Recovering QSVT Polynomials from Side-Channel Information on Quantum Computers

Kidus Tessma
Northwestern University
Evanston, IL, USA
kidustessma2027@u.northwestern.edu

Hrvoje Kukina
TU Wien
Vienna, Austria
hrvoje.kukina@student.tuwien.ac.at

Jakub Szefer
Northwestern University
Evanston, IL, USA
jakub.szefer@northwestern.edu

***Abstract*—Quantum Singular Value Transformation (QSVT) is a powerful framework that can be applied across a wide range of quantum applications, including solving linear systems of equations, phase estimation, or amplitude amplification as employed in Grover's search algorithm. QSVT is in effect a meta-algorithm that can be used to realize various other functionalities or applications. QSVT is typically configured with different polynomials to realize the different functions, and the polynomials are in turn encoded into quantum gate operations that execute on the quantum computer. If an adversary is able to recover the gate operations, specifically gate rotations and their angles, from the quantum computer, they can then attempt to recover the polynomial used, and from the polynomial they can attempt to recover the function that the victim is executing. This paper evaluates different functions implemented in QSVT and how they map to different polynomials and the phase angles. It focuses on the correlation between the phase angle values and the number of phase angles for the different functions implemented using QSVT. The paper shows that knowing the phase angle values, or even just the number of phase angles, something an attacker can learn through a side channel, allows an attacker to learn the type of functionality being implemented by QSVT.**

***Index Terms*—quantum computing, side channels, QSVT**

I. Introduction

In the current Noisy Intermediate-Scale Quantum (NISQ) era, quantum hardware is mainly accessed through cloud-based platforms. Leading providers, including IBM Quantum [1], Amazon Braket [2], or Microsoft Azure Quantum [3], provide remote execution services on their Quantum Processing Units (QPUs). In addition to major cloud providers giving access to quantum computers, many quantum computer developers themselves have built their own cloud access, for example, Quantinuum built Quantinuum Nexus [4] to give access to their QPUs, or IQM built IQM Resonance [5] to give access to their QPUs. In all the settings, QPUs operate under control of the cloud providers and are multiplexed among different users. This remote, shared-access model, while enabling broad access, also introduces novel security considerations.

Emerging research has begun to explore various side-channels in quantum computers, especially when considering such cloud-based quantum computers. Broadly, research has explored side-channels in two directions. First, research has focused on side channels in quantum computer controllers [6], [7]. A malicious cloud provider or an insider could collect power traces from the controllers to attempt to recover the quantum gate operations being executed. Second, research has focused on side channels realized via crosstalk within QPUs [8]. An attacker co-located with the victim learn the two-qubit gate operations being executed by the victim, for example. The second direction assumes multi-tenant quantum computers, not deployed today, but widely researched.

[†] This work was supported in part by NSF grant 2332406.

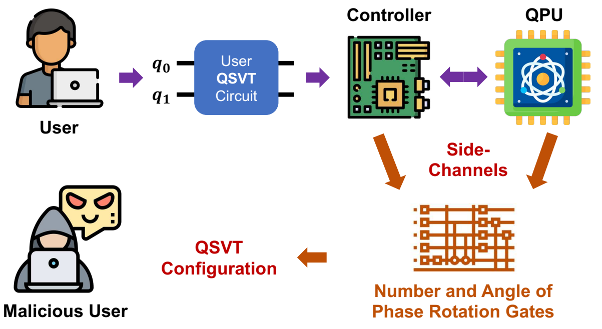

Fig. 1. Overview of the threat model and attack on QSVT algorithm: user submits a QSVT circuit to the quantum computer, while attacker may use side channels in controller or QPU to try to learn about the gates being executed and then the QSVT configuration information.

