miners.utep.edu

3rd Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
vladik@utep.edu

4th Olga Kosheleva
Department of Teacher Education
University of Texas at El Paso
El Paso, TX 79968, USA
olgak@utep.edu

***Abstract*—For cyber-physical systems, cyber-security is vitally important. There are many cyber-security tools that make communications secure – e.g., communications between sensors and the computers processing the sensor's data. Most of these tools, however, are based on RSA encryption, and it is known that with quantum computing, this encryption can be broken. It is therefore desirable to use an unbreakable alternative – quantum cryptography – for such communications. In this paper, we discuss possible consequences of this option. We also explain how quantum computers can help even more: namely, they can be used to optimize the system's design – in particular, to maximize its security, and to make sure that we do not waste time on communicating and processing irrelevant information.**

***Index Terms*—cyber-physical systems, security, quantum cryptography, quantum computations**

I. Security of Cyber-Physical Systems: A Quantum Challenge

What are cyber-physical systems: a brief reminder. Many modern complex systems include both computational parts and physical parts. For example:

- A power station includes actual electricity generators and transformers as well as computational devices that control the generators, transformers, and communications.
- A city-wide system includes computers on all levels, from microprocessors controlling individual devices to computers providing, e.g., city-wide optimization of transportation flows.

Such systems are known as *cyber-physical* systems.

For cyber-physical systems, cyber-security is vital. It is known that many computing system have been successfully attacked, with information stolen or corrupted. In general, cyber-security is an important problem.

This problem is especially vital for cyber-physical systems, since by hacking into these systems, an adversary can cause catastrophic damage: e.g., blow up a nuclear power station.

How cyber-security is provided now. In general, there are two main directions in providing cyber-security of the current cyber-physical systems. On the one hand, there are consistent efforts to educate users, so that adversaries will not use social engineering (as they do now) to penetrate systems. For this purpose, users should create strong passwords, avoid disclosing them, never send them by email, etc.

On the technical side, cyber-security is (or at least should be) provided by making sure that all communications between sensors and computers (and between computers themselves) are encrypted.

This encryption is usually based on the RSA algorithm; see, e.g., [5]. In this algorithm:

- An agent interested in receiving messages selects two very large (up to 100 decimal digits long) prime numbers p and q, and sends their product $n = p \cdot q$ openly to everyone interested.
- Once a recipient knows the value n, he/she can encrypt any message.
- Any agent who knows the values p and q can decrypt this message.
- However, without knowing p and q, decryption does not seem possible.

The security of this algorithm is based on the fact that no efficient algorithm is known for factoring large integers – other than trying all possible prime factors from 1 to $\sqrt{n}$, which would require about 10^{50} computational steps – more than

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science) and HRD-1242122 (Cyber-ShARE Center of Excellence).

the number of moments of time in the Universe; see, e.g., [7], [30].

Quantum challenge to cyber-security. The main problem with existing cyber-security is that a quantum algorithm designed by Peter Shor enables us to factor large integers in feasible time – and thus, to break the RSA encryption; see, e.g., [26]–[28], [31]. Similar algorithms can break all similar encryptions algorithms; see, e.g., [26].

This result practically guaranteed that this challenge has to be taken seriously.

- Before this result, quantum computing was mostly an academic research topic close to science fiction.
- However, once it turned out that a quantum computer will enable to us to read all the messages sent so far, all the governments and all big companies have invested billions of dollars into development of such computers.

Whoever get there first will be the first to read all the information – and thus, to gain a tremendous advantage over others. As a result, thousands of researchers and practitioners all over the world are working on designing a quantum computer – which practically guarantees that it will be eventually built.

It may take 5 years, it may take 20 years, but it will be built. And so, we must be ready for this challenge.

Quantum cryptography: a secure alternative to RSA encoding. The situation with cyber-security is not as gloomy as it may seem after reading the previous subsection. Yes, quantum algorithms make RSA vulnerable, but quantum algorithms also provide an unbreakable (so far) alternative to RSA, called *quantum cryptography*; see, e.g., [26], [31].

Another good news is that, in contract to general quantum computing algorithms – most of which cannot yet be practically implemented – quantum cryptography is perfectly practical, and it *has* been implemented. For example:

- for many years already, there is a quantum computing-protected communication line between the White House and the Pentagon, and
- a reasonable recent Chinese experiment successfully implements quantum cryptography when communicating with a satellite several hundred kilometers above the Earth.

