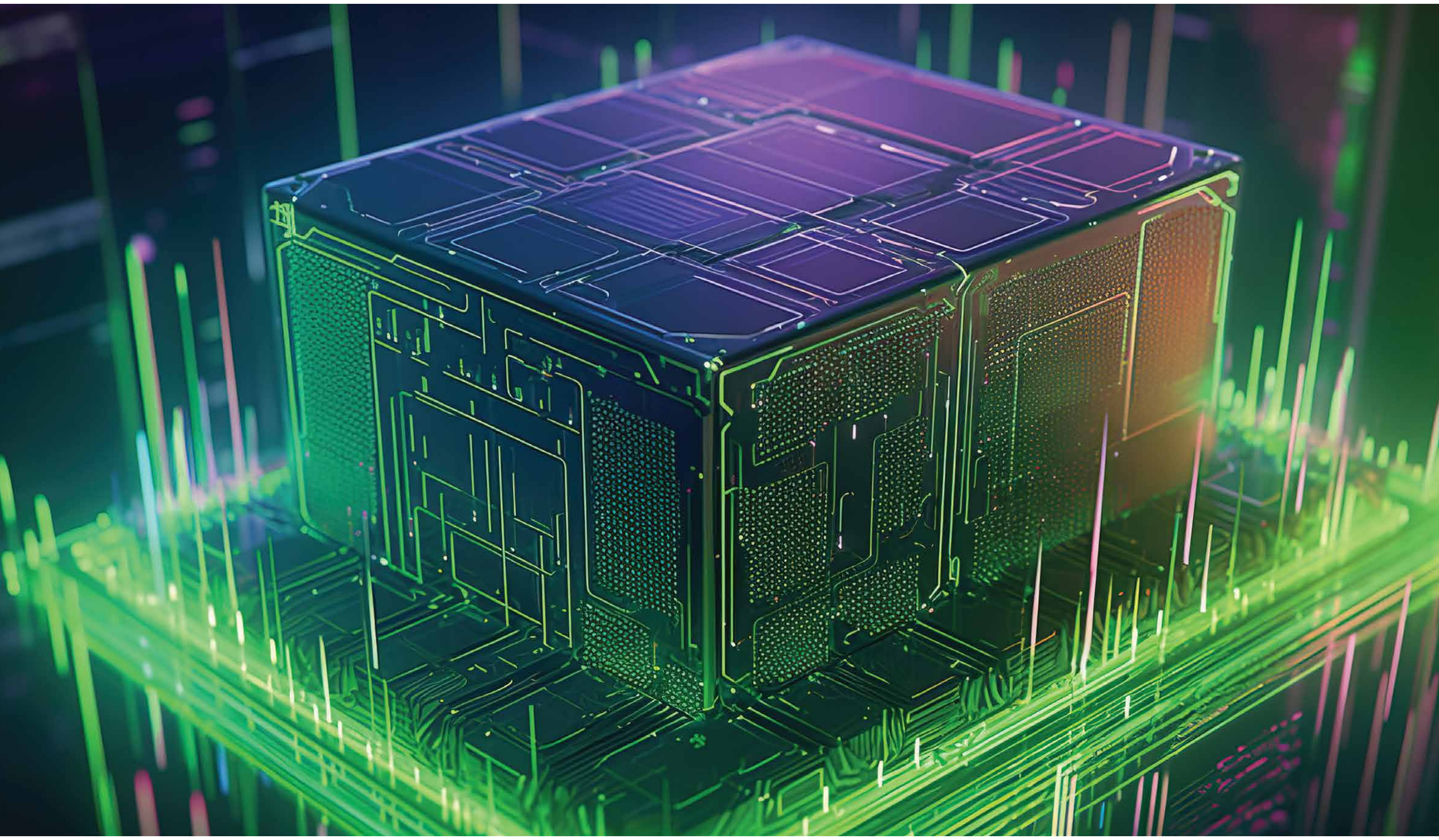


IMAGE LICENSED BY INGRAM PUBLISHING

NISQ Quantum Computing: A Security-Centric Tutorial and Survey

Fan Chen, Lei Jiang, Hausi Müller, Philip Richerme, Cheng Chu, Zhenxiao Fu, and Min Yang

Abstract

Quantum computing (QC) demonstrates substantial theoretical promise in addressing classically intractable problems. Recent investments and advancements across QC system stacks, including hardware, software, and algorithms, underscore a pivotal shift from theoretical exploration to the practical realization of applications. Despite this progress, the prevailing emphasis has predominantly centered on performance enhancement, often overlooking security considerations. In response to this gap, our article presents a comprehensive tutorial and survey aimed at identifying and categorizing vulnerabilities inherent in quantum computing systems. Beginning with an overview encompassing essential principles, ecosystem components, and unique attributes in the quantum computing system stack, we also provide a summary of development resources to facilitate efficient initiation in this domain. Building on this foundational knowledge, we introduce a taxonomy of QC security organized by victim layer and security attack objectives. Utilizing this taxonomy as a guiding framework, the article delivers an extensive survey of the latest advancements in QC security, with the overarching goal of equipping the reader with a comprehensive understanding of quantum

Digital Object Identifier 10.1109/MCAS.2024.3349665
Date of current version: 6 March 2024

computing system principles and an informed awareness of diverse and dynamic QC security threats. The intention is to benefit both industry stakeholders and research communities, ultimately aiming to proactively identify and mitigate security concerns within QC systems, thereby establishing a robust foundation for secure quantum computing environments.

Index Terms—Quantum computing, quantum security, NISQ.

I. Introduction

Quantum Computing (QC), grounded in established theoretical computational models [1], [2], [3], [4], possesses the remarkable potential to exceed the capabilities of the most powerful classical computers. The availability of cloud-based [5], [6], [7], [8], [9] Noisy Intermediate-Scale Quantum (NISQ) [10] computers, coupled with recent enhancements in crucial QC toolflows [11], [12], [13], [14], [15], [16], [17] has empowered quantum computing to showcase quantum advantage across a variety of applications and platforms [18], [19], [20], [21], [22], even before achieving fault tolerance.

With the escalating qubit counts and the growing fidelity of quantum computers, their potential to execute innovative algorithms and generate sensitive intellectual property has become increasingly compelling. In this context, the security of quantum computing systems is of paramount importance, as an insecure QC system not only jeopardizes users but also poses a significant risk to our broader society. However, there is a notable absence of a systematic research effort addressing the evolving landscape of quantum threats, exploring potential vulnerabilities, and establishing robust countermeasures to safeguard the integrity of quantum systems and the sensitive information they process.

In this article, we take the first step in providing a comprehensive tutorial and survey focused on identifying and categorizing vulnerabilities inherent to quantum computing systems. Our ultimate objective is to establish a strong foundation for secure quantum computing environments. This article serves as an initial stride towards this goal by proactively illuminating the landscape of quantum security threats, benefiting both industry stakeholders and research communities.

A. Organization of This Article

This article is organized as follows.

- Section 2 provides an overview of quantum computing systems, encompassing essential principles of quantum computing (Section 2.1), ecosystem components (Section 2.2), and QC design system stack (Section 2.3-2.5). This entails an emphasis on the unique attributes that differentiate QC from classical computing systems. We also summarize various development resources that facilitate an efficient commencement process for researchers and practitioners in this domain.
- Section 3 introduces an initial taxonomy for organizing QC security research. This taxonomy employs a two-dimensional metric, encompassing both the victim logical layers of the QC system and the objectives of security attacks. Aligned with classical taxonomies, it facilitates seamless knowledge transfer while maintaining the flexibility to accommodate ongoing expansion and adaptation as our understanding and research in quantum computing and quantum security advance.
- Section 4 surveys the latest advancements in QC security, utilizing our proposed taxonomy as a guiding framework. We classify these works based on victim layers within quantum computing: QC hardware (Section 4.1), QC software (Section 4.2), and QC algorithms (Section 4.3). Within each victim layer, works are further categorized according to their security objectives, encompassing information leaks, untargeted fault injection, and targeted attacks. Special emphasis is placed on highlighting the unique characteristics of security threats and their corresponding attack vectors. Our analysis of existing research also explores potential defense strategies, detailed in Section 4.4.
- Section 5 concludes this survey.

II. Background on Quantum Computing

In this section, we initially present the fundamental aspects of QC systems (Section 2.1) and outline the configuration of the existing QC ecosystem (Section 2.2). Following that, we offer a detailed introduction to the QC system stack, covering QC hardware (Section 2.3), QC software (Section 2.4), and QC algorithms (Section 2.5). For each layer of the stack, we emphasize state-of-the-art development resources, elaborate on key features, and summarize their characteristics or highlight challenges in the NISQ era. This section is designed to provide fundamental insights into QC systems, serving as a valuable resource for researchers and practitioners as they embark on their exploration of quantum computing.

Fan Chen, Lei Jiang, Cheng Chu, Zhenxiao Fu, and Min Yang are with the Department of Intelligent Systems Engineering, Indiana University, Bloomington, IN 47405 USA (e-mail: fc7@iu.edu; jiang60@iu.edu; chu6@iu.edu; zhfu@iu.edu; my36@iu.edu).
Hausi Müller is with the Department of Computer Science, University of Victoria, Victoria, BC V8P 5C2, Canada (e-mail: hausi@uvic.ca).
Philip Richerme is with the Department of Physics, Indiana University, Bloomington, IN 47405 USA (e-mail: richerme@indiana.edu).

A. Basics of Quantum Computing

1) Foundational Principles

Quantum computing [23] explores information processing through quantum mechanical systems, leveraging principles like superposition, entanglement, and interference. This burgeoning field holds the promise of revolutionizing computation, enabling the solution of complex problems previously beyond the scope of classical paradigms.

Qubit. A *qubit* is the simplest quantum mechanical system, characterized by a linear combination of a two-dimensional state space represented as $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $|0\rangle$ and $|1\rangle$ denote an orthonormal basis, and α, $\beta \in \mathbb{C}$ with $|\alpha|^2 + |\beta|^2 = 1$. The "superposition" of basis states allows an n-qubit system to represent a 2^n-dimensional complex Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$. In contrast, a classical n-bit register can store only one of these 2^n states.

Quantum Gates. A *quantum gate* operating on a n-qubit state defines a $2^n \times 2^n$ unitary matrix (i.e., U) that transforms the input state $|\phi\rangle$ to $|\psi\rangle = \mathtt{U}|\phi\rangle$. One-qubit gates manipulate individual qubits, with common gates like the Hadamard gate (i.e., `H` gate) for superpositions and the Pauli-`X`, `Y`, and `Z` gates for fundamental quantum flips and rotations. Multi-qubit gates operate on two or more qubits concurrently, enabling the creation of entangled states, i.e., a unique quantum phenomenon where qubit states become correlated in ways unattainable in classical systems. The CNOT (Controlled-NOT) gate, a fundamental two-qubit gate, establishes entanglement between a control qubit and a target qubit. Advanced multi-qubit gates, like the Toffoli gate, facilitate controlled operations across multiple qubits.

Quantum Algorithms. A *quantum algorithm* is represented as a circuit comprising a sequence of quantum gates executed on a suitably initialized set of qubits, as illustrated in Figure 1. In the context of quantum algorithms that take classical data as input, a classical-to-quantum encoder, denoted as $E(x)$, is employed to embed the classical data into a quantum state. Subsequently, the resultant quantum state undergoes processing within the following quantum circuit. The core quantum circuit can be constructed with gates of fixed configuration or variational gates, the parameters of which are subject to optimization for their configuration. This flexibility in gate configuration allows for adaptability and optimization of the quantum algorithm's performance.

