Classically Verifiable Quantum advantage from a computational bell test

BY -

Apurva Dhingra (Phd AI and ML @ IIT Bombay, India)

Barnokhon Tashpulotova(BTech CS @ Uni of Washington)

Kush Dhuvad(MSc Physics @ IIT Jodhpur)

Submission Date : 9 August,2024

Efficiently Verifiable Quantum Computational Advantage

Key Points:

Current Limitations in Quantum Advantage:

Existing demonstrations require exponentially costly classical computations for verification.

Proposed Interactive Protocol:

A new protocol for quantum computational advantage that is efficiently verifiable using classical methods.

Trapdoor Claw-Free Functions:

The protocol uses cryptographic tools known as trapdoor claw-free functions.

Unlike previous approaches, this protocol eliminates the need for the adaptive hardcore bit property.

Connection to Bell's Inequality:

The protocol employs a novel connection to Bell's inequality.

This allows for simplified cryptographic assumptions without increasing quantum circuit complexity.

Innovative Trapdoor Claw-Free Function Constructions:

Constructions based on: **Rabin's Function**

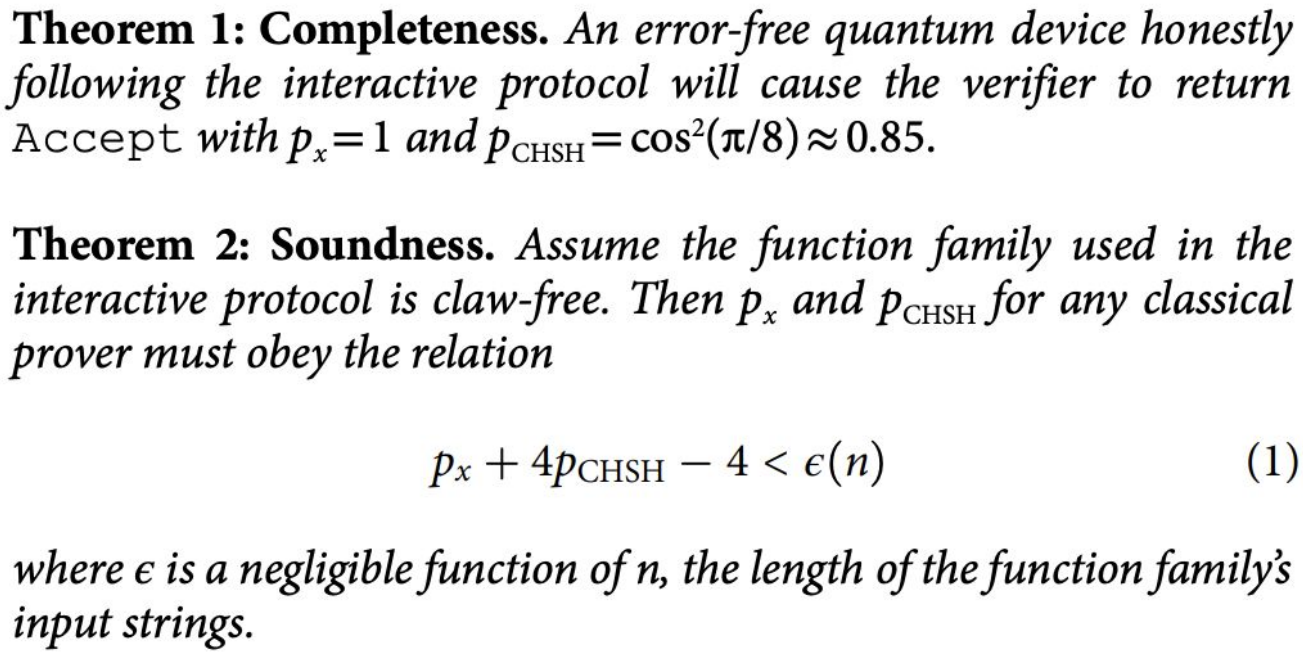
Trap-door Claw-free functions

- An interactive protocol was introduced that serves as both a test for quantum advantage and a generator of certifiable quantum randomness. The protocol's foundation is a two-to-one function, f.

