Towards Secure Classical-Quantum Systems

Daniel Volya*, Tao Zhang†, Nashmin Alam†, Mark Tehranipoor† and Prabhat Mishra*
*Department of Computer & Information Science & Engineering
†Department of Electrical & Computer Engineering
University of Florida, Gainesville, Florida, USA

***Abstract*—Quantum computing has emerged as a promising paradigm, offering significant advancements in solving complex problems that are intractable for classical computers. These systems often involve integrated classical-quantum architectures, where classical components control and communicate with quantum devices. While this integration unlocks the potential of quantum computing, it also introduces new security vulnerabilities and challenges that must be addressed to ensure secure and reliable classical-quantum computing. This paper provides a comprehensive overview of the security concerns related to classical-quantum systems and discusses potential countermeasures. Specifically, we first investigate secure communication with a quantum device through side-channel analysis of post-quantum encryption algorithms. Next, we analyze security vulnerabilities in quantum devices. Finally, we explore mitigation strategies as well as the role of quantum compilation for securing quantum devices. By examining and addressing these critical security concerns, we aim to contribute to the development of a secure and robust foundation for the future of quantum computing. This work will be a stepping stone in secure and trustworthy deployment of integrated classical-quantum systems across various application domains.**

***Index Terms*—Quantum computing, quantum security, post-quantum cryptography.**

I. Introduction

The advent of quantum computing represents a confluence of computer science and physics, with the potential to revolutionize computational capabilities. Exploiting quantum mechanical phenomena, such as superposition and entanglement, quantum devices can address complex problems that are currently intractable for classical systems. However, the development and implementation of quantum systems are fraught with engineering challenges. The construction of quantum devices necessitates intricate engineering to harness and manipulate quantum mechanical effects. Figure 1 shows an example classical-quantum system. In this example, the Google Sycamore quantum computer is connected to classical computers that provides various mechanisms critical for quantum computing, such as preparation of the initial state of the quantum computer, mapping of the required functionality, measurement of the result, and providing feedback, if needed.

The delicate nature of quantum phenomena makes the devices susceptible to errors, sensitivity to environmental factors, and various types of vulnerabilities, including cross-talk among qubits. Cross-talk, for example, poses a critical concern in the context of shared quantum systems, as multiple users accessing the same quantum computer may inadvertently or intentionally obtain information from one another, compromising the security and confidentiality of the data being processed.

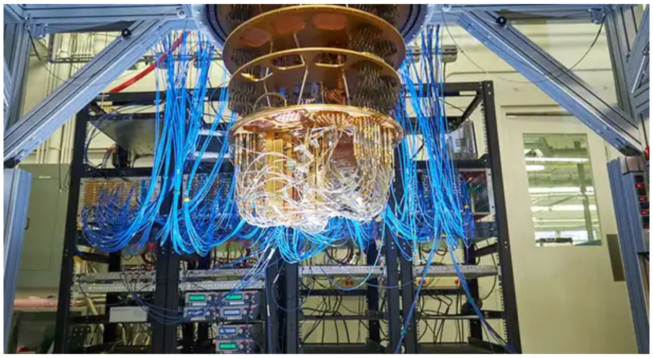


Fig. 1: Google's Sycamore quantum computer [1]. The quantum computer is connected to classical computers as well as other hardware devices to enable classical-quantum computing. The classical computer performs various activities, including quantum state preparation, pulse generation for mapping of quantum circuits, measure the result of the quantum computation, and provide feedback.

To address the security vulnerabilities inherent in quantum devices, researchers must develop new techniques and protocols that safeguard against cross-talk and other forms of information leakage. For instance, employing isolation and partitioning techniques to separate users' quantum operations can ensure that each user's computation is confined to a designated portion of the quantum system. This approach would minimize the potential for cross-talk and information leakage among users, although it may entail additional resource overhead and reduced computational efficiency. Moreover, developing quantum error correction techniques that can detect and mitigate the effects of cross-talk and other sources of noise is critical for enhancing the security of quantum devices. By implementing robust error correction methods, it becomes possible to minimize the impact of cross-talk on the accuracy and reliability of quantum computations, reducing the likelihood of unauthorized information access. However, it is a major challenge to develop an effective error correction strategy since it must account for the unique properties and behaviors of quantum systems.

In addition to addressing specific quantum-related vulnerabilities, classical hardware, such as CPUs, FPGAs, and ASICs, plays a crucial role in the operation of quantum computing systems, as they are responsible for interfacing with quantum devices and orchestrating the necessary control and data processing tasks. As the availability of quantum computers in the cloud becomes more widespread, classical computing components remain a vital puzzle piece in the successful implementation and utilization of these powerful systems. The cloud-based nature of quantum computing services necessitates the secure transmission of jobs and data between classical and quantum systems, further emphasizing the importance of classical computing components in the overall operation of quantum systems. To ensure the confidentiality, integrity, and availability of data exchanged between classical computers and quantum devices, it is essential to employ post-quantum cryptography (PQC) techniques that can withstand the computational power of quantum machines. In this context, classical computing components not only serve as indispensable tools for interfacing with quantum devices but also a potential point of failure in the rapidly evolving security landscape of quantum computing.

As we progress towards realizing the full potential of quantum computing, it is imperative to consider the security implications of the entire quantum system – both classical and quantum components. This holistic perspective allows us to tackle the complex challenges that arise from the interactions between these distinct computing paradigms. In this paper, we focus on two main aspects: the security of classical computers that communicate with quantum devices, and the security of quantum computers themselves. By examining and addressing these critical security concerns, we aim to contribute to the development of a more secure and reliable foundation for the future of quantum computing, ensuring the safe and effective operation of integrated classical-quantum systems.

The remainder of this paper is organized as follows. Section II provides an overview of quantum computing and surveys security vulnerabilities. Section III explores the security vulnerabilities associated with communication between classical and quantum computers. Section IV investigates security vulnerabilities in quantum computers. Section V surveys various mitigation strategies for securing quantum devices. Section VI explores the role of quantum compilers for enabling secure quantum computing. Finally, Section VII concludes the paper.

II. Background

A. Overview of Quantum Computing

Quantum computing is a novel approach to processing information that leverages the principles of quantum mechanics, utilizing qubits, superposition, and entanglement to perform complex calculations. In this section, we provide a brief overview of the fundamental concepts in quantum computing and physics.

