Protecting Quantum Computers with an Antivirus

Sanjay Deshpande*, Chuanqi Xu*, Theodoros Trochatos*, Hanrui Wang†, Ferhat Erata*, Song Han†, Yongshan Ding*, Jakub Szefer*

*Yale University, †Massachusetts Institute of Technology

caslab.csl.yale.edu

Project Overview

We present a first-of-a-kind quantum computer antivirus [1]. The development of quantum computers is advancing rapidly in recent years. In addition to building bigger machines, researchers are exploring models for cloud-based usage of quantum computers, as well as multi-tenancy. As quantum computers become more widely accessible, possibly to malicious users, it is critical to start developing an understanding of quantum computer security. This work first explores and demonstrates crosstalk-based attacks on multi-tenant quantum computers and types of attacker circuits that could affect the operation of victim co-tenant circuits. It then proposes a method to detect such malicious circuits in quantum programs through a quantum computer antivirus. This work further demonstrates how the antivirus software can be run inside Intel SGX enclaves, to ensure not only that users cannot bypass the antivirus during circuit compilation, but also that users can compile programs on their own classical machines without exposing the circuit details to the quantum computer providers.

Threat Model

Our work assumes a multi-tenant quantum computer environment, where different users can request programs to be run on the quantum computers. Any malicious users may this way become co-located on the same quantum computer as a victim user. As multi-tenant quantum computers are not available from IBM yet, we emulate multi-tenancy by assigning a few qubits of a quantum computer to be "attacker" circuit and the remaining qubits to be "victim" circuit, and the two are executed in parallel as shown in Figure 1. We assume the objectives of the malicious users is to leverage crosstalk to affect computation of the victim users, such as to get victim users to generate incorrect results, e.g., misclassify inputs if the victim is running a quantum machine learning algorithm.

Crosstalk Attack and Evaluation

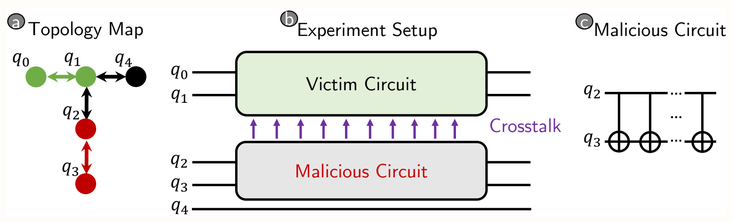

Figure 1: (a) Topology of `ibm_lima` machine, (b) Our experiment setup, (c) Example malicious circuit (Series of CNOT gates).

We use the experimental setup shown in Figure 1b for our experiments. We evaluate three 2-qubit benchmark circuits as victim namely, Grover's Algorithm, Deutsch-Josza Algorithm, and Bernstein-Vazirani Algorithm against ten variants of malicious circuits. For demonstration purpose, we only showcase evaluation based on one malicious circuit here. We use the 5-qubit IBM quantum computer `ibmq_lima` to perform our evaluation, whose topology is shown in Figure 1a. We quantify the amount of crosstalk by using the output probability as a metric. In these experiments, we increase the duration of the malicious circuit (i.e., increase in the number of gates in the malicious circuit) and observe the output probability of the victim circuit.

Figure 4 demonstrates the effectiveness of our 2-qubit malicious circuit (shown in Figure 1c) across different benchmark circuits.

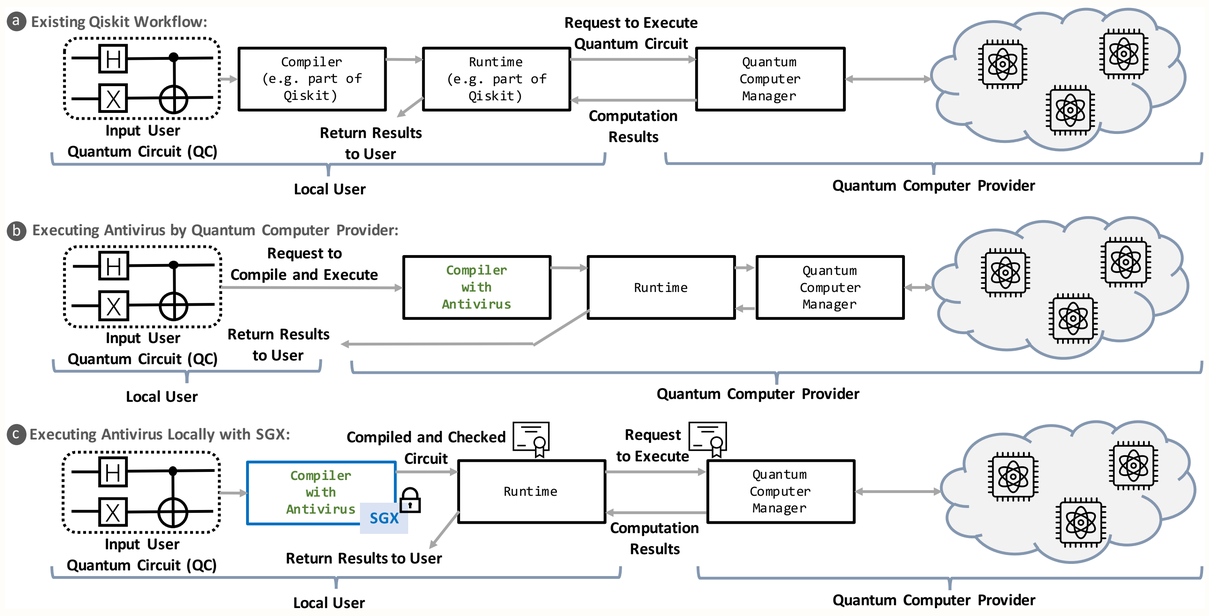

Figure 2: (a) Typical workflow of today's cloud-based quantum computer infrastructures, without any antivirus protection. Compilation process without (b) and with (c) SGX feature.