- "**Claw-free**" refers to the property that it is computationally difficult to find a pair of inputs (x0 , x1) such that f(x0) = f(x1).

- The "**trapdoor**" property means that, given some secret data t, it becomes possible to efficiently invert f and reveal the pair of inputs corresponding to any given output.

Test which only Quantum Computers can pass



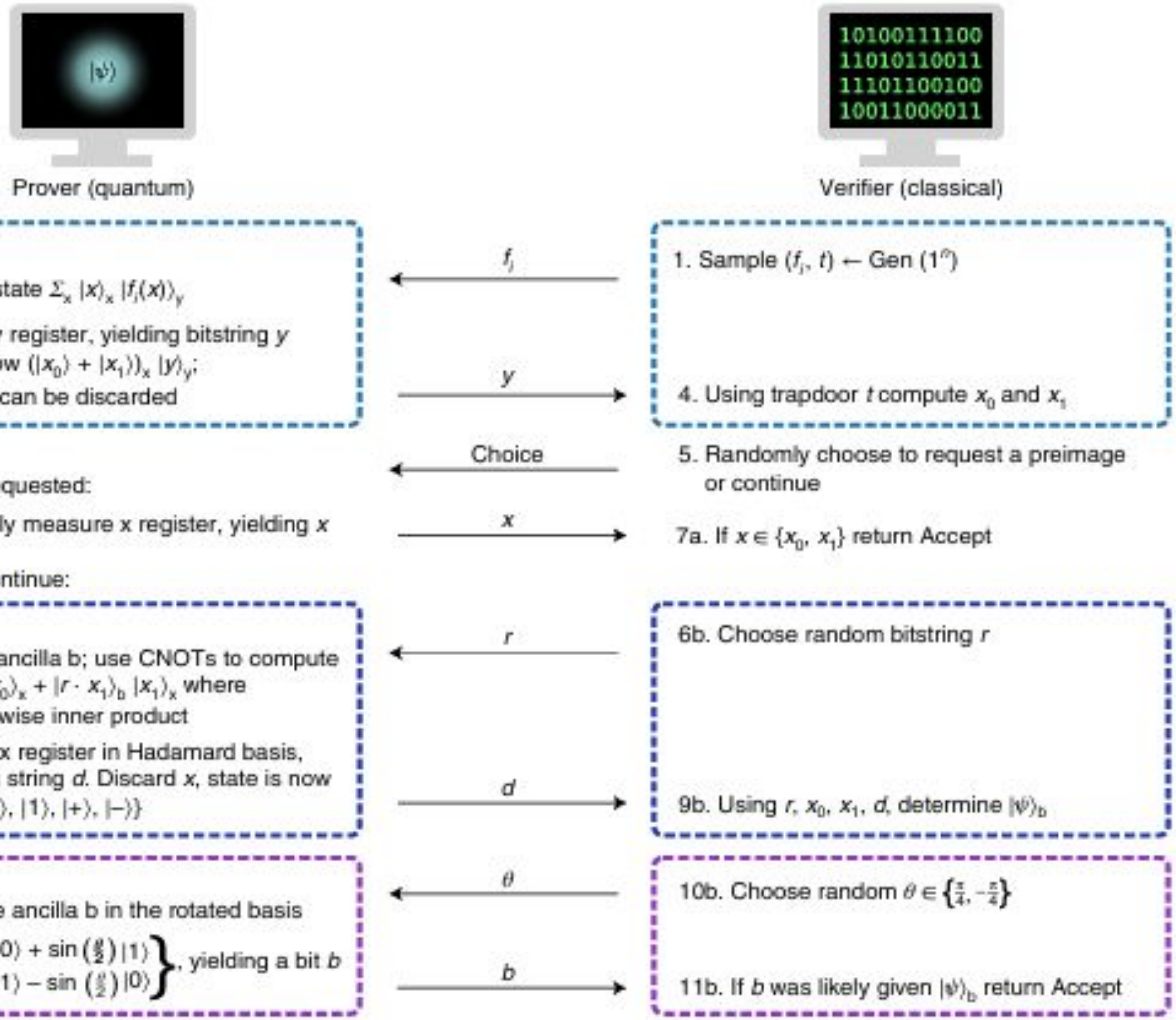
Theorem 1: Completeness. *An error-free quantum device honestly following the interactive protocol will cause the verifier to return* `Accept` *with* $p_x = 1$ *and* $p_{\text{CHSH}} = \cos^2(\pi/8) \approx 0.85$.