Considering all the side-channel threats in quantum computers, this work explores how such threats could affect the Quantum Singular Value Transformation (QSVT) algorithm executed on remote, cloud-based quantum computers. The overview of the threat is shown in Figure 1. QSVT does not solve a single problem, rather it is a general framework that realizes many algorithms by choosing an appropriate target polynomial within its configuration. Different configurations, i.e. different polynomials, are used to implement different functionality. The polynomials are in turn encoded into quantum gate operations that execute on the quantum computer. If an adversary is able to recover the gate operations from the quantum computer via a side-channel, they can then aim to recover the polynomial used, and from the polynomial they can aim recover the functionality or algorithm. Unique feature of QSVT is that the structure of the algorithm is fixed for all the different functionalities, while the only difference is different number and type of rotation gates (corresponding to different polynomials). This can simplify potential security attacks as the attacker only has to learn about the rotation gates and their angles from the side channels, as discussed later in this paper.

A. Contributions

The contributions of this work are:

- This work is the first study to consider how side channels can be abused against QSVT algorithm.
- This work analyzes three different polynomials used with QSVT: one for matrix inversion, one for cubic function, and one for Grover's amplitude amplification function.
- This work analyzes the rotation angle values and their number and how they are related to the polynomials.
- This work further performs sensitivity study to understand how noisy side-channel (resulting in rotation angle values learned by the attacker containing some noise) could affect the attacker's efforts to learn the QSVT functionality.

II. Background on QSVT

QSVT is a powerful framework that enables quantum algorithms to apply polynomial transformations to the singular values of matrices embedded within unitary operators. By leveraging this technique, quantum computers can efficiently manipulate matrix properties such as eigenvalues and singular values with exponential advantages over classical approaches. This allows QSVT to be applied across a wide range of quantum applications, including for matrix inversion via $f(x) = 1/x$, phase estimation via $f(x) = sign(x)$, Hamiltonian simulation for approximating quantum time evolution, or amplitude amplification as employed in Grover's search algorithm, among others. This versatility makes QSVT a foundational tool in the development of advanced quantum algorithms.

The approximation degree is the primary driver of query complexity and gate count in QSVT. In Hamiltonian simulation, if H is provided via an $(\alpha, \cdot, \cdot)$ block-encoding, the necessary and sufficient number of block-encoding invocations to implement e^{-iHt} within error ε is

$$\Theta\left(\alpha|t| + \frac{\log(1/\varepsilon)}{\log\bigl(e + \log(1/\varepsilon)/(\alpha|t|)\bigr)}\right),$$

matching optimal dependence on t and ε up to iterated logarithms [9]. Within QSVT-based simulation, the choice of amplitude-amplification subroutine affects constants and ε-dependence: oblivious amplitude amplification (OAA) yields $O\bigl(\alpha|t| + \log(1/\varepsilon)\bigr)$ queries, while fixed-point amplitude amplification (FPAA) incurs $O\bigl(\alpha|t| \log(1/\varepsilon) + \log^2(1/\varepsilon)\bigr)$. Analyses and numerics consistently favor OAA in query count across a broad (t, ε) range [10]. On the approximation side, the boundedness constraint inherent to QSVT (polynomials must remain bounded on $[-1, 1]$) governs the attainable degree for target functions. A practical "bounded Chebyshev truncation" meta-theorem provides degree bounds of the form $O\bigl(\frac{b}{\delta} \log \frac{b}{\delta\varepsilon}\bigr)$ for piecewise-smooth functions on $[-1, 1]$ (after rescaling), directly translating into $O(d)$ query complexity with d given by the bound [11]. Moreover, for functions such as $\exp(\beta x)$ used in Gibbs-style subroutines, imposing boundedness may necessarily increase the required degree (and hence queries). The imposed boundedness in some regimes increases the required degree by nearly a quadratic factor relative to unconstrained approximation, which explains observed runtime overheads in QSVT-based optimization pipelines [11]. Recent generalized variants (GQSP/GQET/GQSVT) relax structural constraints on implementable polynomials (allowing complex coefficients and indefinite parity) while preserving the $O(d)$ query rule-of-thumb. Crucially, they improve the classical phase-finding cost from $\tilde{O}(d^2)$ in QSP/QSVT to nearly linear $\tilde{O}(d)$, which can dominate end-to-end runtime when many different polynomials are synthesized [12]. These generalized schemes may require down-scaling by a factor β relating the monomial- and Chebyshev-basis norms of the polynomial; β is at most $O(\log d)$ in general and can be bounded by 2 in certain structured cases, so scaling losses are modest in practice [12]. Overall, across standard and generalized frameworks, degree/accuracy tradeoffs (with boundedness), block-encoding normalization α, and amplification strategy together determine the concrete query/gate complexity of QSVT-style algorithms.