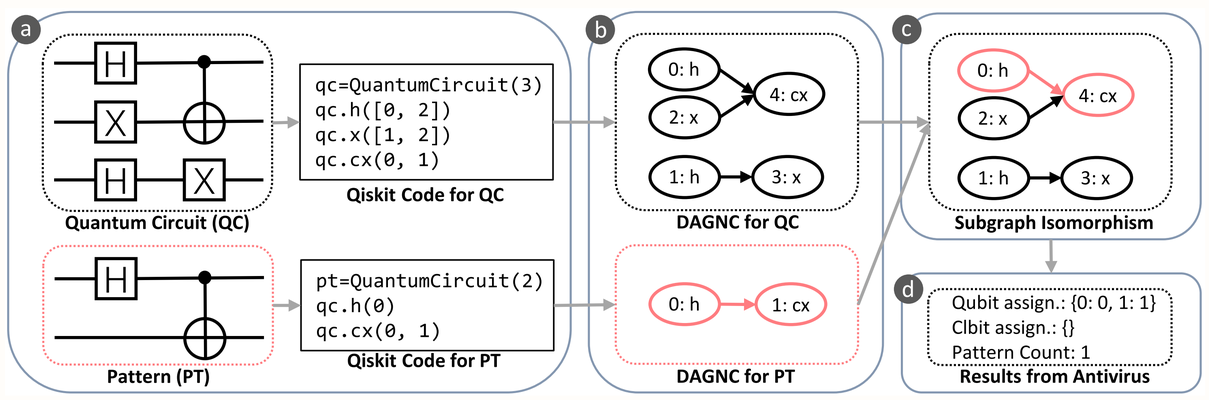

Figure 3: Overview and workflow of the quantum computer antivirus.

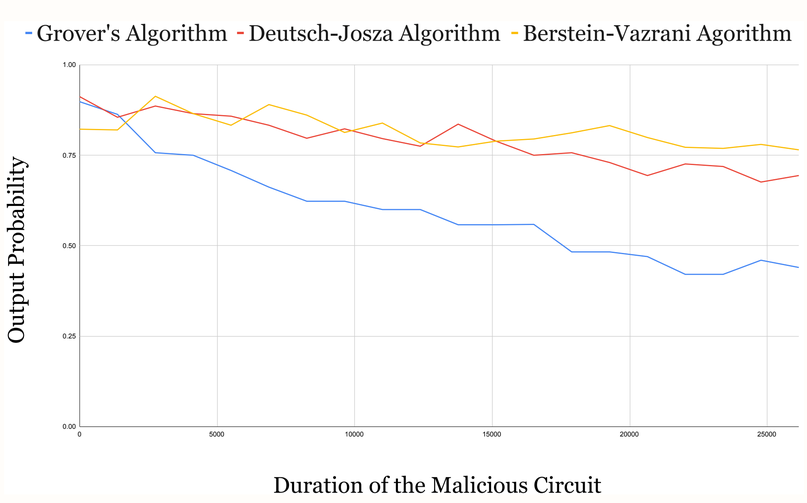

Figure 4: Duration of quantum computer circuits versus output probability of victim circuits.

We test the output probability on three different victim circuits: Grover's, Deutsch-Josza and Bernstein-Vazirani algorithms. We make an observation that, as the duration of the malicious circuit increases, the output probability of all the benchmark circuits is lowered.

Circuit	Circuit Duration (dt)	Scanning Time without SGX (s)	Scanning Time with SGX (s)
Malicious	20K	1.07	249.23
Grover	10K	1.06	248.73
Deutsch-Jozsa	10K	0.90	229.91
Bernstein-Vazirani	10K	1.06	248.95

Table 1: Antivirus scan time of malicious and victim circuits, without and with use of SGX.

Conclusion

This work presented the first proposed antivirus for quantum computers. The work evaluated a number of malicious circuits that use different patterns of gates to trigger crosstalk errors on adjacent qubits. The attack experiments were evaluated on real, publicly accessible cloud-based IBM quantum computers. The antivirus was developed as means to proactively check for malicious circuits in the source code of the quantum programs. It was shown to have 100% detection with no false positives on the tested malicious and victim circuits.

Quantum Antivirus

The malicious circuits can be detected at compile time by using the antivirus to scan input quantum computer programs for occurrences of malicious patterns. In our antivirus program shown in Figure 3, a directed acyclic graph is used to represent a quantum circuit, and thus the pattern matching problem can be reduced to the *subgraph isomorphism problem*. In Figure 2, we present the workflow of Qiskit, and our proposed antivirus scheme running by quantum computer providers and locally, without and with SGX feature.

As Table 1 shows, the pattern scanning operation is very quick without SGX. It can be observed that SGX has a very large, but almost constant overhead, which comes from setting up the SGX enclave and loading all the necessary Qiskit and Python code into the enclave.

Acknowledgements

This work was supported in part by NSF grants 1901901. We would like to thank Dmitrii Kuvaiskii from Intel for the help with Intel SGX and Gramine tools.

References

[1] S. Deshpande, C. Xu, T. Trochatos, Y. Ding, and J. Szefer, "Towards an antivirus for quantum computers," in *2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2022, pp. 37–40.