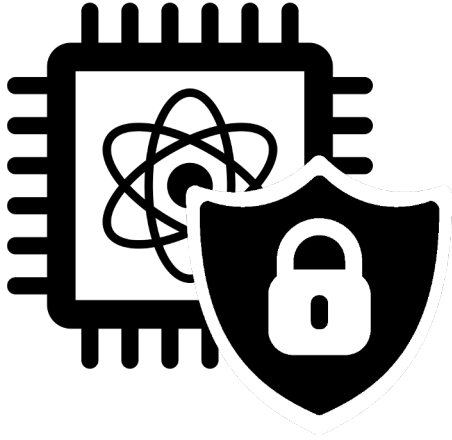




Tutorial on Security of Quantum Computing Systems

QCE Tutorial

Sept. 17, 2026

Prof. Jakub Szefer (jakub.szefer@northwestern.edu)

Electrical and Computer Engineering

Northwestern University



Computer Architecture
and Security Lab (CASLAB)

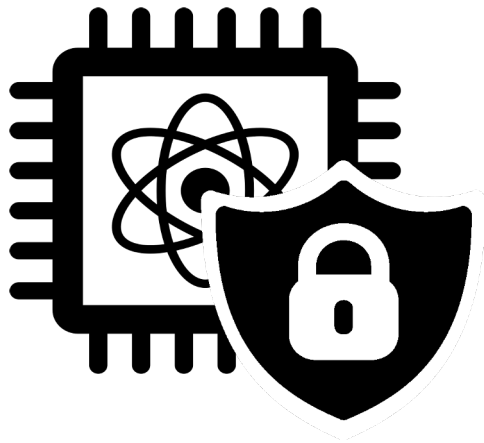


Northwestern
University

Tutorial Outline

- 1:00pm – 1:30pm: Introduction and Motivation
- 1:30pm – 2:30pm: Crosstalk and Fault Injection in NISQ Quantum Computers
- 2:30pm – 3:00pm: Coffee Break
- 3:00pm – 3:45pm: Side Channels in NISQ and FTQC Quantum Computers
 - Side Channels in NISQ Controllers
 - Side Channels in FTQC
- **3:45pm to 4:30pm: Trusted Execution Environments for Quantum Computers**
 - **Design of QC-TEEs for Quantum Computers**
- **4:00pm: End**