Yet another good news is that not only the current quantum cryptography algorithm unbreakable, but this algorithm is also, in some reasonable sense, the best possible (see, e.g., [8]). Not only it is the best possible for two-agent communication, it is also clear how to use it in the most efficient way for multi-agent communications; see, e.g., [22].

What we do in this paper. First, we provide a brief description of quantum cryptography. Our main objective is to analyze how quantum cryptography can be implemented to make cyber-physical systems more secure. We will also analyze how, more generally, quantum computing can help in the design of cyber-physical systems – in particular, in providing their security.

II. Quantum Cryptography: A Brief Reminder

Basic facts from quantum mechanics: a brief reminder. In quantum mechanics (see, e.g., [7], [30]), in addition to the usual classical states $s_1, \ldots, s_n$, we also have *superpositions*, i.e., states of the type

$$s = c_1 \cdot |s_1\rangle + \ldots + c_n \cdot |s_n\rangle,$$

where $c_1, \ldots, c_n$ are complex numbers for which

$$|c_1|^2 + \ldots + |c_n|^2 = 1.$$

These states can be viewed as vectors $(c_1, \ldots, c_n)$ in the n-dimensional complex-valued vector space $\mathbb{C}^n$. In particular, each of the original states s_i corresponds to a vector $(0, \ldots, 0, 1, 0, \ldots, 0)$ with 1 in the i-th place.

If, for a system in this state, we perform a measurement to determine in which of the states $s_1, \ldots, s_n$ is this system, then we will get:

- the state s_1 with probability $|c_1|^2$,
- $\ldots$, and
- the state s_n with probability $|c_n|^2$,

where $|c|$ is an absolute value of a complex number: for a complex number $c = a + b \cdot \mathrm{i}$ (where i denotes $\sqrt{-1}$), the absolute value is defined as $|c| = \sqrt{a^2 + b^2}$.

Each probability can be alternatively described as $|\langle s, s_i\rangle|^2$. Here, the scalar (= dot) product $\langle a, b\rangle$ of two complex-valued vectors $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ is defined in the usual way, as

$$\langle a, b\rangle = a_1 \cdot b_1^* + \ldots + a_n \cdot b_n^*,$$

where a^* means complex conjugate, i.e., an operation that transforms each complex number $z = a + b \cdot \mathrm{i}$ into $z^* = a - b \cdot \mathrm{i}$.

The probabilities of getting n possible outcomes should add up to 1, which explains the above constraint

$$|c_1|^2 + \ldots + |c_n|^2 = 1.$$

After the measurement, if we get the result s_i, then the original state s transforms into the state s_i.

Instead of an instrument for measuring one of the states s_i, we can have a different instrument that measures the original state against a different set of mutually orthogonal vectors $s'_1, \ldots, s'_n$; in this case, the probability to get the i-th result when in state s is equal to $|\langle s, s'_i\rangle|^2$.

Bits and qubits. The main part of a usual computer is a *bit* (which is short of *binary digit*). A bit can be in two possible states: 0 and 1. A natural quantum analog of a bit – known as a *quantum bit* (*qubit*, for short) can be in one of the states $c_0 \cdot |0\rangle + c_1 \cdot |1\rangle$, with $|c_0|^2 + |c_1|^2 = 1$. Quantum cryptography uses only four of these states: the two original states $|0\rangle$ and $|1\rangle$, and two new states:

$$|0'\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}} \cdot |0\rangle + \frac{1}{\sqrt{2}} \cdot |1\rangle \text{ and } |1'\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}} \cdot |0\rangle - \frac{1}{\sqrt{2}} \cdot |1\rangle.$$

One can easily check that the two new vectors are orthogonal, so we can use them for measurement. Let us denote:

- the original basis, consisting of the states $|0\rangle$ and $|1\rangle$, by $+$, and
- the new basis, consisting of the states $|0'\rangle$ and $|1'\rangle$, by $\times$.

One can easily check that:

- If we prepare a state in the original $+$ basis, i.e., prepare a state $|0\rangle$ or $|1\rangle$, and measure this state with respect to the same basis, we get exactly the prepared state: 0 or 1.
- Similarly, if we prepare a state in the $\times$ basis, i.e., prepare a state $|0'\rangle$ or $|1'\rangle$, and we measure this state with respect to the same basic, we also get back the prepared state.