Theorem 2: Soundness. *Assume the function family used in the interactive protocol is claw-free. Then* p_x *and* p_{CHSH} *for any classical prover must obey the relation*

$$p_x + 4p_{\text{CHSH}} - 4 < \epsilon(n) \tag{1}$$

where ϵ *is a negligible function of* n*, the length of the function family's input strings.*

Protocol



Prover (quantum)

Verifier (classical)

Round 1

1. Sample $(f_i, t) \leftarrow \text{Gen}(1^\eta)$

2. Generate state $\Sigma_x |x\rangle_x |f_i(x)\rangle_y$

3. Measure y register, yielding bitstring y
State is now $(|x_0\rangle + |x_1\rangle)_x |y\rangle_y$;
y register can be discarded

4. Using trapdoor t compute x_0 and x_1

5. Randomly choose to request a preimage or continue

If preimage requested:

6a. Projectively measure x register, yielding x

7a. If $x \in \{x_0, x_1\}$ return Accept

Otherwise, continue:

Round 2

6b. Choose random bitstring r

7b. Add one ancilla b; use CNOTs to compute
$|r \cdot x_0\rangle_b |x_0\rangle_x + |r \cdot x_1\rangle_b |x_1\rangle_x$ where
$r \cdot x$ is bitwise inner product

8b. Measure x register in Hadamard basis, yielding a string d. Discard x, state is now
$|\psi\rangle_b \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$

9b. Using r, x_0, x_1, d, determine $|\psi\rangle_b$

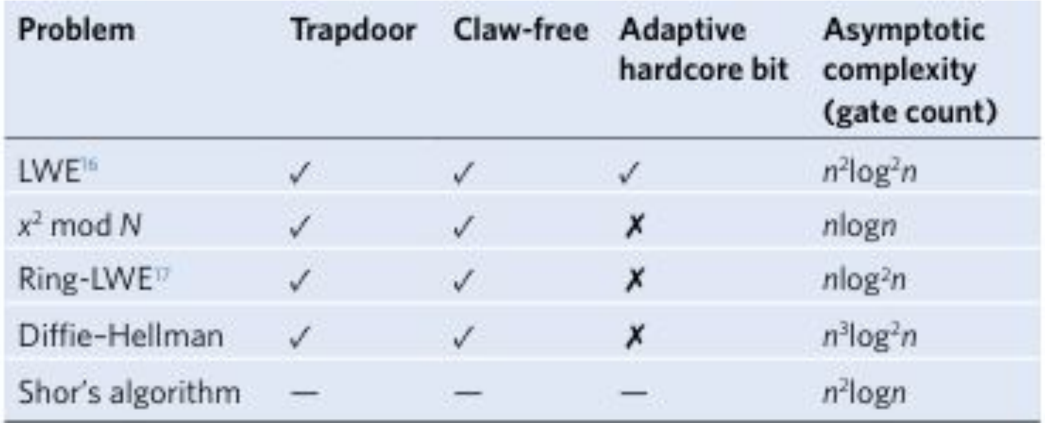
Round 3

10b. Choose random $\theta \in \left\{\frac{\pi}{4}, -\frac{\pi}{4}\right\}$

11b. Measure ancilla b in the rotated basis
$\left\{\begin{matrix} \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)|1\rangle \\ \cos\left(\frac{\theta}{2}\right)|1\rangle - \sin\left(\frac{\theta}{2}\right)|0\rangle \end{matrix}\right\}$, yielding a bit b