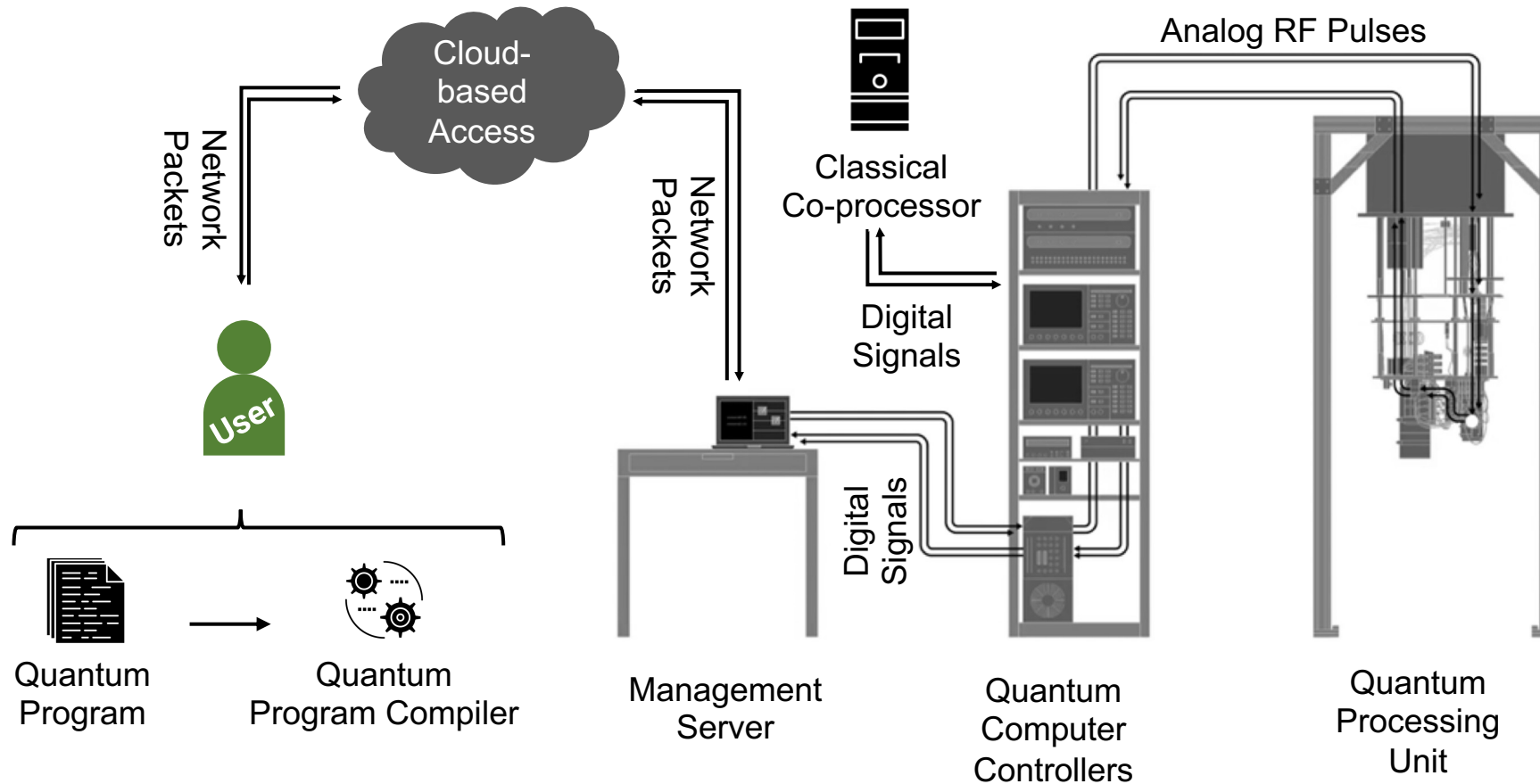




Secure Architectures for Quantum Computers

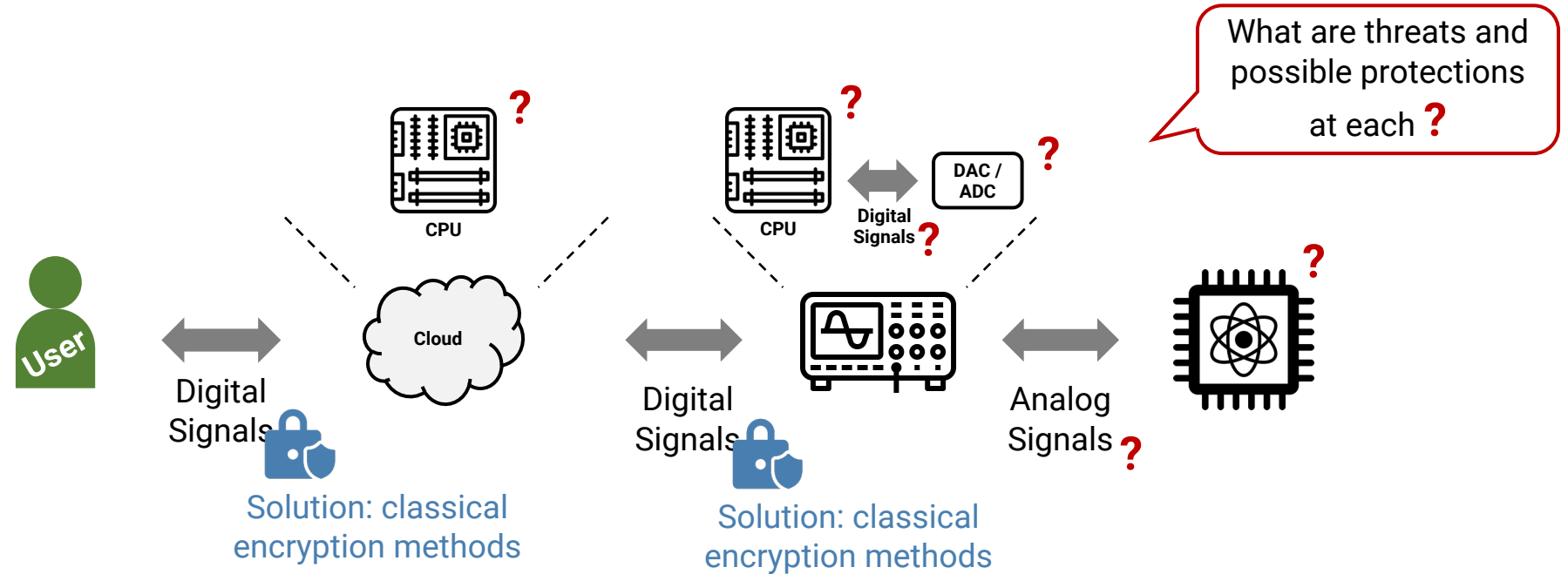


Typical Quantum Computer System Overview



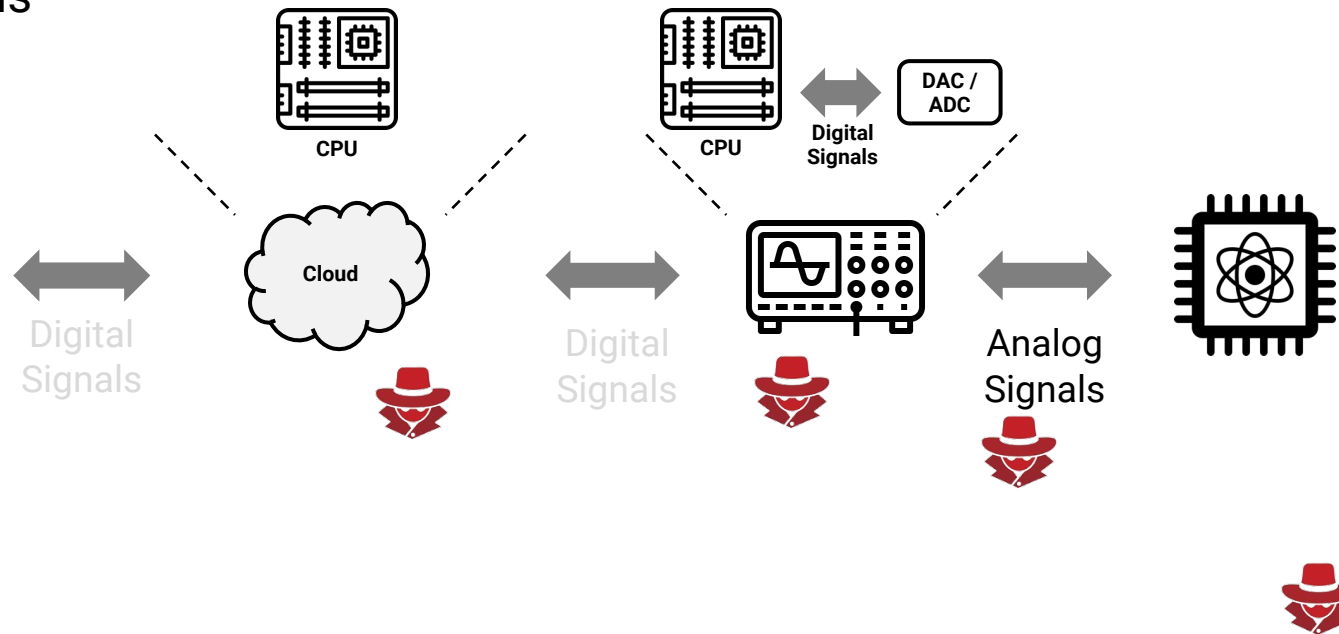
Typical Quantum Computer System Overview

- Abstract view of the quantum computer system can aid in analyzing the potential security threats:



Assumptions and Threats

- Assume honest-but-curious attackers
 - Do not inject errors or manipulate signals
 - Only eavesdrop on operation of the quantum computer and any signals
- Resulting attack surface:
 - Cloud (CPU)
 - Controller (CPU)
 - Controller (DAC)
 - QPU

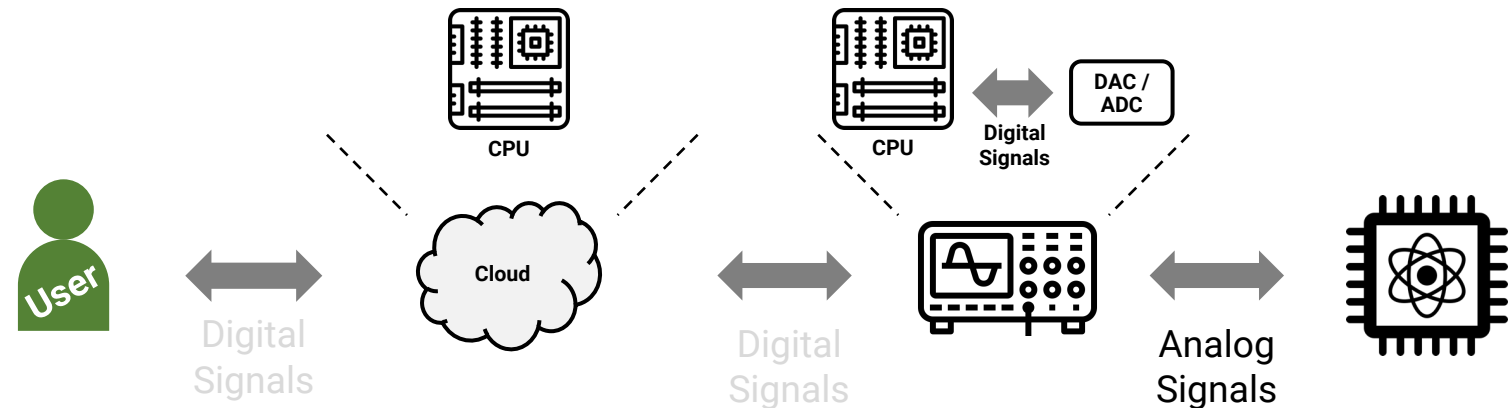


Assumptions and Threats

Solution: Blind Quantum Computation 

	Cloud (CPU)	Controller (CPU)	Controller (DAC)	QPU	
?	Untrusted	Untrusted	Untrusted	Untrusted	?
?	Untrusted	Untrusted	Untrusted	Trusted	?
?	Untrusted	Untrusted	Trusted	Untrusted	?
?	Untrusted	Untrusted	Trusted	Trusted	?
?	Untrusted	Trusted	Untrusted	Untrusted	?
?	Untrusted	Trusted	Untrusted	Trusted	?
?	Untrusted	Trusted	Trusted	Untrusted	?
?	Untrusted	Trusted	Trusted	Trusted	?

	Cloud (CPU)	Controller (CPU)	Controller (DAC)	QPU	
	Trusted	Untrusted	Untrusted	Untrusted	?
	Trusted	Untrusted	Untrusted	Trusted	?
	Trusted	Untrusted	Trusted	Untrusted	?
	Trusted	Untrusted	Trusted	Trusted	?
	Trusted	Trusted	Untrusted	Untrusted	?
	Trusted	Trusted	Untrusted	Trusted	?
	Trusted	Trusted	Trusted	Untrusted	?
	Trusted	Trusted	Trusted	Trusted	?