On the other hand:

- If we prepare a state in the $+$ basis and measure it in the $\times$ basis, then we get either 0 or 1 with probability 1/2.
- Similarly, we prepare a state in the $\times$ basis and measure it in the $+$ basis, then we get either 0 or 1 with probability 1/2.

Now, we are ready to describe the quantum cryptography algorithm.

Quantum physics naturally leads a random number generator. The quantum cryptography algorithm uses a random number generator that produces either 0 or 1 with probability 1/2.

With quantum physics, there is no need – as many computers do now – to use *pseudo-random* numbers, i.e., numbers that are generate by a complex algorithm. Indeed, in quantum physics – as we have just saw – there are plenty of processes that produce actually random results.

Quantum cryptography algorithm: first step. Let us show how the random number generator can be used if an agent A wants to send a message x consisting of m bits $x_1, \ldots, x_m$ to another agent B.

First, for some integer n (to be described later), A runs a random generator $2n$ times, and generates $2n$ random numbers $a_1, \ldots, a_n, a_{n+1}, \ldots, a_{2n}$. Here:

- the values $a_1, \ldots, a_n$ will be used as bits to be sent, and
- the values $a_{n+1}, \ldots, a_{2n}$ will be used to decide which basis we use.

Specifically:

- if $a_{n+i} = 0$, A will use the $+$ basis to send the i-th bit, and
- if $a_{n+i} = 1$, A will use the $\times$ basis to send the i-th bit.

Then, for each i from 1 to n, A sends to B the bit a_i encoded in the basis a_{n+i}, i.e.:

- if $a_i = 0$ and $a_{n+i} = 0$, A sends the state $|0\rangle$;
- if $a_i = 0$ and $a_{n+i} = 1$, A sends the state $|0'\rangle$;
- if $a_i = 1$ and $a_{n+i} = 0$, A sends the state $|1\rangle$; and
- if $a_i = 1$ and $a_{n+i} = 1$, A sends the state $|1'\rangle$.

The agent B also runs a random number generator, but only n times and gets the values $b_1, \ldots, b_n$. For each bit i, B uses the measurement corresponding to the value b_i, i.e.:

- if $b_i = 0$, B measures the i-th signal in the $+$ basis, and
- if $b_i = 1$, B measures the i-th signal in the $\times$ basis.

B then records the measurement results $m_1, \ldots, m_n$.

Second step. After B finishes the measurement process, A openly sends, to B, all the values $a_{n+1}, \ldots, a_{2n}$ that describe the basis of the sent signal. For some number c of these indices, A also sends the original values a_i.

In half of the cases, the sending and measuring basis coincide, i.e., $a_{n+i} = b_i$. So, as we have mentioned earlier, for these values i, the measurement result should reconstruct the original signal, i.e., we will have $m_i = a_i$. In particular, this should happen for approximately $c/2$ of the indices for which A sent the values a_i.

If for some of these i, we have $m_i \neq a_i$, this means that something interfered with the communication process, i.e., that we have an eavesdropper. Vice versa, suppose that there is an eavesdropper who listens to the conversation – i.e., who measures the signals while they go from A to B. Then, since the eavesdropper does not know the orientation a_{n+i}, in half of the cases, its measurement basis will be different from the one used for sending. For such i, the transmitted signal will be changed – so after B's measurement, instead of the original signal a_i, we will have 0 or 1 with equal probability.

So, if there is an eavesdropper, then, out of c bits:

- for half of them, i.e., for $c/2$ bits, the signal will be changed;
- thus, for a half of this half – i.e., for $c/4$ bits – we will get $a_i \neq m_i$.

For large c, there is a high probability that at least in one of these cases, we will have $a_i \neq m_i$. Thus, with high probability, the eavesdropper will be detected.

If there is an eavesdropper, then we need to physically inspect the communication path.

Comment. Remember that in our case, we do not talking about sending a signal several hundred kilometers into space. We are talking about *short-distance* communications: from the reactor to the control room, from the in-city weather sensor to the in-city computer, etc.

In such cases, the path *can* be physically inspected.

Third step. If no eavesdropper was detected, then the agent B sends, to A, the list of al the values $i_1, \ldots, i_m$ for which $a_{n+i} = b_i$ (with the exception of those indices for which a_i was sent by A via an open channel). For all these indices, we have $a_i = m_i$.

There are approximately $m \approx n/2 - c/2$ such indices. Now, both A and B know $m \approx n/2 - c/2$ values $a_{i_k} = m_{i_k}$, $k = 1, \ldots, m$ that no one else knows. These values can be used for the final step.