1) Qubits, superposition, and entanglement: A qubit, or quantum bit, represents the basic unit of quantum information. Unlike classical bits, which can only take on a value of 0 or 1, qubits exist in a state that can be described as a linear combination of basis states $|0\rangle$ and $|1\rangle$. Mathematically, a qubit's state can be represented as:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \tag{1}$$

Here, α and β are complex numbers that determine the probability amplitudes for the qubit to be in the $|0\rangle$ and $|1\rangle$ states, respectively. The probabilities of measuring a qubit in either state are given by the squared magnitudes of the coefficients, i.e., $|\alpha|^2 + |\beta|^2$. Due to the conservation of probability, the sum of these probabilities must equal 1:

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2}$$

Superposition is a fundamental principle of quantum mechanics that allows particles to exist in multiple states simultaneously. In the context of quantum computing, this property allows qubits to be in a superposition of $|0\rangle$ and $|1\rangle$ states. This unique feature enables quantum computers to perform parallel computations on multiple inputs simultaneously, leading to potential speedups in solving certain computational problems, such as factorization and optimization.

Entanglement is another key concept in quantum mechanics, characterized by the strong correlations that can exist between two or more quantum particles. When particles are entangled, the quantum state of one particle cannot be described independently of the others, regardless of the physical distance between them. Mathematically, entanglement can be represented using the tensor product of individual qubit states:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle . \tag{3}$$

In quantum computing, entanglement is harnessed to create non-classical correlations between qubits, which can lead to more efficient algorithms and enhanced computational power [2]. Notably, entanglement plays a crucial role in the implementation of quantum error correction techniques and the realization of quantum communication protocols, such as quantum teleportation and quantum key distribution.

2) Quantum gates and circuits: Quantum gates and circuits are essential components in the manipulation and processing of quantum information. Analogous to classical logic gates, quantum gates perform operations on qubits, transforming their states while adhering to the principles of quantum mechanics. However, unlike classical gates, quantum gates are reversible and represented by unitary matrices. Some common quantum gates include the Pauli-X, Pauli-Y, Pauli-Z, Hadamard, CNOT, and Toffoli gates, among others.

Quantum circuits are designed to execute quantum algorithms by sequentially applying a series of quantum gates to an initial set of qubits. These circuits, like their classical counterparts, are composed of wires and gates, with the critical difference being that they operate on qubits rather than classical bits. The quantum circuit's output is obtained by measuring the final state of the qubits after the execution of the gates.

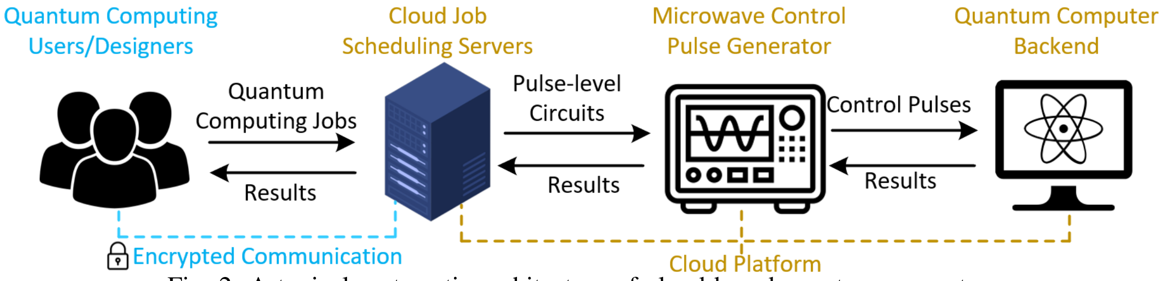


Fig. 2: A typical systematic architecture of cloud-based quantum computers.

One notable feature of quantum circuits is their ability to exploit the principles of superposition and entanglement, enabling the performance of parallel computations and the generation of non-classical correlations. This property is crucial for quantum algorithms, such as Shor's algorithm for integer factorization and Grover's algorithm for unsorted database search, which can offer significant speedups compared to classical algorithms.

B. *Quantum Coherence and Decoherence*

Understanding the underlying device physics of quantum systems is crucial for addressing the unique challenges posed by quantum computing, including the issues of coherence, decoherence, and qubit interactions.

Coherence is a fundamental property of quantum systems that arises from the superposition principle. In a coherent quantum system, the phase relationship between different states remains constant over time, allowing the system to exhibit interference effects. Coherence is essential for the proper functioning of quantum algorithms, as it enables the manipulation of quantum states through quantum gates and the utilization of the superposition principle to achieve computational speedups. The degree of coherence in a quantum system is quantified by the coherence time, which represents the timescale over which the coherence is maintained. The longer the coherence time, the more resilient the quantum system is to noise and other sources of error, allowing for more complex and accurate quantum computations [3], [4].

Decoherence, on the other hand, refers to the loss of coherence in a quantum system due to its interaction with the surrounding environment. As a result of decoherence, quantum superpositions and entangled states collapse into classical mixtures, ultimately leading to the loss of the unique quantum properties that enable quantum computing. Decoherence is one of the primary challenges in the development of practical quantum computers, as it imposes limits on the accuracy and scalability of quantum computations. Several factors can cause decoherence, including thermal fluctuations, electromagnetic radiation, and material imperfections. To mitigate the effects of decoherence, researchers employ various strategies, such as maintaining low temperatures, isolating quantum systems from external sources of noise, and developing quantum error correction techniques that can detect and correct errors introduced by decoherence.

While coherence is a vital property that enables the manipulation of quantum states and the realization of quantum algorithms, decoherence presents a major challenge that must be overcome to achieve practical quantum computing.

C. *Classical Security Vulnerabilities*

This section provides an overview of classical security vulnerabilities that are relevant to quantum systems. An example flow of a cloud-based quantum computer is illustrated in Figure 2 where local users design their quantum circuits and submit computing jobs to the cloud through encrypted communication. The cloud job scheduling servers then send pulse-level circuits to microwave electronics to generate corresponding control pulses. Qubits in quantum computers must be regulated and controlled by control pulses to implement quantum algorithms and get results back to users. Clearly, a cloud-based quantum computer system is *hybrid*, consisting of not only promising quantum backends but also classical platforms and instruments. Therefore, we introduce hardware security vulnerabilities in classical platforms which have implications to compromise cloud-based quantum systems.