Assumptions and Threats

New Solutions:
QC-TEE & CASQUE

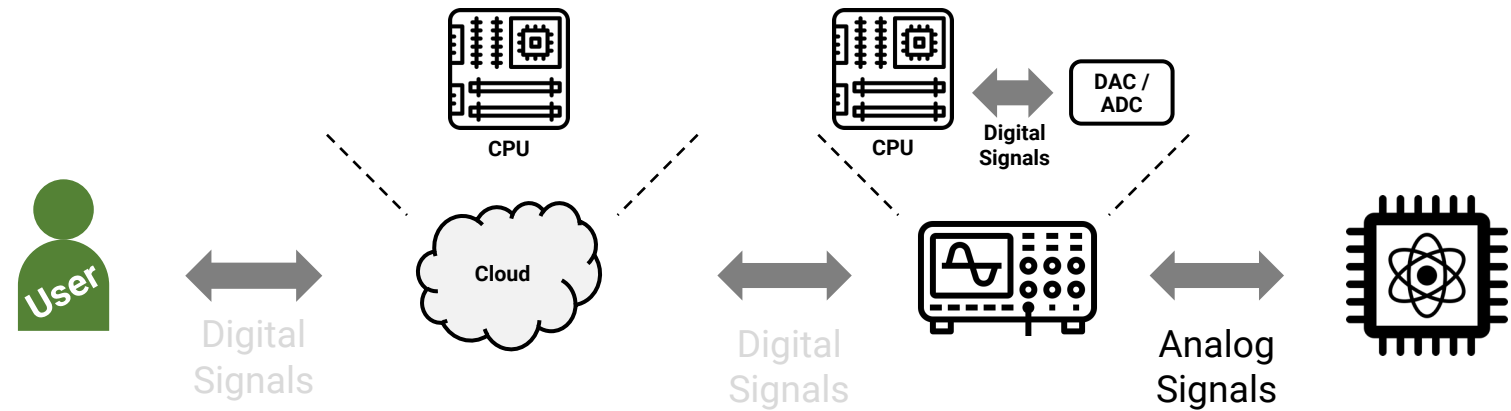


?
?
?
?
?
?
?

Cloud (CPU)	Controller (CPU)	Controller (DAC)	QPU
Untrusted	Untrusted	Untrusted	Untrusted
Untrusted	Untrusted	Untrusted	Trusted
Untrusted	Untrusted	Trusted	Untrusted
Untrusted	Untrusted	Trusted	Trusted
Untrusted	Trusted	Untrusted	Untrusted
Untrusted	Trusted	Untrusted	Trusted
Untrusted	Trusted	Trusted	Untrusted
Untrusted	Trusted	Trusted	Trusted

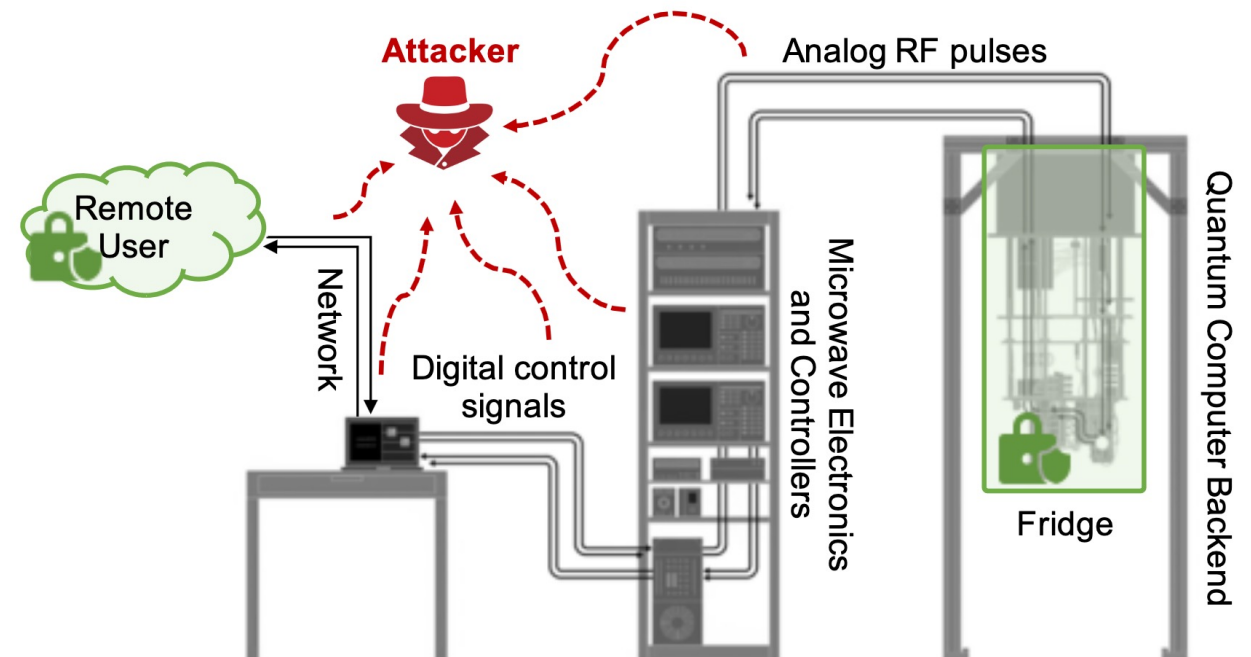
Cloud (CPU)	Controller (CPU)	Controller (DAC)	QPU
Trusted	Untrusted	Untrusted	Untrusted
Trusted	Untrusted	Untrusted	Trusted
Trusted	Untrusted	Trusted	Untrusted
Trusted	Untrusted	Trusted	Trusted
Trusted	Trusted	Untrusted	Untrusted
Trusted	Trusted	Untrusted	Trusted
Trusted	Trusted	Trusted	Untrusted
Trusted	Trusted	Trusted	Trusted

?
?
?
?
?
?
?
?



QC-TEE Assumptions

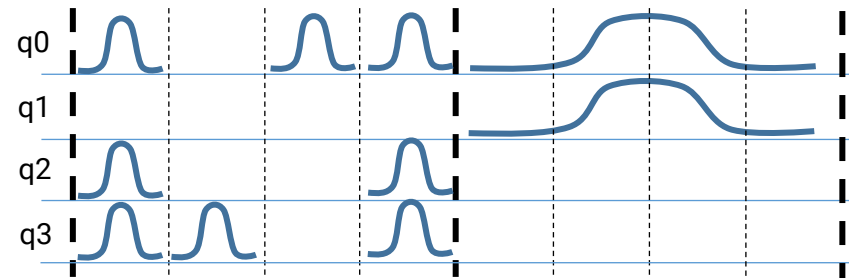
- Focus on superconducting qubit quantum computers
- Fridge of superconducting qubit quantum computers can form natural trust boundary
 - Temperature and pressure changes can be easily detected when fridge is opened
- **Key idea:** need to protect or obfuscate the analog RF pulses going into the (trusted) QPU
 - Digital signals (going through untrusted cloud or controller CPU) are equivalent to the RF pulses
 - Assume we cannot put signal generators or other large electronics in the fridge



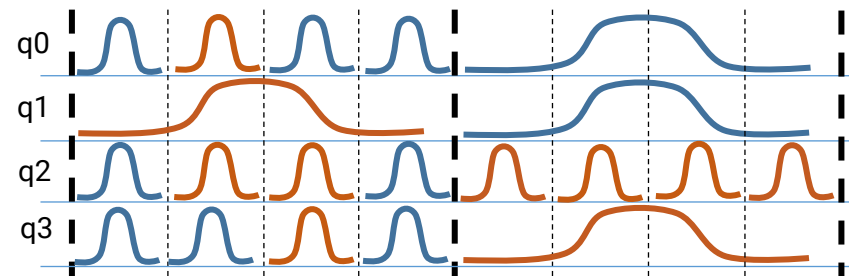
High-Level Idea of Hiding Control Pulses

- Analog control pulses cannot be “encrypted” unlike digital information, but we can hide or obfuscate them instead

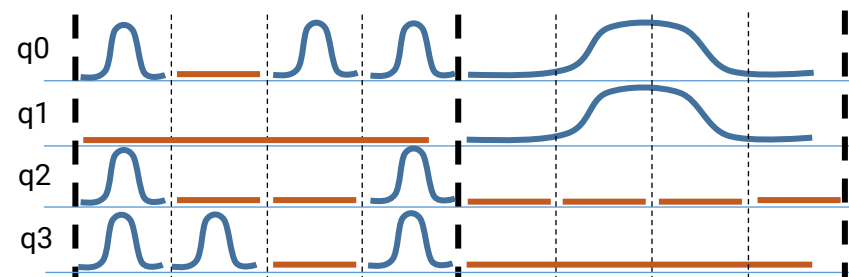
1. Input user circuit:



