Protecting Quantum Computers with a Trusted Controller

Theodoros Trochatos
Yale University
New Haven, CT, USA
theodoros.trochatos@yale.edu

Chuanqi Xu
Yale University
New Haven, CT, USA
chuanqi.xu@yale.edu

Sanjay Deshpande
Yale University
New Haven, CT, USA
sanjay.deshpande@yale.edu

Yao Lu
Yale University
New Haven, CT, USA
physics.lu@yale.edu

Yongshan Ding
Yale University
New Haven, CT, USA
yongshan.ding@yale.edu

Jakub Szefer
Yale University
New Haven, CT, USA
jakub.szefer@yale.edu

***Abstract*—Rapidly evolving research on quantum computers brings closer reality of practical quantum computers that can solve previously untractable problems. However, as these computers become more powerful, they will process and generate valuable data and intellectual property, making them targets of security attacks. As a result, there is a need to develop architectures for securing quantum computers and the computation. This work presents emerging ideas on design of a trusted quantum computer controller as a pivotal component in securing quantum computing systems. At its core, a trusted controller acts to protect the privacy and security of user's circuits executing on the quantum computer itself. In this work, the classical controller used to generate control pulses for quantum computers is augmented with trusted hardware. The trusted classical hardware can analyze and manipulate circuits and instructions before generating control pulses sent to the untrusted, quantum processing unit. This work analyzes the promises of trusted controller and presents ideas for design of secure cloud-based quantum computers.**

***Index Terms*—quantum computing, security, defenses, quantum computer controller**

I. Introduction

Quantum computing holds the promise of unprecedented computational power, promising to revolutionize various industries, from materials research to discovery of new medicines. As quantum technologies advance, the realization of practical quantum cloud computing becomes increasingly feasible, presenting both novel opportunities but also challenges. To help expand availability of quantum computing, multiple cloud providers have opened up access to small and medium-scale quantum computers to customers as Infrastructure as a Service (IaaS). For example, cloud-based services such as IBM Quantum [22], Amazon Braket [10] and Azure Quantum [20] already provide access to the NISQ (Noisy Intermediate Scale Quantum) quantum computers remotely for users. Amidst the excitement of harnessing quantum capabilities in the cloud, there is an unsolved critical issue of security of the quantum computers and the quantum circuits they execute.

To help protect from the untrusted quantum computer cloud providers, several blind quantum computation protocols have been proposed [1], [3], [7]–[9], [12]–[17], [23]. However, most of the work remains theoretical, due to the fact that BQC requires a local, trusted quantum computer, as well as, quantum networking used to connect the remote and local quantum computers, which is not available today.

An alternative approach to protect quantum computers and circuits they execute is to develop security architectures where the quantum computer is augmented with trusted components or hardware. Instead of putting trust in mathematical constructs (which would be preferred, but has tremendous overheads), the trust is put in the manufacturer of the equipment. Assuming the added, trusted hardware is indeed secure, it can offer protections for the system. This is well known approach in classical computers, for example in confidential computing and architectures such as Intel SGX [4].

Existing work in this area has focused on design of secure quantum processing units [24]–[26], which require modification of the quantum processing unit. In the existing work, components outside the quantum processing unit are untrusted, while the quantum processing unit is trusted along with its modifications. As an emerging idea, this paper introduces a dual concept of having a trusted quantum controller, but untrusted quantum processing unit. This work explores promises of having a trusted controller, while the quantum processing unit is untrusted. The controller is typically simpler, and easier to modify, making it more realistic that it could be augmented with new security features. Also, it uses classical components such as CPU or FPGA, which are well researched and have established ideas regarding design of secure architectures for both CPU and FPGA.

This paper delves into the key components of a trusted quantum controller, exploring how the integration of a trusted controller could act as a guardian for the quantum computing systems. We examine the architecture, mechanisms and ben-

This work was supported in part by National Science Foundation grant 2312754. This work was performed while authors were with Yale University.

efits of a trusted quantum controller, shedding light on their role in mitigating security risks while facilitating the broader adoption of quantum cloud computing. We also analyze the limitations of such design.

A. Contributions

The contributions of this work are as follows:

- We present trusted quantum computer controller as a new design paradigm for developing future, secure quantum computers.
- We define a realistic threat model for exploring protections offered by trusted quantum computer controller.
- We analyze the benefits and limitations of such design, and give direction for emerging research on secure quantum computers.