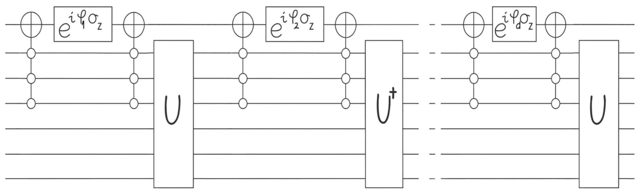

Fig. 2. Example of a QSVT circuit.

QSVT encodes the $f(x)$ functions through three main steps: it first block-encodes the target matrix $\boldsymbol{A}$ into a larger unitary U, then approximates the target function $f(x)$ using a polynomial, then converts it to a Chebyshev polynomial, and finally uses Quantum Signal Processing (QSP) to generate a sequence of controlled phase rotations, i.e. angles, to implement the polynomial transformation on the singular values. The QSP phase angles for the coefficients in the Chebyshev polynomial are computed using methods such as Laurent method [13]. Based on these phase angles, the resulting quantum circuit applies phase rotation gate (typically a Z-rotation). This is repeated for all the phase angles. Note that if the target function $f(x)$ is even, then only even Chebyshev terms appear; if the target function $f(x)$ is odd, then only odd Chebyshev terms appear. However, the number of phase rotations corresponds to the degree of the polynomial, and generally for d-degree polynomial there will be $d + 1$ angles; zero coefficient in the polynomial does not mean the corresponding phase rotation angle is zero. The final QSVT quantum circuit incorporates the sequence of phase rotation gates with other gates in-between.

If an attacker gains access to these phase rotation angles or their number, such as through side-channel analysis, they may be able to infer type of the underlying function. In this study, we explore the extent to which an attacker could recover function from leaked phase angles, and analyze how sensitive these angles are to small perturbations in $f(x)$. These perturbations are meant to represent noise in the side channel.

III. Threat Model

This work assumes an attacker has access to a side-channel information about the quantum circuit executing on the remote quantum computer. Existing work [6], [7], [8] has already demonstrated various side-channel attacks in controllers or QPUs them self. The attack setup is shown in Figure 1. From these side channels, we assume the attacker is able to get information about the types of gates being executed on the quantum computer, especially the phase angle rotations.

To launch the attack, we assume the attacker knows that the victim is executing a QSVT algorithm and thus knows the typical QSVT circuit structure, but he or she does not know specific target function $f(x)$ being implemented. The threat model in particular does not assume the attacker has access to the full polynomial representation, the Chebyshev coefficients, or direct measurements of the internal quantum state. All they can access is the phase rotations and their number through the side channel. The goal of the attacker is exactly to try to recover the type of the target function being executed from the side channel information about the phase rotations.

Due to noisy nature of side channels, we also consider case where the attacker does not have exact information about the quantum gates and the phase rotations, but has partial or noisy access to the phase angle sequence $\{\phi_0, \phi_1, \ldots, \phi_d\}$. We later simulate different noise levels in our experiments.

IV. Experimental Methodology

The experimental methodology in this work followed five steps in order to analyze vulnerability of QSVT to side-channel attacks:

(A) *Code Survey:* We identified publicly available QSVT implementations from repositories such as GitHub.

(B) *Function Approximation:* Target functions were approximated using Chebyshev polynomials, and their coefficients were extracted.