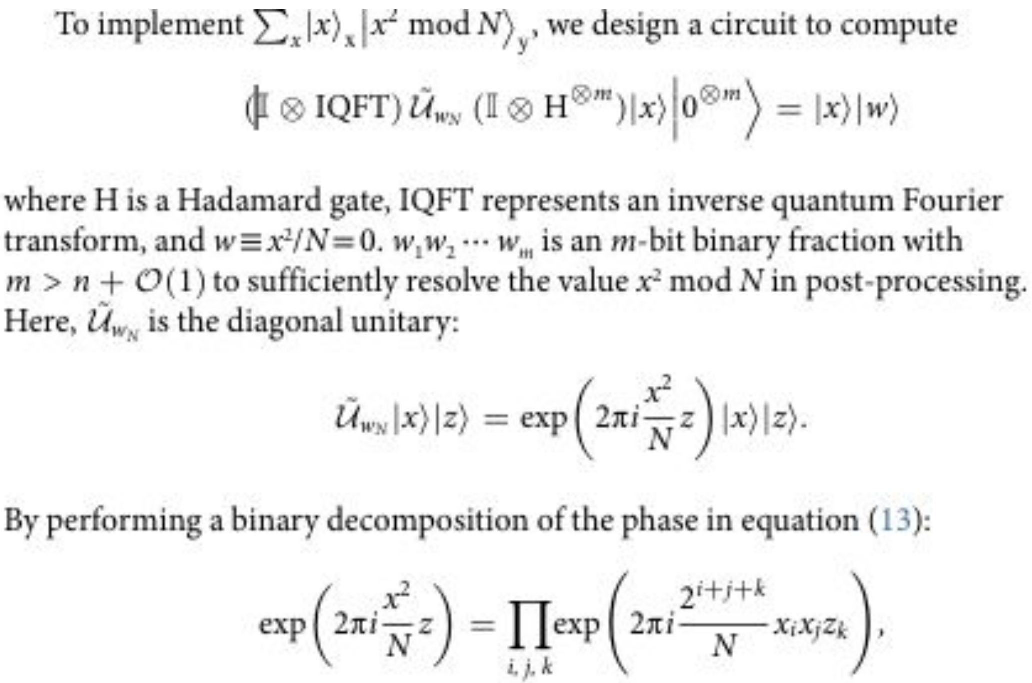
11b. If b was likely given $|\psi\rangle_b$ return Accept

Cryptographic constructions for interactive quantum advantage protocols



Problem	Trapdoor	Claw-free	Adaptive hardcore bit	Asymptotic complexity (gate count)
LWE[16]	✓	✓	✓	$n^2\log^2 n$
x^2 mod N	✓	✓	✗	$n\log n$
Ring-LWE[17]	✓	✓	✗	$n\log^2 n$
Diffie–Hellman	✓	✓	✗	$n^3\log^2 n$
Shor's algorithm	—	—	—	$n^2\log n$

Implementation Phase circuits for x^2 mod N (Quantum Phase estimation QPE)



To implement $\sum_x |x\rangle_x |x^2 \bmod N\rangle_y$, we design a circuit to compute