Quantum Measurement. A quantum circuit is measured multiple times to estimate the expectation of a particular observable. Generalized *quantum measurement* is modeled through Positive-Operator Valued Measurement (POVM) [23]. A POVM with n outcomes on a d-dimensional Hilbert space is described by an n-element vector M of operators M_i such that $M = [M_1 \ldots M_n]^T$ ($M_i \geq 0$ and $\Sigma_{i=1}^{n} M_i = 1$). By default, qubits are typically measured in the z-basis for simplicity of implementation. For a quantum state denoted by $|\phi\rangle$, the probability of observing a result M_i is determined by the Born rule: $p_i = \langle\phi|M_i{}^{\dagger} M_i|\phi\rangle$. A proficient quantum algorithm yields a superposition state that provides a useful answer with high probability upon measurement. Following measurement, the quantum system's state collapses to the observed outcome M_i.

2) Physical Implementation

Similar to classical computing, a practical quantum computer utilizes a native gate set capable of universal quantum computations. This set usually includes several one-qubit gates and normally only one type of two-qubit gate. To execute a logical quantum circuit on a quantum computer, as illustrated in Figure 2, the user must compile the quantum circuit and its input data into a sequence of physical pulses [24], [25] using a designated configuration file. A pulse is defined by parameters such as an integer duration, a complex amplitude, and standard deviation. Notably, different quantum computers may support distinct pulse durations, maximum pulse amplitudes, and pulse channel numbers. The same quantum computer may also require varying values for pulse error calibration at different times. A configuration file [5], [8], providing up-to-date information on a quantum computer, allows the compiler to generate a high-quality pulse sequence.

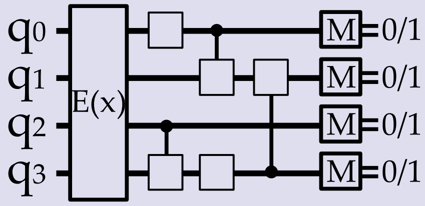


Figure 1. A quantum circuit.

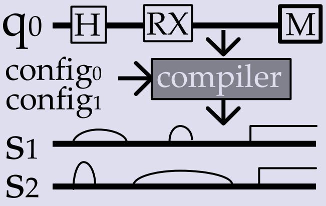


Figure 2. Quantum compilation.

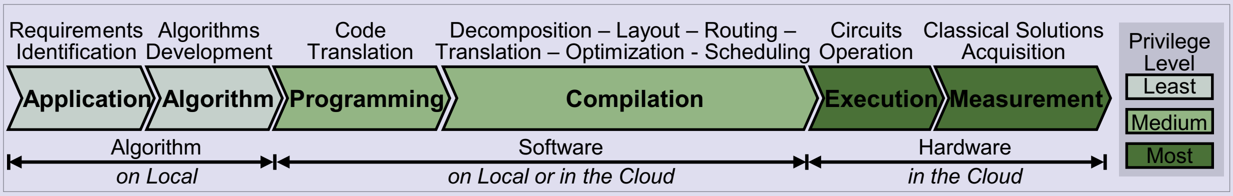


Figure 3. An overview of standard quantum computing workflow and ecosystem.

B. Emerging Quantum Computing Ecosystem

Figure 3 shows the standard workflow of a QC system. Similar to classical computing systems, a QC system can be vertically divided into abstract layers, including algorithm, software, and hardware. The algorithm layer identifies applications suitable for QC systems and designs corresponding quantum algorithms. The software layer encompasses the toolchain that facilitates the translation of QC algorithms into executable code, ultimately generating physical pulses sequences. The compiled circuit is then mapped onto the physical hardware layer, typically with limited connectivity. The circuit undergoes execution multiple times using the noisy devices and quantum gates, and measurements are performed according to a specific basis. The results are postprocessed to obtain classical outcomes.

Execution Model. Quantum computers currently operate and are anticipated to persist in their role as coprocessors [26], [27], with classical computers acting as controllers. This collaborative setup is essential for ensuring accurate measurements and executing fault-tolerant computations, particularly considering the inherent errors in NISQ quantum devices. In the "quantum co-processor" model, a classical microprocessor takes charge of a quantum computer, overseeing its execution. Unlike co-processor models in classical computing where operations can proceed without microprocessor intervention, the microprocessor sends instructions to the quantum co-processor at every cycle [26], [27]. Throughout each cycle of operation, the quantum unit remains precisely controlled by the microprocessor.

Quantum Cloud Computing. Due to the high cost of quantum computers, researchers and professionals often turn to cloud-based Quantum-as-a-Service (QaaS) for access to universal quantum devices. Users can design quantum algorithms locally or choose to download free or paid algorithms from specific QC algorithm providers. The management of quantum software toolchains is commonly facilitated by third parties. Users can select software tools through paid services or utilize open-source software, either locally or by sending their designed quantum algorithm to the cloud, depending on the available service. The generated quantum pulse sequences are then submitted to QaaS clouds as instructions. The server subsequently applies the physical pulse sequence to the qubits and communicates the measured results back to the user.

Privilege Levels in QC Systems. As color-coded in Figure 3, similar to classical computing, privilege levels in a quantum computing system denote varying access and control within its hierarchical architecture. Designed on the principle of least privilege, entities receive the minimum access needed for their functions. High-level QC algorithms at the top provide overarching guidance with limited direct control. As we descend through layers, there's a progressive increase in privilege, each layer building on the functionalities above. The QC hardware layer at the lowest level enjoys the highest privilege, exercising direct control over system resources. This privilege concept ensures a structured, secure environment by restricting access based on specific layer requirements, thereby preventing unauthorized access and minimizing security vulnerabilities.

C. QC Hardware

1) QC Hardware Development Resources

Table 1 summarizes NISQ hardware with various qubit technologies. Natural qubits (e.g., trapped ions, neutral atoms, photonics) scale well, while synthetic qubits (e.g., superconducting, quantum dots) benefit from easy fabrication using existing silicon integration. Each quantum technology has unique strengths and limitations, and there is no clear winner. Currently, superconducting and trapped-ion are two leading technologies, receiving significant support from academia and industry. They are also accessible as cloud services through companies like IBM [5], Google [6], Rigetti [7], IonQ [8], and Quantinuum [9].

2) Quantum Noises

Noises [36], [37], [38], [39], [40] are inherent in NISQ devices, varying in origin and characteristics across different quantum computing technologies. Notably, certain types of quantum noise exhibit time-dependent behavior, causing fluctuations in noise levels over time.

Table 1.
A summary on NISQ hardware development resources (SC: superconducting; TI: Trapped Ions; NA: Neutral Atoms; PT: Photonic; QD: Quantum Dots).

Tech.	Ref.	Key Features	Companies
SC	[24] [28]	+ Fabrications, Latency - Fidelity, Cryogenic	IBM, Google, Rigetti
TI	[25] [29]	+ Fidelity, Scalability - Latency, Controllability	IonQ, AQT, Quantinuum
NA	[30] [31]	+ Scalability, 3D - Fidelity, Controllability	Atom Computing, Infleqtion, QuEra
PT	[32] [33]	+ On-Chip Integration - Programability	Xanadu, PsiQuantum
QD	[34] [35]	+ Fabrications - Connectivity, Variations	Intel, HRL, SQC, Nanosys

In superconducting NISQ systems, for instance, variations in the population of unpaired electrons lead to notable temporal variations in the decoherence rate [41] throughout the day, resulting in random fluctuations in key parameters of a NISQ computer. From a logical standpoint, errors arising from NISQ noises can be systematically categorized into three main types: *readout*, *gate*, and *crosstalk* errors.

- Readout errors occur when there is a misidentification of the qubit value, such as observing $|1\rangle$ instead of the actual $|0\rangle$ state, and vice versa. Quantum information measurements can be effectively represented as population measurements within a specific basis [42], enabling a concise description through a probabilistic matrix model. Mitigation of readout errors is primarily achieved through post-hoc corrections [36], [37].
- Gate errors can be classified as *coherent* and *incoherent* errors. *Incoherent errors* are easier to handle than *coherent errors* because they can often be modeled as depolarizing noise [23]:

$$\mathrm{Err}(\rho) = (1-p)\rho + p\frac{I}{2^n} \tag{1}$$

  where Err(·) denotes the noise channel, ρ is the original density matrix, and n represents the number of qubits. The probabilistic error rate, denoted as p, depends on both the NISQ computer and the specific quantum circuit being executed. *Coherent errors* are typically miscalibrations in control parameters that produce similar or even drifting [43] errors during consecutive executions of a quantum circuit, introducing a systematic bias in the final output. Even though coherent errors are dominant gate errors [38], [39], they are often overlooked in noise-aware QC system designs due to their more intricate and hard-to-identify feature.
- Crosstalk errors encompass a diverse range of physical phenomena that become pronounced in largescale NISQ systems, typically consisting of more than 10 qubits. The coupling phenomena causing crosstalk among qubits vary significantly across diverse physical technologies. Examples include but are not limited to, crosstalk between superconducting qubits sharing a common readout resonator [44], crosstalk between trapped-ion qubits located in the same trap [45], and other instances yet to be explored. Interested readers seeking a comprehensive understanding of this topic are encouraged to refer to related works [46], [47], [48], [49]. A widely accepted metric [40] for crosstalk measurement highlights observable deviations between real quantum devices and their ideal behavior, formalizable and capturable in an architecture-independent manner. A recent study [50] has demonstrated that crosstalk results in a 20% increase in error for quantum gate phase flip and a 33% decrease in gate fidelity in a circuit employing only 9 `CNOT` gates on the `IBM_Melbourne` processor. Furthermore, this research has unveiled and emphasized that the displacement of two-qubit gates significantly contributes to the overall crosstalk errors in a quantum circuit.

3) Common Characteristics in NISQ Technologies

NISQ technologies, diverse in nature, typically operate by applying operations to individual qubits at designated locations using physical pulses. Moving qubits or their states within the system is a standard operation, and multi-qubit gates necessitate physical adjacency. State-of-the-art NISQ computers share common notable shortcomings, outlined as follows.

- NISQ computers feature a restricted universal native gate set, comprising only a few types of one-qubit gates and a single type of two-qubit gate. For instance, IBM quantum computers exclusively support two one-qubit gates (`U2`, `U3`) and one two-qubit gate (`CNOT`). Consequently, executing an n-qubit quantum circuit that relies on multi-input complex gates can only be approximated [16] using the limited gate set provided by the NISQ computer.
- Various measurable errors arise on NISQ devices due to imperfections in fabrication, qubit control, external interference, and measurement. The current error rate, approximately ~10^{-3} [51],

[52], significantly surpasses the 10^{-15} error rate required for many quantum algorithms [53], [54] to achieve computational advantages. While theoretically feasible, practical implementation of fault-tolerant quantum computers through quantum error-correction protocols [55], [56], [57] with millions of physical qubits may require decades of further research.

- A qubit on a NISQ computer experiences a short coherence time, allowing it to maintain its superposition state for a limited time, e.g., approximately ~100μs [52] in superconducting qubits. This imposes a stringent constraint on the duration and complexity of circuits running on the NISQ computer, necessitating their completion within this coherence window to prevent any loss of information.

