Securing HHL Quantum Algorithm against Quantum Computer Attacks

Yizhuo Tan
yizhuo.tan@yale.edu
Yale University
New Haven, CT, USA

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

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

Abstract

The advent of shared, cloud-based quantum computers introduces critical security vulnerabilities. This work identifies and demonstrates two novel attacks against the important HHL algorithm. The two attacks are the Improper Initialization Attack (IIA) and the Higher Energy Attack (HEA), and this work shows that both can be abused to cause HHL to output incorrect results. To address this new threat, this work presents design and implemention of a novel, low-overhead defense circuit for HHL. By adding a single ancilla qubit and minimal gates, the proposed defense reliably detects both IIA and HEA regardless which qubits (ancilla, clock, b) they target. The proposed defense is validated in simulation and on IBM quantum hardware, demonstrating its effectiveness and resilience to noise, providing a practical pathway to securing HHL against the two types of attacks.

CCS Concepts

• **Security and privacy → Side-channel analysis and countermeasures**.

Keywords

Quantum Computing, HHL Algorithms, Improper Initialization Attack, Higher Energy Attack

ACM Reference Format:
Yizhuo Tan, Hrvoje Kukina, and Jakub Szefer. 2025. Securing HHL Quantum Algorithm against Quantum Computer Attacks. In *Hardware and Architectural Support for Security and Privacy 2025 (HASP 2025), October 19, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3768725.3768736

1 Introduction

The advent of cloud-based quantum computing has made quantum processors (QPUs) widely accessible as shared, multi-tenant resources. This new paradigm, where circuits from different, potentially distrusting, users are executed on the same hardware, introduces critical security vulnerabilities not present in classical computing [5, 7, 17]. This paper investigates these threats by targeting the Harrow-Hassidim-Lloyd (HHL) algorithm [10]. As a cornerstone algorithm for solving linear systems with applications in fields like quantum machine learning [3, 6, 15], its security is paramount. While HHL has seen numerous performance and complexity optimizations [1, 4], its fundamental vulnerability to runtime attacks in a shared environment remains unexplored. Our work addresses this gap by demonstrating specific attacks against HHL on today's Noisy Intermediate-Scale Quantum (NISQ) computers and, more importantly, proposing a practical defense.

This work analyzes and addresses two primary threats [17] [19] recently explored: the Improper Initialization Attack (IIA) and the Higher Energy Attack (HEA). The Improper Initialization Attack (IIA) involves maliciously setting a qubit's initial state to $|1\rangle$ when it should be $|0\rangle$. This can be launched via physical effects like crosstalk on a shared device or through a compromised software supply chain that modifies the victim's circuit code. The Higher Energy Attack (HEA) leverages the fact that superconducting qubits have non-computational energy states (e.g., $|2\rangle$, $|3\rangle$, ...). An attacker with pulse-level control can excite a qubit into one of these states. Standard quantum gates are not calibrated for these higher states and will malfunction. Crucially, this improper state can persist even after reset operations, allowing an attack to affect subsequent computations. Both attacks can corrupt HHL's output by compromising just a single qubit.

To counter these threats, this work presents a novel, fortified HHL circuit design with built-in defenses against IIA and HEA. Our approach augments the standard HHL circuit with a dedicated defense register and carefully placed gates that act as tripwires for malicious activity on the ancilla, clock, and b qubits. Extensive testing on real quantum hardware confirms that our strategies effectively neutralize these attacks, providing a clear signal to detect and discard corrupted results and ensuring the algorithm's output can be trusted.

1.1 Contributions

This work makes the following contributions to analysis and defense of the HHL algorithm:

(1) We identify and analyze two potent attacks IIA and HEA that can corrupt HHL's output on shared cloud QPUs.
(2) We present the complete design of a novel, low-overhead defense circuit to counter these threats.
(3) We provide a comprehensive evaluation that validates both the severity of the attacks and the effectiveness of our solution through extensive testing on simulators and real IBM quantum hardware.

HASP 2025, October 19, 2025, Seoul, Republic of Korea

ACM ISBN 979-8-4007-2198-4/25/10
https://doi.org/10.1145/3768725.3768736

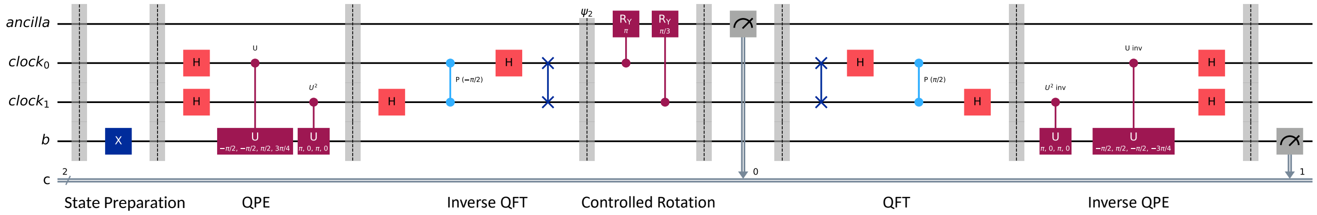


Figure 1: Quantum circuit for the HHL algorithm, the vertical barriers are used to separate the different phases of the algorithm; the phases are labeled at the bottom of the circuit.

2 Background

This section provides the necessary background on the HHL algorithm, as well as the concepts of Improper Initialization Attacks and Higher Energy Attacks.