Final step. The agent A send m bits $y_k = x_k \oplus a_{i_k}$, where $a \oplus b$ is exclusive "or", or, what is the same, addition modulo 2:

$$0 \oplus 0 = 0, \quad 0 \oplus 1 = 1 \oplus 0 = 1, \text{ and } 1 \oplus 1 = 0.$$

This operation is associative and has the property that $b \oplus b = 0$ for all b. Thus, we always have

$$(a \oplus b) \oplus b = a \oplus (b \oplus b) = a \oplus 0 = a.$$

Since $a_{i_k} = m_{i_k}$ for all k, this means that upon receiving these encrypted bits, B can easily decrypt them as

$$x_k = y_k \oplus m_{i_k}.$$

The secure communication is completed.

So how do we select n? The only thing about the algorithm that we did not describe yet is how to select n. The above description leads to the following procedure for selecting n:

- First, we select c based on the degree of confidence that we want to have that there is no eavesdropper.
- Then, we select n for which $m = n/2 - c/2$, i.e., we select $n = 2m + c$.

III. How Quantum Cryptography Can Help Cyber-Security of Cyber-Physical Systems: Analysis

Main idea. The main idea of using quantum cryptography in cyber-physical systems is straightforward: all the communications between sensors and computers (and between computers themselves) must be encrypted by using quantum cryptography.

Important issue. As we can see from the above description of quantum cryptography, there is an important issue with its practical implementation. Indeed:

- Traditional communication means sending bits. A simple cable can easily send hundreds of millions of bits per second.
- In contrast, quantum cryptography means sensing qubits, i.e., quantum states. This is not so easy, and the current speed with which we can send qubits is many orders of magnitude smaller. As a result, we cannot send as much information from the sensors as we send now.

How to deal with this issue. At present, since communications are fast, we usually send raw data from the sensors to the processors. If we switch to quantum cryptography, we will not be able to send as much data as before. Thus, if we want to still send all the information, we need to first compress the raw data, so that sending this information would require fewer bits.

Compression requires a significant amount of computational power. For example, the best known image compression algorithms – as implemented in the JPEG'2000 standard [29] and its modifications (see, e.g., [16]) – are based on using *wavelets* (see, e.g., [2], [3] and references therein). There are many algorithms that provide fast computations with wavelets, such as Fast Wavelet Transform, but still, these algorithms are beyond the ability of simple processors usually embedded in sensors. Even more sophisticated algorithms are needed if we want to implement 3-D generalizations of wavelet compression algorithms; see, e.g., [4], [12]–[15], [17]–[20], [23], [25].

So, to make sure that the quantum-related cyber-security enhancement works for cyber-physical systems, we must add, to each sensor, computational power – with an embedded efficient compression algorithm.

IV. How Quantum Algorithms in General Can Help in Designing Cyber-Physical Systems

Do we need all the sensor data? At present, sensors are cheap, communication is cheap. As a result, when designing a system, we add as many sensors as possible, even though some of the information may be duplicate – or even irrelevant.

For example, when we predict weather, we try our best to use as much information about the current weather as possible. In practice, data from reasonably faraway regions is rarely useful for predicting next day's weather, since weather changes rarely travel fast. However, it is easier to just add a few extra sensors than to perform a detailed and time-consuming analysis of which locations are relevant and which are not.

This issue becomes important if we use quantum communications. When we switch to quantum communications, communication becomes slower and more expensive. It is therefore desirable to detect which data points are relevant and which are not.

In this analysis, quantum computing can help. Interestingly, quantum computing can help in this analysis. Namely, there are quantum algorithms – such as the Deutsch-Jozsa algorithm – that help us decide where certain bits are relevant; see, e.g., [6], [22].

The most impressive example is an algorithm for the case when the input has only 1 bit, i.e., when the data processing algorithm computes the function $f(x)$ of an 1-bit data x. In this case, the question is whether this bit is relevant at all:

- if it is not relevant, this would mean that the result $f(x)$ of the computation does not depend on x at all, i.e., that

$$f(0) = f(1);$$

- if the input bit is relevant, then we will have $f(0) \neq f(1)$.

In non-quantum computing, the only way to check whether $f(0) = f(1)$ is:

- to apply the algorithm f to 0 and to 1 and
- to compare the results of these two applications.

