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

Deepak Tosh, Oscar Galindo,
Vladik Kreinovich, and Olga Kosheleva

University of Texas at El Paso
El Paso, Texas 79968, USA
dktosh@utep.edu, ogalindomo@utep.edu,
vladik@utep.edu, olgak@utep.edu

1. What Are Cyber-Physical Systems: A Brief Reminder

- Many modern complex systems include both computational parts and physical parts.
- E.g., 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.

2. For Cyber-Physical Systems, Cyber-Security Is Vital

- 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.

3. 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.

4. Cyber-Security Now (cont-d)

- An agent selects two very large (up to 100 decimal digits long) prime numbers p and q.
- He 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.

5. Cyber-Security Now (cont-d)

- This algorithm is secure since no efficient algorithm is known for factoring large integers:
  - other than trying all possible prime factors from 1 to $\sqrt{n}$,
  - but this would require about 10^{50} computational steps,
  - this is more than the number of moments of time in the Universe.

6. Quantum Challenge to Cyber-Security

- A quantum algorithm designed by Peter Shor enables us to factor large integers in feasible time.
- Thus, it can break the RSA encryption.
- Similar algorithms can break all similar encryptions algorithms.
- This result practically guaranteed that this challenge has to be taken seriously.
- Before this result, quantum computing was mostly an academic topic close to science fiction; but:
  - 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 quantum computing.

7. Quantum Challenge (cont-d)

- Whoever gets there first will be the first to read all the information.
- Thus, this person will gain a tremendous advantage over others.
- Thousands of researchers and practitioners all over the world are working on designing a quantum computer.
- This 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.

8. Quantum Cryptography: A Secure Alternative to RSA Encoding

- The situation with cyber-security is not as gloomy as it may seem at first glance.
- Yes, quantum algorithms make RSA vulnerable.
- However, quantum algorithms also provide an unbreakable (so far) alternative to RSA.
- It is called *quantum cryptography*.
- Another good news is that:
  - while general quantum computing algorithms cannot yet be practically implemented
  - quantum cryptography is perfectly practical, and it *has* been implemented.
- There is a quantum computing-protected communication line between the White House and the Pentagon.

9. Quantum Cryptography (cont-d)

- China used quantum cryptography it to communicating with a satellite.
- Yet another good news is that:
  - not only the current quantum cryptography algorithm unbreakable;
  - this algorithm is also, in some reasonable sense, the best possible.
- 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.

10. What We Do in This Talk

- First, we provide a brief description of quantum cryptography.
- Our main objective is to use quantum cryptography for making cyber-physical systems more secure.
- We will also analyze how quantum computing can help in the design of cyber-physical systems.

11. Basic Facts From Quantum Mechanics: A Brief Reminder

- In quantum mechanics:
  - in addition to the usual classical states $s_1, \ldots, s_n$,
  - we also have *superpositions*, i.e., states of the type

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

- Here $c_1, \ldots, c_n$ 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.

12. Quantum Mechanics (cont-d)

- If 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 $|c_n|^2$.
- 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

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

- Here a^* means complex conjugate: for $z = a + b \cdot \mathrm{i}$, we have $z^* = a - b \cdot \mathrm{i}$.
- The probabilities of getting n possible outcomes should add up to 1, which explains the above constraint

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

13. Quantum Mechanics (cont-d)

- After the measurement, if we get the result s_i, then the original state s transforms into the state s_i.
- We can measure 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 $|\langle s, s'_i \rangle|^2$.

14. 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 can be in one of the states $c_0 \cdot |0\rangle + c_1 \cdot |1\rangle$, with $|c_0|^2 + |c_1|^2 = 1$.
- It is known as a *quantum bit*, or *qubit*, for short.
- 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.

15. Bits and Qubits (cont-d)

- Let us denote:
  - the original basis $|0\rangle$ and $|1\rangle$, by $+$, and
  - the new basis $|0'\rangle$ and $|1'\rangle$ by $\times$.
- For states $|0\rangle$ or $|1\rangle$:
  - if we measure them with respect to the same basis,
  - we get exactly the prepared state: 0 or 1.
- Similarly, For states $|0'\rangle$ or $|1'\rangle$:
  - if we measure them in the $\times$ basis,
  - we also get back the prepared state.
- If we prepare a state in the $+$ basis and measure it in the $\times$ basis, we get 0 or 1 with probability $1/2$.
- If we prepare a state in the $\times$ basis and measure it in the $+$ basis, we also get 0 or 1 with probability $1/2$.