2.1 HHL Algorithm

The Harrow-Hassidim-Lloyd (HHL) algorithm is a foundational quantum algorithm that provides a potential exponential speedup for solving systems of linear equations compared to its classical counterparts [11] [10]. The algorithm is conceptually divided into three main stages: (1) Quantum Phase Estimation to determine the eigenvalues of the system matrix, (2) a controlled rotation to invert these eigenvalues, and (3) an inverse Quantum Phase Estimation to uncompute the qubits. An example of a 2x2 HHL circuit is shown in Figure 1. The most complex stage is Quantum Phase Estimation (QPE), which is a critical subroutine for many quantum algorithms. The goal of QPE in HHL is to estimate the eigenvalues of the Hermitian matrix A. This is achieved by implementing the unitary time-evolution operator $U = e^{iAt}$, where the state vector $|b\rangle$ (held in the b qubit) is evolved for a duration controlled by a set of ancillary qubits known as the clock qubit. After applying an inverse Quantum Fourier Transform, this clock qubit holds a binary representation of the eigenvalues. The number of qubits in the clock qubit determines the precision of this estimate, creating a direct trade-off between accuracy and the circuit depth required for the controlled unitary operations. For the algorithm to succeed, two conditions are critical. First, the matrix A must be s-sparse and well-conditioned to ensure the Hamiltonian simulation of e^{-iAt} is efficient and the results are numerically stable. Second, the entire process is probabilistic. A successful computation is heralded only by measuring a dedicated ancilla qubit in the state $|1\rangle$. If the measurement yields $|0\rangle$, the result is invalid, and the entire algorithm must be repeated. This operational model, with its reliance on the precise states of the ancilla, clock, and b qubits, creates multiple points of failure. The integrity of these components is paramount, as a fault or malicious manipulation can corrupt or invalidate the entire computation, making them key targets for security analysis.

2.2 Improper Initialization Attacks

The Improper Initialization Attack (IIA) occurs when an attacker maliciously alters a qubit's initial state, typically from the expected $|0\rangle$ to $|1\rangle$. This can be achieved through two primary vectors. First, at the hardware level on a shared multi-tenant QPU, an attacker can exploit physical crosstalk from their own operations to manipulate a victim's physically adjacent qubit [2, 9]. Second, at the software level, an attacker can adapt well-known software supply-chain attack techniques [13] to modify a victim's quantum circuit code before execution.

2.3 Higher Energy Attacks

The second threat we analyze is the Higher Energy Attack (HEA), which exploits leakage into non-computational states of the qubit Hilbert space. While quantum algorithms operate within the $|0\rangle$, $|1\rangle$ subspace, physical qubits possess a ladder of higher energy states (e.g., $|2\rangle$, $|3\rangle$, ...). An adversary with low-level pulse control, a feature available on most major quantum platforms [12], can craft a pulse to drive a target qubit into one of these illicit states.

The HEA is a potent threat due to three effects demonstrated in prior works [17, 19]: (1) State Misclassification: The quantum measurement discriminator, calibrated only for the $|0\rangle$ and $|1\rangle$ subspace, typically misinterprets any higher energy state as a $|1\rangle$ outcome. (2) Gate Failure: Standard single- and two-qubit gates, which are precisely calibrated microwave pulses targeting the $|0\rangle$ to $|1\rangle$ transition frequency, have no effect on a qubit in a higher energy state. (3) Reset Resilience: Most critically, these higher energy states are persistent; they are not cleared by standard reset protocols, creating a persistent fault that can corrupt subsequent circuits allocated to the same physical qubit.

3 Threat Model

Our threat model assumes an adversary with the following access, knowledge, and capabilities:

Adversary's Access: The attacker is a standard, non-privileged user with remote access to a shared quantum computer. They can use publicly available tools, such as Qiskit Pulse, to gain low-level control over qubit operations.

Adversary's Knowledge and Position: We assume the attacker operates in a multi-tenant environment, sharing the QPU either spatially or temporally with the victim. This co-tenancy is a prerequisite for HEA and one possible vector for IIA (the other being a software supply-chain attack). The attacker is assumed to know the physical indices of the victim's qubits but does not need to know their exact logical function (e.g., ancilla vs. clock).

Adversary's Capabilities: The adversary's goal is to set a target qubit to the $|1\rangle$ state (for an IIA) or a higher energy state like $|2\rangle$ (for an HEA). Based on prior work [16], we assume these capabilities

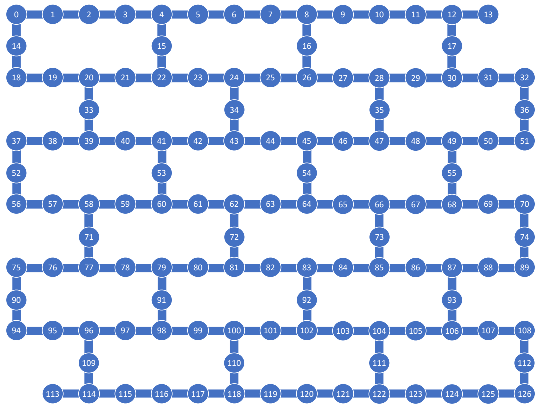


Figure 2: Topology of a IBM Quantum QPU, the Eagle r3 processor, with 127 qubits. Circles represent qubits, thick lines represent fixed couplings between the qubits.

are feasible. The scope of this paper is therefore not on developing new attack vectors, but on analyzing the impact of these attacks on the HHL algorithm and presenting a practical defense.

4 Experimental Setup

Our experimental setup includes the quantum platforms used for evaluation, the specific methods for generating higher energy states and tested benchmarks.

4.1 Quantum Platforms Used

Our experiments were conducted using IBM Qiskit on two distinct platforms: quantum simulators for ideal-case analysis and a real IBM superconducting device for practical hardware validation.