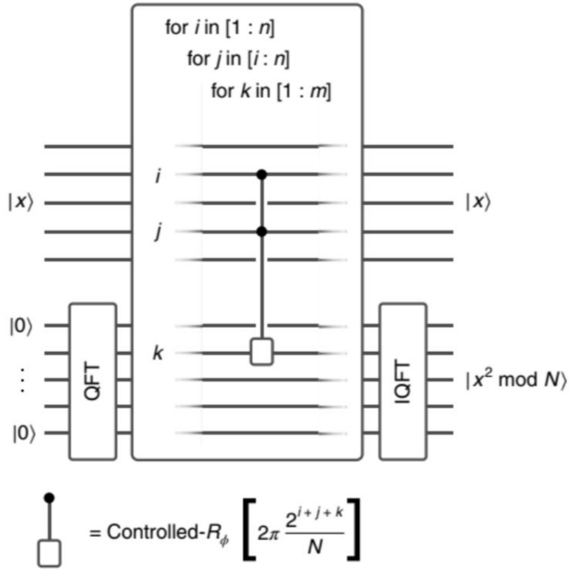
$$(\mathbb{I} \otimes \mathrm{IQFT})\, \tilde{\mathcal{U}}_{w_N}\, (\mathbb{I} \otimes \mathrm{H}^{\otimes m})|x\rangle \left|0^{\otimes m}\right\rangle = |x\rangle|w\rangle$$

where H is a Hadamard gate, IQFT represents an inverse quantum Fourier transform, and $w \equiv x^2/N = 0.\, w_1 w_2 \cdots w_m$ is an m-bit binary fraction with $m > n + \mathcal{O}(1)$ to sufficiently resolve the value $x^2 \bmod N$ in post-processing. Here, $\tilde{\mathcal{U}}_{w_N}$ is the diagonal unitary:

$$\tilde{\mathcal{U}}_{w_N}|x\rangle|z\rangle = \exp\left(2\pi i \frac{x^2}{N} z\right)|x\rangle|z\rangle.$$

By performing a binary decomposition of the phase in equation (13):

$$\exp\left(2\pi i \frac{x^2}{N} z\right) = \prod_{i,\, j,\, k} \exp\left(2\pi i \frac{2^{i+j+k}}{N} x_i x_j z_k\right),$$



Idea for a better Quantum Phase

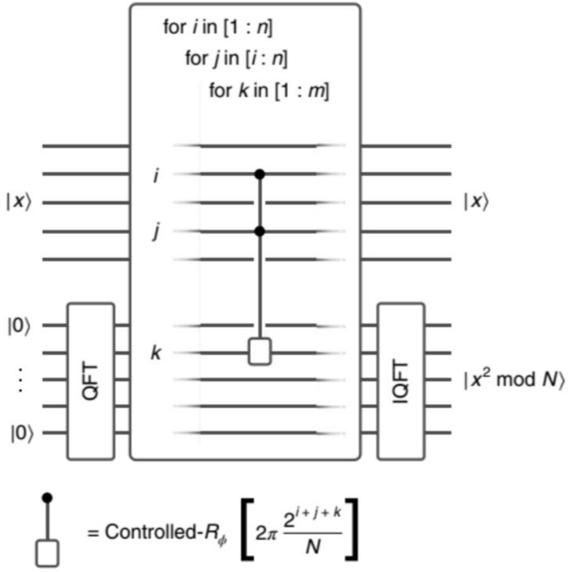
- Note when QFT is taken it is with respect to 2^$y.size$, where $y.size$ is the number of qubits in $|y\rangle$ register.

- For better Estimation from QPE, we propose using QFT_N (QFT with respect to N != 2^n.)