(C) *Noise Injection:* Controlled noise was introduced into the coefficients; modifying coefficients results in changes to the phase angles and this simulates noise in the side channel.

(D) *Phase Angle Computation:* The PyQSP library's `laurent` method was used to convert the Chebyshev polynomials (with coefficients with and without noise) into a phase angle sequence.

(E) *Phase Angle Comparison:* The sequences of perturbed phase angles were compared to the original sequences of phase angles to evaluate the impact of the perturbations.

A. Code Survey

We identified a small number of publicly available repositories with QSVT algorithms that could be executed. We mainly leveraged Classiq's code [14] from which we took the optimization formula and the adjusting function. However, we used the `pyqsp` library [15] for all three functions to generate the phase angle sequences.

B. Function Approximation

The functions evaluated in this work include matrix inversion ($f(x) = 1/x$), a cubic function ($f(x) = x^3$), and Grover's amplitude amplification function. Each function was represented in Chebyshev polynomial form.

For the matrix inversion case ($f(x) = 1/x$), we adapted an implementation from Classiq's library [16] which includes QSVT functionality that approximates the function over a bounded singular value interval $[\sigma_{\min}, \sigma_{\max}]$, typically constrained to $(0, 1]$. Due to the difficulty of approximating $1/x$ near zero, the method required a high-degree polynomial specifically, a 30-degree Chebyshev approximation. Classiq's approach uses convex optimization to minimize the maximum approximation error over the specified spectral interval, resulting in a high-fidelity polynomial tailored to the singular value distribution of the input matrix. Note that the inversion polynomial must be odd, since $1/x$ itself is an odd function and QSP only works with polynomials that respect this parity. When the input is a degree-30 polynomial, Classiq treats the input degree as an upper bound and automatically rounds it down to the nearest odd degree so that the approximation stays inside the odd-only constraint. The result is a degree-29 polynomial with corresponding 30 phase angle rotations; recall that for degree-d polynomial there will be $d+1$ angles.

For the cubic function case ($f(x) = x^3$), we constructed the polynomial exactly using its Chebyshev expansion. In particular,

$$f(x) = \tfrac{1}{4}\big(T_3(x) + 3T_1(x)\big),$$

so the target polynomial is degree-3 with nonzero coefficients only for the odd Chebyshev terms T_1 and T_3, which are:

$$T_1(x) = x$$

and

$$T_3(x) = 4x^3 - 3x$$

Because x^3 is already an odd function, it naturally satisfies the parity requirements of the Laurent-based QSP synthesis, and no adjustment to enforce parity is needed. Also, since the representation is exact and minimal, no convex optimization or high-degree approximation is required here. The QSP Laurent synthesis therefore produces a degree-3 polynomial, which corresponds to 4 phase rotations in the final sequence.

For the Grover's amplitude amplification function we use cubic Chebyshev basis function that has support only on the T_3 term, i.e.,

$$f(x) = T_3(x)$$

so the target polynomial is degree-3, and the T_3 term is:

$$T_3(x) = 4x^3 - 3x,$$

with coefficients $\{0, 0, 0, 1\}$ in the Chebyshev expansion basis. Because $T_3(x)$ is an odd polynomial, it is naturally compatible with the parity constraints required by the Laurent-based QSP synthesis. Similar to the x^3 case, the synthesis produces a degree-3 polynomial, resulting in 4 phase rotations in the final sequence.

TABLE I
FIRST FOUR PHASE ANGLES FOR THE TESTED ALGORITHMS.

Alg.	Angle 1	Angle 2	Angle 3	Angle 4
Matrix Inv.	4.9294	3.1011	3.1754	3.0951
Cubic	4.3741	2.0309	2.0309	-8.1923
Grover's	3.1591	3.1398	3.1398	-9.4073

TABLE II
CONFUSION MATRIX FOR CLASSIFICATION OF THE THREE ALGORITHMS.