Quantum Simulators: For simulation, we used IBM's BasicSimulator and AerSimulator with 1000 shots per circuit execution. As these are gate-level simulators that only support the $|0\rangle$, $|1\rangle$ computational subspace, they were used exclusively to evaluate the IIA. The HEA requires a physical device.

Quantum Hardwares: Hardware validation was performed on *IBM_brisbane*, a 127-qubit processor with a heavy-hexagonal topology in Figure 2. To ensure proper qubit connectivity and prevent unwanted transpiler optimizations, we used physical qubits 0, 1, 2, 3 or 0, 1, 2, 14 with the compiler optimization level set to 0. Due to significant device noise, our hardware tests were limited to smaller HHL instances. All hardware experiments were run for 10000 shots, and the final results were averaged over 3 independent runs to ensure statistical robustness.

4.2 Generating Higher Energy States

To implement the HEA, we leveraged Qiskit Pulse [12] to deliver custom microwave pulses to specific physical qubits. The parameters for these pulses (e.g., frequency and amplitude) were calibrated for each target qubit using standard frequency sweep and Rabi experiments via cloud access.

A critical challenge in targeting a specific physical qubit is the Qiskit transpiler, which may insert SWAP gates to optimize circuit layout, effectively moving a logical qubit to a different physical location during execution. As a standard SWAP gate is ineffective on a qubit in a higher energy state, the malicious state would not follow the logical qubit. To prevent this, we explicitly disabled transpiler optimizations and manually designed our initial qubit layouts (0,1,2,3 and 0,1,2,14) to match the *IBM_brisbane* hardware topology. We then verified the transpiled circuit to ensure the attack pulse was correctly applied to the intended physical qubit in all experiments.

4.3 Benchmarks

The circuits used to evaluate IIA and HEA are shown in Fig. 3 and Fig. 4, respectively. The examples in the figures demonstrate the attacks on the ancilla qubits, but the same approach is taken for the attack on all other qubits.

For the HHL part of the testing circuits, we use the following matrix A and vector b:

$$A = \begin{bmatrix} 3/4 & 1/4 \\ 1/4 & 3/4 \end{bmatrix}, b = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{1}$$

The result of the HHL circuit is interpreted from the output probabilities when the b register is measured. The way a solution encoded in a quantum state can be compared to a classical solution vector for a particular system of linear equations is through the ratio of the squares of the magnitudes of its components. In our case, the correct output should have a ratio of 1:9, meaning that among all the measurements when the ancilla qubit is 1, the number of measurements of the b register that yield 0 versus the number that yield 1 should be in the ratio 1:9.

For emulating the IIA attack, an additional X gate is inserted at the beginning of the circuit in order to set the target qubit to $|1\rangle$ state before the circuit executes. To test the HEA attack, we first insert an X gate at the beginning of the circuit to set the target qubit to $|1\rangle$, then apply a custom pulse to excite the qubits from $|1\rangle$ to $|2\rangle$. This is also to emulate the attack only. In practice, attackers could use different schemes to achieve IIA and HEA.

We quantify the attack and defense results by measuring the variational distance between the baseline (attack-free) output distribution and the one produced under attack. A distance ranging from 0 to 0.2 indicates a minimal impact of the attack, while a distance between 0.2 and 0.4 suggests a mild influence. A distance of 0.4 to 0.6 implies a significant impact, and a distance from 0.6 to 1 denotes a very high impact of the attack.

5 Defenses

Before evaluating the attacks, we explain the design and rationale behind our defenses. Then, both the attacks and defenses are evaluated together.

5.1 Defense Idea

The goal of our work is to detect when an attack has occurred and allow the user to determine from the output of the quantum circuit if there is an attack (and results should be ignored) or if there was no attack. The approach to detecting the presence of attacks involves incorporating additional measurements for the different qubits used by the HHL algorithm: ancilla, clock, and b qubits.

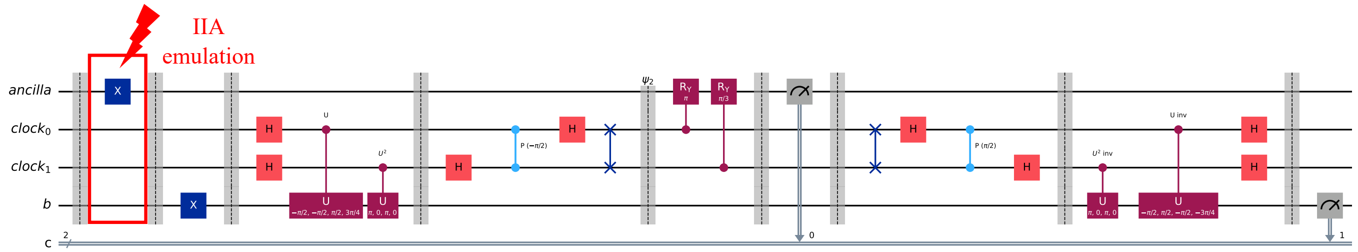


Figure 3: Emulation of Improper Initialization Attack (IIA) on HHL ancilla qubit. To emulate the attack, an additional `X` gate is inserted at the beginning of the circuit in order to set the ancilla qubit to $|1\rangle$ state before the circuit executes.