II. Background

A. Cloud-Based Quantum Computers

Most of the cloud-based quantum computer deployments available today follow a similar workflow: users submit jobs, i.e. quantum circuits also called quantum programs, to the cloud provider, then jobs are queued for execution, when the user's job reaches the head of the queue it is executed on the real quantum computer and measurements are returned to the user. The remote user is able to develop their code in a high-level language, such as Python. The code is then transpiled into native quantum gates, which can be executed by the target quantum computer, also called "backend". The transpiled circuit is then sent to the quantum computer provider as a job that is to be executed on the backend. Remote users can also specify their own custom control pulses. Both native basis gates and custom pulses are supported by our security architecture.

Based on the request to execute the provided quantum circuit, the quantum computer manager is responsible to schedule the job on the quantum computer, when it is available. Further, the transpiled circuit is converted into control RF pulses, and the controller FPGAs and arbitrary waveform generators (AWGs) are instructed to generate the RF pulses actually sent to the quantum computer hardware. The readout of the results, also called "measurement", is likewise done by issuing a number of RF control pulses and reading out the response. During measurement operation, the qubit state is collapsed into the classical state and the classical bits are sent back to the user as the computation results. Each job typically consists of thousands of "shots". The users collect measurement statistics of all the shots to compute the probabilities of the different classical bit results and typically find the result with the highest probability.

B. Control Pulses

Superconducting qubits are typically manipulated using microwave pulses. To execute native gates on a quantum computer, the requisite control pulses must be generated and supplied to each specific gate. IBM Quantum provides the pulses for all native gates as part of the quantum computer specifications. These parameters undergo regular updates through calibrations to ensure consistent fidelity over time. A pulse in this context is typically characterized by its envelope, frequency, and phase. Taking the example of superconducting qubit control, the envelope outlines the signal shape generated by the arbitrary waveform generator (AWG), a commonly used laboratory instrument. Meanwhile, the frequency and phase define a periodic signal employed to modulate the envelope signal.

C. Quantum Computer Controller

The quantum computer controller is a set of classical equipment that is used to control the qubits. It is typically composed of classical processors, CPUs or FPGAs, as well as AWGs and other components. More integrated controllers may contain all the components in one system, such as an FPGA with RF connections. In all cases, there is need for analog to digital (ADC) and digital to analog (DAC) conversions. We can abstract the controller as a unit that receives digital information about the circuit that should be executed, and transforms it into analog pulses sent to the quantum processing unit. While there is research on cryogenic CMOS circuits, the controllers today are typically located outside the quantum processing unit. When considering non-superconducting quantum computers, the controller may be involved in generation of control signals for lasers, rather than RF signals, for example. But the key idea remains that the controller is an intermediary between the digital representation of the circuits that users submit, and analog signals used to actually actuate the qubits to execute the circuits on the quantum processing unit.

D. Pulse-Level Circuit Description

To fully describe a quantum program, all pulses for all the channels need to be defined, including when the pulses should start relative to the starting point of the circuit, to what qubits the pulses will be applied, and other physical operations like frequency or phase change, need to be specified. This information, referred to as pulse information, along with other useful information forms a so-called pulse-level circuit description. Pulse-level circuits and pulse information are important and valuable to be provided to users, because they enable users to verify quantum circuits and check execution details. The quantum computer controller effectively generates the control pulses based on the provided pulse-level description.

E. Gate-Level and Pulse-Level Circuits

In order to actually generate pulse-level circuits, a number of steps are needed. The first step in developing a quantum circuit or program is to build a logic-level circuit with a quantum development kit, such as Qiskit [5], Amazon Braket SDK [10], Q# [20] and Cirq [18]. Analogous to classical computing, logic-level quantum circuits usually contain high-level descriptions. A series of operations need to be done to transform them into low-level and hardware-specific instructions, which is similar to the preprocessing, compilation