D. QC Software

1) QC Software Development Resources

The evolution of quantum computing software, though slightly delayed compared to hardware and algorithms, has witnessed rapid progress. It encompasses the design of quantum programming languages and compilers, as outlined in Table 2. Major industry leaders contribute to a vibrant developer community by providing free software development kits. In the realm of quantum computing software solutions, these industry players offer tailored services, specializing in the integration of quantum solutions with High-Performance Computing (HPC) for specific business use cases. Additionally, they assist clients in identifying suitable applications and developing customized solutions.

2) Quantum Programming Language

Quantum programming languages transform mathematical descriptions of a QC algorithm into executable implementations, serving as the initial input for the QC software toolchain. Operating across various abstraction levels, options like QASM [58] enable direct specification of operations on a QC device, resembling assembly language in classical computing. However, challenges include reduced readability and increased complexity compared to higherlevel languages. Programmers require a profound understanding of QC hardware architecture, and the code is often device-specific, limiting portability across platforms.

Similar to classical programming languages, quantum languages broadly fall into two categories: functional and imperative. Functional languages prohibit direct variable modification, emphasizing an abstract, mathematical approach. In contrast, imperative languages permit direct manipulation of variables and require explicit instructions for computations, potentially leading to the development of more efficient programs. However, this approach may introduce greater complexity for programmers. Examples of imperative quantum languages include Scaffold [59] and Q# [63], while Microsoft LIQUi|⟩ [61] represents a purely functional quantum language. A recent trend in programming languages, as seen in Quil [62] and Silq [64], emphasizes functional programming with added flexibility to support imperative programming when needed.

3) Quantum Compiler

Quantum compilers convert abstract quantum computing algorithms into executable instructions, often represented as physical pulses, processed step by step on quantum hardware. This comprises a five-step process: logical-level decomposition, layout, routing, logical-to-physical translation, and physical-level optimization and scheduling. Iterative refinement in these steps yields a synthesized quantum circuit meeting specified criteria like qubit count, gate counts, circuit depth, and other relevant parameters. Here, we briefly introduce the key objective for each step.

- **Decomposition.** This initial stage involves logical passes before embedding the circuit into the backend. It typically includes unrolling custom

Table 2.
A summary on QC software development resources.

	Ref.	Key Features
Programming Language	QASM [58]	Assembly, Gate-Level
	Scaffold [59]	C, LLVM [60] Infrastructure
	LIQUi|⟩ [61]	Functional
	Quil [62]	Python, Imperative & Functional
	Q# [63]	C#, Imperative
	Silq [64]	High-Level, Imperative & Functional
Open-Source Compiler	Qiskit [11]	IBM's Quantum Software Platform
	Cirq [12]	Google's Quantum Software Platform
	pyQuil [13]	Rigetti's Quil-based Python Library
	t|ket⟩ [14]	Quantinuum's NISQ Development Kit
	Braket [15]	Amazon's Quantum Service Platform
	BQSKit [16]	LBNL's Compiler Framework
	PennyLane [17]	PennyLane' QML Framework

instructions and converting the circuit to a logical universal gate set, comprising one-qubit and a few two-qubit gates. For instance, the Clifford group (i.e., `X, Z, P, H, CX`), combined with any non-Clifford gates (e.g., `T` or `CCX`), enables universal quantum computations. An illustrative work in this domain is detailed in [65], presenting a framework for compiling and optimizing Clifford+T quantum circuits to minimize the T count. The article introduces and compares efficient quantum compilers for multi-qubit Clifford+T circuits.

- **Layout & Routing.** These two stages facilitate the translation of virtual qubits in a logical algorithmic quantum circuit to the physical qubits on a backend. This is achieved by applying a suitable layout and inserting gates, such as SWAP, into the original circuit to ensure compatibility with the connectivity of the target QC hardware. The primary objective is to minimize additional operations and insertions, preserving the fidelity of the QC algorithm.
- **Translation.** This step manages gate-set transpilation, approximating any unitary transformation describing the quantum computation as a sequence of gates chosen from the native gate set of the target backend.
- **Optimization & Scheduling.** These two stages represent the primary and final optimization loop before deployment on hardware. Error rates of the quantum hardware are normally taken into account in these processes where the compiler elects physical qubits and their movements, which minimize the circuit error rates. The output of the quantum circuit can be obtained through several runs on NISQ computers.

The compilation process in quantum computing presently bears strong resemblances to classical compilation techniques. It can be categorized into static compilation, where executable instructions are generated before runtime, and dynamic compilation, where instructions are generated during runtime. Since mid-circuit measurements are still in the early stages of development for many quantum computers [9], [66], [67], most compilers [11], [12] [13], [14], 15], [16], [17] listed in Table 2 operate under the assumption that data input and control are statically known and can thus be managed through static compilation. Furthermore, despite the ongoing challenges in practically implementing quantum error correction firmware for NISQ devices, certain quantum computing software frameworks, like Mitiq [68], are integrating cost-effective error mitigation techniques into their platforms.

4) Challenges in NISQ QC Software

Quantum computing software serves as an intermediary, connecting low-level physical quantum hardware with high-level mathematical quantum algorithms. Below we summarize the challenges in NISQ QC software designs.

- Driven by strict hardware resource constraints, QC software prioritizes efficiency over abstraction and modularity [27]. This contrasts with classical computing system software, necessitating the low-level QC hardware layer to reveal more physical details to quantum toolflows for improved QC circuit compilation. This heightened access and control over system resources in QC software pose potential security threats if exploited by adversaries due to increased privilege and direct access to system resources.
- The noise and variations in low-level quantum hardware contribute to uncertainty and notable inaccuracies in the information exchange between quantum hardware and software layers. Furthermore, the intricate nature of quantum systems, coupled with the inherent noise in NISQ devices, adds complexity to debugging in quantum computing software.
- The dynamic evolution of algorithms and hardware in the quantum computing landscape mandates adaptive and complementary approaches in software design. The software must flexibly adjust to these changes, ensuring optimal performance and compatibility in the ever-changing quantum environment.

E. QC Algorithms

1) QC Algorithm Development Resources

In Table 3, we present a summary detailing the evolution of quantum algorithms and their practical applications. Despite the strong theoretical foundations underpinning several seminal quantum algorithms [1], [2], [3], [4], [53], [54], [69], [70], [71], [72], [73], [74], their implementation requires an error rate of approximately 10^{-15} to effectively showcase their computational superiority over classical algorithms. This stringent error threshold significantly exceeds the current capabilities of NISQ computers. While the realization of fault-tolerant quantum computers is theoretically viable, incorporating quantum error-correction protocols [55], [56], [57] presents a formidable practical challenge. Achieving this goal may necessitate sustained research efforts spanning potentially decades.

NISQ algorithms [84] exploit error-prone qubits and imperfect quantum gates to address classically

Table 3.
A summary on QC algorithm development.

	Ref.	Applications
Theoretical Algorithms	[1], [2]	Quantum Turing Machine
	[3], [4]	Universal Quantum Simulator
	[53], [69]	Prime Factorization
	[54], [70]	Database Search
	[71], [72]	Cryptography
	[73], [74]	Linear Equation Solver
NISQ Algorithms	[18], [75]	Quantum Supervised Learning
	[76], [77]	Quantum Unsupervised Learning
	[78], [79]	Quantum Reinforcement Learning
	[19], [80]	Variational Quantum Eigensolver
	[20], [81]	Variational Quantum Simulator
	[21], [82]	Combinatorial Optimization
	[22], [83]	Numerical Solver

challenging problems, garnering significant interest across diverse disciplines. This spans applications in Quantum Machine Learning (QML) [18], [75], [76], [77], [78], [79], variational quantum eigensolvers [19], [80], quantum simulators [20], [81], combinatorial optimization [21], [82], and numerical solvers [22], [83]. The prevailing strategy in crafting NISQ algorithms revolves around a hybrid quantum-classical framework. This method delegates the computationally intensive or intricate portion of a task to the quantum computer, with the remainder managed by a robust classical computing system. These algorithms iteratively refine the parameters of a parameterized quantum circuit, officially recognized as Variational Quantum Algorithms (VQAs) [85]. This framework enables the leverage of quantum advantages within the existing constraints of quantum hardware, offering a practical avenue for addressing complex problems across diverse domains.

2) Building Blocks of VQAs

While the building blocks of various VQAs share similarities to varying extents, this survey specifically employs quantum machine learning as a representative use case to elucidate the fundamental building block and optimization process in VQAs.

The algorithm typically starts with a data encoding module, specifically a classical-to-quantum encoder denoted as $\mathbf{E}(\mathbf{x})$, when the inputs are classical data. The primary function of this encoder is pivotal because it transforms a classical input vector $\mathbf{x}$ into a quantum state represented by an N-qubit quantum state, denoted as $|\mathrm{x}\rangle$. The formulation of the encoder model can be expressed as follows:

$$\mathcal{E}: \mathbf{x} \to |\mathbf{x}\rangle = \mathbf{E}(\mathbf{x})|0\rangle^{\otimes N} \tag{2}$$

When the input involves quantum data, a dedicated quantum circuit module is utilized to prepare the input quantum state instead. Subsequently, the generated $|\mathbf{x}\rangle$ state undergoes manipulation by a parametrized quantum circuit (PQC) denoted as $\mathbf{U}(\theta)$:

$$\mathcal{U}: |\mathbf{x}\rangle \to |\mathbf{y}(\theta)\rangle = \mathbf{U}(\theta)|\mathbf{x}\rangle \tag{3}$$

where $U(\theta)$ is realized through either a singular layer or a composition of multiple-layered circuit ansatz, with θ representing a set of parameters subject to optimization. The final output results are obtained through quantum state measurement, denoted as $\mathbf{M}$, which maps the output quantum state $|\mathbf{y}(\theta)\rangle$ to a classical vector $\mathbf{y}(\theta)$:

$$\mathcal{M}: |\mathbf{y}(\theta)\rangle \to \mathbf{y}(\theta) = \langle \mathbf{y}(\theta)|\mathbf{M}^{\dagger}\mathbf{M}|\mathbf{y}(\theta)\rangle \tag{4}$$

By convention, qubits are measured in the z-basis for ease of implementation by default. From a global perspective, the comprehensive formulation of a VQA can be represented as:

$$\mathcal{Q}: \mathbf{Q} = \mathbf{M} \circ \mathbf{U}(\theta) \circ \mathbf{E}(\mathbf{x}) \tag{5}$$

A VQA is subjected to evaluation through a predefined objective function, denoted as $\mathbf{L}(\cdot)$, and undergoes iterative optimization to attain optimal parameters. The minimization of this objective is facilitated through hybrid quantum-classical gradient descent [86]. The optimization process can be succinctly formulated as follows:

$$\mathcal{L}: \mathbf{y}(\theta) \to \text{ Loss } = \mathbf{L}(\mathbf{y}(\theta)) \tag{6}$$

$$\text{Update rule}: \theta_j^{t+1} = \theta_j^t - \eta \frac{\partial \mathbf{L}(\mathbf{y}(\theta))}{\partial \theta_j} \tag{7}$$

3) Datasets for VQA Benchmarking

While VQAs are still in early development, studies have unveiled insights into their enhanced generalization capabilities and the transferability of variational quantum machine learning models [87], [88], [89], [90]. The effectiveness and adaptability of VQAs depend crucially on the quality and diversity of datasets for benchmarking. In the initial stages of VQA development, especially in QML models, prevalent small-scale classical datasets like [91] and [92] often required down-sampling or preprocessing to accommodate the limited qubit capacity of real quantum devices.