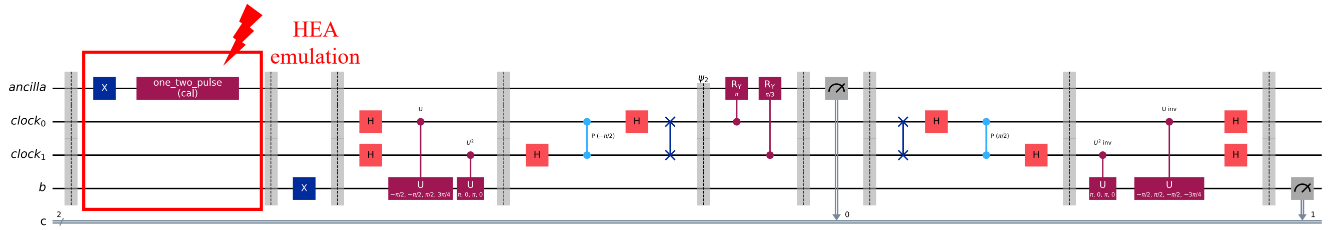


Figure 4: Emulation of Higher Energy Attack (HEA) on HHL ancilla qubit. To emulate the attack, an additional `X` gate is inserted at the beginning of the circuit in order to set the ancilla qubit to $|1\rangle$ followed by a custom pulse used to excite the qubits from $|1\rangle$ to $|2\rangle$ state, i.e. the higher energy state, before the circuit executes.

To detect IIA, the intuition is that we need to confirm if a qubit was initialized into $|0\rangle$ state (no attack) or $|1\rangle$ (attack). This can be done, for example, by directly measuring the qubit at beginning of the execution and checking it. To differentiate between IIA and HEA, we leverage the property of higher-energy states: the predefined quantum gates are not effective in presence of higher energy states. Both improper initialization and higher-energy states results in the qubits being set into initial states that would be both measured as '1'. However, improper initialization sets the qubit(s) into $|1\rangle$ state that is modified when different quantum gates are applied, while for higher-energy states, the qubit(s) are set into $|2\rangle$ or higher states, that are not readily affected by the quantum gates – no matter what gates are applied to the qubits, the measurement of the qubit(s) will always be '1' (until the qubits begin to decay). When possible, the defense aims to differentiate if the quantum states are or are not affected by the HHL algorithms gates, thus pointing to the type of attack.

5.2 Defenses Circuit

Our defense strategy is implemented in a modified HHL circuit 5 that incorporates low-overhead checks against both IIA and HEA. Requiring only one additional ancilla qubit, the design enables a multi-part "attack signature", or attack checks, to be obtained from targeted measurements to protect all critical HHL qubits. The mechanism integrates three primary checks into the HHL workflow:

b Qubit Defense: An initial check that uses entanglement to verify the state of the b register before the main computation protects from attacks on b register (c.f. Part 2 in Figure 5).

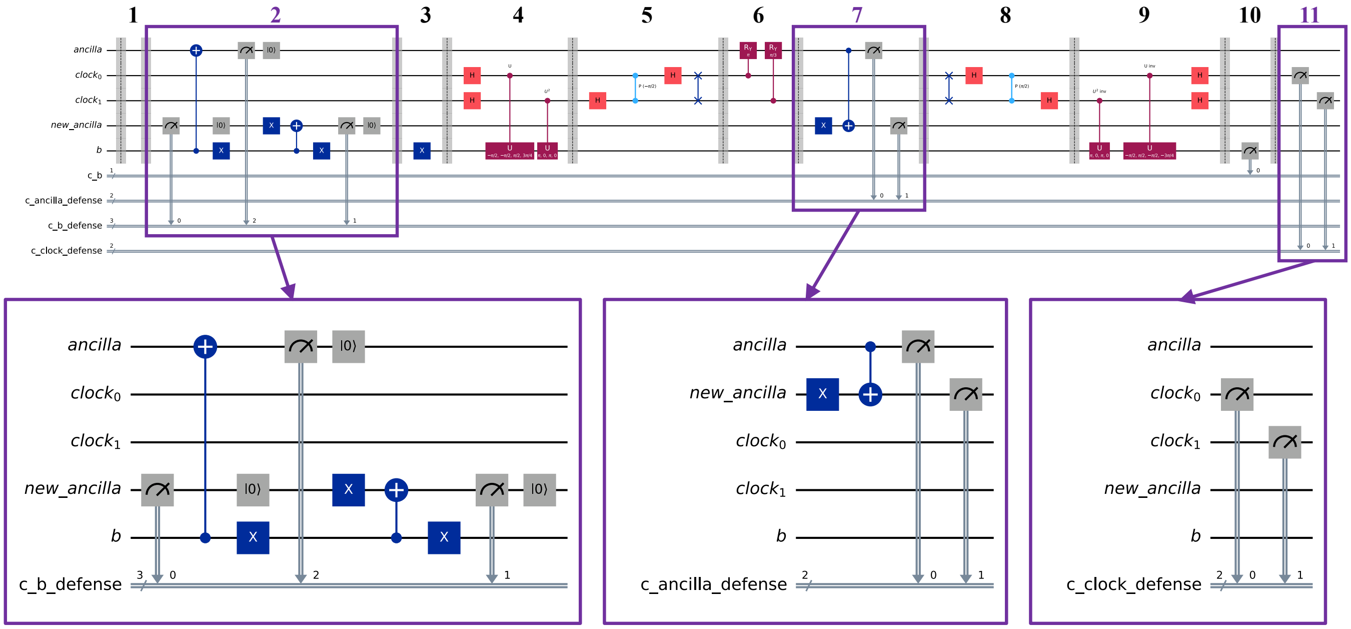
Table 1: Determining if an attack occured on the ancilla and new ancilla qubits. The `c_ancilla_defense` output is measured in Part 7 in Fig. 5. While output 01 seems ambiguous, by using `c_b_defense` discussed later we can fully determine if 01 means no attack or HEA on new ancilla.

Attack Type	Expected `c_ancilla_defense` Output
No attack, HHL converges	`10`
No attack, HHL continues to update	`01`
HEA on ancilla	`11`
HEA on new ancilla	`01` or `11`
HEA on both ancilla and new ancilla	`11`

Ancilla Qubit Defense: A modified measurement that replaces the standard one to verify the state of the primary ancilla to protect it (c.f. Part 7 in Figure 5).