True / Predicted	Matrix Inv.	Cubic	Grover's
Matrix Inv.	100%	0	0
Cubic	0	100%	0
Grover's	0	0	100%

C. Noise Injection

To simulate noisy side-channel information, noise was added to the coefficients. Section VI contains more details for the sensitivity study. In brief, we tested various magnitudes of noise from 1×10^{-1} to 1×10^{-4} introduced into the coefficients of the polynomials.

D. Phase Angle Computation

The Laurent method is standard in Quantum Signal Processing (QSP) for computing the phase angles that implement a target polynomial. Starting from a polynomial expressed in the Chebyshev basis, it recursively factorizes the associated unitary matrix into a sequence of single-qubit rotations (phase angles) interleaved with the signal operator. This method produces exact phase sequences that satisfy unitarity and parity constraints, enabling precise implementation of polynomials on quantum circuits without iterative optimization. It is implemented by PyQSP library's `laurent` method. This was done for the polynomials with coefficients with and without noise. Resulting in two different sets of angles for each polynomial, with and without noise respectively.

E. Phase Angle Comparison

The sequences of perturbed phase angles were compared to the original sequences of phase angles to evaluate the impact of the perturbations. A simple difference between the angle values was computed for this.

V. CLASSIFYING QSVT FUNCTIONALITY FROM PHASE ANGLE VALUES

To demonstrate the potential for side channels, we developed a very simple classifier. The key observation is that the phase rotation angles are unique to each of the function: matrix inversion, cubic function, and amplitude amplification from Grover's search. Since each function may contain different number of phases, we put in a limit that attacker has access to only first four phase angles. The first four angles for the three functions used in this work are shown in Table I.

The classifier simply computes the Euclidean distance between the input vector being tested and each of the three reference vectors. Each vector contains four phase angles. The output of the classifier is the class of the reference vector that is closest to the input being tested. We tested the classifier by adding small amounts of noise to the values of the reference vectors and using these as test inputs. More detailed study of the noise is presented in Section VI.

Table II shows the confusion matrix. It can be seen that with small changes in the values in the vectors, i.e. small changes to the rotation angles, the predicted class always matches the true class. This indicates that an attacker who has access to a small subset of the rotation angle values, e.g., obtained through a side channel, can easily classify the algorithm being used by the victim – assuming the attacker has access to the reference values.

VI. NOISE SENSITIVITY ANALYSIS

In this section, we further analyze how noise affects the ability of attacker to learn about the type of function being executed by the victim. To simulate noisy side channel, we add noise to the coefficients in the polynomial, and then generate new rotation gate angles. This simulates noisy side channels where the phase angles obtained by the attacker are different from the actual ones used in the algorithm.

A. Matrix Inversion: $f(x) = 1/x$

First, we evaluated the phase sensitivity of the QSVT implementation for the matrix inversion function ($f(x) = 1/x$) under two types of perturbations to the phase sequence. In the first experiment, additive noise with a magnitude of 1×10^{-1} was applied to 15 odd-indexed coefficients at indices $[1, 3, 5, \ldots, 29]$. Recall that if target function $f(x)$ is odd, then only odd Chebyshev terms appear, and that due to rounding down we have degree-29 polynomial with 30 phase angles $\{\phi_0, \phi_1, \ldots, \phi_{29}\}$. This modification resulted in a maximum phase deviation of approximately 0.2126 radians. In the second experiment, additive noise with a magnitude of 1×10^{-2} was applied to 15 odd-indexed coefficients at indices. This rounding introduced a smaller maximum deviation of 0.0203 radians.

The results are shown in Figures 3 and 4. With the added noise, the QSVT phase sequences still look almost identical. Further, the added noise does not change the number of phase angles used. With or without noise, the attacker is able to identify the 30 phase rotations, corresponding to the degree-29 polynomial that was used.