1) Confidentiality Vulnerabilities: Side-channel attacks pose serious security concerns on (cryptographic) implementations by deducing assets, e.g., private keys, from observable physical properties of running devices such as power, electromagnetic, and timing, regardless of algorithmic strength. The encrypted communication between local users and cloud servers (see Figure 2) may be eavesdropped if the session key can be extracted by exploiting side-channel vulnerabilities of classical microelectronic platforms, such as CPU, ASIC, and FPGA.

Power side-channel attacks measure the power consumption of the target application, preprocess the profiles, and deduce the key by following statistical methodologies like simple power analysis (SPA) [5] and differential/correlation power analysis (DPA/CPA) [6]. For example, the key-dependent square and multiplication operations in the RSA algorithm are likely to be easily distinguished in the power traces to infer the secret key [5]. DPA/CPA assumes a set of (sub)key guesses first and then computes values of selected intermediate variables under each guess; the statistical power (e.g., hamming distance) of the only correct (sub)key guess case would exhibit the highest correlation with the actual power consumption when the number of plaintexts is enough.

Electromagnetic (EM) side-channel attacks [7] share similar flows with their power side-channel counterparts but can be more effective because EM emanation is highly localized and less obfuscated by non-crypto operations, yielding a higher signal-to-noise ratio. It is worth noting that recent research on cloud FPGA platforms has shown that physical access/proximity may not be a must in side-channel attacks [8], [9]. The adversaries assume a multi-tenant model and demonstrate the possibility that the malicious power sensor can capture side-channel profiles from physically isolated victim designs (e.g., AES accelerator) on the same FPGA fabric because of the sharing of the power distribution network (PDN). Besides power and EM threats, timing side-channel is drawing more and more attention given that Spectre [10] and Meltdown [11] vulnerabilities have affected nearly all mainstream CPU vendors including Intel, AMD, and ARM. These micro-architectural vulnerabilities and their variants allow malicious programs to steal security assets, e.g., session keys, from the memory of other running programs [12]. The root cause of these vulnerabilities is the speculative execution feature available in most modern processors; by abusing this feature, Meltdown can enable user programs to access kernel memory while Spectre can break the isolation between user applications in different ways like *Flush+Reload* and *Evict+Reload* [13].

2) Integrity Vulnerabilities: Malicious insiders of cloud service providers may intentionally cause integrity violations to falsify the message or alter compliant behaviors. There are two major categories of attacks to achieve their goals, i.e., physical fault injection and tampering. Physical fault injection refers to inducing drastic environmental disturbance on the target device such that timing faults can be injected to gain privilege escalation, bypass built-in security mechanisms, and modify registered data [14]. The particular attacks entail premature clocks, voltage drops, EM disturbances, and laser pulses at different costs, precision, and expertise requirements [15]. For instance, sudden voltage drops or power glitches would increase the gate delay and there is a chance those on critical paths may consequently fail to meet the timing constraints, resulting in setup/hold time violations.

Similar to side-channel attacks on cloud FPGA, fault injection can also be fully remote by exploiting the so-called *Rowhammer* vulnerability where the adversaries repeatedly access a specific row of memory cells in a dynamic random-access memory (DRAM) device. The accessing or hammering can render electrical interference on adjacent rows of memory cells and finally cause bit flips for data corruption [16], even in a cross-VM manner on cloud servers [17]. As for tampering attacks on data or firmware, although such behaviors at the physical layer are mostly prohibited by error detection schemes, there are still successful attack cases. For example, despite active bitstream encryption, compromised (encrypted) configuration bitstreams can be still loaded on Xilinx Virtex-5 FPGAs to generate faulty outputs which can be used to deduce secret information [18]. More targeted bit-level manipulations [19] can be further enabled with reverse-engineered bitstream formats [20].

III. Securing Communication between Classical and Quantum Computers

There are security risks associated with the growing popularity of cloud-based quantum computing services since the users submit quantum computing workloads to a job management server via network channels. To safeguard the confidentiality, integrity, and availability (CIA) requirements of network data in the quantum era, a contest was launched by NIST in 2016 to establish new post-quantum cryptography (PQC) algorithms that can withstand both conventional and quantum attacks.

The very first standards for PQC in the categories of public key encryption (PKE), key encapsulation mechanisms (KEM), and digital signatures (DS) were recently announced by NIST in 2022 after three rounds of assessment. The two widely recognized learning with error (LWE) and learning with rounding (LWR) problems, which work on structured lattice-based schemes, are the root of the difficulty for three of the seven finalist candidates. In this regard, the LWE problem-based CRYSTALS-Kyber and CRYSTALS-Dilithium standards were chosen as the initial standards for KEMs and signature schemes, respectively [21]. However, even if the existing lattice-based KEM/PKE methods are shown to be secure against conventional mathematical crypt-analysis techniques, there is still a possibility that they could be deciphered using a variety of side-channel attacks (SCA).

In this section, we first describe the vulnerabilities and side-channel attacks on current PQC algorithms. Next, we briefly outline some possible countermeasures to protect the PQC algorithms against these vulnerabilities.

A. Side-Channel Attacks on PQC Algorithms

In side-channel attacks, security assets can be revealed by analyzing physical properties of hardware that the cryptographic applications operate on. In particular, power and electromagnetic radiation leakages have been used to extract secret information [22], [23]. These leakages also pose the greatest threat to the PQC implementations; thereby necessitating further investigation and mitigation strategies.

CRYSTALS-Kyber [24] is a cryptographically IND-CCA2-secure algorithm, indicating that it is not identifiable when subjected to an adaptable chosen ciphertext attack (CCA). The difficulty of the module learning with errors problem, often known as the Mod-LWE problem, is critical to the safety of the CRYSTALS-Kyber system. Based on a post-quantum version of the Fujisaki-Okamoto transform [25], Kyber includes a chosen plaintext attack (CPA)-secure PKE scheme (KYBER.CPAPKE) and a CCA-secure KEM scheme (KYBER.CCAKEM). However, recent investigations demonstrate that Kyber KEM has been subjected to a range of vulnerabilities, enabling a wide variety of side-channel attacks. These attacks depend on their attack methodologies, target modules, and the operating mode of CRYSTALS-Kyber.