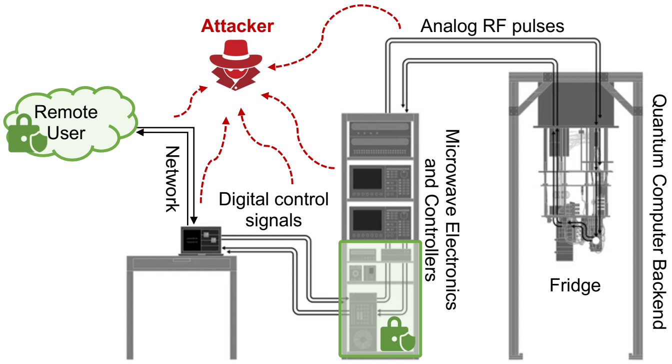

Fig. 1. Schematic of a typical superconducting quantum computer, showing an honest-but-curious cloud provider attempting to spy on the control pulses and equipment outside the trusted controller. The green boundaries are trusted components. Note that our threat model can easily expand to non-superconducting quantum computers.

and assembly process for classical computing programs. The second step is then to *transpile* the circuits, which is the term used by Qiskit to represent the operations and transformations that are like preprocessing and compilation. The process of transpiling involves many steps, including decomposing non-native quantum gates into groups of native gates, grouping and removing quantum gates to reduce the number of gates, mapping the logic qubits in the original circuits to the physical qubits on the specified quantum computers, routing the circuit under limited topologies, potentially optimizing circuits to lower error, and so on. The third step is termed *schedule* in Qiskit, which transforms gate-level circuits into pulse-level circuits. Scheduling further maps quantum circuits to microwave pulses, which are the ultimate physical operations used to regulate and control qubits. Based on previously calibrated data for each basis gate on each qubit or qubit pair, scheduling creates microwave pulse sequences that are ready to be carried out for quantum programs. The end result is a circuit composed only of control pulses representing basis gates that can be executed on the target quantum computer.

F. Intellectual Property Contained in the Circuits

Quantum computer circuits are realizations of intellectual property. Proprietary algorithms need to be translated into circuits, and actually executed on quantum computers to obtain the results. Thus, cloud providers naturally have access to all the intellectual property when they receive a circuit for execution. Mechanisms need to be introduced somewhere in the quantum computing system to protect the user's circuits.

III. Goal of this Paper

Previous techniques aim to protect arbitrary quantum programs, necessitating a classical client to be augmented with quantum capabilities and quantum networking (BQC), or incurring exponential computational overhead (QHE). For the aforementioned reasons, these techniques are rendered impractical. Trochatos et al. [26] recently proposed a novel hardware architecture for a trusted execution environment tailored for superconducting quantum computers. However, this method comes with its own set of limitations. The technique is specific to superconducting quantum computers, as other quantum computer types do not incorporate a dilution refrigerator. In this paper, we aim to minimize the trusted area in a quantum computer. Rather of having the dilution refrigerator as a large trusted computing base, we examine different assumptions and components of the quantum computer that can operate as the trusted area.

IV. Threat Model

Our work considers the threat model of an honest-but-curious cloud provider. Other threat models, ranging from fully malicious cloud provider to fully trusted cloud provider, are possible, but they are orthogonal to our work. The honest-but-curious cloud provider assumption encompasses different attacks. The cloud provider could outright aim to spy on users to learn their intellectual property or data contained in the quantum circuits. Or, if the cloud provider as a whole may not be malicious, one of the employees, the so called malicious insider, could try to spy on the operation of the quantum computer. Such two scenarios are isomorphic in our threat model and we simply consider them together under the honest-but-curious cloud provider model. In the honest-but-curious the cloud providers are untrusted, they can spy on operation of the quantum computers, but do not modify them. The setup of a typical superconducting quantum computer with the quantum controller as the trusted area is shown in Figure 1.

V. Possible Trusted Controller Architectures

Our work aims to analyze what protections a trusted quantum computer controller could give users from honest-but-curious cloud providers who want to learn what quantum programs or data, i.e., quantum circuits or algorithms, are being executed by the user. We assume the cloud provider has access to all the information and electronics, including the interior of the dilution refrigerator (in case of superconducting quantum machines). Our threat model applies to both superconducting and non-superconducting quantum machines. We assume the honest-but-curious provider can observe any communication (digital and analog), but does not tamper with it (such tampering can be detected by checking the quality of the solution). Thus, our goal is to protect from passive attacks, such as information leakage. Active attacks such as fault injection are orthogonal and future work.