B. Cubic Function: $f(x) = x^3$

Second, we performed a phase sensitivity analysis on the QSVT implementation for the cubic function ($f(x) = x^3$), using a degree-3 Chebyshev polynomial approximation. In this experiment, noise with a magnitude of 1×10^{-2} was added to three odd-indexed coefficients at positions $[1, 3]$. The resulting phase difference yielded a maximum deviation of approximately 0.0773 radians

The results are shown in Figure 5. The added noise did not significantly change the rotation angles. For the noise level of 1×10^{-4}, the maximum deviation in radians was bigger for this cubic function compared to the matrix inversion case. This

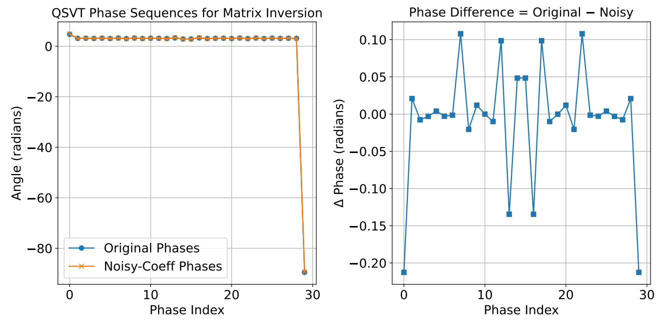

Fig. 3. Phase rotation angles for the matrix inversion function ($f(x) = 1/x$) where 30 rotations are used. This figure shows original phases and the perturbed, noisy phases where the side-channel noise is emulated by adding random additive noise with a magnitude of 1×10^{-1} into the coefficients of the polynomial and then regenerating the phase angles.

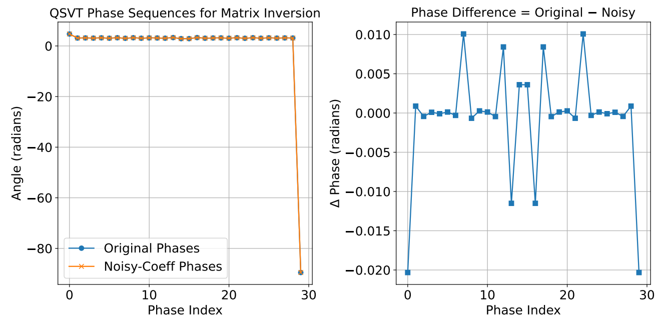

Fig. 4. Phase rotation angles for the matrix inversion function ($f(x) = 1/x$) where 30 rotations are used. This figure shows original phases and the perturbed, noisy phases where the side-channel noise is emulated by adding random additive noise with a magnitude of 1×10^{-2} into the coefficients of the polynomial and then regenerating the phase angles.

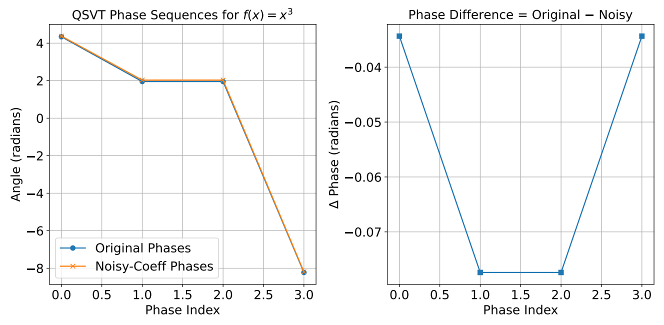

Fig. 5. Phase rotation angles for the cubic function ($f(x) = x^3$) where 4 phase rotations are used. This figure shows original phases and the perturbed, noisy phases where the side-channel noise is emulated by adding random additive noise with a magnitude of 1×10^{-2} into the coefficients of the polynomial and then regenerating the phase angles.