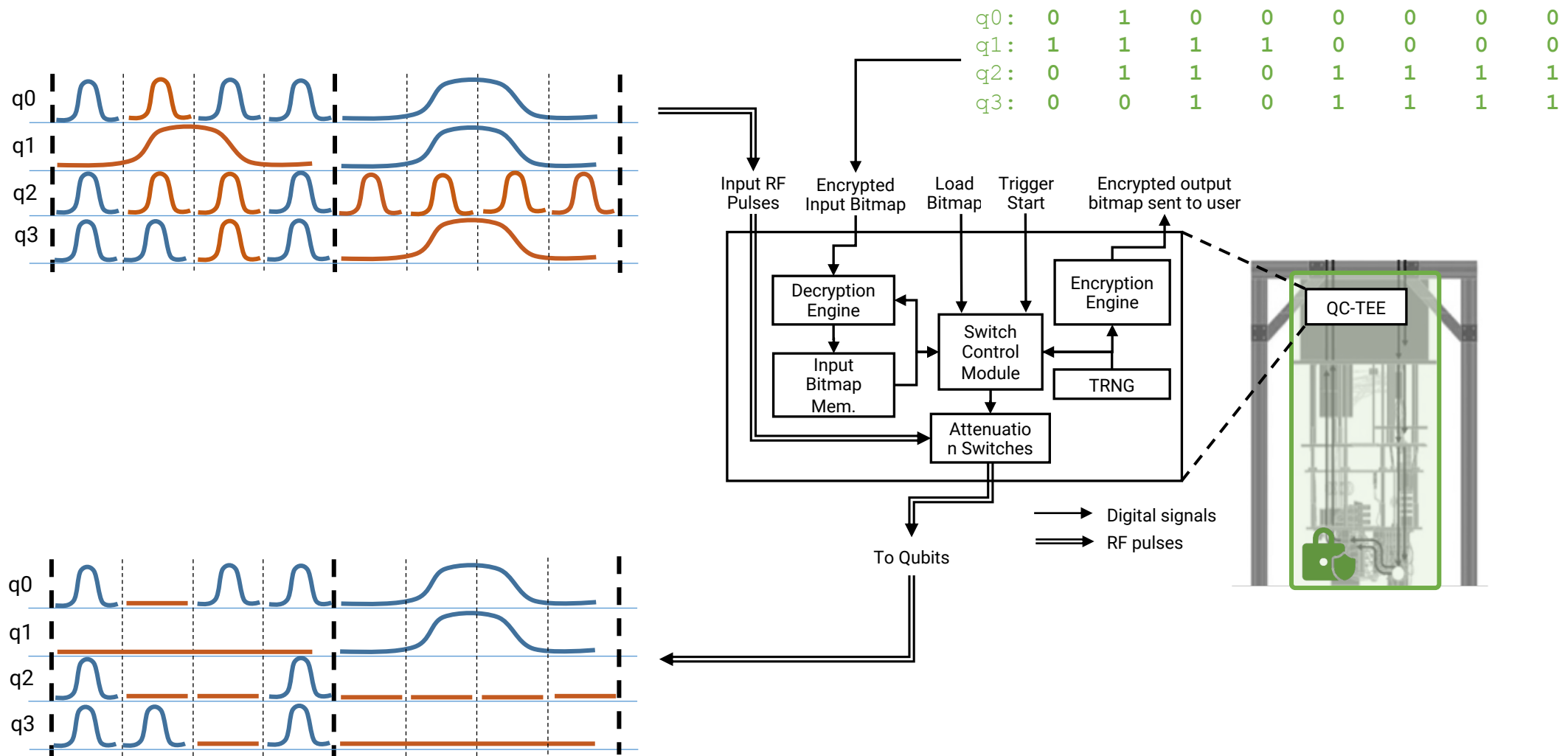
2. Circuit with decoy pulses:



3. Decoy pulses attenuated in the QPU

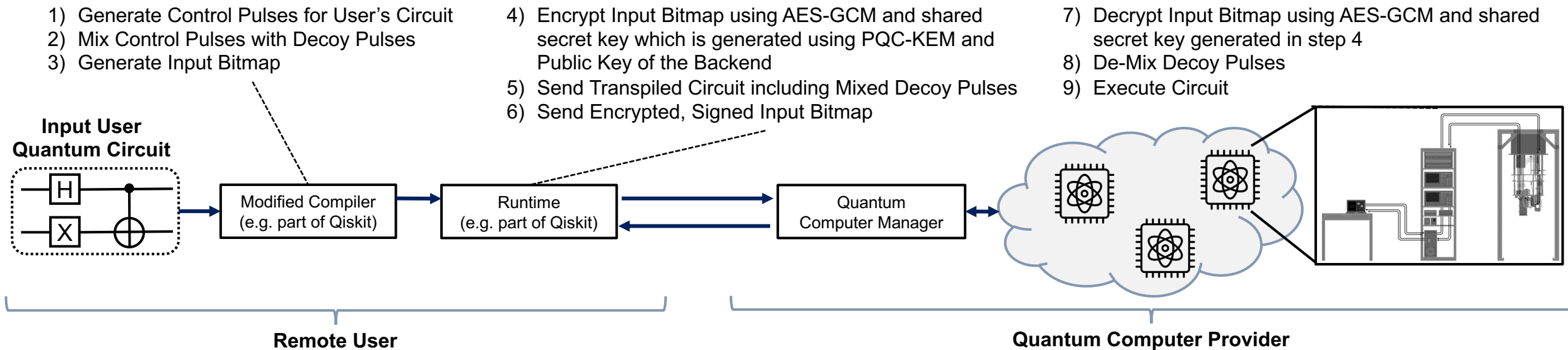


Trusted Hardware for Attenuation of Control Pulses



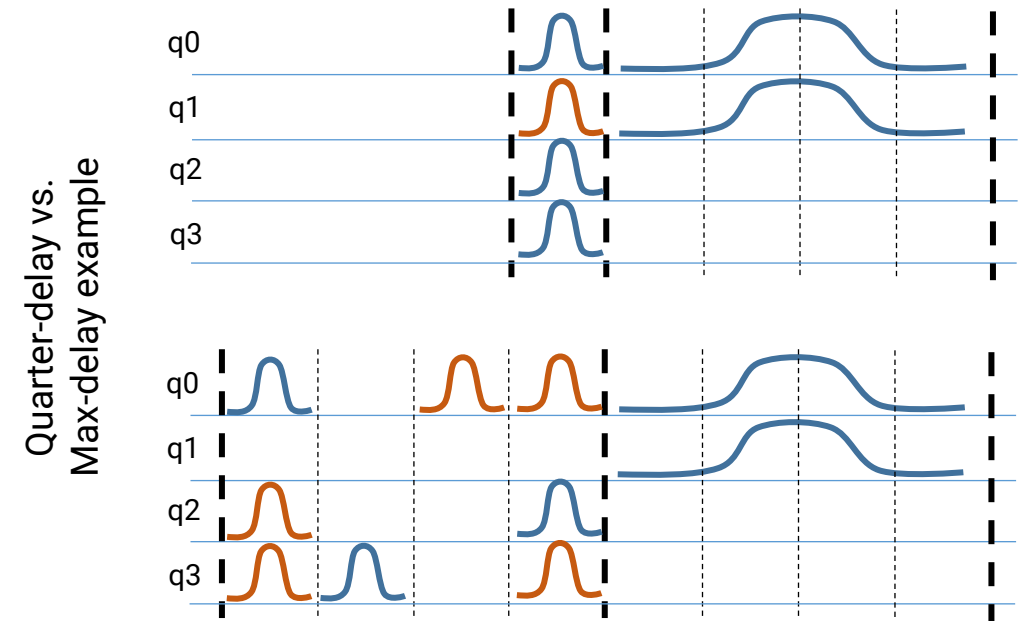
End-to-End Operation

- Software needs to generate the decoy pulses and the bitmap, while hardware attenuates the pulses inside the fridge:



Evaluation Setup

- We evaluated different configurations:
 - Baseline
 - One-sixteenth-delay
 - Quarter-delay
 - Max-delay
- Further for all the levels, except baseline, we analyze results with and without a randomize-output option
- Evaluated on:
 - 7-qubit IBM Perth, to test small-scale benchmarks
 - 27-qubit IBM Algiers, through the Pay-As-You-Go plan on IBM Cloud to test medium-scale benchmarks
 - Aer simulator



Evaluation Using Variational Distance

- Worst case Variational Distance (VD) for the QASM-Bench benchmarks considering perfect RF switches and emulating imperfect RF switches (99.99% amplitude attenuation)

Obfuscation Level:	max-delay	
	w/o rand.-out.	w/ rand.-out.
Avg. VD (perfect RF switches)	0.2099	0.2453
Avg. VD (imperfect RF switches)	0.2652	0.2691

- Worst case Variational Distance (VD) for the QASM-Bench benchmarks for machines with different Quantum Volume (QV)

Obfuscation Level:	max-delay	
	w/o rand.-out.	w/ rand.-out.
Real IBM Perth ($QV = 32$)	0.2099	0.2453
Sim. IBM Perth ($QV = 32$)	0.1461	0.1481
Sim. IBM Mumbai ($QV = 64$)	0.1653	0.1705
Sim. IBM Cairo ($QV = 128$)	0.1286	0.1299



Evaluation of Benchmark Correctness

- Evaluation of correctness of 22 QASM-Bench benchmarks assuming imperfect RF switches (99.99% amplitude attenuation), all benchmarks show correct results

Benchmark	Qubits	Gates	CX	Depth	one-sixteenth-delay		quarter-delay		max-delay	
					w/o rand.-out.	w/ rand.-out.	w/o rand.-out.	w/ rand.-out.	w/o rand.-out.	w/ rand.-out.
deutsch	2	5	1	5	✓	✓	✓	✓	✓	✓
iswap	2	9	2	8	✓	✓	✓	✓	✓	✓
quantumwalks	2	11	3	8	✓	✓	✓	✓	✓	✓
grover	2	16	2	12	✓	✓	✓	✓	✓	✓
dnn	2	226	42	155	✓	✓	✓	✓	✓	✓
teleportation	3	8	2	7	✓	✓	✓	✓	✓	✓
qaoa	3	15	6	12	✓	✓	✓	✓	✓	✓
toffoli	3	18	6	13	✓	✓	✓	✓	✓	✓
linearsolver	3	19	4	12	✓	✓	✓	✓	✓	✓
fredkin	3	19	8	12	✓	✓	✓	✓	✓	✓
basis_change	3	53	10	22	✓	✓				
adder	4	23	10	12	✓	✓	✓	✓	✓	✓
bell	4	33	7	14	✓	✓	✓	✓	✓	✓
qft	4	36	12	9	✓	✓	✓	✓	✓	✓
variational	4	54	16	34	✓	✓				
vqe	4	89	9	28	✓	✓	✓	✓	✓	✓
basis_trotter	4	1626	582	815	✓	✓	✓	✓	✓	✓
qec_en	5	25	10	18	✓	✓	✓	✓	✓	✓
error_correctiond3	5	114	49	78	✓	✓	✓	✓	✓	✓
qaoa	6	270	54	110	✓	✓	✓	✓	✓	✓
bv	19	56	18	22	✓	✓	✓	✓	✓	✓
wstate	27	157	52	55	✓	✓	✓	✓	✓	✓

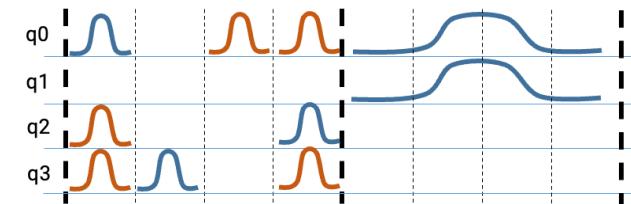
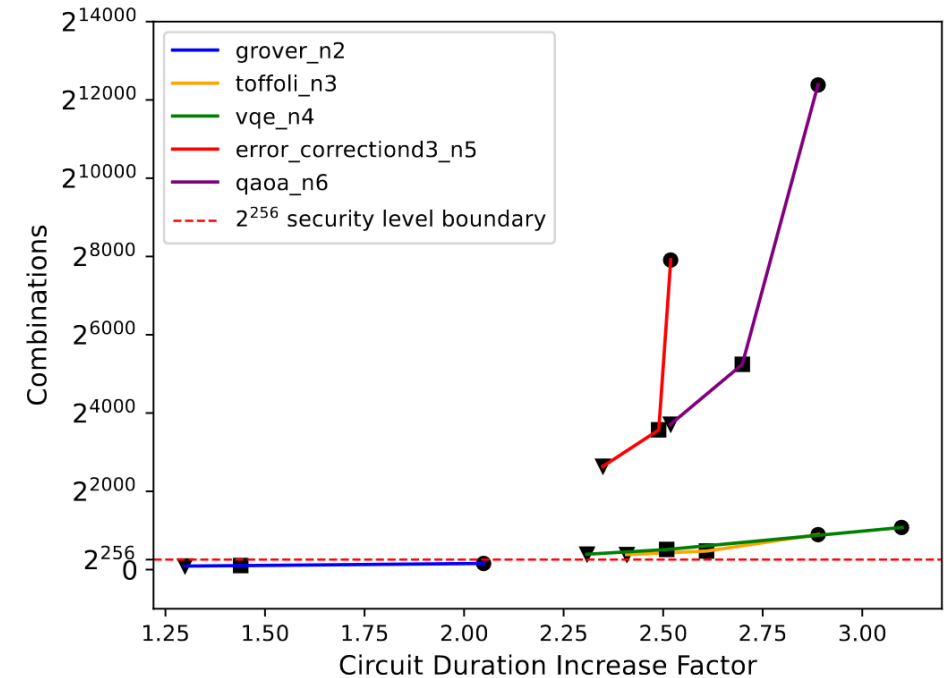


Security Analysis

- The cloud provider is able to see all the (real and decoy) control pulses, but needs to figure out which ones are real and which are decoy
- The complexity based on our analysis:

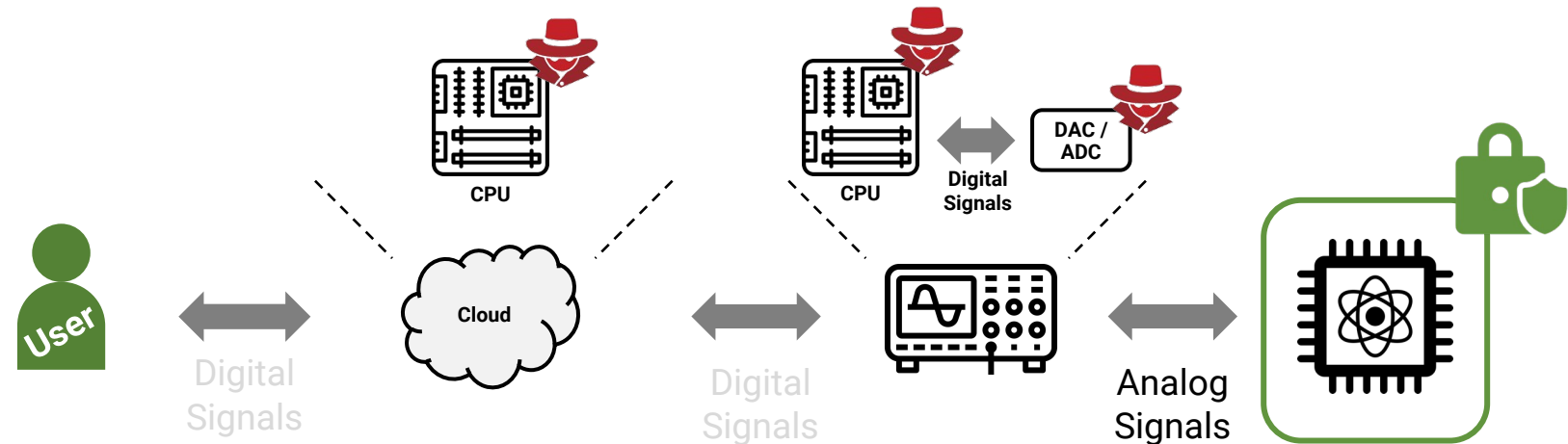
$$Comb = ((2^{n_{SubSlots}})^{n_{SlotSQ}})^{n_{qubits}} \times ((2)^{n_{SlotCX}})^{n_{SubCXInSlotCX}} \times ((2^{n_{SubSlotsInSlotCX}})^{n_{SlotCX}})^{(n_{qubits} - 2 \times n_{SubCXInSlotCX})}$$