Clock Qubit Defense: A final measurement to ensure the clock qubits were correctly returned to $|0\rangle$ after the uncomputation phase (c.f. Part 11 in Figure 5).

The measurements from these three stages form a composite 7-bit signature. A normal execution yields a specific baseline value (e.g., 1000000), while any deviation provides an unambiguous signal of a fault, prompting the user to discard the corrupted output. This design is scalable to larger HHL instances.



Defense for the b qubit. Defense for the ancilla qubit. Defense for the clock qubits.

Figure 5: HHL circuit including our defenses. Part 2 serves as a defense for the b qubit, Part 7 serves as a defense for the ancilla qubit, while Part 11 serves as a defense for the clock qubits.

Table 2: Determining if an attack occured on the b qubit. The `c_b_defense` output is measured in Part 2 in Fig. 5.

Attack Type	Expected `c_b_defense` Output
No attack	000
HEA on b	010
IIA on b	011
HEA on ancilla	001
IIA on ancilla	001
HEA on new ancilla	110
IIA on new ancilla	100
HEA on ancilla and b	011
IIA on ancilla and b	010
HEA on ancilla and new ancilla	111
IIA on ancilla and new ancilla	101
HEA on new ancilla and b	110
IIA on new ancilla and b	111
HEA on ancilla, new ancilla and b	111
IIA on ancilla, new ancilla and b	110

5.3 Details of Defense for the ancilla Qubit

We explain the ancilla defense first because this protects both the ancilla qubit and the new ancilla qubit, which are used in the defense for the b qubit. A new ancilla qubit was added to the original HHL circuit as shown bottom-middle of Fig. 5. An `X` gate was applied to the new ancilla qubit, followed by a `CNOT` with the ancilla as control and the new ancilla as target. Finally, both qubits were measured. The measurement results are stored in the classical register `c_ancilla_defense`, with the left bit for the ancilla qubit and the right bit for the new ancilla qubit. Normally, these qubits will be measured as 10 when HHL converges or 01 when further updates are required, while under attack they will be measured as 11 because HEA results in a '1' readout and disables the `CNOT` gate. Consequently, the error introduced by IIA or HEA on the ancilla qubit can result in unexpected outputs from `c_ancilla_defense`. Table 1 shows how the output of `c_ancilla_defense` can be used to determine if there was an attack, and what type of attack. While output 01 seems ambiguous, by using `c_b_defense` discussed later we can fully determine if 01 means no attack or HEA on new ancilla.

Table 3: Determining if an attack occured on the clock qubits. The `c_clock_defense` output is measured in Part 11.

Attack Type	Expected `c_clock_defense` Output
No attack	00
HEA on clock0	10
IIA on clock0	10
HEA on clock1	01
IIA on clock1	01
HEA on clock0 and clock1	11
IIA on clock0 and clock1	11

5.4 Details of Defense for the b Qubit

Bottom-left of Fig. 5 shows details of the defense for the b qubit using two ancilla qubits. We apply `X` gates to the b qubit twice. Initially, the qubit should be in $|0\rangle$ state. If the b qubit is not under attack, two `X` gates (which are analogous to NOT gates in classical computers) will cancel each other out and the state of the b qubit will be |0⟩ and can be used by the remainder of the algorithm as normal. To detect a possible attack, the b qubit is entangled with the ancilla and new_ancilla qubits.

This efficient defense can distinguish the state of the b qubit simply by applying `CNOT`, `X`, and `Reset` gates, along with measurements on two ancilla qubits stored in `c_b_defense`, thereby avoiding direct measurement of b itself. The leftmost bit corresponds to the first measurement of the new ancilla qubit, the second bit to its second measurement, and the rightmost bit to the ancilla qubit. The `Reset` gates in this defense can also protect ancilla and new ancilla qubits from IIA, which is why we do not mention IIA in 1.

Table 2 outlines how `c_b_defense` can be used to determine if there was an attack. Some attack types may seem ambiguous due to identical expected `c_b_defense` output. However, by combining both `c_b_defense` and `c_ancilla_defense`, we can distinguish these attack types to a certain degree. For example, HEA on ancilla, IIA on ancilla both yield the same expected `c_b_defense` output 10. However, when `c_ancilla_defense` = 11, it indicates that an HEA has occurred on the ancilla qubit. Regardless, only one output of `c_b_defense`, i.e. 000, indicates no attack.

5.5 Details of Defense for the Clock Qubits

The defense for the clock qubits is a simple but effective state-verification check performed at the end of the circuit as shown in bottom-right of Fig. 5. The HHL algorithm's uncomputation phase (Parts 8-9 of Figure 5) is designed to deterministically return the clock qubits to their initial |0⟩ state. We leverage this by adding a final measurement of these qubits into the `c_clock_defense` register. Under normal operation, the expected outcome is 00. However, both an IIA and an HEA will cause the attacked qubit(s) to end in a non-|0⟩ state, resulting in a measurement of '1'. Consequently, as summarized in Table 3, any output other than the 00 baseline in this register provides an unambiguous signal of an attack.

5.6 Combined Defense and Attack Detection

All the defenses combined in the HHL circuits are again shown in Fig. 5. To detect if an attack has affected the circuit, the user need to read out the concatentaed value of `c_ancilla_defense` (2 bits), `c_b_defense` (3 bits) and `c_clock_defense` (2 bits) registers. The results is a 7-bit value. Value of 1000000 indicates no attack, HHL converges. Value of 0100000 indicates no attack, HHL continues to update. All other values indicate an attack and computation results should be discarded.