Deep learning-based power side channel analysis was utilized by Ji et al. to demonstrate an effective message recovery attack on CRYSTALS-Kyber (Kyber768, parameter k = 3) [26]. They used Xilinx Artix-7 FPGA device to perform the power side-channel analysis. A message recovery from a successfully generated ciphertext in the CRYSTALS-Kyber key encapsulation mechanism (KEM) automatically implies the recovery of the session key since the session key is extracted from the message through the use of hash functions. The disclosed attack is possible by means of an expansion of the multi-bit error injection technique first introduced in [27] for side-channel analysis of software implementations of lattice-based PKE and KEM: the sliced multi-bit error injection technique. Because software implementations carry out their instructions in a sequential fashion, slicing is unnecessary. Hardware implementations, however, require slicing. Only 10% of messages may be retrieved with enumeration up to 2^{64} without using slicing. Ji et al. proved that the same enumeration may be used to recover all messages after slicing in the hardware implementation of CRYSTALS-Kyber [26].

Polynomial multiplication algorithms, including the number theoretic transform (NTT) algorithm and the Toom-Cook algorithm, which serve as the cornerstones of lattice-based PQC, are also susceptible to a variety of side-channel attacks. Before the results are interpolated, both of these polynomial multiplication schemes undertake point-wise multiplication at the threshold level. As a result, the secret parameters are implicated in these scalar multiplications [28], [29]. The attacker can try a divide-and-conquer strategy in the CPA attack to retrieve the secret key. By providing some correct or valid ciphertexts as inputs in the decapsulation algorithm, Mujdei et al. retrieved all of the NTT coefficients for CRYSTALS-Kyber with only 200 traces [30].

Pessl et al. [29] have effectively presented a sophisticated single-trace attack on the NTT multiplication algorithm of the software implementation of Crystal Kyber (operating on an Arm Cortex-M4 microprocessor). Single-trace key recovery can be achieved via sub-traces corresponding to the target multiplications in a single trace because a single trace can involve numerous multiplications of an intended subkey. Moreover, Primas et al. [28] have shown the whole private key can be successfully extracted from the NTT operation during the ring lattice LWE decryption process.

The attackers can not only target the multiplication algorithms but also try to attack the message encoding functions of the encryption operation of PQC algorithms. A secret message generated at random can be revealed by attacking the message encoding algorithm during the encapsulation phase. The retrieved message and public key can be used to create a shared temporary session key.

When a bit message is 0, the coefficient of the polynomial is encoded into 0x0000, and when a bit message is 1, the coefficient of the polynomial is encoded into 0xFFFF. This is how the message encoding function works in the CRYSTALS-Kyber algorithm. Since power usage depends on the Hamming weight of the encoded coefficient, each message can be seen using power/EM signatures. Since the target message is randomly produced each time, this attack scenario can only employ a single trace. Sim et al. applied CRYSTALS-Kyber and Saber SW implementations on an Arm Cortex-M4 core to successfully complete a 100% single-trace attack to restore the message [31].

Along with the message encoding functions, vulnerabilities have been found in message recovery decoding functions. By attacking message-recovery decoding functions, Xu et al. presented power side-channel assaults on CRYSTALS-Kyber to retrieve the entire secret key with fewer than 960 traces [32].

Besides the CRYSTALS-Kyber algorithm, side channel vulnerabilities have been found on other lattice-based PQC algorithms, such as Saber, NTRU, NewHope KEM, etc. Using EM side-channel assisted CCAs with templates to categorize a single bit/byte message, Ravi et al. [33] successfully retrieved the secret key for Kyber KEM, NewHope KEM, Saber KEM, and Round5 PKE running on an ARM Cortex-M4 core. Moreover, the private key could be recovered by Soojung et al. [34] on both NTRU implementations (e.g., NTRUEncrypt and NTRU Open Source) by leveraging a single power usage discovered during the decryption.

By drastically altering Hamming weights based on particular secret key components, Askeland et al. [35] took advantage of NTRU leakage that the target processor caused when handling information. They identified a single-trace side-channel approach that successfully recovered a sizable portion of the secret key, and later lattice reduction techniques extracted the remaining pieces. Ngo et al. [36] demonstrated that masked implementation of IND-CCA-safe Saber KEM is also susceptible to side-channel attacks. They provided a power analysis method based on deep learning to extract the persistent secret key as well as the session key from a small set of traces.

B. Possible Countermeasures for Side-Channel Attacks on PQC Algorithms

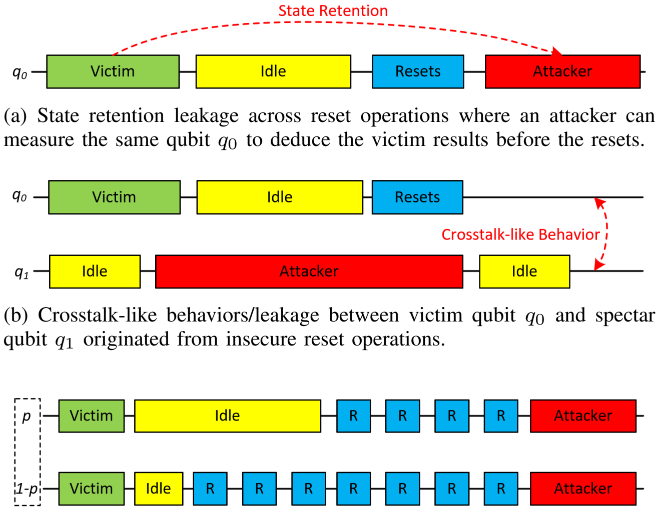
PQC algorithms are susceptible to various side-channel attacks, thereby compromising the security of quantum computing systems. Before implementing PQC algorithms in a quantum device, countermeasures must be taken to thwart side-channel attacks. A number of masking techniques have been proposed to prevent side-channel attacks. However, none of the methods have yet proven to be fully effective [37], [38]. Recent studies have discussed effective solutions to discover the leaky modules at the pre-silicon stage [39]–[41], thereby reducing the likelihood of side-channel leakage at the post-silicon stage by allowing the designer to make changes at the RTL level. These approaches analyzed Saber algorithm and demonstrated that multiplication modules (e.g., polynomial multiplication and vector multiplication) are vulnerable [39], [40]. Future research could focus on the pre-silicon analysis of the other PQC algorithms in order to secure these algorithms against side-channel attacks.

IV. Vulnerabilities in Quantum Devices

In this section, we discuss security vulnerabilities in quantum computer systems. Quantum cybersecurity is an emerging field. We highlight two works on insecure reset operations and power side-channel attacks which have been demonstrated on real-world systems.