- Number of possibilities is much beyond 2^{256} combinations that cloud provider would have have to consider



Trusted QPU with Dynamic Pulse Switching

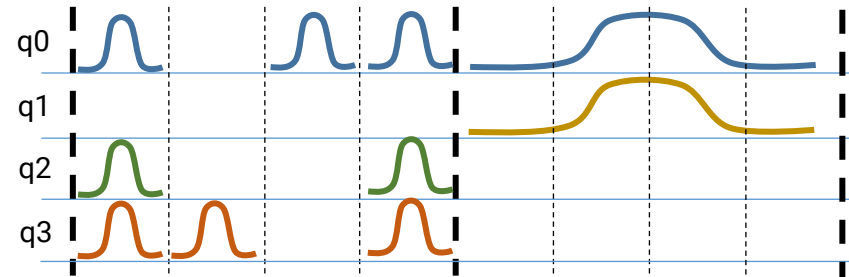
- Maintain same threat model as QC-TEE
 - Trusted QPU
 - Components outside of QPU are not trusted
- **Key idea:** increase complexity (i.e. number of guesses attacker has to consider) by enabling switching of pulses between different channels



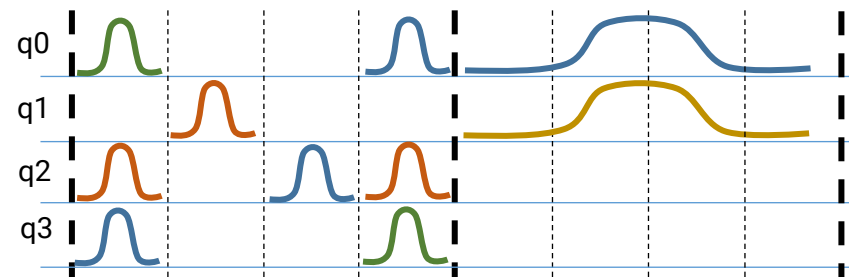
High-Level Idea of Switching Pulses

- We can increase confusion of the attacker by internally switching the control pulses

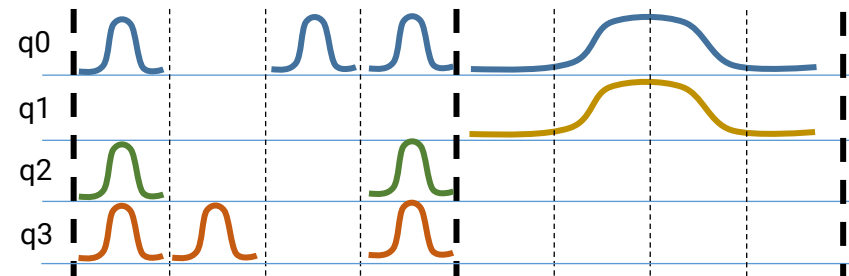
1. Input user circuit:



2. Circuit with pulses switched (as seen by cloud provider):

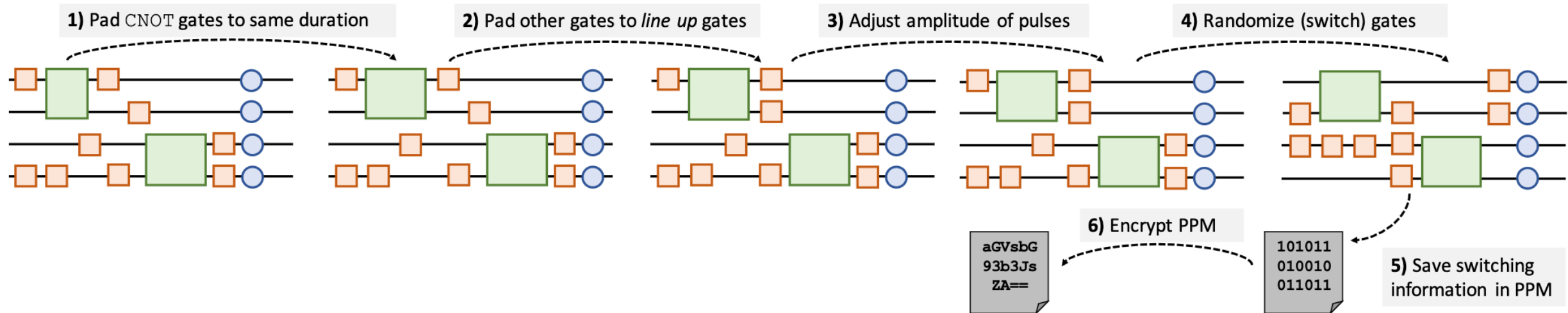


3. Pulses switched back inside the QPU:



CASQUE – Software Circuit Preparation

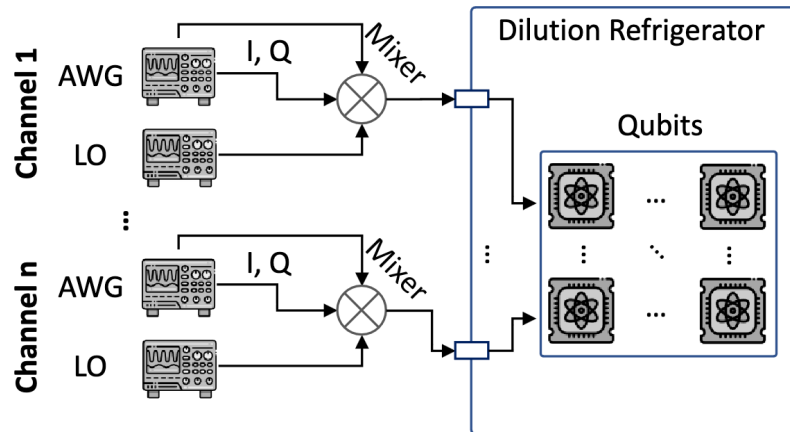
- On software side, quantum circuits need to be adjusted to enable the pulse switching, then pulses need to be randomly switched and the switching information encrypted for the QPU:



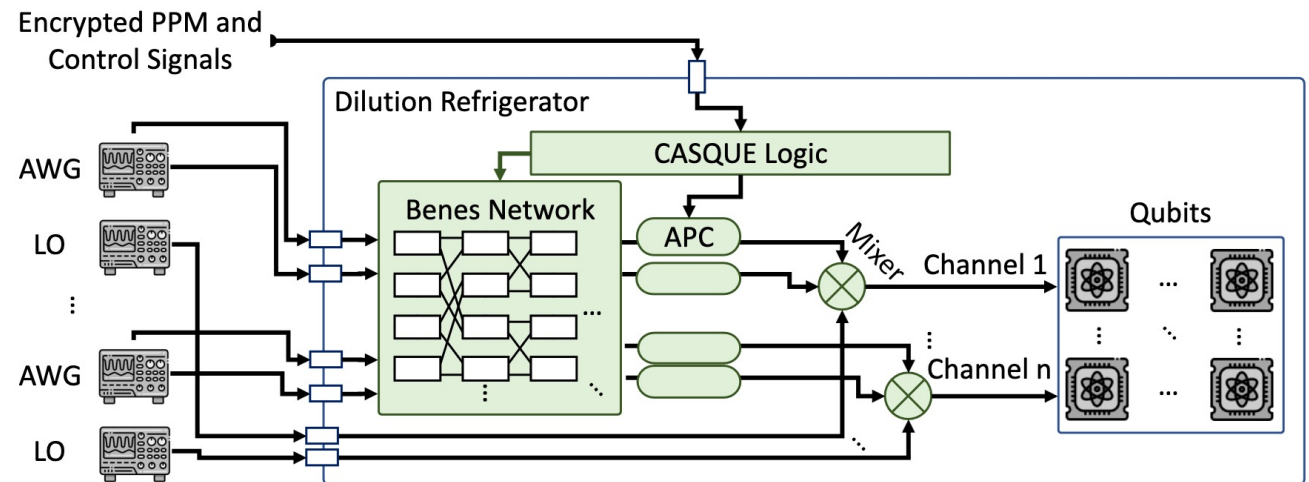
CASQUE – Proposed Hardware Changes

- Key challenge of switching the pulses between drive channels are the frequency, amplitude and phase
- CASQUE keeps signal generators outside the (trusted) fridge and the QPU, but adds new logic and mixers into the fridge

Typical QPU Setup



Proposed CASQUE Hardware Modifications



Evaluation Using Variational Distance

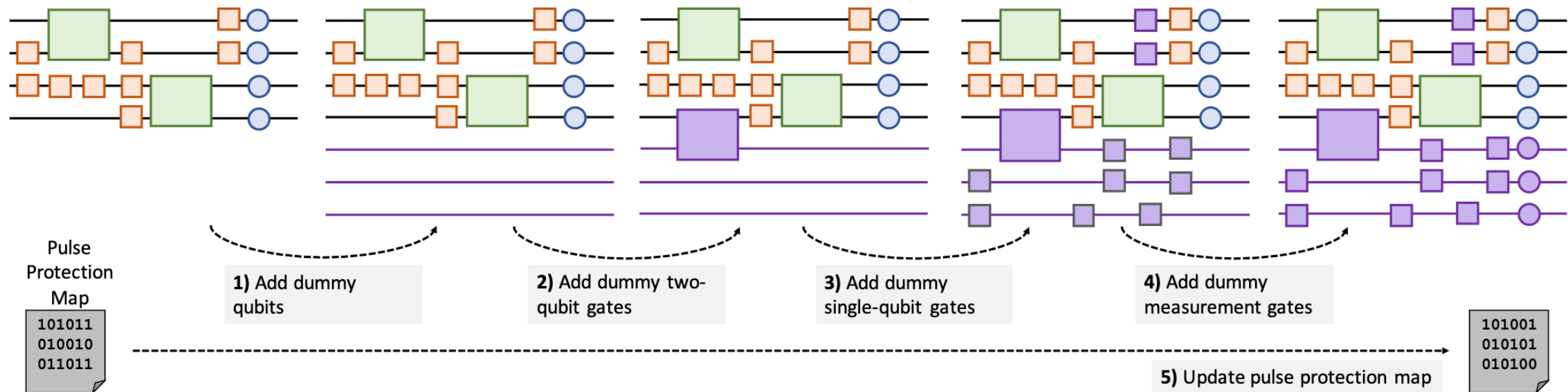
- Variational Distance (VD) for the QASM-Bench benchmarks considering perfect switching, decrease in fidelity is due to the extended circuit duration

Benchmark	Qubits	Gates	CNOT	Ref	VD
wstate	3	30	9	[27]	0.126
basis_change	3	53	10	[33]	0.116
variational	4	54	16	[33]	0.080
vqe	4	89	9	[27]	0.195
qec_en	5	25	10	[50]	0.642
error_correctiond3	5	114	49	[35]	0.184
simon	6	44	14	[4]	0.195
qaoa	6	270	54	[15]	0.203



CASQUE+ with Dummy Qubits

- CASQUE+ introduces idea of dummy qubits, any control pulses on the dummy qubits, or pulses switched to dummy qubits have no effect on the outcome of computation



Security Evaluation of CASQUE+

- Approximate attack complexity on selected QASM-Bench benchmark

Benchmark	Qubits	Gates	CNOT	Complexity			
				0 dummy qubits (CASQUE)	w/ 2 dummy qubits (CASQUE+)	w/ 4 dummy qubits (CASQUE+)	w/ 8 dummy qubits (CASQUE+)
wstate	3	30	9	2^8	2^{15}	2^{19}	2^{25}
basis_change	3	53	10	2^{43}	2^{66}	2^{80}	2^{100}
variational	4	54	16	2^{20}	2^{26}	2^{30}	2^{36}
vqe	4	89	9	2^{45}	2^{64}	2^{77}	2^{95}
qec_en	5	25	10	2^{17}	2^{22}	2^{26}	2^{32}
error_correctiond3	5	114	49	2^{103}	2^{132}	2^{155}	2^{191}
simon	6	44	14	2^{20}	2^{25}	2^{28}	2^{33}
qaoa	6	270	54	2^{268}	2^{324}	2^{365}	2^{428}