16. Quantum Physics Naturally Leads to a Random Number Generator

- The quantum cryptography algorithm uses a random number generator that produces 0 or 1 with prob. $1/2$.
- With quantum physics, there is no need – as many computers do now – to use *pseudo-random* numbers.
- Such numbers are usually generated by a complex algorithm.
- Indeed, in quantum physics, many processes produce actually random results.

17. Quantum Cryptography Algorithm: First Step

- Suppose that 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, generating $a_1, \ldots, a_n, a_{n+1}, \ldots, a_{2n}$.
- 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 $|1'\rangle$.

18. First Step (cont-d)

- 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;
  - if $b_i = 1$, B measures the i-th signal in the $\times$ basis.
- B then records the measurement results $m_1, \ldots, m_n$.

19. Second Step

- After B finishes the measurement process, A openly sends, to B, all the values a_{n+o}, $i = 1 \ldots, n$.
- For some number c of the indices i, A also sends the original values a_i.
- In half of the cases, the sending and measuring basis coincide, i.e., $a_{n+i} = b_i$.
- For these values i, the measurement result should reconstruct the original signal: $m_i = a_i$.
- 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.
- In other words, we have an eavesdropper.

20. Second Step (cont-d)

- Vice versa, suppose that there is an eavesdropper who listens to the conversation.
- The eavesdropper measures the signals while they go from A to B.
- The eavesdropper does not know the orientation a_{n+i}.
- So, 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.

21. Second Step (cont-d)

- 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 sufficiently 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.

22. Second Step (cont-d)

- 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.

23. Third Step

- Suppose that no eavesdropper was detected.
- Then the agent B sends, to A, the list of all the values $i_1, \ldots, i_m$ for which $a_{n+i} = b_i$.
- Of course, there is no need re-send the values a_i previously 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.

24. 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:

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

- 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.

25. 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 $n = 2m + c$.

26. How Quantum Cryptography Can Help Cyber-Security: Main Idea and Related Issue

- All the communications between sensors and computers must be encrypted by using quantum cryptography.
- There is an important issue with practical implementation.
- 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.

27. 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.

28. How to Deal with This Issue (cont-d)

- For example, the best known image compression algorithms such as JPEG'2000 use *wavelets*.
- 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; 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.

29. 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.
- E.g., in weather prediction, we 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.
- However, it is easier to add a few extra sensors than to analyze which locations are relevant.

30. 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.

31. Quantum Computing Can Help

- Quantum computing can help in this analysis:
  - there are quantum algorithms – such as the Deutsch-Jozsa algorithm,
  - that help us decide where certain bits are relevant.
- The most impressive example is an algorithm for the case when the input has only 1 bit.
- Then, 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, then the result $f(x)$ of the computation does not depend on x: $f(0) = f(1)$.
- If the input bit is relevant, then $f(0) \neq f(1)$.

32. Quantum Computing Can Help (cont-d)

- 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.
- 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.
- It is proven that the current quantum scheme for checking $f(0) = f(1)$ is, in effect, the only possible one.

33. Other Possible Applications of Quantum Computing to Cyber-Physical Systems

- In designing a cyber-physical system, we look for 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.

34. Applying Quantum Computing (cont-d)

- 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 enables us to find the desired alternative in $\sqrt{N}$ steps.
- For large N, this is much faster.
- E.g., when $N \approx 10^6$, the quantum search is three orders of magnitude faster.

35. Comment About Parallelization

- An additional speed-up can be obtained 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: 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*.
- It has been shown that the usual teleportation algorithm is, in some reasonable sense, unique
- Thus, it cannot be improved.

36. 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 $F(d)$.
- For complex systems, we rarely know the exact consequences of selecting each alternative.

37. Optimization (cont-d)

- 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.
- 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.

38. Optimization (cont-d)

- This 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
  - we 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 $m \in [M, \overline{M}]$.
- So, we can take $[M, \overline{M}]$ as the new value of the interval containing the actual maximum m.

39. Optimization (cont-d)

- If such a design does not exist, we conclude that $m \in [\underline{M}, M]$.
- So, 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: then:
  - 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.

40. Optimization (cont-d)

- 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 $\le \varepsilon$, we need $k = \left\lceil \log_2 \left(\dfrac{\overline{F} - \underline{F}}{\varepsilon} \right) \right\rceil$.
- Here, $\lceil x \rceil$ denotes the smallest integer which is greater than or equal to x.

41. Optimization (cont-d)

- 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, in comparison with the N-step exhaustive search:
  - we get almost the same speed-up
  - as for Grover's algorithm.

42. Acknowledgments

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).