6 Analysis of IIA and HEA

This section summarizes the effectiveness of IIA and HEA.

6.1 Evaluation of IIA on Simulator

To compare the results without and with attack, we used the variational distance metric. Table 4 shows the variational distance between original HHL probability distribution and IIA probability distributions under the different attacks on ancilla, clock, and b qubits. It can be seen that attacking any of the qubits results in significant variational distance. Interestingly, attacking clock1 or both clock0 and clock1 has less impact than attacking the other qubits. Nevertheless, we can surmise that attacking any one qubit is sufficient to generate incorrect results.

Table 4: Variational distances of HHL outputs under IIA on BasisSimulator and AerSimulator.

Victim qubit	BasicSimulator	AerSimulator
no attack	0	0
ancilla	0.4980	0.5180
clock0	0.5010	0.4450
clock1	0.2489	0.2520
clock0 and clock1	0.2300	0.2210
b	0.5080	0.5360

Table 5: Ratio and variational distances of HHL outputs under HEA on *IBM_brisbane*.

Victim Qubit	Ratio	Variational Distance
No attack	1 : 1.1759	0
ancilla	1 : 0.8095	0.1863
clock0	1 : 0.6849	0.1002
clock1	1 : 1.8335	0.0713
b	1 : 2.7800	0.2065
Attack 4 qubits	1 : 1.2169	0.1191

Table 6: Ratio and variational distances of HHL outputs under IIA on *IBM_brisbane*.

Victim Qubit	Ratio	Variational Distance
No attack	1 : 1.1759	0
ancilla	1 : 0.8910	0.1048
clock0	1 : 1.2912	0.1125
clock1	1 : 1.1868	0.0923
b	1 : 1.0105	0.0616
Attack 4 qubits	1 : 1.3788	0.1388

6.2 Evaluation of IIA and HEA on Quantum Hardware

Table 5 and table 6 summarize these two attacks on HHL algorithm on *IBM_brisbane* machine. We use the average ratio and the variational distance to quantify the difference between probability distribution of original HHL hardware output and that of attacked output. On real hardware, HEA seem to have bigger impact in terms of the variational distance metric. In table 7, we evaluate the success of an attack by determining whether it causes the ratio to greatly deviate from the ratio observed without an attack. As shown in the table, all the HEA and IIA succeeded except for IIA on clock1, demonstrating the overall effectiveness of our HEA and IIA. On the other hand, the noisy nature of the NISQ computers means that effects of the attacks and the noise both affect the output, and further study of the attacks on real hardware is necessary.

Table 7: Summary of HHL outputs without defense under HEA and IIA on *IBM_brisbane*. We test attacks on the original HHL circuit. The checkmark represents the success of the attack detection while the cross means it fails. For the defense, we assume that the defense is successful if the ratio with defense is closer to the baseline (without attack) of 1:1.1759 or the theoretical ratio of 1:9, compared to the outcome without defense.

Victim Qubits		HEA		IIA	
Num. Victim Qubits	Victim	Ratio	Attack Succeeded	Ratio	Attack Succeeded
0	**No attack**	1 : 1.1883	-	1 : 1.1759	-
1	**ancilla**	1 : 0.8095	✓	1 : 0.8910	✓
1	**clock0**	1 : 0.6849	✓	1 : 1.2912	✓
1	**clock1**	1 : 1.8335	✓	1 : 1.1868	✗
1	**b**	1 : 2.7800	✓	1 : 1.0105	✓
4	**All HHL qubits**	1 : 2.2500	✓	1 : 1.2169	✓

Table 8: Summary of HHL with defense outputs under HEA and IIA on *IBM_brisbane*. We test attacks on the defense circuit and compare the output we get with the expected output. The 7-bit output of the detection registers includes, from left to right, 2 bits for `c_ancilla_defense`, 3 bits for `c_b_defense`, and 2 bits for `c_clock_defense`. The baseline expected outputs are `10 000 00` when HHL converges and `01 000 00` when HHL is still updating. The checkmark represents the success of the attack detection while the cross means it fails.

Victim Qubits		HEA		IIA	
Num. Victim Qubits	Victim	Actual Output	Defense Succeeded	Output of Detection Registers	Defense Succeeded
0	**No attack**	`10 000 00`	✓	`10 000 00`	✓
1	**ancilla**	`11 010 00`	✓	`01 001 00`	✓
1	**new ancilla**	`01 111 00`	✓	`01 100 00`	✓
1	**clock0**	`11 011 11`	✓	`01 001 01`	✓
1	**clock1**	`11 011 11`	✓	`10 001 00`	✓
1	**b**	`01 001 11`	✓	`01 001 00`	✓
4	**All HHL qubits**	`10 011 01`	✓	`10 010 10`	✓

7 Evaluation of the Defense

Table 8 compares the performance of our defense circuit under attack against its baseline functionality. The first two columns summarize the victim qubits being attacked. Columns 3 and 5 denote the actual outputs obtained by our experiments for HEA and IIA with and without attack, respectively, while Columns 4 and 6 indicate whether the attacks were successful. The 7-bit output of the detection registers includes all the measurements from the defense mechanisms as shown in part 2, 7, and 11 in Fig. 5, which are, from left to right, 2 bits for `c_ancilla_defense`, 3 bits for `c_b_defense`, and 2 bits for `c_clock_defense`. The measurement of the b qubit in part 10 of Fig.5 is excluded, as our focus is solely on testing the defense circuit rather than the actual ratio of the HHL algorithm.