This 2-calls-to-f idea sounds simple until we realize that the algorithm f may be very complicated: e.g., algorithms for weather prediction usually take hours on a high performance computer.

With this in mind, quantum computing indeed helps: namely, by using quantum computing, we can check whether $f(0) = f(1)$ in only *one* call to f. In this call, the input will be neither 0 not 1 but rather a superposition of these two states.

Comment. It is worth mentioning that, as shown in [24], the current quantum scheme for checking whether $f(0) = f(1)$ is, in effect, the only possible one.

Other possible applications of quantum computing to cyber-physical systems and their security. In designing a cyber-physical system – and, in particular, in designing cyber-security part of the system – we try to find a design d that satisfies certain specifications. In some cases, there are efficient algorithms for finding such a design, However, in many other

cases, we have to use methods similar to exhaustive search: let the computer try all possible options until we find one that satisfies the desired specifications.

In this search, quantum computing can help. Indeed:

- If we need to look through N possible options, then in non-quantum computing, we need to perform, in the worse case, N computational steps – by looking at all these options (and, on average, we need $N/2$ steps).
- Interestingly, a quantum algorithm proposed by Grover [10], [11], [26] enables us to find the desired alternative in $\sqrt{N}$ steps.

For large N, this is much much faster: e.g., when $N \approx 10^6$, the quantum search is three orders of magnitude faster.

Comment about parallelization. An additional speed-up can be obtained if we parallelize the algorithm, i.e., if we have several computers working in parallel. Parallelization necessitates sending preliminary results from one computer to another. As we already know, for quantum computing, communication is not as easy as in the non-quantum case. Good news is that there is an efficient quantum method of sending signals without a need for quantum channels. This method is known by a somewhat misleading science-fiction name of *teleportation*; see, e.g., [26].

It is worth mentioning that, similar to the uniqueness of the Deustch-Jozsa algorithm, it is possible to show that the usual teleportation algorithm is, in some reasonable sense, unique – and thus, cannot be further improved.

What about optimization. Usually, there are several different designs that satisfy all the given constraints. In such situations, it is desirable to select the best of these designs. In precise terms, this means that:

- the user has to provide us with an objective function F that described the quality of each design d, and
- we should select the design with the largest possible value of $F(d)$.

It should be mentioned that for complex systems, we rarely know the exact consequences of selecting each alternative. At best, we know these consequences with some accuracy ε. Thus:

- we are not looking for the exact maximum of the objective function $F(d)$,
- it is sufficient to look for a design which is ε-close to this maximum $m \stackrel{\text{def}}{=} \max\limits_{d} F(d)$.

In finding such an optimal design, quantum computing can also help; see, e.g., [1]. Indeed, usually, we know the range $[\underline{F}, \overline{F}]$ of possible values of the objective function. For each value F from this range, we can use the Grover's algorithm, and in time $\sqrt{N}$, either find a design for which $F(d) \geq F$ or conclude that there is no such design.

This possibility leads to the following bisection algorithm for finding a narrow interval $[\underline{M}, \overline{M}]$ that contains m:

- We start with the interval $[\underline{M}, \overline{M}] = [\underline{F}, \overline{F}]$.
- On each step, we compute the midpoint $M = \dfrac{\underline{M} + \overline{M}}{2}$, and use Grover's algorithm to check whether there exists a design d for which $F(d) \geq M$.
- If such a design exists, this means that $m \geq M$, so we can conclude that $m \in [M, \overline{M}]$, and we can take $[M, \overline{M}]$ as the new value of the interval containing the actual maximum m.
- If such a design does not exist, we conclude that $m \in [\underline{M}, M]$, and we can take $[\underline{M}, M]$ as the new value of the interval containing the actual maximum m.
- In both cases, we decrease the width of the interval $[\underline{M}, \overline{M}]$ by half.
- We stop this procedure when the width of the interval $[\underline{M}, \overline{M}]$ becomes smaller than or equal to ε. In this case, since this interval contains the actual (unknown) maximum m, we can conclude that all the values M from this interval are ε-close to this maximum m.
- We know that there is a design d for which $F(d)$ is in the final interval $[\underline{M}, \overline{M}]$, so we can use Grover's algorithm to find one of such designs. The value $F(d)$ corresponding to this design will indeed be ε-close to the actual (unknown) maximum m.

How many steps do we need?