Recent initiatives introduce datasets explicitly tailored for VQA benchmarking across various quantum computing tasks. For example, [93] explores the impact of data in QML models, showcasing quantum advantages through classical data engineering approaches, albeit with results that may not be easily interpretable by humans. Recognizing the need for dedicated quantum datasets in VQA benchmarking, there is a growing interest in utilizing quantum datasets composed of quantum states. Notable examples include the NTangled dataset [94], featuring quantum states with diverse multipartite entanglement, the QDataSet [95] comprising 52 high-quality datasets derived from simulations of one- and two-qubit systems, and the Alchemy dataset [96], a quantum chemistry dataset with 12 properties of 119,487 organic molecules. Additionally, the tmQM dataset [97] contains geometries and properties of a large transition metal-organic compound space.

4) Challenges in NISQ Algorithms

NISQ variational algorithms share several common challenges:

- NISQ variational algorithms employ parameterized quantum circuits as a fundamental component with applications across diverse domains. However, the construction of these PQCs is typically empirically designed, relying on prior knowledge or random assignment. Consequently, the explainability and expressivity of these circuits in specific applications are uncertain. The design of PQCs lacks a principled approach.
- NISQ computers encounter challenges such as a restricted number of qubits, short coherence times, unavoidable noise, and imprecise control. These limitations significantly constrain the width and depth of VQA circuits, limiting their applicability to practicalscale problems. Investigating strategies to realize quantum advantages in addressing real-world problems using NISQ devices is an area of considerable research significance.
- The intricate optimization landscape inherent in VQAs underscores the imperative for the development of more efficient and quantum-specific optimizers. At present, VQA optimizers predominantly rely on classical methods without integrating quantum-specific insights and adaptations. This reliance on classical approaches causes notable variations in the performance of VQAs.

III. Quantum Security Taxonomy

In this section, we present an initial quantum security taxonomy employed for categorizing the existing works in this survey. Firstly, our deliberate alignment with the classical taxonomy serves to facilitate knowledge transfer from the domain of security in classical computing. It is crucial to emphasize that, while the conceptual foundations exhibit similarities with classical security paradigms, the vulnerabilities, specified attack vectors, and potential defenses are intricately linked to the distinctive nature of quantum computing, as detailed in the subsequent sections. Secondly, in recognition of the dynamic landscape of quantum computing, our taxonomy deliberately prioritizes the use of stable components within the QC system as metrics for categorization, ensuring durability and minimizing changes over time. Additionally, our objective is to maintain the taxonomy's breadth for broad applicability and flexibility, allowing it to accommodate ongoing expansion and adaptation as our understanding and research in quantum computing and quantum security progress.

To this end, we classify existing works using a two-dimensional metric. One dimension involves the victim logical layers of the QC system, specifically identifying where attack vectors are present, namely QC hardware, QC software, and QC algorithms. On the other dimension, we cover the objective of security attacks, incorporating the following taxonomy:

- **Information Leak.** In an information leak scenario, the objective of an adversary or unauthorized user is to acquire sensitive or confidential information. The adversary exploits system vulnerabilities, whether intentionally or unintentionally. The extent of their prior knowledge about QC systems may vary. For instance, [98], [99] describes a scenario where a completely untrusted QC software vendor, lacking prior knowledge of the QC circuits, resorts to direct theft. In contrast, [44], [100] assume the adversary has profiled the victim QC hardware to construct a statistical correlation model to facilitate future information inference.
- **Untargeted Attack.** Untargeted attacks refer to situations where an adversary opportunistically exploits vulnerabilities, leading to the corruption or degradation of functionality in a quantum computing system. These attacks occur without a specific target or a predetermined way in which the attacker intends to manipulate the system. We prefer "untargeted attack" over "fault injection" (used in [101]) because "fault injection" in classical computing [102] typically denotes a physical attack on hardware, excluding the entire computing system stack. To prevent confusion and provide a comprehensive term for incorporating

quantum security research works beyond the physical layer of quantum computers, as demonstrated in [103], [104], [105], we use the term "untargeted attack."

- **Targeted Attack.** Targeted attacks denote sophisticated and focused attacks orchestrated by an adversary with a specific goal, necessitating a higher level of knowledge about the victim system compared to an untargeted attack. In this context, the attack vectors are meticulously crafted and customized based on the adversary's prior knowledge, tailored to achieve a particular objective. For instance, the QDoor attack [104] assumed that the adversary has access to the training dataset of a quantum machine learning model. With this access, the adversary can introduce a customized malicious loss item during the model's training process, allowing precise control over the targeted class they aim to degrade in terms of model accuracy.

As pointed out in [101], vulnerabilities in the victim layer can trigger malicious changes manifesting in one or multiple different layers within the QC system. For instance, [103] illustrates that modifications to QC software (e.g., compiler configuration files) can introduce backdoor gates added in the QC hardware. In contrast to the taxonomy methodology in [101], our focus is on the identification and categorization of vulnerabilities within QC systems rather than examining the observable effects or consequences. We believe that providing insights to industry professionals and researchers in identifying security vulnerabilities within this emerging computing paradigm is particularly crucial, especially at the current early research stage.

IV. Security Threats in NISQ Quantum Computing Systems

In this section, we present a comprehensive survey of existing works in quantum security, systematically organized using our established two-dimensional taxonomy. More specifically, the related works are categorized based on the victim layers within the QC system: QC Hardware (Section 4.1), QC Software (Section 4.2), and QC Algorithms (Section 4.3). Within each victim layer, we identify acknowledged vulnerabilities and elaborate on the employed attack vectors from prior works. Following this, we conduct a detailed analysis of works categorized by their objectives, specifically addressing information leaks, untargeted attacks, and targeted attacks. In addition, we explore potential defense strategies as detailed in Section 4.4.

A. Security Threats in QC Hardware

Vulnerabilities in QC Hardware. As shown in Table 4, existing works emphasize two main attack vectors in the QC hardware layer: quantum crosstalk noises and the exploration of vulnerabilities through side channels like qubit reset, power trace, or laser damage. To enhance comprehension of these works, recent advancements in the foundational principles of quantum systems engineering related to crosstalk noises and side channels are summarized below:

- **Crosstalk Characterization.** Early investigations into quantum crosstalk [46], [47], [48], [49] focused on examining the noticeable discrepancies between the expected behavior of ideal quantum gates and circuits and the actual behavior in large-qubit systems caused by crosstalk. However, to consider crosstalk as a potential quantum security attack vector, a more precise understanding of the mechanisms and causal relationships associated with crosstalk between qubits is essential. Significant contributions in this domain include [111] and [112], both dedicated to measuring and analyzing crosstalk effects in superconducting quantum computers. Abrams et al. [111] introduced three measurements to quantify DC (i.e., configuring a qubit to a specific frequency) and AC flux (i.e., operating two-qubit gates) crosstalk between tunable transmons. The research in [112] categorized crosstalk effects into quantum crosstalk, as investigated in [111], and classical crosstalk, addressing electromagnetic interference between microwave lines and on-chip electromagnetic fields. The article presented corresponding approaches for characterizing these crosstalks. With these crosstalk characterizations and principled measurements, an attacker can intentionally manipulate crosstalk for malicious purposes, and hardware designers can also devise corresponding defense techniques.
- **Side Channels.** Side-channel attacks exploit unintended information leaked from the physical implementation of a QC system, rather than directly targeting its intended functionality. Current

Table 4.
A summary of QC hardware vulnerability research.

	Attack Vector	
	Noises	Side Channels
Info Leak	[44], [106]	[44], [100]
Untageted Attack	[107], [108]	[109], [110]
Tageted Attack	[44]	[106]

works exploiting quantum side channels for security attacks typically involve profiling quantum systems on specific types of side channels and constructing statistical models. Information is then retrieved by leveraging these models, as demonstrated in profiling nonuniform qubit reset latency in [44], power consumption variation in [100], and laser damage in [109], [110]. However, a notable drawback is the absence of systematic research from the quantum physics perspective, focusing on quantum side channel characterization or analysis. This gap raises concerns about the effectiveness and comprehensiveness of the constructed statistical models. Further exploration and understanding of quantum side channels are essential for developing robust security measures in a quantum computing system.

1) Information Leak

Crosstalk Induced by Reset. The work in [44] explored crosstalk induced by reset operations in IBM superconducting H7 quantum computers. In this device generation, a reset operation involves a measurement followed by a conditional X gate, transitioning the post-measurement state from $|1\rangle$ to $|0\rangle$ on outcome 1. Profiling the IBM quantum computer using this reset model, the authors observed qubit coupling during reset, resulting in crosstalk between qubits sharing the same readout resonator. Specifically, if a victim qubit q_0 and an attacker qubit q_1 share the same readout resonator, reset operations on q_0 induce a crosstalk-like effect on q_1, allowing the inference of information about q_0 through measurements on q_1. Even when the victim qubit q_0 undergoes multiple reset operations, an adversary measuring the q_1 can still deduce the prior state of q_0 based on the resulting measurements.

Crosstalk Induced by Measurement. The work in [106] explores qubit crosstalk, evident in observable readout error dependencies. The study adopts a multi-programming approach, running two programs concurrently on the same quantum computer. Profiling on IBM quantum computers uncovers a correlation between readout errors on qubits in an attack program and those in a victim program. Consequently, the researchers construct a statistical model capturing this readout error-dependent signature for a specific quantum computer. They leverage this model to infer results from the victim program via measurements on the attack program, achieving an impressive 96% accuracy in identifying victim output on three public IBM computers. However, the experimental configurations, such as the criteria for selecting victim and attack qubits, lack clarity.

Side-Channel via Reset. The research discussed in [44] explores reset operations in IBM superconducting H7 quantum computers. It uncovers a correlation between the measured probability of outputting '1' on the attack qubit q_1 and the measurement time on the victim qubit q_0, which shares the same readout resonator with q_1. This previously overlooked yet crucial quantifiable indicator closely related to qubit reset gains significance, especially in the context of side-channel attacks. This insight allows an attacker to measure q_1, record its probability of outputting '1', and consequently infer both the duration between the start of the victim program and its last measurement and the duration between the victim's last measurement and the end of its allocated share.

Side-Channel via Power Traces. The study in [100] explores the vulnerability of electronic systems, where the power consumption trace generated by a quantum computer becomes a potential target for attackers seeking detailed information about a victim program. This susceptibility stems from the distinctive processing pathways of various pulses corresponding to different gates in a quantum computer circuit. In their study, the researchers carried out preliminary experiments, capturing power traces for various quantum pulses and constructing a comprehensive model. This model forms the foundation for illustrating power-centric side-channel attacks on quantum computers, showcasing the capability to extract information about the control pulses dispatched to these machines. Analyzing these control pulses empowers adversaries to reverseengineer the equivalent gate-level circuit description, potentially disclosing the confidential algorithms in operation. The study introduces five attack strategies and assesses their effectiveness, leveraging control pulse information acquired from cloud-hosted quantum computers.