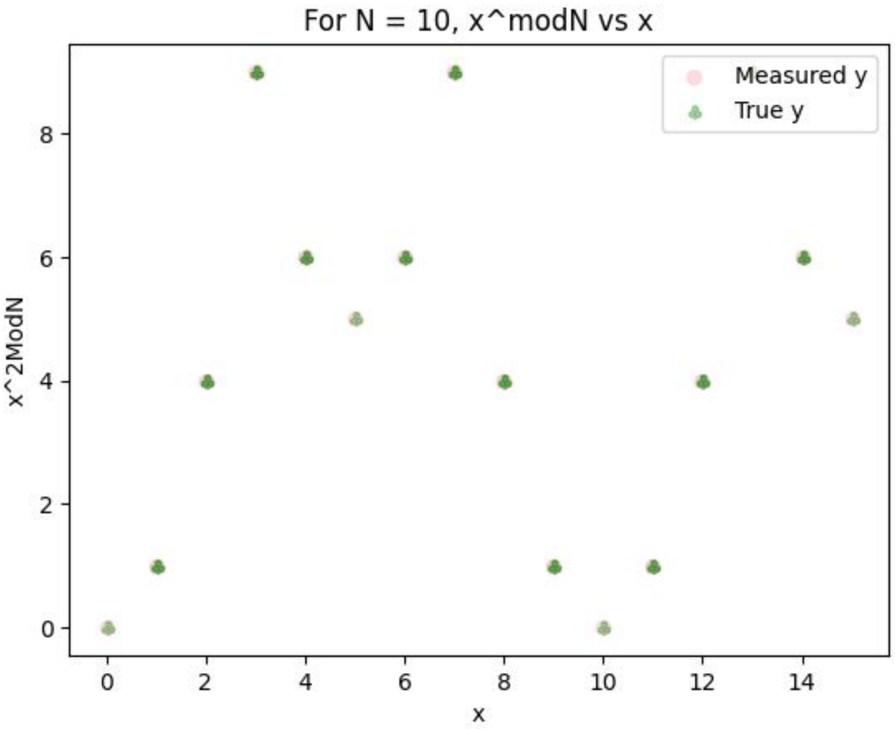
$$\frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} x_n \omega_N^{nk}, \qquad k = 0, 1, 2, \ldots, N-1,$$

$$\mathrm{QFT} : |x\rangle \mapsto \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} \omega_N^{xk} |k\rangle.$$

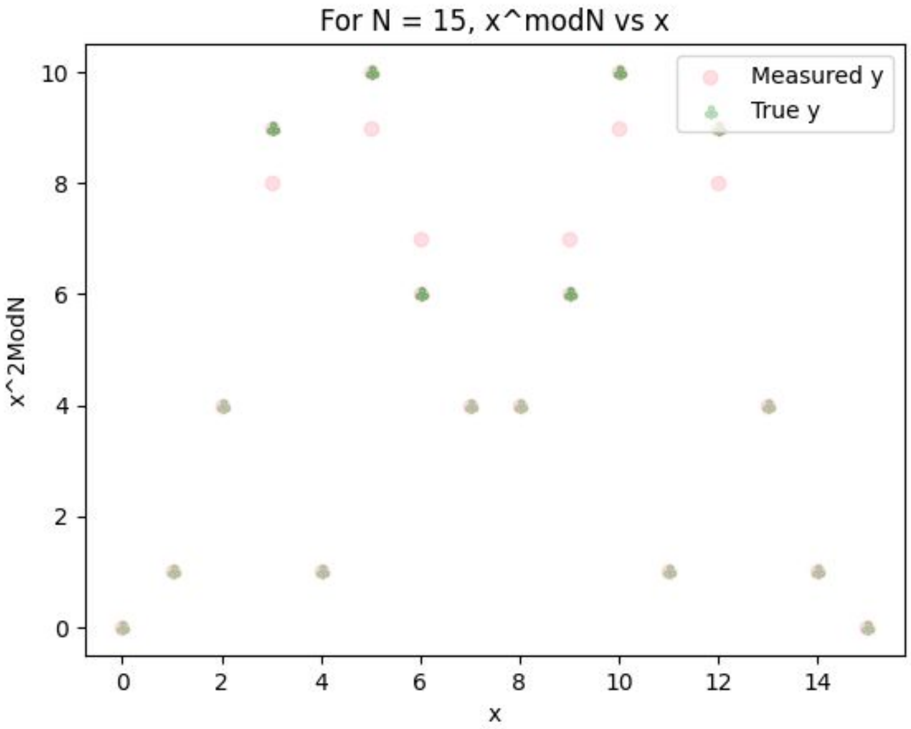
- We weren’t able to implement it during the Project duration, we wish to test it out when we can.