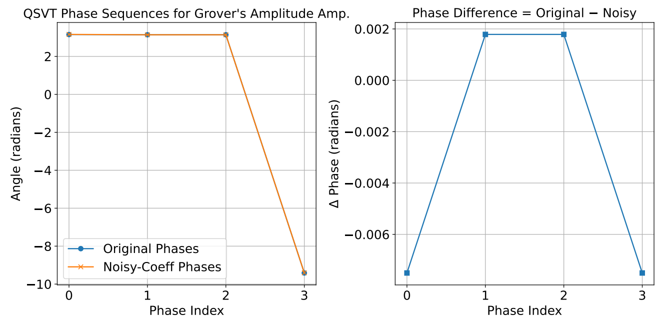

Fig. 6. Phase rotation angles for Grover's amplitude amplification function where 4 phase rotations are used. This figure shows original phases and the perturbed, noisy phases where the side-channel noise is emulated by adding random additive noise with a magnitude of 1×10^{-4} into the coefficients of the polynomial and then regenerating the phase angles.

may be related to having fewer terms in the polynomial. Still, the added noise does not change the number of phase angles used. With or without noise, the attacker is able to identify the 4 phase rotations, corresponding to the degree-3 polynomial.

C. Grover's Amplitude Amplification Function

Third, we analyzed the phase sensitivity of the QSVT implementation for Grover's amplitude amplification function, represented using a Chebyshev polynomial approximation. A degree-3 polynomial was used. To evaluate the effect of perturbations, noise with a magnitude of 1×10^{-4} was added to two odd-indexed coefficients at positions $[1, 3]$. This modification resulted in a maximum phase deviation of approximately 0.0075 radians.

The results are shown in Figure 6. The cubic function and Grover's amplitude amplification function use polynomials of same degree. But noise added to coefficients in Grover's amplitude amplification function was less than that in cubic function, and the resulting maximum phase deviation was less as well, as could be expected. This time again, the added noise does not change the number of phase angles used. With or without noise, the attacker is able to identify the 4 phase rotations, corresponding to the degree-3 polynomial.

VII. Discussion

Our findings suggest that Quantum Signal Processing phase angles encode the target polynomial in a structured manner. While this regularity contributes to the robustness of QSVT circuits, it also introduces potential side-channel risks. An adversary with partial access to the phase sequence could exploit this structure, particularly if the function's properties, e.g., parity or degree are known, to reconstruct the original function. Ironically, the inherent resilience of QSP to noise may increase its susceptibility to reverse engineering via side-channel or statistical attacks.

In particular we have found that:

1) For same phase index, phase angles have different magnitudes for different functions, for example cubic function and Grover's have different angles by about 1 radian or more, and are quite unique. A simple classifier can classify the function type based on knowledge of first few angles (Section V).
2) Even with noisy phase angles, the attacker who can find just the number of rotations gains the knowledge of the degree of the polynomial. This can be easily exploited as, for example, matrix inversion has vastly

larger polynomial than Grover's algorithm, attacker could identify these two just by observing number of rotations without knowing detailed angles (Section VI).

VIII. Defenses

To mitigate these risks, we recommend that QSVT implementations adopt strategies of phase obfuscation. One such approach is angle randomization, where a secret global or local shift is applied to the phase angles and later reversed. Another defense involves function padding or camouflage, wherein harmless modifications are introduced to the target function. For example, approximating a modified function such as $f(x) = \frac{1}{x} + e^{x^3}$ instead of $\frac{1}{x}$ alone can obscure the original computational intent and hinder reverse engineering. Additionally, alternative phase synthesis methods, such as using nonlinear basis functions or Fourier-type expansions, may provide increased resilience against side-channel attacks while preserving approximation fidelity. These techniques represent promising directions for future work in secure and privacy-preserving quantum algorithm design.

Our findings were based on assumption of availability of side-channel information. Hardware methods could be used to limit the side-channels. For example, control pulses could be obfuscated [17] to limit ability of attackers to launch side-channel attacks in the first place. On the software side, if the side-channel is due to co-location of the attacker and the victim, an anti-virus like solutions could be used [18] to detect malicious attackers who are trying to collect side-channel information, and prevent from executing.