A. Abstracted View of a Quantum Computer Controller

The only assumption that we make is the quantum controller, which converts the digital control pulses to analog RF pulses, to be trusted. We develop a simple abstraction of a quantum computer controller: A classical processing unit (CPU/FPGA), which is required to manage and orchestrate the quantum operations and a digital-to-analog and analog-to-digital converter unit (DAC/ADC), which converts the digital signals into analog signals suitable for manipulating the qubits, and also converts analog signals back to digital ones that can be processed on the controller. Considering the two units: CPU/FPGA and DAC/ADC, there are four possible combinations:

B. Untrusted CPU/FPGA and Untrusted DAC/ADC

If both the CPU/FPGA and DAC/ADC in a quantum computing system are untrusted, clearly the units cannot be leveraged to provide protection mechanisms. This is equivalent to today's cloud-based quantum computers, where there are no added security features nor trust in the controller.

C. Untrusted CPU/FPGA and Trusted DAC/ADC

An untrusted CPU/FPGA is not able to provide any security features. However, a trusted DAC/ADC, on the other hand, could be leveraged to protect the circuits during the digital to analog, or analog to digital conversion. As DAC/ADC is involved in conversion of signals, we assume it has no computation power on its own, so it may not be able to add, remove, or modify the quantum circuits. However, in theory, it is possible for a trusted DAC/ADC to manipulate its operation as it generates analog signals (DAC), or when it recovers digital signals from analog inputs (ADC).

- **Hiding Power Signatures:** The power consumption of electronic devices, including DAC/ADC, can exhibit characteristic patterns during different operations. By intentionally introducing variations in power consumption that are independent of the actual digital data (for DAC) or analog signal (for ADC), the DAC/ADC unit may attempt to make it more difficult for an attacker to analyze the power side-channel and extract sensitive information.
- **Injecting Noise:** A trusted DAC/ADC unit could dynamically adjust output signal (for DAC) to create noise in the RF lines and mitigate EM based side channels, or it could manipulate digital data (for ADC) to confuse the cloud provider about what is actually read from the quantum computer signals.

D. Trusted CPU/FPGA and Untrusted DAC/ADC

A trusted CPU/FPGA can be leveraged to add, remove, or modify the quantum gates in a circuit before it is sent to the quantum computer. It can also manipulate the received measurements before they are passed onto the rest of the cloud computing infrastructure. Importantly, trusted CPU/FPGA can be used to decrypt circuits received from users, and encrypt the results.

- **Adding Circuit and Its Inverse:** The trusted CPU/FPGA could insert quantum gates and their inverses [21] into a circuit. The trusted CPU/FPGA could in this way obfuscate the structure of the circuit. The output analog signals observed by the cloud provider would correspond to the obfuscated circuit.
- **Swapping Qubits:** Quantum swapping operations allow the rearrangement of qubits [24]. The trusted CPU/FPGA, could dynamically modify quantum circuits to swap qubits or channels. The resulting circuit executed on the quantum computer would have different qubit orders, adding some obfuscation.
- **Executing Dummy Shots:** The trusted CPU/FPGA could execute dummy or decoy shots. These dummy shots would later have to be discarded. The cloud provider, not knowing which are right and which are wrong shots, would have more difficulty guessing what is the actual computation being done by the user.

E. Trusted CPU/FPGA and Trusted DAC/ADC

Having both a trusted CPU/FPGA and a trusted DAC/ADC in a quantum computing system combines the benefits of both a trusted CPU/FPGA and a trusted DAC/ADC.

VI. Discussion

In this section, we provide a discussion, exploring the trade-off between minimizing the trusted computing base and maximizing security in the context of protecting quantum circuits. The decision to designate the quantum controller as the trusted boundary presents a compelling case for reducing the area that needs to be trusted, but may introduce significant security challenges, as the attack complexity from an honest-but-curious server's perspective is $O(n)$. In this scenario, a malicious server potentially can gain access to critical outputs and operations of the quantum controller by brute-forcing the outputs, which could compromise the confidentiality of the quantum computing process.

On the other hand, leveraging the dilution refrigerator as the trusted boundary [24]–[26] expands the trusted computing base, yet offers enhanced protection. Our analysis suggests that despite the increased complexity associated with securing a larger area, the dilution refrigerator provides a robust shield against various attack vectors and thus bolstering the overall security posture of quantum computing systems, as the attacker requires exponential number of guesses to brute-force and retrieve the outputs of the circuit.

It's essential to acknowledge that the effectiveness of these approaches depends significantly on the threat landscape and the capabilities of potential adversaries. For certain applications or scenarios, an $O(n)$ level of complexity required by the attacker's perspective might be deemed acceptable, especially if coupled with stringent monitoring and mitigation strategies.