A. *Insecure Reset Operations in Quantum Infrastructure*

The reset operation is an integral part of cloud-based quantum computing infrastructure in between circuit shots to erase qubit status. Mi et al. [42] explore the security vulnerabilities of reset operations in the real-world IBM Quantum cloud. It assumes that the prevalent multi-tenant user model is on its way to the Quantum cloud in the foreseeable future where multiple users can share the Quantum computing resources on mutually disjoint sets of qubits at flexible time slots. Under this assumption, the existing system-level wipe, i.e., all qubits need to be cleared at the same time, cannot suffice the requirements of resetting individual qubits without interrupting the computations on others. Besides, the security vulnerabilities of reset operations may lead to a variety of information leakage [42] which are illustrated in Figure 3.



(a) State retention leakage across reset operations where an attacker can measure the same qubit q_0 to deduce the victim results before the resets.

(b) Crosstalk-like behaviors/leakage between victim qubit q_0 and spectar qubit q_1 originated from insecure reset operations.

(c) Example secure reset mechanisms where a random variable controls how many resets are applied at a specific shot, e.g., 4 and 7 resets at probability p and $1-p$, respectively.

Fig. 3: Security vulnerabilities of reset operations in cloud-based quantum computing infrastructure and proposed security mechanisms in [42].

A reset operation typically consists of two parts, i.e., a measurement operation followed by a conditional X gate. As stated in Section II, the status $|\psi\rangle$ of a qubit will collapse to a deterministic classic binary bit **0** or **1** after measurements [43]. If the classic bit is **1**, the X quantum gate, similar to *NOT* gate in digital circuitry, will be invoked to flip $|1\rangle$ to $|0\rangle$; otherwise, the X gate is bypassed. Ideally, the output of a reset operation is always $|0\rangle$ whereas it is not perfect in real-world scenarios as discussed in [42].

As depicted in Figure 3(a), there is state retention leakage which can be measured by the adversaries operating the same qubit q_0 as the victim even if the reset operation(s) occur between the sessions. The root cause is the existing reset solution cannot fully eliminate the victim's results. As quantified in [42], there still remains around 5% remnant which can be exploited by attackers as information leakage even after multiple times of resets.

Figure 3(b) presents the crosstalk-like behavior between the victim qubit q_0 and its adjacent attacker qubit q_1. It has been found that the reset or other operations on q_0 can affect the measurement results on q_1 to some extent. In light of this, the adversaries can figure out two important pieces of timing information, i.e., the duration between victim qubit initialization and its last measurement as well as the duration between the last victim measurement operation and the end of the victim computing session. The timing information can effectively assist in inferring the output distribution of victims by estimating how many reset operations have been applied in the victim program. In order to secure reset operations, [42] also proposes a mechanism based on existing (insecure) reset where a random variable controls how many resets are applied for a specific shot. As shown in Figure 3(c), 4 and 7 reset operations can be inserted at the end of victim circuits at the probability of p and $1-p$, respectively, which can significantly eliminate the statistical distance (leakage) between victim cases.

B. *Power Side-channel Attacks on Quantum Computers*

Power side-channel attacks are common threats in classical platforms as introduced in Section II-C. Xu et al. [44] further extends the scope of victims to quantum computer systems. As shown in Figure 2, the physical quantum computer is controlled and driven by low-level control pulses, i.e., the signals generated by modulating the circuit shot waveform on a low-noise high-frequency microwave. In other words, the control pulses encode the quantum circuit design and intellectual property (IP) of local users/designers which may be reconstructed by malicious insiders within the cloud service providers through the power side channel. In fact, the target victim platform in [44] is the microwave control pulse generator instead of the quantum computer. Therefore, the attack is essentially focusing on conventional instruments like arbitrary waveform generator (AWG) and modulator. However, the asset control pulses are highly relevant to quantum circuits.

The input to the microwave control pulse generator is the pulse-level circuit that contains all pulses to be performed to fulfill a quantum program on the backend qubits. The AWG will produce the corresponding circuit shot signals accordingly and modulate them on a high-frequency carrier wave. In this procedure, the pulse generation consumes energy and thus establishes a power side channel that can be gauged by malicious insiders who possess physical access. Specifically, the side-channel profiles can be measured as per-shot timing, per-shot power, per-channel power, total power, and single power-related statistics under different assumptions of adversarial capabilities. Multiple attack cases can be enabled, such as determining the specific circuit from a given set and finding unknown circuit oracles. Furthermore, a completely unknown circuit may be reconstructed from power traces, as only a limited number of basis gates/pulses are supposed to be utilized, thus greatly reducing the search space.

V. Intrinsically Secure Quantum Devices

The sensitivity of quantum systems to environmental factors and noise has significant implications for the security of quantum devices. Noise and environmental sensitivity can introduce errors in the transmitted quantum states, leading to data leakage and an increased risk of eavesdropping. We first survey mitigation strategies. Next, we outline how measurement-induced steering can be useful for isolation.

A. Mitigation Strategies

To address the challenges posed by environmental sensitivity and noise, researchers employ various techniques to minimize their impact on quantum devices. These strategies include:

1) Physical isolation and shielding: Quantum systems can be isolated from external sources of noise through the use of shielding materials, such as superconducting enclosures, and the implementation of vibration isolation platforms.
2) Cryogenic cooling: Maintaining quantum devices at ultra-low temperatures can suppress thermal fluctuations and reduce the effects of noise on the quantum states.
3) Quantum error correction: Developing error correction techniques that can detect and correct errors introduced by noise and environmental factors is essential for maintaining the security and performance of quantum devices.
4) Fault-tolerant quantum computing: The design of quantum algorithms and circuits that are inherently robust to noise and errors can help ensure the security and accuracy of quantum computations, even in the presence of environmental sensitivity.

By understanding the nature of noise factors and developing strategies to mitigate their effects, researchers can advance the development of secure and reliable quantum technologies.

B. Security by Steering

Following mitigation by isolation, measurement-induced steering of quantum systems [45], a phenomenon closely related to entanglement, offers a unique approach to enhancing the security of quantum devices. In a quantum steering scenario, the entanglement between an exposed detector and a protected system qubit (or qubits) can be exploited to remotely control the quantum state of the system qubit while keeping it isolated from the environment. The steering process allows a trusted party to manipulate the quantum state of the protected system qubit by performing measurements on the entangled exposed detector. By observing the correlations between the measurement outcomes of the detector and the system qubit, the trusted party can assess the security of the quantum system and detect potential eavesdropping or tampering attempts. In this way, quantum steering provides a robust method for verifying the integrity of quantum devices and communication protocols, ultimately enhancing their security against potential attacks while maintaining the isolation of the system qubits from the environment.