- We start with an interval $[\underline{F}, \overline{F}]$ of width $\overline{F} - \underline{F}$.
- On each step, we divide the width by half.
- So, in k steps, we get the width $2^{-k} \cdot (\overline{F} - \underline{F})$.
- To reach width $\leq \varepsilon$, we need

$$k = \left\lceil \log_2 \left(\frac{\overline{F} - \underline{F}}{\varepsilon} \right) \right\rceil,$$

where $\lceil x \rceil$ denotes the smallest integer which is greater than or equal to x.

Each iteration involves using Grover's algorithm and thus, requires $\sqrt{N}$ steps. So overall, we need $k \cdot \sqrt{N}$ steps.

As we have mentioned earlier, usually, the accuracy with which we know the consequences of each selection is not so good. So, the value ε is not very small and thus, the number k of iterations is small. Thus, by using this algorithm, we get almost the same speed-up in comparison with the N-step exhaustive search as for Grover's algorithm itself.

References

[1] C. Ayub, M. Ceberio, and V. Kreinovich, "How quantum computing can help with (continuous) optimization", In: M. Ceberio and V. Kreinovich (eds.), Decision Making under Constraints, Cham, Switzerland: Springer Verlag, to appear.

[2] A. E. Brito and O. Kosheleva, "Interval + image = wavelet: for image processing under interval uncertainty, wavelets are optimal", Reliable Computing, Vol. 4, No. 3, pp. 291–301, 1998.

[3] A. E. Brito, O. M. Kosheleva, and S. D. Cabrera, "Multi-resolution data processing is optimal: case study of detecting Surface Mounted Devices", Proceedings of the International Conference on Intelligent Systems and Semiotics (ISAS'97), National Institute of Standards and Technology Publ., Gaithersburg, MD, 1997, pp. 157–161.

[4] S. D. Cabrera, "Three-dimensional compression of mesoscale meteorological data based on JPEG2000", In: Battlespace Digitization and Network-Centric Warfare II, Proceedings of SPIE, Vol. 4741, pp. 239–250, 2002.

[5] Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, MIT Press, Cambridge, Massachusetts, 2009.

[6] D. Deutsch and R. Jozsa, "Rapid solutions of problems by quantum computation", Proceedings of the Royal Society of London, Ser. A, Vol. 439, pp. 553–558, 1992.

[7] R. Feynman, R. Leighton, and M. Sands, The Feynman Lectures on Physics, Boston, Massachusetts: Addison Wesley, 2005.

[8] O. Galindo, V. Kreinovich, and O. Kosheleva, "Current quantum cryptography algorithm is optimal: a proof", Proceedings of the IEEE Symposium on Computational Intelligence for Engineering Solutions CIES'2018, Bengaluru, India, November 18–21, 2018.

[9] O. Galindo, O. Kosheleva, and V. Kreinovich, "Towards parallel quantum computing: standard quantum teleportation algorithm is, in some reasonable sense, unique", In: H. Seki, C. H. Nguyen, V.-N. Huynh, and M. Inuiguchi (eds.), USB Proceedings of the 7th International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making IUKM'2019, Nara, Japan, March 27–29, 2019, pp. 23–34.

[10] L. K. Grover, "A fast quantum mechanical algorithm for database search", Proceedings of the 28th ACM Symposium on Theory of Computing, 1996, pp. 212–219.

[11] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack", Physical Reviews Letters, Vol. 79, No. 2, pp. 325–328, 1997.

[12] V. Jayaram, B. Usevitch, and O. Kosheleva, "Detection from hyperspectral images compressed using rate distortion and optimization techniques under JPEG2000 Part 2", Proceedings of the 11th IEEE Digital Signal Processing Workshop DSP'04, Taos Ski Valley, New Mexico, August 1–4, 2004, pp. 111–114.

[13] Olga Kosheleva, Task-Specific Metrics and Optimized Rate Allocation Applied to Part 2 of JPEG2000 and 3-D Meteorological Data, PhD Dissertation, Department of Electrical and Computer Engineering, University of Texas at El Paso, 2003.

[14] O. Kosheleva, "Towards optimal compression of meteorological data: a case study of using interval-motivated overestimators in global optimization", In: A. Torn and J. Zilinskas (eds.), Models and Algorithms for Global Optimization, Springer, New York, 2007, pp. 59–71.