VII. Prior Work

To help protect from the untrusted quantum computer cloud providers, several blind quantum computation protocols have been proposed [1], [3], [7]–[9], [12]–[17], [23]. Most of the work remains theoretical, due to the fact that BQC requires a local, trusted quantum computer, as well as, quantum networking used to connect the remote and local quantum computers, which is not available today. Qenclave [11] shows that Remote State Rotation can be used in the context of the BQC protocol to delegate a universal quantum computation with perfect blindness. In this concept, the clients' quantum abilities are restricted to receiving single-qubit states, applying a random z-rotation to it, and forwarding it to the next client or server. However, as of now, the question whether states prepared by rotation-only clients are sufficient for verification remains open, as [11] only showed that these kinds of states are enough to achieve blindness.

Alternatively, one area of research focuses on the security threats posed by untrusted compilers in quantum circuits. In [6] the authors propose inserting a small random circuit into the original circuit for obfuscation, which is then sent to the untrusted compiler and concatenate the inverse of the random circuit in the compiled circuit to recover the original functionality. However, the quantum circuits are still vulnerable to an untrusted quantum cloud provider, who may be able to observe the control pulses of the circuit and retrieve sensitive intellectual from the running circuit or algorithm.

Approaches which do not use blind quantum computation include [19] which adds at compile time pairs of `RX` gates into the circuit to obfuscate it to the cloud provider. The work however does not leverage any trusted hardware, and all the inserted gates are actually executed, severely limiting the amount of possible obfuscation compared to our work. Trochatos et al. [25], [26] recently proposed a novel hardware architecture for a trusted execution environment tailored for superconducting quantum computers. However, this method comes with its own set of limitations. The technique is specific to superconducting quantum computers, as other quantum computers types do not incorporate a dilution refrigerator. Additionally, the approach assumes that the quantum provider cannot access the trusted region within its own refrigerator. As an alternative direction, Ayanzadeh et al. propose an application-specific solution to protect the coefficients of QAOA [2]. Although promising, this approach suffers from an extremely limited scope on a single application.

Another relevant work [27] concentrates on the obfuscation of quantum hybrid-classical algorithms, specifically QAOA. The primary objective is to safeguard confidential information encoded within the circuit parameters against potential threats from untrusted quantum hardware. Moreover, in [28], a strategy is presented that advocates the equal distribution of computations across various hardware alternatives. Additionally, an adaptive heuristic is introduced to discern instances of tampering with the hardware, enhancing the overall security of quantum computations.

VIII. Conclusion And Future Work

Our research shows the gap between current architectures and the need for confidentiality in order to make quantum computation widely acceptable. In this paper, we explored different assumptions for the trusted region of a quantum controller. We classify four different, realistic threat model scenarios and we propose a concept architecture to ensure the confidentiality of the quantum computers, keeping the trusted computing base as minimal as possible. Finding a realistic solution to these problems of the previous works is essential for the acceptance of quantum computing and toward the goal of a truly confidential quantum computing landscape.

References

[1] Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations, 2008. https://arxiv.org/pdf/0810.5375.pdf.

[2] Ramin Ayanzadeh, Ahmad Mousavi, Narges Alavisamani, and Moinuddin Qureshi. Enigma: Privacy-preserving execution of qaoa on untrusted quantum computers. *arXiv preprint arXiv:2311.13546*, 2023.

[3] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 517–526, 2009. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5438603.

[4] Victor Costan and Srinivas Devadas. Intel sgx explained. *Cryptology ePrint Archive*, 2016.

[5] Andrew Cross. The ibm q experience and qiskit open-source quantum computing software. In *APS March meeting abstracts*, volume 2018, pages L58–003, 2018.

[6] Subrata Das and Swaroop Ghosh. Randomized reversible gate-based obfuscation for secured compilation of quantum circuit. *arXiv preprint arXiv:2305.01133*, 2023.

[7] Vedran Dunjko, Elham Kashefi, and Anthony Leverrier. Blind quantum computing with weak coherent pulses. *Physical Review Letters*, 108(20), may 2012. https://arxiv.org/pdf/1108.5571.pdf.

[8] Joseph F. Fitzsimons and Elham Kashefi. Unconditionally verifiable blind quantum computation. *Physical Review A*, 96(1), jul 2017. https://arxiv.org/pdf/1203.5217.pdf.