2) Untargeted Attack

Fault Injection via Crosstalk. Both [107] and [108] investigate crosstalk-induced, untargeted fault injection in a multi-programming scenario, wherein victim and attacker programs operate simultaneously on a shared quantum computer. The study in [107] centers on large trapped-ion quantum computers with multiple interconnected traps. When implementing multi-programming on these devices, qubits from both victim and attacker programs may share the same trap and become coupled. The research illustrates that repeated shuttle operations, employed to communicate an attacker qubit to another trap, can result in fidelity degradation in the coupled victim qubit. Consequently, this process has the potential to inject faults into the victim program. In

[108], crosstalk error rates are derived from IBM quantum computers through repeated program executions. The researchers also developed a simulator to quantify crosstalk effects across various tasks. Using this model, their experiments demonstrated that intentional repetition of CNOT operations on specific qubits - coupled with those in the victim program - facilitates the injection of faults into the victim program.

Fault Injection via Side Channels. Both studies presented in [109] and [110] center on quantum key distribution using photonic quantum computers, employing a laser followed by an optical attenuator for data communication. The authors highlight the vulnerability of these quantum computers to high-power laser side-channel attacks, wherein injecting light from the communication line into the laser significantly compromises the fidelity of the prepared quantum states. Recent research [113] explores qubit performance in the presence of environmental factors like radioactive materials and cosmic rays. As our understanding advances, we anticipate an increasing number of studies identifying vulnerabilities in quantum computing hardware, whether through laser, radiation, or other means, as potential sidechannel attacks or beyond.

3) Targeted Attack

While both [44] and [106] focused on information leaks, their findings have the potential for targeted attacks on QC systems. In the case of [44], rather than relying on measured results from attack qubits (q_1) to infer information about the victim qubit (q_0), an adversary could intentionally perform reset operations on q_1. The resulting crosstalk via a shared resonator, as per the static model established, would induce a predictable state change on q_0. Similarly, the demonstrated readout error dependency between a victim qubit and an attacker qubit in [106] can be strategically leveraged. Intentional operations on an attacker qubit would manifest correlated readout error rates on the victim qubit.

B. Security Threats in QC Software

Vulnerabilities in QC Software. As outlined in [27], the design of quantum computing software draws parallels with classical computing toolchains and hardware synthesis. This alignment emphasizes the inherent complexity in developing and using QC software tools, requiring specialized domain expertise and creating a significant entry barrier for average users in the quantum computing domain.

In the following section, we explore state-of-the-art QC software designs and highlight potential security vulnerabilities that attackers could exploit. It is crucial to recognize that toolflow designs are continually evolving, keeping pace with the rapid advancements in quantum technology and algorithms. Despite this dynamic landscape, these designs reveal common techniques and strategies. Subsequently, we delve into a detailed discussion of recent works exploiting attack vectors in the QC software layer, encompassing both the toolflow software itself and its configuration, as summarized in Table 5.

Table 5.
A summary of QC software vulnerability research (red highlights related works with potential quantum applications).

	Attack Vector	
	Toolflow	Configuration
Info Leak	[98], [99]	NA
Untageted Attack	[114], [115]	[103]
Tageted Attack	[98], [99]	[103], [116]

- **Approximation in the QC Compilation.** In the QC compilation, crucial steps often involve approximations rather than precise translation. For instance, during circuit logic decomposition, the original unitary matrix is represented through matrix and tensor products of specific elementary gates. The errors introduced by these approximations are considered acceptable if they fall within a predefined threshold. Similarly, in the translation step that replaces logical gates with physical gates, approximation is also allowed. Deliberately leveraging these approximations can become an attack vector, as discussed in [103] and [117].
- **Gate Insertion due to Limited Connectivity.** Limited connectivity poses a significant challenge in the layout and routing of QC compilation, transforming it into a complex, constrained, multi-objective optimization problem. Despite several research efforts [114], [115] aimed at addressing this issue through improved optimization algorithms or the development of fully-connected quantum hardware, the insertion of SWAP gates remains inevitable. The gates insertion step is beyond the control of average users and could potentially be exploited for untargeted attacks to degrade the fidelity of a victim QC circuit.
- **Quantum Noises.** Each quantum server typically monitors and records its temporal and spatially varying quantum noises. A configuration file is commonly used to describe the latest information of a NISQ server, enabling the compiler to generate a high-quality pulse sequence for a specific quantum circuit and its input data. When the

same quantum circuit receives new input data, the compiler must recompile the circuit with the updated input. However, an adversary with domain knowledge can stealthily modify these configuration files, thereby altering the functionality and/or behavior of the quantum server, as demonstrated in [103] and [116].

1) Information Leak

Information Leak via QC Toolflow. A quantum circuit can contain sensitive information, spanning from crucial financial data to proprietary algorithms. Consequently, relying on unverified compilers for quantum circuits creates a vulnerability that potential adversaries could exploit to steal intellectual property (IP). However, currently, there are no covert strategies for exploiting vulnerabilities in the quantum software layer to facilitate information leaks. Existing studies [98], [99] assume a wholly untrusted third-party software vendor that indiscriminately appropriates quantum IPs once users transmit their quantum circuits through a software-as-a-service platform.

Relying solely on the configuration for information leakage is not a feasible strategy. No previous work has explored this, and we do not consider it a viable approach.

2) Untargeted Attack

Potential Attacks via QC Toolflow. Although the main emphasis of layout and routing schemes in [114] and [115] is not on quantum computing security, they unveil a crucial aspect of QC compilation–the introduction of additional operations, which inevitably reduces algorithm fidelity. These insights highlight a potential vulnerability, as an untrusted third party or adversary could intentionally manipulate the compilation process by introducing more operations in the synthesized circuit to corrupt the quantum circuit output. Despite the feasibility of these potential attack approaches, enhancing the stealthiness of such attacks remains a research challenge.

Attacks via Configuration Modification. QTrojan [103] introduced an untargeted attack method that uses specific lines in a server-specific configuration file as triggers to manipulate the compilation process. This manipulation leads to the synthesized circuit disabling the actual data encoding, achieved by incorporating an additional circuit layer for pre- and post-encoding. As a result, the victim QC circuit malfunctions. QTrojan is a stealthy attack approach as the trigger can be disguised as pulse error calibration for the data encoding layer, and the resulting implementation on QC hardware introduces no additional circuit depth. The only noticeable difference lies in the pulse amplitudes, making the attack harder to detect.

Table 6.
A summary of QC algorithm vulnerability research.

	Attack Vector	
	VQA Blocks	Configuration
Info Leak	-	[118],[119],[120],[121]
Untageted Attack	[105],[122]	[104]
Tageted Attack	[123],[124]	[104],[125]

3) Targeted Attack

Attacks via QC Toolflow. For the study presented in [98] and [99], the entirely untrusted third-party software vendor has the capability to execute targeted attacks, but this threat model is not practical, lacking any form of stealth.

Attacks via Configuration Modification. The QTrojan approach [103] can also be utilized for targeted attacks. The authors demonstrated the feasibility of encoding a set of predefined data into the circuit during actual computation, effectively replacing the original data. This introduces an additional layer of quantum gates alongside the pre- and post-encoding layers used for untargeted fault injection. While technically achievable, we argue that this added layer may introduce unstealthy and noticeable discrepancies. Acharya and Saeed [116] employs a similar attack model to QTrojan by manipulating the compilation configuration file but with a focus on altering quantum error rate parameters in the compile-time calibration file. The design approach is rooted in the observed dependency between the output of a quantum circuit and the error parameters of the quantum hardware. However, the sophisticated nature of this dependency, combined with the compilation process, makes it challenging to precisely predict how the error rates on a device will impact the output for a specific algorithm. Although this research highlights the vulnerability, we believe it would be more realistic and relevant to consider applying such attacks in an untargeted scenario.

C. Security Threats in QC Algorithm

Vulnerabilities in QC Algorithm. The development of quantum computing algorithms, especially NISQ variational algorithms, is progressing rapidly. As depicted in Table 6, we broadly categorize the utilized attack vectors into VQA building blocks and other factors termed configuration. In the following, we provide an overview of the latest research on NISQ algorithms, highlighting their vulnerabilities and potential threat vectors. Then, a detailed discussion of each work listed in Table 6 is presented.

- **Classical-to-Quantum Encoder.** In VQA methods where classical inputs are employed, the classical-to-quantum encoder stands out as a crucial module, consistently argued to hold greater significance than subsequent trainable ansatz blocks in an expanding body of literature [126], [127], [128]. This module has been identified as the sole element within a VQA capable of introducing nonlinearity into the model [129]. A study by [127] investigates these encoders as partial Fourier series and underscores that baseline encoding, using a single Pauli-rotation encoding, can only effectively learn a sine function. Building on this insight, subsequent works [128], [129] advocate the use of repeated Pauli encoding layers to expand the frequency spectrum and, consequently, enhance the expressivity of quantum models. While this trend has demonstrated improved performance in VQAs, it concurrently complicates the interpretability and explainability of these models. Given the significance of the classical-to-quantum encoder in a QC algorithm, most works leverage the encoder as the main attack vector within all the VQA building blocks.

- **VQA Optimization.** While VQA theory is still in its early stages of development, several studies have contributed valuable insights into the enhanced generalization capabilities [87], [88], [89] and transferability of VQAs [90]. Notably, these advancements are observed when VQA models are trained with minimal data or, in an intriguing context, no training data at all from the target domain, which is referred to as a zero-shot setting. The training optimization process encompasses several critical components, including data preprocessing [93], [130], parameter configuration [131], [132], and initialization [90], [133]. Additionally, decisions such as choosing the training optimizer [134], [135] and addressing the challenge of the gradient barren plateau [135], [136] play pivotal roles in shaping the overall effectiveness of the training process. All these aforementioned factors related to VQA optimization have vulnerabilities that could be exploited by an adversary with domain knowledge, serving as potential attack vectors on the QC system.
- **Quantum Noise in VQAs.** While modeling quantum noise for specific quantum algorithms is inherently complex, it consistently proves to be a significant obstacle to the performance and robustness of QC algorithms on real NISQ hardware. Three noteworthy challenges emerge: (1) The inherent misalignment between ideal simulations and real-device experiments introduces variations in results [137], directly affecting the fidelity of quantum computations. (2) Quantum noise not only interferes with the trainability of algorithms but also leads to noise-induced barren plateaus [136]. As the circuit depth increases, crucial features of the landscape experience exponential suppression due to noise, presenting a substantial obstacle to the reliability of quantum computations. (3) Intrinsically, noise can serve as potential attack vectors impacting model robustness and fairness without noticeably affecting model performance, as theoretically demonstrated in quantum adversarial machine learning [123], quantum differential privacy [138], and fairness verification in quantum machine learning [139]. Experimental validation on NISQ devices has been conducted and reported in [125].

1) Information Leak

Quantum algorithms traditionally undergo empirical local design for fixed inputs. The advent of variational quantum algorithms featuring parameterized circuits has spurred a paradigm shift, enabling the widespread use of quantum algorithms in novel settings. Specifically, a general VQA is designed and initially optimized by a third party, then downloaded by a user who fine-tunes the VQA with sensitive, specific local data. Achieving precision in VQC design necessitates a profound understanding of the domain, with the training process demanding costly data acquisition. Consequently, VQCs represent invaluable intellectual assets deserving robust protection.

Attack via Untrusted Servers. In this evolving quantum computing model, multiple parties participate in computations, introducing new security threats. Existing works commonly assume a semi-honest server, aiming to either pilfer the VQA-optimized parameters [118] or acquire other parties' private training data [119], [120], [121]. While these studies draw upon security theories from classical machine learning, a significant gap exists in comprehending how leaked information is efficiently employed to create meaningful inferences.