IX. Conclusion

In this work, we showed that if an adversary is able to recover the gate operations, specifically gate rotations and their angles, from the quantum computer as a victim QSVT algorithm executes, they could then identify the specific QSVT function that the victim was executing. This paper evaluated three different functions implemented in QSVT (matrix inversion, cubic function, and amplitude amplification from grover's search) and how they mapped to different polynomials and the phase angles. It focused on the correlation between the phase angle values and the number of phase angles for the different functions implemented using QSVT. The paper showed that knowing the phase angle values allows the attacker to identify the function, if they have the reference values for the rotation angles of each candidate function. Further, even just knowing the number of the phase angles allows an attacker to guess the type of functionality being implemented by QSVT, such as they can guess whether it is matrix inversion (due to many rotations) or cubic function (due to few rotations). This work demonstrates the need to consider protection for QSVT running on quantum computers, and presented few initial defense ideas.

References

[1] "IBM Quantum," https://quantum-computing.ibm.com/.

[2] "Amazon Braket," https://aws.amazon.com/braket/.

[3] "Azure Quantum," https://azure.microsoft.com/en-us/products/quantum.

[4] "Quantinuum Nexus," https://nexus.quantinuum.com/.

[5] "IQM Resonance," https://meetiqm.com/products/iqm-resonance/.

[6] F. Erata, C. Xu, R. Piskac, and J. Szefer, "Quantum circuit reconstruction from power side-channel attacks on quantum computer controllers," in *Transactions on Cryptographic Hardware and Embedded Systems*, ser. TCHES, September 2024.

[7] C. Xu, F. Erata, and J. Szefer, "Exploration of power side-channel vulnerabilities in quantum computer controllers," in *Proceedings of the Conference on Computer and Communications Security*, ser. CCS, November 2023.

[8] N. Choudhury, C. N. Mude, S. Das, P. C. Tikkireddi, S. Tannu, and K. Basu, "Crosstalk-induced side channel threats in multi-tenant nisq computers," 2024. [Online]. Available: https://arxiv.org/abs/2412.10507

[9] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics," in *Proceedings of the Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC '19, 2019, p. 193–204.

[10] K. Toyoizumi, N. Yamamoto, and K. Hoshino, "Hamiltonian simulation using quantum singular value transformation: complexity analysis and application to the linearized vlasov-poisson equation," 2023. [Online]. Available: https://arxiv.org/abs/2304.08937

[11] E. Tang and K. Tian, "A cs guide to the quantum singular value transformation," 2023. [Online]. Available: https://arxiv.org/abs/2302.14324

[12] C. Sünderhauf, "Generalized quantum singular value transformation," 2023. [Online]. Available: https://arxiv.org/abs/2312.00723

[13] L. Laneve and S. Wolf, "On multivariate polynomials achievable with quantum signal processing," *Quantum*, vol. 9, p. 1641, 2025.

[14] "Qsvt example notebook," https://github.com/Classiq/classiq-library/blob/main/functions/qmod_library_reference/classiq_open_library/qsvt/qsvt.ipynb.

[15] "pyqsp: Python quantum signal processing," https://github.com/ichuang/pyqsp.

[16] T. Goldfriend, I. Reichental, A. Naveh, L. Gazit, N. Yoran, R. Alon, S. Ur, S. Lahav, E. Cornfeld, A. Elazari *et al.*, "Design and synthesis of scalable quantum programs," 2024. [Online]. Available: https://arxiv.org/abs/2412.07372

[17] T. Trochatos, S. Deshpande, C. Xu, Y. Lu, Y. Ding, and J. Szefer, "Dynamic pulse switching for protection of quantum computation on untrusted clouds," in *International Symposium on Hardware Oriented Security and Trust*, ser. HOST, May 2024.

[18] S. Deshpande, C. Xu, T. Trochatos, H. Wang, F. Erata, S. Han, Y. Ding, and J. Szefer, "Design of quantum computer antivirus," in *Proceedings of the International Symposium on Hardware Oriented Security and Trust*, ser. HOST, May 2023.