[15] O. Kosheleva, A. Aguirre, S. D. Cabrera, and E. Vidal, Jr., "Assessment of KLT and bit-allocation strategies in the application of JPEG2000 to the battlescale forecast meteorological data", Proceedings of the IEEE International Geoscience and Remote Sensing Symposium IGARSS'03, Toulouse, France, July 21–25, 2003, Vol. 6, pp. 3589–3591.

[16] O. Kosheleva and S. D. Cabrera, "Application of task-specific metrics in JPEG2000 ROI compression", Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation, Santa Fe, New Mexico, USA, April 7–9, 2002, pp. 163–167.

[17] O. M. Kosheleva, S. D. Cabrera, B. E. Usevitch, A. Aguirre, and E. Vidal, Jr., "MSE optimal bit-rate allocation in JPEG2000 Part 2 compression applied to a 3-D data set", In: M. S. Schmalz (ed.), Mathematics of Data/Image Coding, Compression, and Encryption VII, with Applications, Proceedings of the SPIE/International Society for Optical Engineering, Vol. 5561, Denver, Colorado, August 2–6, 2004, pp. 51–61.

[18] O. Kosheleva, S. Cabrera, B. Usevitch, and E. Vidal, Jr., "How to best compress 3-D measurement data under given guaranteed accuracy", Proceedings of the 10th IMEKO TC7 International Symposium, Saint-Petersburg, Russia, June 30–July 2, 2004, Vol. 1, pp. 217–222.

[19] O. Kosheleva, S. Cabrera, B. Usevitch, and E. Vidal, Jr., "Compressing 3D measurement data under interval uncertainty", In: J. Dongarra, K. Madsen, and J. Wasniewski (eds.), Proceedings of the PARA'04 Workshop on State-of-the-Art in Scientific Computing, Springer Lecture Notes in Computer Science, 2006, Vol. 3732, pp. 142–150.

[20] O. Kosheleva, S. Cabrera, and E. Vidal Jr., "Optimal bit allocation for maximum absolute error distortion in the application of JPEG2000 Part 2", Proceedings of the 11th IEEE Digital Signal Processing Workshop DSP'04, Taos Ski Valley, New Mexico, August 1–4, 2004, pp. 134–138.

[21] O. Kosheleva and V. Kreinovich, "How to introduce technical details of quantum computing in a theory of computation class: using the basic case of the deutsch-jozsa algorithm", International Journal of Computing and Optimization, Vol. 3, No. 1, pp. 83–91, 2016.

[22] O. Kosheleva and V. Kreinovich, "Secure multi-agent quantum communication: towards the most efficient scheme (a pedagogical remark)", Mathematical Structures and Modeling, Vol. 49, pp. 119–125, 2019.

[23] O. M. Kosheleva, B. E. Usevitch, S. D. Cabrera, and E. Vidal, Jr., "Distortion optimal bit allocation methods for volumetric data using JPEG 2000", IEEE Transactions On Image Processing, Vol. 15, No. 8, pp. 2106–2112, 2006.

[24] V. Kreinovich, M. Ceberio, and R. Alvarez, "How to use quantum computing to check which inputs are relevant: a proof that Deutsch-Jozsa algorithm is, in effect, the only possibility", Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence SSCI'2019, Xiamen, China, December 6–9, 2019, pp. 828–832.

[25] J. L. Melchor, Jr., S. D. Cabrera, A. Aguirre, O. M. Kosheleva, and E. Vidal, Jr., "JAVA implemented MSE optimal bit-rate allocation applied to 3-D hyperspectral imagery using JPEG2000 compression", In: B. Huang, R. W. Heymann, C. C. Wang (eds.), Satellite Data Compression, Communications, and Archiving, Proceedings of the SPIE/International Society for Optical Engineering, Vol. 5889, San Diego, California, August 2005, pp. 24–34.

[26] M. Nielsen and I. Chuang, Quantum Computation and Quantum Information, Cambridge: Cambridge University Press, 2000.

[27] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, November 20–22, 1994.

[28] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", SIAM J. Sci. Statist. Comput., Vol. 26, pp. 1484–1509, 1997.

[29] D. S. Taubman and M. W. Marcellin, JPEG2000 Image Compression Fundamentals, Standards and Practice, Boston, Dordrecht, London: Kluwer, 2002.

[30] K. S. Thorne and R. D. Blandford, Modern Classical Physics: Optics, Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics, Princeton, New Jersey: Princeton University Press, 2017.

[31] C. P. Williams and S. H. Clearwater, Ultimate Zero and One, New York: Copernicus, 2000.