2) Untargeted Attack

Attack via VQA Blocks. The vulnerability of VQAs has been extensively explored, particularly in the context of quantum machine learning as a primary use case. In terms of data input, [122] demonstrated that even minimal alterations to input data can lead trained quantum classifiers to make incorrect classifications. Notably, as the dimensionality increases, the required alteration decreases, rendering high-dimensional quantum

To defend against quantum security attacks, we outline several defense techniques, some tailored for quantum computing, others inspired by classical security research.

classifiers susceptible to misclassification with small perturbations. Focusing on the final measurement and readout phase, [105] presented an attack model wherein less-trusted vendors might manipulate the results or parameters of quantum circuits, resulting in suboptimal solutions or increased costs. They conducted modeling and simulation exercises to illustrate adversarial tampering of input parameters and measurement outcomes, using the Quantum Approximate Optimization Algorithm (QAOA) as an exemplary hybrid quantum-classical algorithm.

Attack via Objective Functions. In the VQA design process, QDoor [104] demonstrated that the formulation of the loss function for optimization can serve as an attack vector. This involves framing the training of a quantum machine learning model as a type of multi-task learning. While minimizing inference errors, QDoor introduces a malicious loss item that is manipulated in the subsequent compilation process, indiscriminately amplifying the inference error of synthesized circuits on a NISQ computer.

3) Targeted Attack

Attack via Input Data. [123] and [124] uncovered the susceptibility of a quantum classifier to deception through adversarial examples, achieved by adding imperceptible perturbations to the original legitimate samples. They not only emphasized the notable vulnerability but also demonstrated the viability of targeted attacks. However, the complexity introduced in data encoding presents a challenge to maintaining the effectiveness of targeted attacks. It is noteworthy that if the input is quantum data, malicious manipulation of the input could remain a highly effective strategy.

Attack via Objective Functions. The QDoor approach [104] can be extended to a targeted attack setting, with or without a specified trigger, granted the adversary has access to the training dataset. The fundamental concept involves introducing a malicious loss item into the loss function during the training process, and the attack is triggered in the compilation process. The key distinction lies in the computation of the malicious loss item, which exclusively occurs on inputs within a target class or a more finely defined set of targeted inputs using a trigger. This work introduces a novel and general approach that can be expanded to incorporate additional attack vectors as our understanding evolves. While QDoor concentrates on the compilation process as its handler, an adversary could broaden this concept by considering decomposition, layout, routing, or mapping as alternative handlers, given the appropriate formulation of a malicious loss item.

D. Potential Defense Techniques

To defend against quantum security attacks, we outline several potential defense techniques. Some of these methods are specifically tailored for quantum computing, while others draw inspiration from classical security research as conceptual techniques without thorough evaluation and implementation.

- To enhance security at the QC hardware level, various strategies can be implemented. One is the integration of cryptographic chips or modules, providing an additional layer of protection. The deployment of Physical Unclonable Functions (PUF) [140], [141], watermarking techniques [117], and the incorporation of obsolete circuit design through the insertion of dummy SWAP gates [142] collectively contribute to fortifying defenses against information leaks. Furthermore, it is crucial to implement robust security measures across various dimensions. Encryption techniques can be employed to secure both data transmission and storage, ensuring that sensitive information remains protected. Access controls play a pivotal role in permitting interactions with the quantum system only to authorized entities. Regular hardware monitoring and security assessments facilitate the early detection of unusual activities or potential intrusions, enabling proactive responses to security threats.
- To enhance security at the QC software level, a noteworthy work [143] discusses the design of a quantum computer antivirus. The authors propose a method to detect viruses in quantum programs by representing both input circuits and malicious virus circuits as graphs. They formulate virus checking as a sub-graph isomorphism finding problem, which can be quickly solved with existing algorithms. Research in [99] proposes splitting the quantum circuit into multiple parts that are sent to a single compiler at different times

or to multiple compilers. This way, the adversary only has access to partial information. The work in [116] advocates for inserting test points into quantum circuits to study their error rates concerning other qubit allocations.

- To enhance security at the QC algorithm level, it is essential to incorporate error-correction techniques. This strategic measure addresses the impact of noise and errors inherent in quantum hardware. Simultaneously, advances in distributed quantum algorithm [144] and quantum cryptographic algorithms are pivotal against potential threats and attacks. In the realm of quantum machine learning, retraining proves crucial in mitigating risks associated with data poisoning attacks [103]. Moreover, ensuring secure quantum computation involves exploring techniques such as quantum one-time pad protocols [145], quantum homomorphic encryption schemes [146], and quantum multiparty computing methods [147].

V. Conclusion

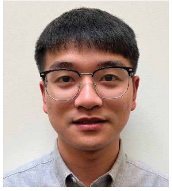
In summary, quantum computing demonstrates substantial theoretical promise, as evidenced by recent advancements in hardware, software, and algorithms. However, the predominant focus on performance has overshadowed critical security considerations. Addressing this imbalance, our article provides a comprehensive tutorial and survey that systematically identifies vulnerabilities in quantum computing systems. The overarching goal is to empower industry stakeholders and research communities, facilitating proactive measures to discern and mitigate security concerns, thereby establishing a secure foundation for the future development of quantum computing environments.

Fan Chen received the Ph.D. degree from the Department of Electrical and Computer Engineering, Duke University, in 2020. She is an Assistant Professor with the Department of Intelligent Systems Engineering, Indiana University Bloomington. Her research interests include quantum computing, quantum security, and machine learning acceleration. She is a recipient of the 2022 NSF Faculty Early Career Development Program (CAREER) Award, the 2022 Best Associate Editor of *IEEE Circuits and Systems Magazine*, the 2021 Service Recognition Award of Great Lakes Symposium on VLSI (GLSVLSI), and the 2019 Cadence Women in Technology Scholarship. Her research has won the Best Paper Award at the 2023 IEEE International Conference on Quantum Computing and Engineering (QCE) and the Best Paper Award and the Ph.D. forum Best Poster Award at the 2018 Asia and South Pacific Design Automation Conference (ASP-DAC). She serves on the Editorial Board of *IEEE Circuits and Systems Magazine*. She also serves as a technical reviewer for over 30+ international conferences/journals.

Lei Jiang received the B.S. and M.S. degrees from Shanghai Jiao Tong University and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Pittsburgh, in 2014. He is an Associate Professor with the Department of Intelligent Systems Engineering, Indiana University Bloomington. His research interests include quantum computing, privacy-preserving machine learning, and hardware accelerator design. Prior to joining IU, he worked for AMD.

Hausi Müller is a Professor of computer science and an Associate Dean of research with the Faculty of Engineering, University of Victoria, British Columbia, Canada. He is an international expert in software engineering, software evolution, cyber-physical systems, adaptive systems, smart and context-aware systems, and program understanding. He served on the IEEE Transactions on Software Engineering editorial board for 12 years. He is the cofounder of the SEAMS conference series (ACM/IEEE International Symposium on Software Engineering for Adaptive and Self-Managing Systems). He was a general chair of the 23rd ACM/IEEE International Conference of Software Engineering (ICSE 2001) and 30th IEEE International Conference on Software Maintenance and Evolution (ICSME 2014), and was recently a technical program co-chair of IEEE World Forum on Internet of Things (WF-IOT 2015 and 2018). He is a Fellow of the Canadian Academy of Engineering and a CS Golden Core member and received the 2016 TCSE Distinguished Service Award.

Philip Richerme received the Ph.D. degree from Harvard University in 2012, under the direction of Gerald Gabrielse, followed by the postdoctoral research with Chris Monroe from the University of Maryland. He is an Associate Professor of physics at Indiana University. His current research focuses on the quantum simulation of materials and chemical systems using trapped ions, which can serve as a well-isolated platform for studying classically intractable problems.



Cheng Chu received the M.S. and B.S. degrees from the Hefei University of Technology, Hefei, Anhui, China, in 2021 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Department of Intelligent Systems Engineering, Indiana University. His research falls primarily in the field of quantum computing, with an emphasis on quantum computer architecture, quantum security, and quantum neural network design.

Zhenxiao Fu received the B.S. degree in physics and the M.S. degree in electronics science and technology from ShanghaiTech University in 2020 and 2023, respectively. He is currently pursuing the Ph.D. degree with the Department of Intelligent Systems Engineering, Indiana University Bloomington, focusing on the research of quantum devices, architectures, and related algorithms.

Min Yang received the B.S. degree in engineering from Xidian University in 2020 and the M.S. degree in engineering from Xi'an Jiaotong University in 2023. She is currently pursuing the Ph.D. degree with the Department of Intelligent Systems Engineering, Indiana University Bloomington. Her research focuses on the fields of quantum computing, quantum security, and quantum machine learning.

References

[1] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines," *J. Stat. Phys.*, vol. 22, no. 5, pp. 563–591, May 1980.

[2] D. Deutsch, "Quantum theory, the Church–Turing principle and the universal quantum computer," *Proc. Roy. Soc. London A*, vol. 400, no. 1818, pp. 97–117, Jul. 1985.

[3] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, nos. 6–7, pp. 467–488, Jun. 1982.

[4] S. Lloyd, "Universal quantum simulators," *Science*, vol. 273, no. 5278, pp. 1073–1078, Aug. 1996.

[5] *IBM Quantum*. Accessed: Dec. 1, 2023. [Online]. Available: https://quantum-computing.ibm.com/

[6] *Google Quantum*. Accessed: Dec. 1, 2023. [Online]. Available: https://quantumai.google/hardware/

[7] *Rigetti Quantum Computing*. Accessed: Dec. 1, 2023. [Online]. Available: https://www.rigetti.com/

[8] *IonQ Quantum Computing*. Accessed: Dec. 1, 2023. [Online]. Available: https://ionq.com/

[9] *Quantinuum*. Accessed: Dec. 1, 2023. [Online]. Available: https://www.quantinuum.com/hardware/h2

[10] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.

[11] A. Cross, "The IBM Q experience and QISKit open-source quantum computing software," in *Proc. APS March Meeting Abstr.*, 2018, Paper no. L58-003.

[12] *Cirq Developers*. Accessed: Dec. 1, 2023. [Online]. Available: https://quantumai.google/cirq

[13] R. S. Smith et al., "An open-source, industrial-strength optimizing compiler for quantum programs," *Quantum Sci. Technol.*, vol. 5, no. 4, Jul. 2020, Art. no. 044001.

[14] S. Sivarajah et al., "t|ket>: A retargetable compiler for NISQ devices," *Quantum Sci. Technol.*, vol. 6, no. 1, Jan. 2021, Art. no. 014003.

[15] *Amazon*. Accessed: Dec. 1, 2023. [Online]. Available: https://github.com/aws/amazon-braket-sdk-python

[16] *Berkeley Quantum Synthesis Toolkit*. Accessed: Dec. 1, 2023. [Online]. Available: http://bqskit.lbl.gov

[17] V. Bergholm et al., "PennyLane: Automatic differentiation of hybrid quantum-classical computations," 2018, *arXiv:1811.04968*.

[18] V. Havlíček et al., "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, Mar. 2019.

[19] A. Peruzzo et al., "A variational eigenvalue solver on a photonic quantum processor," *Nature Commun.*, vol. 5, no. 1, p. 4213, Jul. 2014.

[20] Y. Li and S. C. Benjamin, "Efficient variational quantum simulator incorporating active error minimization," *Phys. Rev. X*, vol. 7, no. 2, Jun. 2017, Art. no. 021050.