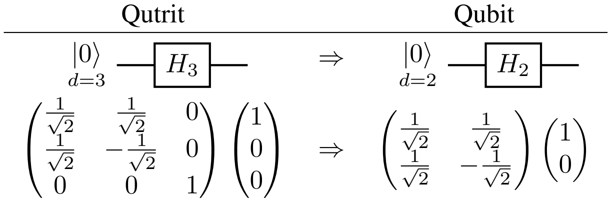

Qutrit
Qubit
$|0\rangle_{d=3}$
H_3
$\Rightarrow$
$|0\rangle_{d=2}$
H_2
$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$
$\Rightarrow$
$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

Fig. 4: Example qutrit gate (H_3) that can be trivially compressed into a qubit gate (H_2) as done in [47].

However, measurements of a detector pose a potential security flaw, as a motivated actor can intercept the readouts to gain information about the state of the entangled system qubit, as noted in Section IV-A. This vulnerability highlights the need for alternative methods to maintain the security of the quantum system without revealing sensitive information through detector readouts. One such recent approach involves a protocol that can prepare the desired quantum state of the system qubit based on the average of readout outcomes – revealing no information about the system. By doing so, the trusted party can still exploit quantum steering to control the system qubit, while minimizing the risk of information leakage to potential adversaries. Steering to an arbitrary state has been recently demonstrated on contemporary cloud-accessible quantum devices [46].

VI. Role of Quantum Compilers in Securing Quantum Devices

Quantum compilers play a critical role in the design of quantum algorithms, as they translate high-level quantum programs into low-level quantum circuits that can be executed on quantum hardware. The low-level circuits are then further translated into electromagnetic pulses that are sent to the qubit. In this section, we discuss the potential of quantum compilers to enhance the security of quantum devices by optimizing circuit implementations, mitigating leakage, and providing fault-tolerant designs.

A. Optimizing Circuit Implementations

Quantum compilers can optimize circuit implementations in different ways, such as by minimizing the number of gates and qubits required to perform a given quantum operation. This optimization process can enhance the security of quantum devices in several ways:

1) Reducing the circuit depth: The compiler reduces the number of gates in a circuit and thereby reducing the

overall circuit depth, leading to a shorter execution time [48]. This reduction in execution time minimizes the window of opportunity for potential attackers to exploit environmental noise or introduce malicious errors into the system.

2) Minimizing the number of qubits: The compiler may find potential reductions in the number of qubits [47]. Reducing the number of qubits required for executing a quantum algorithm can help mitigate the effects of noise and decoherence, as fewer qubits need to be protected and maintained in a coherent state. This reduction in qubits can also lower the risk of eavesdropping or tampering, as each additional qubit provides a potential target for adversaries.

In general, these optimizations amount to reducing the dimensionality of the quantum operations, as demonstrated in Figure 4. Such optimizations may be performed at the circuit level, or at the pulse level as discussed in [47]. We briefly outline two popular compilation strategies.

1) Cosine-Sine Decomposition (CSD) Algorithm: One of the main objectives of the compiler is to rewrite any arbitrary unitary, or quantum gate, to an approximately equivalent quantum circuit which is constructed only out of a finite set of gates. Algorithm 1 shows major steps in Cosine-Sine Decomposition (CSD). Consider a qubit system which is acted on by a unitary matrix U of size $2^N \times 2^N$. CSD will produce the following:

$$U = diag(L_1, L_2) \begin{pmatrix} C & -S \\ S & C \end{pmatrix} diag(R_1, R_2) \qquad (4)$$

where L_1, L_2, R_1, R_2 are block matrices of size $2^{N-1} \times 2^{N-1}$. C and S are $\cos\vec{\theta}$ and $\sin\vec{\theta}$ respectively, where $\vec{\theta}$ is given by the CSD.

Algorithm 1: Cosine-Sine Decomposition

```
Function CSD (Unitary Matrix U, dim d):
    $n \longleftarrow \log_d(U.size)$
    $m_0 \longleftarrow d^n$
    $r_0 \longleftarrow d^{n-1}$
    while $1 <= j <= d - 1$ do
        while Submatrices remaining do
            $CSD(U_i^{(j)}, m_{j-1}, r_{j-1})$
            $m_j = m_{j-1} - r_0$
            $r_j = r_{j-1}$
        end
    end
    Combine all Matrices
End Function
```

2) Solovay-Kitaev (SK) Algorithm: The CSD algorithm discussed in the previous section is an efficient heuristics, but is not guaranteed to produce accurate results. In contrast, Solovay-Kitaev algorithm is optimal. In other words, it is guaranteed to be ϵ-close to the expected output, given a desired error ϵ. However, Solovay-Kitaev algorithm inherently utilizes tree-like structure, and although significantly limits the search space by exploiting algebraic properties, is slower than the CSD algorithm.

Algorithm 2: Solovay-Kitaev Algorithm

```
Function Solovay-Kitaev(Gate U, depth n):
    if $n == 0$ then
        return Basic Approximation of $U$
    else
        $U_{n-1} \longleftarrow$ Solovay-Kitaev$(U, n-1)$
        $V, W \longleftarrow$ Approx-Decompose$(U, n-1)$
        $V_{n-1} \longleftarrow$ Solovay-Kitaev$(V, n-1)$
        $W_{n-1} \longleftarrow$ Solovay-Kitaev$(W, n-1)$
        return $U_n = V_{n-1} W_{n-1} V_{n-1}^\dagger W_{n-1}^\dagger U_{n-1}$
    end
End Function
```

Algorithm 2 shows the major steps in the Solovay-Kitaev algorithm. It uses the observation that for an accuracy of $\epsilon > 0$, a sequence of gates that approximate the unitary can be generated in $O(\log^c(1/\epsilon))$. The underlying strategy is to start at an arbitrary approximation, which can be stored to a table ahead of time. Then, by utilizing the properties of $SU(d)$, keep applying transformation that drives the operation to a closer approximation until a circuit depth of n is reached.

Although theoretically efficient, in practice the Solovay-Kitaev algorithm suffers from large runtime primarily due to the iterative search structure. For this reason, methods such as CSD are more common due to their faster runtime despite them not producing the most optimal solution. Moreover, methods such as CSD exploit stable and well studied matrix decompositions.