Other Possible Secure Quantum Computer Architectures

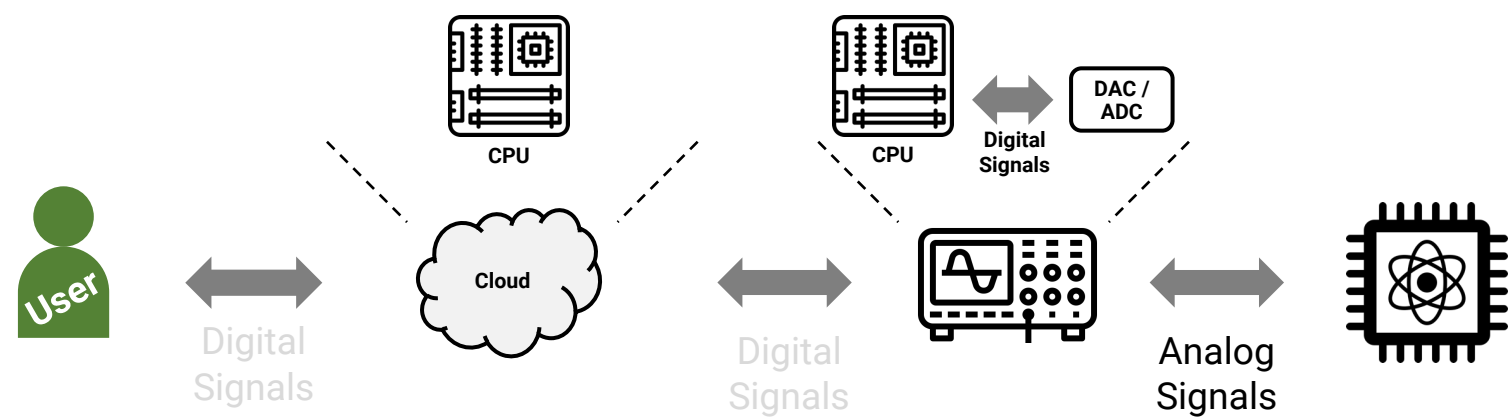
Blind Quantum Computation 

QC-TEE & CASQUE 

New Solution: Trusted Controller 

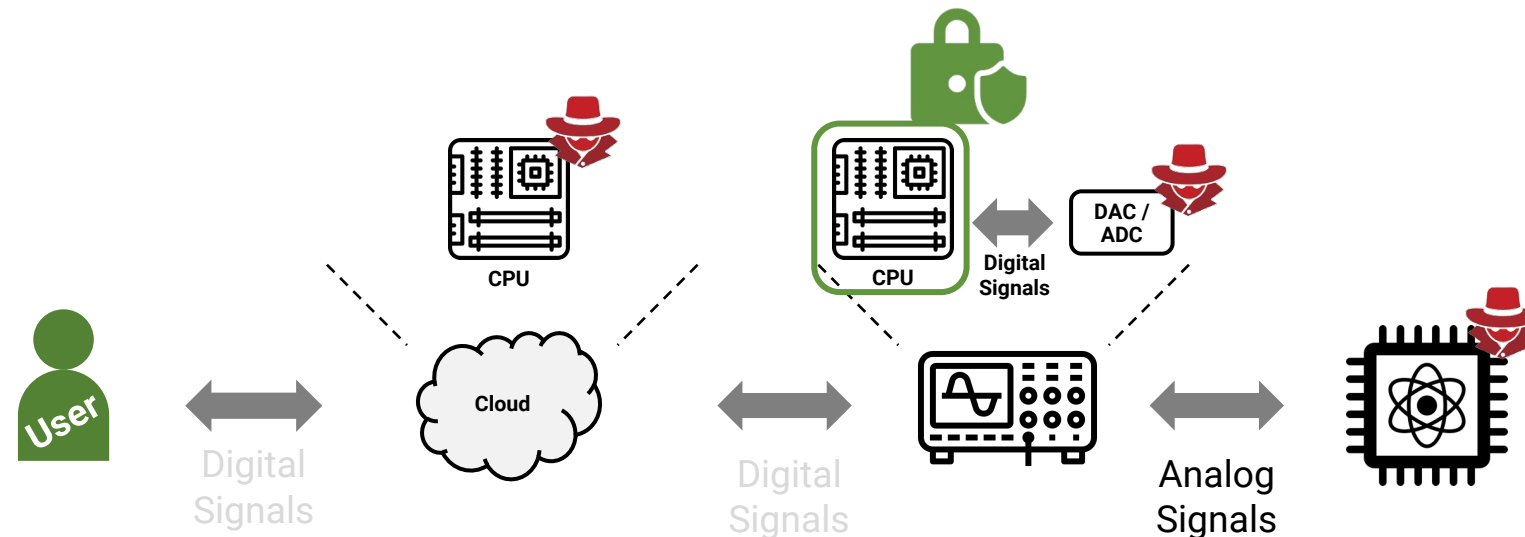
Cloud (CPU)	Controller (CPU)	Controller (DAC)	QPU
Untrusted	Untrusted	Untrusted	Untrusted
Untrusted	Untrusted	Untrusted	Trusted
Untrusted	Untrusted	Trusted	Untrusted
Untrusted	Untrusted	Trusted	Trusted
Untrusted	Trusted	Untrusted	Untrusted
Untrusted	Trusted	Untrusted	Trusted
Untrusted	Trusted	Trusted	Untrusted
Untrusted	Trusted	Trusted	Trusted

Cloud (CPU)	Controller (CPU)	Controller (DAC)	QPU
Trusted	Untrusted	Untrusted	Untrusted
Trusted	Untrusted	Untrusted	Trusted
Trusted	Untrusted	Trusted	Untrusted
Trusted	Untrusted	Trusted	Trusted
Trusted	Trusted	Untrusted	Untrusted
Trusted	Trusted	Untrusted	Trusted
Trusted	Trusted	Trusted	Untrusted
Trusted	Trusted	Trusted	Trusted



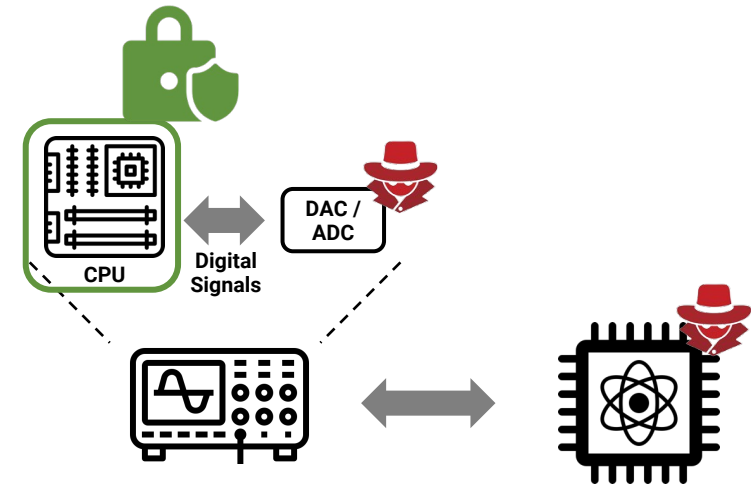
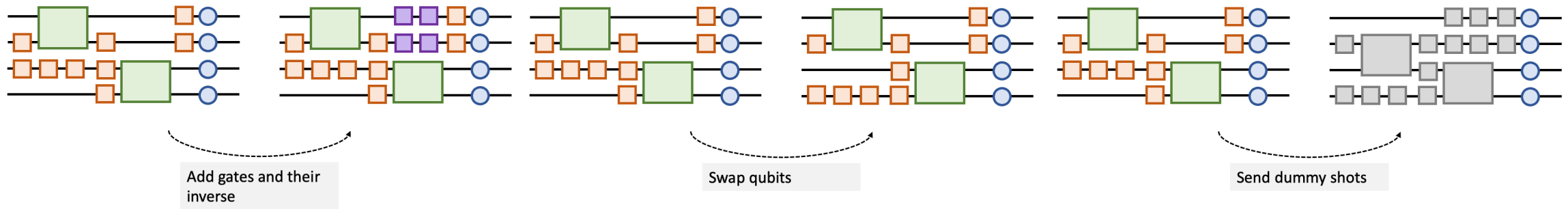
Quantum Program Obfuscation with Trusted Controller

- Demonstrate how trusted controller could provide protections from untrusted cloud provider or malicious insiders
- Leverage existing classical security solutions such as Intel SGX
 - Avoid modifications to QPU
 - Avoid assumptions about trusted QPU
 - Use well-studied Intel SGX



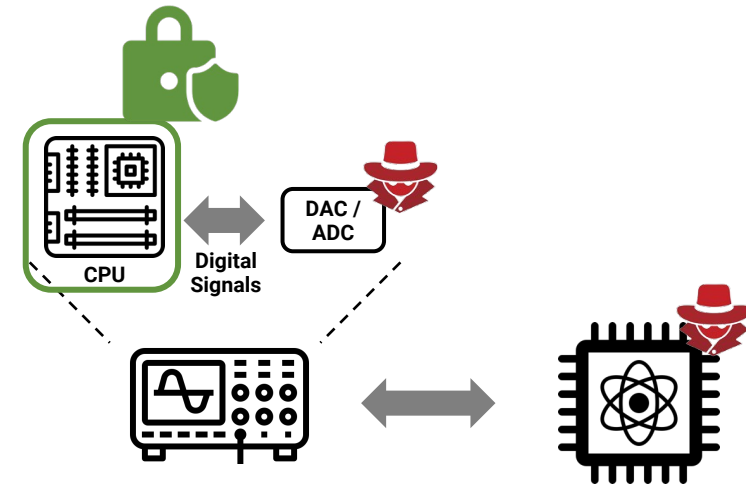
Trusted Controller's Possible Operations

- Adversary (cloud provider) can see all the control pulses leaving the (trusted) controller, limiting what the controller can do
- Possible operations done by the trusted controller:



Informal Security Analysis

- Cloud provider always sees the output and has total knowledge of the pulses going to the QPU
- Informally, at best $O(n)$ complexity for attacker to guess the program, n is number of shots
- Still some benefits
 - Increase n by mixing shots from different users
 - Send different shots to different QPUs
 - Future ideas, leverage metamorphic properties of programs to further hide the true computation
- ... and leverage existing classical security solutions such as Intel SGX
 - Avoid modifications to QPU
 - Avoid assumptions about trusted QPU
 - Use well-studied Intel SGX



Tutorial Outline

- 1:00pm – 1:30pm: Introduction and Motivation
- 1:30pm – 2:30pm: Crosstalk and Fault Injection in NISQ Quantum Computers
- 2:30pm – 3:00pm: Coffee Break
- 3:00pm – 3:45pm: Side Channels in NISQ and FTQC Quantum Computers
- **3:45pm to 4:30pm: Trusted Execution Environments for Quantum Computers**
 - **Design of QC-TEEs for Quantum Computers**
- **4:00pm: End**