The baseline outputs are `10 000 00` when HHL converges and `01 000 00` when HHL is still iterating, all other outputs indicate some sort of error. For our defense mechanism, we consider the defense is successful if the actual output deviates from the baseline output, indicating the detection of an attack. As shown in Table 8, the actual outputs match the baseline when no attack is present, demonstrating that our mechanism does not interfere with the proper functioning of the HHL algorithm. This confirms that our defense strategy, which employs an additional qubit, maintains a high level of output accuracy. Furthermore, Table 8 shows that our defense successfully detects all types of HEA and IIA targeting various victim qubits. Even attacks on multiple qubits are effectively mitigated by our defense mechanism as shown in the last row of the tables where all 4 HHL qubits are attacked.

Our defense mechanism can further distinguish, to some extent, which victim qubits the attacker is targeting by combining the results from table 1, 2 and 3. For instance, outputs such as `01 001 00` or `10 001 00` indicate IIA on the ancilla qubit, while `01 100 00` or `10 100 00` signify IIA on the new ancilla qubit. These outputs align with the experimental results shown in Table 8. However, not all actual outputs match the expected outputs predicted by table 1, 2 and 3. Possible reasons for these discrepancies include noise and the unique characteristics of higher-energy states.

Given the characteristics of higher energy states, it is essential to ensure that after transpilation, the attack pulse and measurement of the corresponding victim qubit are mapped to the same physical qubit on *IBM_brisbane*. In our experiments, we addressed this issue by prioritizing circuit correctness over optimization levels. This involved meticulous inspection of each transpiled circuit and forgoing some optimization steps. While this approach can be further improved by selectively choosing qubits with optimal connectivity for HHL, ensuring that all 7-bit measurement outputs map to the same physical qubits remains difficult on *IBM_brisbane* due to the limitations of its transpilation process.

7.1 Resilience of Defense to Noise

We have further evaluated our defense mechanism to ensure that it works correctly in presence of the noise in today's Noisy Intermediate Scale Quantum (NISQ) computers. Fig. 6 shows the output of the HHL algorithm with our defense when there is no attack. The baseline is the ideal simulator shown as blue bars. As can be seen the dominant outputs of the attack detection registers are `10`

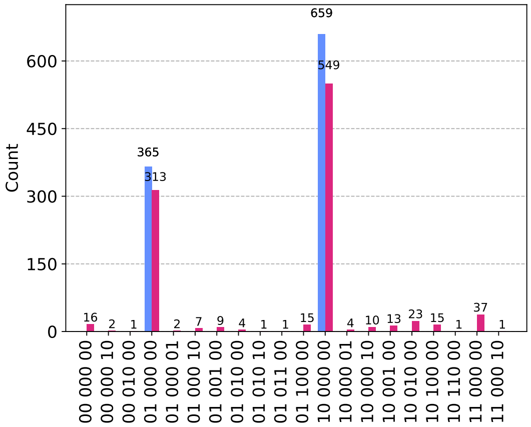


Figure 6: HHL defense experiments using qubit 0,1,2,3,14 on ideal simulator (blue bars) and fake backend *FakeBrisbane* (pink bars) with 1024 shots. The baseline expected outputs are `10 000 00` when HHL converges and `01 000 00` when HHL is still iterating through the optimization process. All other outputs represent noisy outputs.

`000 00` when HHL converges and `01 000 00` when HHL is still iterating, while no incorrect outputs are present. The pink bars show the execution on simulation using *FakeBrisbane*, which includes noise from *IBM_brisbane*. A limited amount of incorrect (noisy) output is observed while the correct `10 000 00` and `01 000 00` outputs remain dominant. Thus our defense is resilient to noise, which should be expected as we do not introduce significant new gates into the circuit nor do we increase the depth of the HHL circuit by any noticeable amount.

8 Related Work

The security of quantum algorithms and quantum information, in general, has become a very interesting field lately. Given that quantum machines are developing at a rapid pace, researchers have started focusing on the security aspects. Therefore, [22] exhibits how to securely transmit information using the HHL algorithm to prevent information leakage, while [8] defines a new measure of information leakage for the quantum encoding of classical data. Error propagation in the HHL algorithm has been studied in [23], where the authors identified three major sources of errors: single-qubit flipping, gate infidelity, and error propagation. There is also a potential way to reduce the demands on physical qubits by evaluating the resource cost of quantum phase estimation, which is a crucial part of the HHL algorithm, before and after quantum error correction [24]. Similarly, quantum phase estimation, being one of the most computationally expensive components of the HHL algorithm, has been tested in terms of scaling properties and related noise resilience [14].

On the security attack side, researchers are actively exploring software supply chain and other attacks on quantum computers. Prior work has demonstrated that the gate-level to pulse-level specification of circuits could be abused to inject attacks in quantum circuits [21]. This is an example of software supply chain attacks that could be leveraged to deploy the IIA and cause HHL to generate incorrect results. In parallel, researchers have explored higher energy state attacks [20]. The HEA used in our work directly leverages higher energy state attacks [20], but applies them to a new victim quantum circuit algorithm, the HHL.

The HHL algorithm was developed as a quantum facilitator for solving large systems of linear equations, offering potential exponential speedups over classical methods. Over time, its time complexity has been improved by Ambainis [1], Childs et al. [4], and Wossnig et al.[18], while various works have optimized circuit depth and efficiency. We believe the defense ideas targeting IIA and HEA can be applied to these HHL variants since the defenses focus on attack mitigation rather than algorithm specifics.

9 Conclusion and Future Work

This work demonstrated two types of attacks that could be performed on the HHL algorithm, both on quantum simulators and on quantum hardware. To address the attacks, this work presented novel defense strategies against these attacks. To the best of our knowledge, this is the first work that shows attacks and explains possible defense strategies for the HHL algorithm, both in theory and in practice. It further demonstrates a practical defense circuit that detects the attack with limited overhead and is resilient to noise.