[21] Z. Wang et al., "Quantum approximate optimization algorithm for MaxCut: A fermionic view," *Phys. Rev. A, Gen. Phys.*, vol. 97, no. 2, Feb. 2018, Art. no. 022304.

[22] C. Bravo-Prieto et al., "Quantum singular value decomposer," *Phys. Rev. A, Gen. Phys.*, vol. 101, no. 6, Jun. 2020, Art. no. 062310.

[23] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[24] P. Krantz et al., "A quantum engineer's guide to superconducting qubits," *Appl. Phys. Rev.*, vol. 6, no. 2, Jun. 2019, Art. no. 021318.

[25] C. D. Bruzewicz et al., "Trapped-ion quantum computing: Progress and challenges," *Appl. Phys. Rev.*, vol. 6, no. 2, Jun. 2019, Art. no. 021314.

[26] B. Valiron et al., "Programming the quantum future," *Commun. ACM*, vol. 58, no. 8, pp. 52–61, 2015.

[27] F. T. Chong, D. Franklin, and M. Martonosi, "Programming languages and compiler design for realistic quantum hardware," *Nature*, vol. 549, no. 7671, pp. 180–187, Sep. 2017.

[28] M. Kjaergaard et al., "Superconducting qubits: Current state of play," *Annu. Rev. Condens. Matter Phys.*, vol. 11, no. 1, pp. 369–395, Mar. 2020.

[29] J. M. Pino et al., "Demonstration of the trapped-ion quantum CCD computer architecture," *Nature*, vol. 592, no. 7853, pp. 209–213, Apr. 2021.

[30] M. Saffman, "Quantum computing with atomic qubits and Rydberg interactions: Progress and challenges," *J. Phys. B, At., Mol. Opt. Phys.*, vol. 49, no. 20, Oct. 2016, Art. no. 202001.

[31] H. Levine et al., "Parallel implementation of high-fidelity multiqubit gates with neutral atoms," *Phys. Rev. Lett.*, vol. 123, no. 17, Oct. 2019, Art. no. 170503.

[32] J. L. O'brien, A. Furusawa, and J. Vučković, "Photonic quantum technologies," *Nature Photon.*, vol. 3, no. 12, pp. 687–695, 2009.

[33] F. Flamini, N. Spagnolo, and F. Sciarrino, "Photonic quantum information processing: A review," *Rep. Prog. Phys.*, vol. 82, no. 1, 2018, Art. no. 016001.

[34] C. Kloeffel and D. Loss, "Prospects for spin-based quantum computing in quantum dots," *Annu. Rev. Condens. Matter Phys.*, vol. 4, no. 1, pp. 51–81, Apr. 2013.

[35] A. Chatterjee et al., "Semiconductor qubits in practice," *Nature Rev. Phys.*, vol. 3, no. 3, pp. 157–177, Feb. 2021.

[36] B. Nachman et al., "Unfolding quantum computer readout noise," *NPJ Quantum Inf.*, vol. 6, no. 1, p. 84, Sep. 2020.

[37] M. R. Geller and M. Sun, "Toward efficient correction of multiqubit measurement errors: Pair correlation method," *Quantum Sci. Technol.*, vol. 6, no. 2, Apr. 2021, Art. no. 025009.

[38] J. Emerson et al., "Symmetrized characterization of noisy quantum processes," *Science*, vol. 317, no. 5846, pp. 1893–1896, Sep. 2007.

[39] M. Urbanek et al., “Mitigating depolarizing noise on quantum computers with noise-estimation circuits,” *Phys. Rev. Lett.*, vol. 127, no. 27, Dec. 2021, Art. no. 27050.

[40] M. Sarovar et al., “Detecting crosstalk errors in quantum information processors,” *Quantum*, vol. 4, p. 321, Sep. 2020.

[41] S. Gustavsson et al., “Suppressing relaxation in superconducting qubits by quasiparticle pumping,” *Science*, vol. 354, no. 6319, pp. 1573–1577, Dec. 2016.

[42] C. Monroe et al., “Programmable quantum simulations of spin systems with trapped ions,” *Rev. Mod. Phys.*, vol. 93, no. 2, Apr. 2021, Art. no. 025001.

[43] K. Rudinger et al., “Probing context-dependent errors in quantum processors,” *Phys. Rev. X*, vol. 9, no. 2, Jun. 2019, Art. no. 021045.

[44] A. Mi, S. Deng, and J. Szefer, “Securing reset operations in NISQ quantum computers,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2022, pp. 2279–2293.

[45] C. Fang et al., “Crosstalk suppression in individually addressed two-qubit gates in a trapped-ion quantum computer,” *Phys. Rev. Lett.*, vol. 129, no. 24, Dec. 2022, Art. no. 240504.

[46] F. Arute et al., “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[47] K. Wright et al., “Benchmarking an 11-qubit quantum computer,” *Nature Commun.*, vol. 10, no. 1, p. 5464,2019.

[48] H.-S. Zhong et al., “Quantum computational advantage using photons,” *Science*, vol. 370, no. 6523, pp. 1460–1463, 2020.

[49] K. Takeda et al., “A fault-tolerant addressable spin qubit in a natural silicon quantum dot,” *Sci. Adv.*, vol. 2, no. 8, Aug. 2016, Art. no. e1600694.

[50] M. Ahsan, S. A. Z. Naqvi, and H. Anwer, “Quantum circuit engineering for correcting coherent noise,” *Phys. Rev. A, Gen. Phys.*, vol. 105, no. 2, Feb. 2022, Art. no. 022428.

[51] *IonQ Forte*. Accessed: Dec. 1, 2023. [Online]. Available: https://ionq.com/quantum-systems/forte/

[52] *IBM Heron*. Accessed: Dec. 1, 2023. [Online]. Available: https://research.ibm.com/blog/ibm-quantum-roadmap-2025/

[53] P. W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134.

[54] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Proc. 28th Annu. ACM Symp. Theory Comput. (STOC)*, 1996, pp. 212–219.

[55] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” *Phys. Rev. A, Gen. Phys.*, vol. 52, no. 4, pp. R2493–R2496, Oct. 1995.

[56] D. A. Lidar and T. A. Brun, *Quantum Error Correction*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[57] B. M. Terhal, “Quantum error correction for quantum memories,” *Rev. Mod. Phys.*, vol. 87, no. 2, pp. 307–346, Apr. 2015.

[58] S. Pakin, “A quantum macro assembler,” in *Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC)*, Sep. 2016, pp. 1–8.

[59] A. J. Abhari et al., “Scaffold: Quantum programming language,” Dept. Comput. Sci., Princeton Univ., Princeton, NJ, USA, Tech. Rep. TR-934-12, 2012.

[60] C. Lattner and V. Adve, “LLVM: A compilation framework for lifelong program analysis & transformation,” in *Proc. Int. Symp. Code Gener. Optim. (CGO)*, Mar. 2004, pp. 75–86.

[61] D. Wecker and K. M. Svore, “Liqui|⟩: A software design architecture and domain-specific language for quantum computing,” 2014, *arXiv:1402.4467.*

[62] R. S. Smith, M. J. Curtis, and W. J. Zeng, “A practical quantum instruction set architecture,” 2016, *arXiv:1608.03355.*

[63] K. Svore et al., “Q: Enabling scalable quantum computing and development with a high-level DSL,” in *Proc. Real World Domain Specific Lang. Workshop*,2018, pp. 1–10.

[64] B. Bichsel et al., “Silq: A high-level quantum language with safe uncomputation and intuitive semantics,” in *Proc. 41st ACM SIGPLAN Conf. Program. Lang. Design Implement.*, Jun. 2020, pp. 286–300.

[65] L. E. Heyfron and E. T. Campbell, “An efficient quantum compiler that reduces *T* count,” *Quantum Sci. Technol.*, vol. 4, no. 1, Sep. 2018, Art. no. 015004.

[66] P. Nation and B. Johnson, *How to Measure and Reset a Qubit in the Middle of a Circuit Execution*. Accessed: Dec. 1, 2023. [Online]. Available: https://research.ibm.com/blog/quantum-mid-circuit-measurement

[67] *Q-CTRL*. Accessed: Dec. 1, 2023. [Online]. Available: https://docs.q-ctrl.com/fire-opal/user-guides/how-to-run-a-circuit-with-mid-circuit-measurements

[68] R. LaRose et al., “Mitiq: A software package for error mitigation on noisy quantum computers,” *Quantum*, vol. 6, p. 774, Aug. 2022.

[69] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, Jan. 1999.

[70] C. Zalka, “Fast versions of Shor’s quantum factoring algorithm,” 1998, *arXiv:quant-ph/9806084.*

[71] C. H. Bennett and G. Brassard, “Quantum cryptography: Public key distribution and coin tossing,” *Theor. Comput. Sci.*, vol. 560, pp. 7–11, Dec. 2014.

[72] H.-K. Lo, M. Curty, and K. Tamaki, “Secure quantum key distribution,” *Nature Photon.*, vol. 8, no. 8, pp. 595–604, Aug. 2014.

[73] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Phys. Rev. Lett.*, vol. 103, no. 15, Oct. 2009, Art. no. 150502.

[74] D. W. Berry, “High-order quantum algorithm for solving linear differential equations,” *J. Phys. A, Math. Theor.*, vol. 47, no. 10, Mar. 2014, Art. no. 105301.

[75] K. Mitarai et al., “Quantum circuit learning,” *Phys. Rev. A, Gen. Phys.*, vol. 98, no. 3, Sep. 2018, Art. no. 032309.

[76] M. H. Amin et al., “Quantum Boltzmann machine,” *Phys. Rev. X*, vol. 8, no. 2, May 2018, Art. no. 021050.

[77] S. Lloyd and C. Weedbrook, “Quantum generative adversarial learning,” *Phys. Rev. Lett.*, vol. 121, no. 4, Jul. 2018, Art. no. 040502.

[78] D. Dong et al., “Quantum reinforcement learning,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1207–1220, Jul. 2008.

[79] G. D. Paparo et al., “Quantum speedup for active learning agents,” *Phys. Rev. X*, vol. 4, no. 3, Jul. 2014, Art. no. 031002.

[80] O. Higgott, D. Wang, and S. Brierley, “Variational quantum computation of excited states,” *Quantum*, vol. 3, p. 156, Jul. 2019.

[81] S. McArdle et al., “Variational ansatz-based quantum simulation of imaginary time evolution,” *NPJ Quantum Inf.*, vol. 5, no. 1, p. 75, Sep. 2019.

[82] A. Bengtsson et al., “Improved success probability with greater circuit depth for the quantum approximate optimization algorithm,” *Phys. Rev. Appl.*, vol. 14, no. 3, Sep. 2020, Art. no. 034010.

[83] M. Lubasch et al., “Variational quantum algorithms for nonlinear problems,” *Phys. Rev. A, Gen. Phys.*, vol. 101, no. 1, Jan. 2020, Art. no. 010301.

[84] K. Bharti et al., “Noisy intermediate-scale quantum algorithms,” *Rev. Mod. Phys.*, vol. 94, Feb. 2022, Art. no. 015004.

[85] M. Cerezo et al., “Variational quantum algorithms,” *Nature Rev. Phys.*, vol. 3, no. 9, pp. 625–644,2021.

[86] J. R. McClean et al., “The theory of variational hybrid quantum-classical algorithms,” *New J. Phys.*, vol. 18, no. 2, Feb. 2016, Art. no. 023023.