[9] Vittorio Giovannetti, Lorenzo Maccone, Tomoyuki Morimae, and Terry G. Rudolph. Efficient universal blind quantum computation. *Physical Review Letters*, 111(23), dec 2013. https://arxiv.org/pdf/1306.2724.pdf.

[10] Constantin Gonzalez. Cloud based qc with amazon braket. *Digitale Welt*, 5:14–17, 2021.

[11] Yao Ma, Elham Kashefi, Myrto Arapinis, Kaushik Chakraborty, and Marc Kaplan. Qenclave-a practical solution for secure quantum cloud computing. *npj Quantum Information*, 8(1):128, 2022.

[12] Atul Mantri, Carlos A. Pé rez-Delgado, and Joseph F. Fitzsimons. Optimal blind quantum computation. *Physical Review Letters*, 111(23), dec 2013. https://arxiv.org/pdf/1306.3677.pdf.

[13] Tomoyuki Morimae. Continuous-variable blind quantum computation. *Physical Review Letters*, 109(23), dec 2012. https://arxiv.org/pdf/1208.0442.pdf.

[14] Tomoyuki Morimae, Vedran Dunjko, and Elham Kashefi. Ground state blind quantum computation on aklt state, 2011. https://arxiv.org/pdf/1009.3486.pdf.

[15] Tomoyuki Morimae and Keisuke Fujii. Blind topological measurement-based quantum computation. *Nature Communications*, 3(1), sep 2012. https://arxiv.org/pdf/1110.5460.pdf.

[16] Tomoyuki Morimae and Keisuke Fujii. Blind quantum computation protocol in which alice only makes measurements. *Physical Review A*, 87(5), may 2013. https://arxiv.org/pdf/1201.3966.pdf.

[17] Tomoyuki Morimae and Takeshi Koshiba. Composable security of measuring-alice blind quantum computation, 2013. https://arxiv.org/pdf/1306.2113.pdf.

[18] Victory Omole, Akhilesh Tyagi, Calista Carey, AJ Hanus, Andrew Hancock, Austin Garcia, and Jake Shedenhelm. Cirq: A python framework for creating, editing, and invoking quantum circuits, 2020.

[19] Tirthak Patel, Daniel Silver, Aditya Ranjan, Harshitta Gandhi, William Cutler, and Devesh Tiwari. Toward privacy in quantum program execution on untrusted quantum cloud computing machines for business-sensitive quantum needs. *arXiv preprint arXiv:2307.16799*, 2023.

[20] Kumar Prateek and Soumyadev Maity. Quantum programming on azure quantum—an open source tool for quantum developers. In *Quantum Computing: A Shift from Bits to Qubits*, pages 283–309. Springer, 2023.

[21] Abdullah Ash Saki, Aakarshitha Suresh, Rasit Onur Topaloglu, and Swaroop Ghosh. Split compilation for security of quantum circuits. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–7, 2021.

[22] Alan C Santos. The ibm quantum computer and the ibm quantum experience. *arXiv preprint arXiv:1610.06980*, 2016.

[23] Takahiro Sueki, Takeshi Koshiba, and Tomoyuki Morimae. Ancilla-driven universal blind quantum computation. *Phys. Rev. A*, 87:060301, Jun 2013.

[24] Theodoros Trochatos, Sanjay Deshpande, Chuanqi Xu, Yao Lu, Yongshan Ding, and Jakub Szefer. Dynamic pulse switching for protection of quantum computation on untrusted clouds. 2024.

[25] Theodoros Trochatos, Chuanqi Xu, Sanjay Deshpande, Yao Lu, Yongshan Ding, and Jakub Szefer. Hardware architecture for a quantum computer trusted execution environment, 2023.

[26] Theodoros Trochatos, Chuanqi Xu, Sanjay Deshpande, Yao Lu, Yongshan Ding, and Jakub Szefer. A quantum computer trusted execution environment. *IEEE Computer Architecture Letters*, 22(2):177–180, 2023.

[27] Suryansh Upadhyay and Swaroop Ghosh. Obfuscating quantum hybrid-classical algorithms for security and privacy. *arXiv preprint arXiv:2305.02379*, 2023.

[28] Suryansh Upadhyay, Rasit Onur Topaloglu, and Swaroop Ghosh. Trustworthy computing using untrusted cloud-based quantum hardware. *arXiv preprint arXiv:2305.01826*, 2023.