B. Mitigating Leakage

Quantum compilers can also be designed to mitigate the effects of quantum leakage, a phenomenon where qubits unintentionally transition to states outside the computational basis. Leakage can severely degrade the performance of quantum algorithms and increase their susceptibility to errors. By employing leakage reduction techniques, such as utilizing leakage-resilient gates or encoding the quantum information in a protected subspace, a quantum compiler can minimize the impact of leakage.

C. Pulse-level Compilation

While high-level quantum compilers translate quantum programs into quantum circuits composed of gates, there exists another level of compilation that focuses on the pulse level. Pulse-level quantum compilation deals with the translation of gate-based quantum circuits into precise control pulses that directly manipulate the underlying quantum hardware. For example, pulse shaping involves designing the appropriate waveforms for control pulses, which are used to manipulate the quantum states of qubits and their interactions. These control pulses must be tailored to the specific hardware platform, taking into account the unique characteristics of the qubits, such as their energy levels, transition frequencies, and coupling strengths. Pulse-level quantum compilers must generate control pulses that accurately implement the desired quantum operations while minimizing the adverse effects of noise, control errors, and crosstalk between qubits.

However, before appropriate waveforms may be developed it is essential to calibrate control pulses. Calibration involves adjusting the amplitude, phase, and timing of control pulses to optimize their performance on the quantum hardware. This process typically requires iterative feedback loops, comparing the results of quantum operations with the desired outcomes and fine-tuning the control parameters accordingly. Accurate calibration is crucial for mitigating systematic errors and ensuring the reliable execution of quantum algorithms on physical hardware.

Quantum compilation can also play a role in error mitigation at the pulse level. By designing robust control pulses and incorporating error-mitigation techniques, such as dynamical decoupling, composite pulses, or optimal control methods, pulse-level quantum compilers can minimize the impact of noise and control errors on quantum computation. These strategies can help enhance the fidelity of quantum operations and improve the overall performance of the quantum hardware, even in the presence of environmental noise and imperfections in the control pulses.

VII. Conclusion

This paper provided an overview of the security concerns related to classical-quantum systems and discussed effective countermeasures. We have explored topics such as the security of classical communication with quantum devices, the importance of post-quantum cryptography, security flaws and mitigation strategies in post-quantum cryptography, environmental sensitivity leading to security faults on quantum devices themselves, a mitigation strategy via an isolation mechanism using entanglement, and the role of quantum compilers for optimizing security guarantees.

As we progress towards harnessing the full potential of quantum computing, it is essential to maintain a holistic perspective that encompasses the entire quantum system, ensuring the safe and effective operation of these integrated systems. This ongoing effort includes securing classical communication channels with quantum devices to protect against eavesdropping or tampering, employing quantum key distribution and other cryptographic techniques, and leveraging post-quantum cryptography to safeguard the exchange of sensitive information when sending data to quantum devices on the cloud.

By addressing these critical security concerns and continually developing strategies to mitigate potential risks, we contribute to building a more secure and reliable foundation for the future of quantum computing. By tackling these challenges, we can pave the way for further advancements in quantum technologies and their applications across various fields, ultimately enhancing the security, reliability, and efficiency of integrated classical-quantum systems.

Acknowledgments

This work was partially supported by the National Science Foundation (NSF) grant CCF-1908131.

References

[1] Matthew Sparkes. Google demonstrates vital step towards large-scale quantum computers. *New Scientist*, 2021.

[2] Daniel Volya and Prabhat Mishra. Quantum spectral clustering of mixed graphs. In *ACM/IEEE Design Automation Conference (DAC)*, pages 463–468, 2021.

[3] Daniel Volya and Prabhat Mishra. Impact of noise on quantum algorithms in noisy intermediate-scale quantum systems. In *IEEE International Conference on Computer Design (ICCD)*, 2020.

[4] Daniel Volya and Prabhat Mishra. Modeling of noisy quantum circuits using random matrix theory. In *IEEE International Conference on Computer Design (ICCD)*, pages 132–138, 2022.

[5] Paul C Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *CRYPTO*, pages 104–113, 1996.

[6] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *CRYPTO*, pages 388–397. Springer, 1999.

[7] Dakshi Agrawal, Bruce Archambeault, Josyula R Rao, and Pankaj Rohatgi. The EM Side—channel(s). In *Cryptographic Hardware and Embedded Systems(CHES)*, pages 29–45. Springer, 2003.

[8] Mark Zhao and G Edward Suh. FPGA-based Remote Power Side-channel Attacks. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 229–244. IEEE, 2018.

[9] Falk Schellenberg, Dennis RE Gnad, Amir Moradi, and Mehdi B Tahoori. An Inside Job: Remote Power Analysis Attacks on FPGAs. *IEEE Design & Test*, 38(3):58–66, 2021.

[10] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, et al. Spectre Attacks: Exploiting Speculative Execution. *Communications of the ACM*, 63(7):93–101, 2020.

[11] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, et al. Meltdown: Reading Kernel Memory from User Space. *Communications of the ACM*, 63(6):46–56, 2020.

[12] Hasini Witharana and Prabhat Mishra. Speculative load forwarding attack on modern processors. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2022.

[13] Some Notes on Meltdown and Spectre. [Online]. https://blog.f-secure.com/some-notes-on-meltdown-and-spectre/, Accessed Apr. 20, 2023.

[14] Anubhab Baksi, Shivam Bhasin, Jakub Breier, Dirmanto Jap, and Dhiman Saha. A Survey on Fault Attacks on Symmetric Key Cryptosystems. *ACM Computing Surveys*, 55(4):1–34, 2022.

[15] Tao Zhang, Md Latifur Rahman, Hadi Mardani Kamali, Kimia Zamiri Azar, Mark Tehranipoor, and Farimah Farahmandi. FISHI: Fault Injection Detection in Secure Heterogeneous Integration via Power Noise Variation. In *2023 IEEE 73rd Electronic Components and Technology Conference (ECTC)*. IEEE, 2023.

[16] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors. *ACM SIGARCH Computer Architecture News*, 42(3):361–372, 2014.

[17] Yuan Xiao, Xiaokuan Zhang, Yinqian Zhang, and Radu Teodorescu. One Bit Flips, One Cloud Flops: Cross-VM Row Hammer Attacks and Privilege Escalation. In *USENIX Security Symposium*, pages 19–35, 2016.

[18] Pawel Swierczynski, Georg T Becker, Amir Moradi, and Christof Paar. Bitstream Fault Injections (BiFI)–Automated Fault Attacks against SRAM-based FPGAs. *IEEE Transactions on Computers*, 67(3):348–360, 2017.