[87] L. Banchi, J. Pereira, and S. Pirandola, “Generalization in quantum machine learning: A quantum information standpoint,” *PRX Quantum*, vol. 2, no. 4, Nov. 2021, Art. no. 040321.

[88] M. C. Caro et al., “Encoding-dependent generalization bounds for parametrized quantum circuits,” *Quantum*, vol. 5, p. 582, Nov. 2021.

[89] M. C. Caro et al., “Generalization in quantum machine learning from few training data,” *Nature Commun.*, vol. 13, no. 1, p. 4919, Aug. 2022.

[90] R. Wang, P. Richerme, and F. Chen, “A hybrid quantum–classical neural network for learning transferable visual representation,” *Quantum Sci. Technol.*, vol. 8, no. 4, Oct. 2023, Art. no. 045021.

[91] Y. LeCun. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[92] A. Krizhevsky et al., *Learning Multiple Layers of Features From Tiny Images*. Accessed: Dec. 1, 2023. [Online]. Available: https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf

[93] H. Huang et al., “Power of data in quantum machine learning,” 2020, *arXiv:2011.01938.*

[94] L. Schatzki et al., “Entangled datasets for quantum machine learning,” 2021, *arXiv:2109.03400.*

[95] E. Perrier, A. Youssry, and C. Ferrie, “QDataSet, quantum datasets for machine learning,” *Sci. Data*, vol. 9, no. 1, p. 582, Sep. 2022.

[96] G. Chen et al., “Alchemy: A quantum chemistry dataset for benchmarking AI models,” 2019, *arXiv:1906.09427.*

[97] D. Balcells and B. B. Skjelstad, “tmQM—Dataset-quantum geometries and properties of 86 k transition metal complexes,” *J. Chem. Inf. Model.*, vol. 60, no. 12, pp. 6135–6146, 2020.

[98] A. Suresh et al., “Short paper: A quantum circuit obfuscation methodology for security and privacy,” in *Proc. Workshop Hardw. Architectural Support Secur. Privacy (HASP)*. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 1–5.

[99] A. A. Saki et al., “Split compilation for security of quantum circuits,” in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*,Nov. 2021, pp. 1–7.

[100] C. Xu, F. Erata, and J. Szefer, “Exploration of power side-channel vulnerabilities in quantum computer controllers,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*. New York, NY, USA: ACM, Nov. 2023, pp. 579–593.

[101] C. Xu, F. Erata, and J. Szefer, “Classification of quantum computer fault injection attacks,” 2023, *arXiv:2309.05478*.

[102] H. Ziade et al., “A survey on fault injection techniques,” *Int. Arab J. Inf. Technol.*, vol. 1, no. 2, pp. 171–186, 2004.

[103] C. Chu et al., “QTROJAN: A circuit backdoor against quantum neural networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[104] C. Chu et al., “QDoor: Exploiting approximate synthesis for backdoor attacks in quantum neural networks,” in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Sep. 2023, pp. 1098–1106.

[105] S. Upadhyay and S. Ghosh, “Robust and secure hybrid quantum-classical computation on untrusted cloud-based quantum hardware,” in *Proc. 11th Int. Workshop Hardw. Architectural Support Secur. Privacy*,Oct. 2022, pp. 1–9.

[106] A. A. Saki and S. Ghosh, “Qubit sensing: A new attack model for multi-programming quantum computing,” 2021, *arXiv:2104.05899*.

[107] A. A. Saki, R. O. Topaloglu, and S. Ghosh, “Shuttle-exploiting attacks and their defenses in trapped-ion quantum computers,” *IEEE Access*, vol. 10, pp. 2686–2699, 2022.

[108] A. Ash-Saki, M. Alam, and S. Ghosh, “Analysis of crosstalk in NISQ devices and security implications in multi-programming regime,” in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design*, Aug. 2020, pp. 25–30.

[109] A. Huang et al., “Laser-seeding attack in quantum key distribution,” *Phys. Rev. Appl.*, vol. 12, no. 6, Dec. 2019, Art. no. 064043.

[110] A. Huang et al., “Laser-damage attack against optical attenuators in quantum key distribution,” *Phys. Rev. Appl.*, vol. 13, no. 3, Mar. 2020, Art. no. 034017.

[111] D. M. Abrams et al., “Methods for measuring magnetic flux crosstalk between tunable transmons,” *Phys. Rev. Appl.*, vol. 12, no. 6, Dec. 2019, Art. no. 064022.

[112] A. Ash-Saki, M. Alam, and S. Ghosh, “Experimental characterization, modeling, and analysis of crosstalk in a quantum computer,” *IEEE Trans. Quantum Eng.*, vol. 1, pp. 1–6, Sep. 2020.

[113] A. P. Vepsäläinen et al., “Impact of ionizing radiation on superconducting qubit coherence,” *Nature*, vol. 584, no. 7822, pp. 551–556, Aug. 2020.

[114] G. Li, Y. Ding, and Y. Xie, “Tackling the qubit mapping problem for NISQ-era quantum devices,” in *Proc. 24th Int. Conf. Architectural Support Program. Lang. Operating Syst.*, Providence, RI, USA, Apr. 2019, pp. 1001–1014.

[115] J. Liu, P. Li, and H. Zhou, “Not all SWAPs have the same cost: A case for optimization-aware qubit routing,” in *Proc. IEEE Int. Symp. High-Perform. Comput. Archit. (HPCA)*,Seoul, South Korea, Apr. 2022, pp. 709–725.

[116] N. Acharya and S. M. Saeed, “A lightweight approach to detect malicious/unexpected changes in the error rates of NISQ computers,” in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2020, pp. 1–9.

[117] V. Saravanan and S. M. Saeed, “Decomposition-based watermarking of quantum circuits,” in *Proc. 22nd Int. Symp. Qual. Electron. Design (ISQED)*, Apr. 2021, pp. 73–78.

[118] Y. Zhang et al., “Federated learning with quantum secure aggregation,” 2022, *arXiv:2207.07444*.

[119] Q. Xia and Q. Li, “QuantumFed: A federated learning framework for collaborative quantum training,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2021, pp. 1–6.

[120] R. Huang, X. Tan, and Q. Xu, “Quantum federated learning with decentralized data,” *IEEE J. Sel. Topics Quantum Electron.*, vol. 28, no. 4, pp. 1–10, Jul. 2022.

[121] C. Chu, L. Jiang, and F. Chen, “CryptoQFL: Quantum federated learning on encrypted data,” 2023, *arXiv:2307.07012*.

[122] N. Liu and P. Wittek, “Vulnerability of quantum classification to adversarial perturbations,” *Phys. Rev. A, Gen. Phys.*, vol. 101, no. 6, Jun. 2020, Art. no. 062331.

[123] S. Lu, L.-M. Duan, and D.-L. Deng, “Quantum adversarial machine learning,” *Phys. Rev. Res.*, vol. 2, no. 3, Aug. 2020, Art. no. 033212.

[124] S. Lu, L. Duan, and D. Deng, “Quantum adversarial machine learning,” *Phys. Rev. Res.*, vol. 2, no. 3, Aug. 2020, Art. no. 033212.

[125] R. Wang, F. Baba-Yara, and F. Chen, “JustQ: Automated deployment of fair and accurate quantum neural networks,” in *Proc. Asia South Pacific Design Autom. Conf. (ASP-DAC)*, 2024, pp. 1–5.

[126] R. LaRose and B. Coyle, “Robust data encodings for quantum classifiers,” *Phys. Rev. A, Gen. Phys.*, vol. 102, no. 3, Sep. 2020, Art. no. 032420.

[127] M. Schuld, R. Sweke, and J. J. Meyer, “Effect of data encoding on the expressive power of variational quantum-machine-learning models,” *Phys. Rev. A, Gen. Phys.*, vol. 103, Mar. 2020, Art. no. 032430.

[128] A. Pérez-Salinas et al., “Data re-uploading for a universal quantum classifier,” *Quantum*, vol. 4, p. 226, Feb. 2020.

[129] C. Chu et al., “QMLP: An error-tolerant nonlinear quantum MLP architecture using parameterized two-qubit gates,” in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design*,Boston, MA, USA, Aug. 2022, pp. 4:1–4:6.

[130] J. M. Kubler, S. Buchholz, and B. Scholkopf, “The inductive bias of quantum kernels,” in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2021, pp. 12661–12673.

[131] B. T. Kiani, S. Lloyd, and R. Maity, “Learning unitaries by gradient descent,” 2020, *arXiv:2001.11897*.

[132] M. Larocca et al., “Theory of overparametrization in quantum neural networks,” *Nature Comput. Sci.*, vol. 3, no. 6, pp. 542–551, Jun. 2023.

[133] K. Zhang et al., “Gaussian initializations help deep variational quantum circuits escape from the barren plateau,” in 2022, *arXiv:2203.09376*.

[134] W. Lavrijsen et al., “Classical optimizers for noisy intermediate-scale quantum devices,” in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*,Oct. 2020, pp. 267–277.

[135] J. R. McClean et al., “Barren plateaus in quantum neural network training landscapes,” *Nature Commun.*, vol. 9, no. 1, p. 4812, Nov. 2018.

[136] S. Wang et al., “Noise-induced barren plateaus in variational quantum algorithms,” *Nature Commun.*, vol. 12, no. 1, p. 6961, Nov. 2021.

[137] S. Resch and U. R. Karpuzcu, “Benchmarking quantum computers and the impact of quantum noise,” *ACM Comput. Surv.*, vol. 54, no. 7, pp. 1–35, Sep. 2022.

[138] W. M. Watkins, S. Y.-C. Chen, and S. Yoo, “Quantum machine learning with differential privacy,” *Sci. Rep.*, vol. 13, no. 1, p. 2453, Feb. 2023.

[139] J. Guan, W. Fang, and M. Ying, “Verifying fairness in quantum machine learning,” in *Proc. 34th Int. Conf. Comput. Aided Verification (CAV)*, in Lecture Notes in Computer Science, vol. 13372, Haifa, Israel. Cham, Switzerland: Springer, Aug. 2022, pp. 408–429.

[140] M. Arapinis et al., “Quantum physical unclonable functions: Possibilities and impossibilities,” *Quantum*, vol. 5, p. 475, Jun. 2021.

[141] K. Phalak et al., “Quantum PUF for security and trust in quantum computing,” *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 11, no. 2, pp. 333–342, Jun. 2021.

[142] A. Suresh et al., “Short paper: A quantum circuit obfuscation methodology for security and privacy,” in *Proc. Workshop Hardw. Architectural Support Secur. Privacy*, Oct. 2021, pp. 6:1–6:5.

[143] S. Deshpande et al., “Design of quantum computer antivirus,” in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2023, pp. 260–270.

[144] S. Y.-C. Chen and S. Yoo, “Federated quantum machine learning,” *Entropy*, vol. 23, no. 4, p. 460, Apr. 2021.

[145] F.-G. Deng and G. L. Long, “Secure direct communication with a quantum one-time pad,” *Phys. Rev. A, Gen. Phys.*, vol. 69, no. 5, May 2004, Art. no. 052319.

[146] U. Mahadev, “Classical homomorphic encryption for quantum circuits,” *SIAM J. Comput.*, vol. 52, no. 6, pp. FOCS18-189–FOCS18-215, Dec. 2023.

[147] C. Crépeau, D. Gottesman, and A. Smith, “Secure multi-party quantum computation,” in *Proc. 34th Annu. ACM Symp. Theory Comput.*, 2002, pp. 643–652.