For future research, there are several directions that can be pursued thanks to the new understanding of attacks and defenses presented in this work. One new direction is investigating whether more efficient alternatives, in terms of circuit design, exist that could achieve the same goals with fewer resources. We believe our design to be minimal, but further optimization may be possible. Another path would involve exploring newer HHL algorithm versions (both fully quantum and hybrid) and applying existing defense strategies on them.

Acknowledgments

This work was supported in part by NSF grant 2332406.

References

[1] Andris Ambainis. 2010. Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations. *arXiv preprint arXiv:1010.4458* (2010).

[2] Abdullah Ash-Saki, Mahabubul Alam, and Swaroop Ghosh. 2020. Analysis of crosstalk in nisq devices and security implications in multi-programming regime. In *ACM/IEEE International Symposium on Low Power Electronics and Design*. 25–30.

[3] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. 2017. Quantum machine learning. *Nature* 549, 7671 (2017), 195–202.

[4] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. 2017. Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision. *SIAM J. Comput.* 46, 6 (Jan. 2017), 1920–1950.

[5] Sanjay Deshpande, Chuanqi Xu, Theodoros Trochatos, Yongshan Ding, and Jakub Szefer. 2022. Towards an Antivirus for Quantum Computers. In *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. 37–40.

[6] Bojia Duan, Jiabin Yuan, Chao-Hua Yu, Jianbang Huang, and Chang-Yu Hsieh. 2020. A survey on HHL algorithm: From theory to application in quantum machine learning. *Physics Letters A* 384, 24 (2020), 126595.

[7] Anthony D'Onofrio, Amir Hossain, Lesther Santana, Naseem Machlovi, Samuel Stein, Jinwei Liu, Ang Li, and Ying Mao. 2023. Distributed quantum learning with co-management in a multi-tenant quantum system. In *IEEE International Conference on Big Data (BigData)*. 221–228.

[8] Farhad Farokhi. 2024. Maximal information leakage from quantum encoding of classical data. *Phys. Rev. A* 109, 2 (2024), 022608. arXiv:2307.12529 [quant-ph]

[9] Benjamin Harper, Behnam Tonekaboni, Bahar Goldozian, Martin Sevior, and Muhammad Usman. 2024. Crosstalk Attacks and Defence in a Shared Quantum Computing Environment. *arXiv preprint arXiv:2402.02753* (2024).

[10] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. 2009. Quantum algorithm for linear systems of equations. *Physical Review Letters* 103, 15 (2009), 150502.

[11] M R Hestenes and E Stiefel. 1952. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand. (1934)* 49, 6 (Dec. 1952), 409.

[12] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D Nation, Lev S Bishop, Andrew W Cross, et al. 2024. Quantum computing with Qiskit. *arXiv preprint arXiv:2405.08810* (2024).

[13] Piergiorgio Ladisa, Henrik Plate, Matias Martinez, and Olivier Barais. 2023. Sok: Taxonomy of attacks on open-source software supply chains. In *IEEE Symposium on Security and Privacy (SP)*. 1509–1526.

[14] Marc Andreu Marfany, Alona Sakhnenko, and Jeanette Miriam Lorenz. 2024. Identifying Bottlenecks of NISQ-Friendly HHL Algorithms. In *IEEE International Conference on Quantum Computing and Engineering (QCE)*. 275–284.

[15] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. 2015. An introduction to quantum machine learning. *Contemporary Physics* 56, 2 (2015), 172–185.

[16] Jerry Tan, Chuanqi Xu, Theodoros Trochatos, and Jakub Szefer. 2024. Extending and defending attacks on reset operations in quantum computers. In *International Symposium on Secure and Private Execution Environment Design (SEED)*. 73–83.

[17] Yizhuo Tan, Hrvoje Kukina, and Jakub Szefer. 2024. Study of Attacks on the HHL Quantum Algorithm. In *IEEE International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*. 481–488.

[18] Leonard Wossnig, Zhikuan Zhao, and Anupam Prakash. 2018. Quantum Linear System Algorithm for Dense Matrices. *Physical Review Letters* 120, 5 (Jan. 2018).

[19] Chuanqi Xu, Jessie Chen, Allen Mi, and Jakub Szefer. 2023. Securing nisq quantum computer reset operations against higher energy state attacks. In *ACM SIGSAC Conference on Computer and Communications Security*. 594–607.

[20] Chuanqi Xu, Jessie Chen, Allen Mi, and Jakub Szefer. 2023. Securing nisq quantum computer reset operations against higher energy state attacks. In *ACM SIGSAC Conference on Computer and Communications Security*. 594–607.

[21] Chuanqi Xu and Jakub Szefer. 2025. Security Attacks Abusing Pulse-level Quantum Circuits. In *IEEE Symposium on Security and Privacy (SP)*. 222–239.

[22] Xiaolong Yang, Dongfen Li, Jie Zhou, Yuqiao Tan, Yundan Zheng, and Xiaofang Liu. 2023. Research on quantum dialogue protocol based on the HHL algorithm. *Quant. Inf. Proc.* 22, 9 (2023), 340.

[23] Anika Zaman and Hiu Yung Wong. 2022. Study of Error Propagation and Generation in Harrow-Hassidim-Lloyd (HHL) Quantum Algorithm. In *IEEE Latin American Electron Devices Conference (LAEDC)*. 1–4.

[24] Muqing Zheng, Chenxu Liu, Samuel Stein, Xiangyu Li, Johannes Mülmenstädt, Yousu Chen, and Ang Li. 2025. An Early Investigation of the HHL Quantum Linear Solver for Scientific Applications. *Algorithms* 18, 8 (Aug. 2025), 491.