[19] Tao Zhang, Mark Tehranipoor, and Farimah Farahmandi. BitFREE: On Significant Speedup and Security Applications of FPGA Bitstream Format Reverse Engineering. In *2023 IEEE European Test Symposium (ETS)*. IEEE, 2023.

[20] Tao Zhang, Jian Wang, Shize Guo, and Zhe Chen. A Comprehensive FPGA Reverse Engineering Tool-chain: From Bitstream to RTL Code. *IEEE Access*, 7:38379–38389, 2019.

[21] https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4.

[22] Jungmin Park, Xiaolin Xu, Yier Jin, Domenic Forte, and Mark Tehranipoor. Power-based side-channel instruction-level disassembler. In *Proceedings of the 55th Annual Design Automation Conference*, pages 1–6, 2018.

[23] Yunkai Bai, Andrew Stern, Jungmin Park, Mark Tehranipoor, and Domenic Forte. Rascv2: Enabling remote access to side-channels for mission critical and iot systems. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 27(6):1–25, 2022.

[24] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber algorithm specifications and supporting documentation. *NIST PQC Round*, 2(4):1–43, 2019.

[25] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO*, pages 537–554. Springer, 1999.

[26] Yanning Ji, Ruize Wang, Kalle Ngo, Elena Dubrova, and Linus Backlund. A side-channel attack on a hardware implementation of crystals-kyber. Cryptology ePrint Archive, Paper 2022/1452, 2022. https://eprint.iacr.org/2022/1452.

[27] Ruize Wang, Kalle Ngo, and Elena Dubrova. A message recovery attack on lwe/lwr-based pke/kems using amplitude-modulated em emanations. In *ICISC*, pages 450–471. Springer, 2023.

[28] Robert Primas, Peter Pessl, and Stefan Mangard. Single-trace side-channel attacks on masked lattice-based encryption. In *Cryptographic Hardware and Embedded Systems (CHES)*, pages 513–533. Springer, 2017.

[29] Peter Pessl and Robert Primas. More practical single-trace attacks on the number theoretic transform. In *LATINCRYPT*, pages 130–149. Springer, 2019.

[30] Catinca Mujdei, Lennert Wouters, Angshuman Karmakar, Arthur Beckers, Jose Maria Bermudo Mera, and Ingrid Verbauwhede. Side-channel analysis of lattice-based post-quantum cryptography: Exploiting polynomial multiplication. *ACM Transactions on Embedded Computing Systems*, 2022.

[31] Bo-Yeon Sim, Jihoon Kwon, Joohee Lee, Il-Ju Kim, Tae-Ho Lee, Jaeseung Han, Hyojin Yoon, Jihoon Cho, and Dong-Guk Han. Single-trace attacks on message encoding in lattice-based kems. *IEEE Access*, 8:183175–183191, 2020.

[32] Zhuang Xu, Owen Pemberton, Sujoy Sinha Roy, David Oswald, Wang Yao, and Zhiming Zheng. Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of kyber. *IEEE Transactions on Computers*, 71(9):2163–2176, 2021.

[33] Prasanna Ravi, Shivam Bhasin, Sujoy Sinha Roy, and Anupam Chattopadhyay. Drop by drop you break the rock-exploiting generic vulnerabilities in lattice-based pke/kems using em-based physical attacks. *IACR Cryptol. ePrint Arch.*, 2020:549, 2020.

[34] Soojung An, Suhri Kim, Sunghyun Jin, HanBit Kim, and HeeSeok Kim. Single trace side channel analysis on ntru implementation. *Applied Sciences*, 8(11):2014, 2018.

[35] Amund Askeland and Sondre Rønjom. A side-channel assisted attack on ntru. Cryptology ePrint Archive, Paper 2021/790, 2021. https://eprint.iacr.org/2021/790.

[36] Kalle Ngo, Elena Dubrova, Qian Guo, and Thomas Johansson. A side-channel attack on a masked ind-cca secure saber kem implementation. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 676–707, 2021.

[37] Joppe W Bos, Marc Gourjon, Joost Renes, Tobias Schneider, and Christine Van Vredendaal. Masking kyber: First-and higher-order implementations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 173–214, 2021.

[38] Nils Paulsrud. A side channel attack on a higher-order masked software implementation of saber, 2022.

[39] Jungmin Park, N Nalla Anandakumar, Dipayan Saha, Dhwani Mehta, Nitin Pundir, Fahim Rahman, Farimah Farahmandi, and Mark M Tehranipoor. Pqc-sep: Power side-channel evaluation platform for post-quantum cryptography algorithms. *IACR Cryptol. ePrint Arch.*, 2022:527, 2022.

[40] Nitin Pundir, Jungmin Park, Farimah Farahmandi, and Mark Tehranipoor. Power side-channel leakage assessment framework at register-transfer level. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 30(9):1207–1218, 2022.

[41] Tao Zhang, Jungmin Park, Mark Tehranipoor, and Farimah Farahmandi. Psc-tg: Rtl power side-channel leakage assessment with test pattern generation. In *ACM/IEEE Design Automation Conference (DAC)*, pages 709–714, 2021.

[42] Allen Mi, Shuwen Deng, and Jakub Szefer. Securing Reset Operations in NISQ Quantum Computers. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2279–2293, 2022.

[43] Zachery Utt, Daniel Volya, and Prabhat Mishra. Quantum measurement discrimination using cumulative distribution functions. In *Design Automation and Test in Europe (DATE)*, 2023.

[44] Chuanqi Xu, Ferhat Erata, and Jakub Szefer. Exploration of Quantum Computer Power Side-Channels. *arXiv preprint arXiv:2304.03315*, 2023.

[45] Sthitadhi Roy, J. T. Chalker, I. V. Gornyi, and Yuval Gefen. Measurement-induced steering of quantum systems. *Phys. Rev. Research*, 2(3):033347, September 2020.

[46] Daniel Volya and Prabhat Mishra. State Preparation on Quantum Computers via Quantum Steering. *arXiv preprint:2302.13518*, March 2023.

[47] Daniel Volya and Prabhat Mishra. Quantum Data Compression for Efficient Generation of Control Pulses. In *Asia and South Pacific Design Automation Conference (ASPDAC)*, 2023.

[48] Daniel Volya and Prabhat Mishra. Qudcom: Towards quantum compilation for qudit systems. In *arXiv*, April 2023.