Results and analysis - Round 1


For N = 10, x^modN vs x

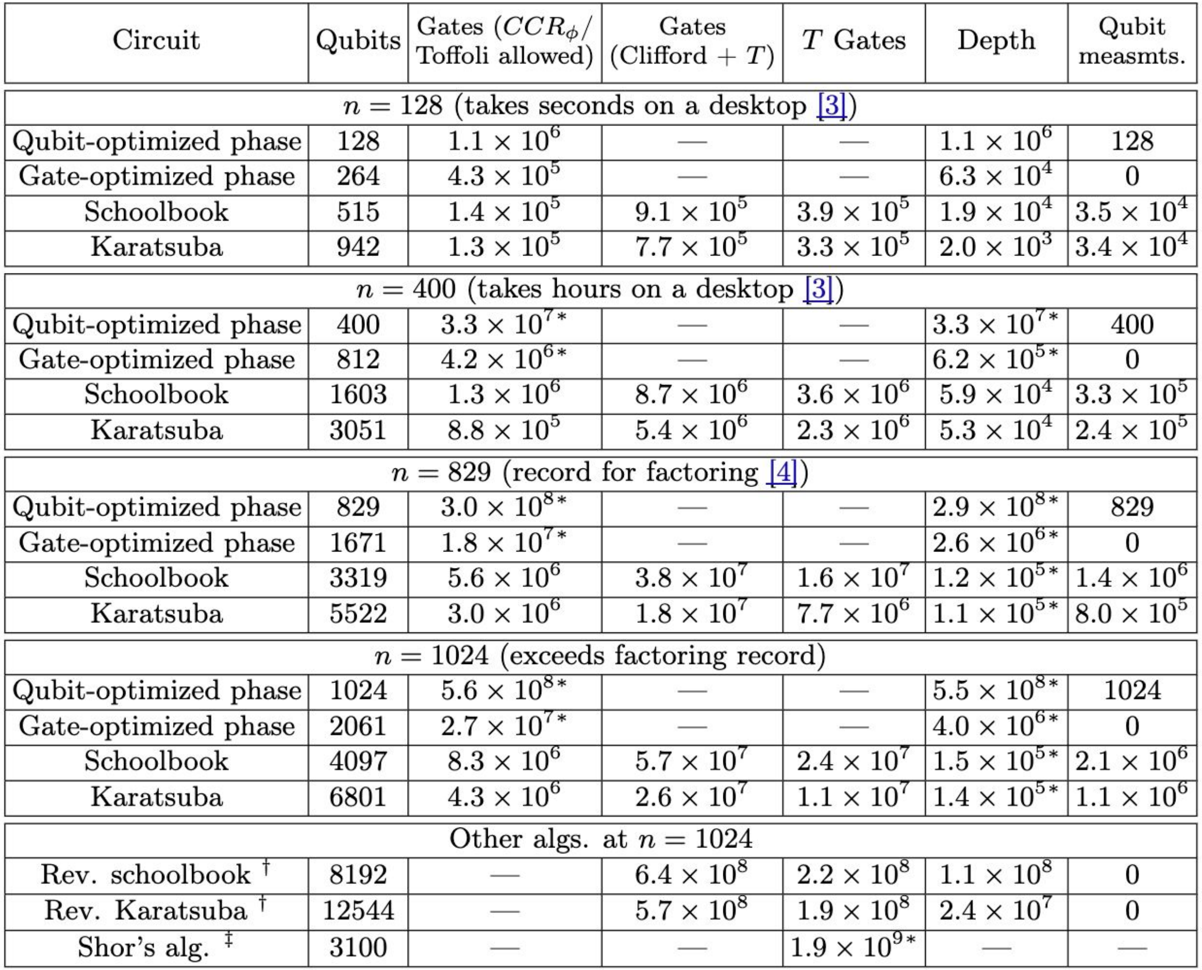
For $N =10$, the $ y = x^2 mod N$ values match the true values as shown.


For N = 15, x^modN vs x

For $N =15$, the $ y = x^2 mod N$ (Here 5 of the measured values differ from the true)

Opening:https://platform.classiq.io/circuit/b019810c-826e-4a7c-bf2c-e93516875fee?version=0.44.0

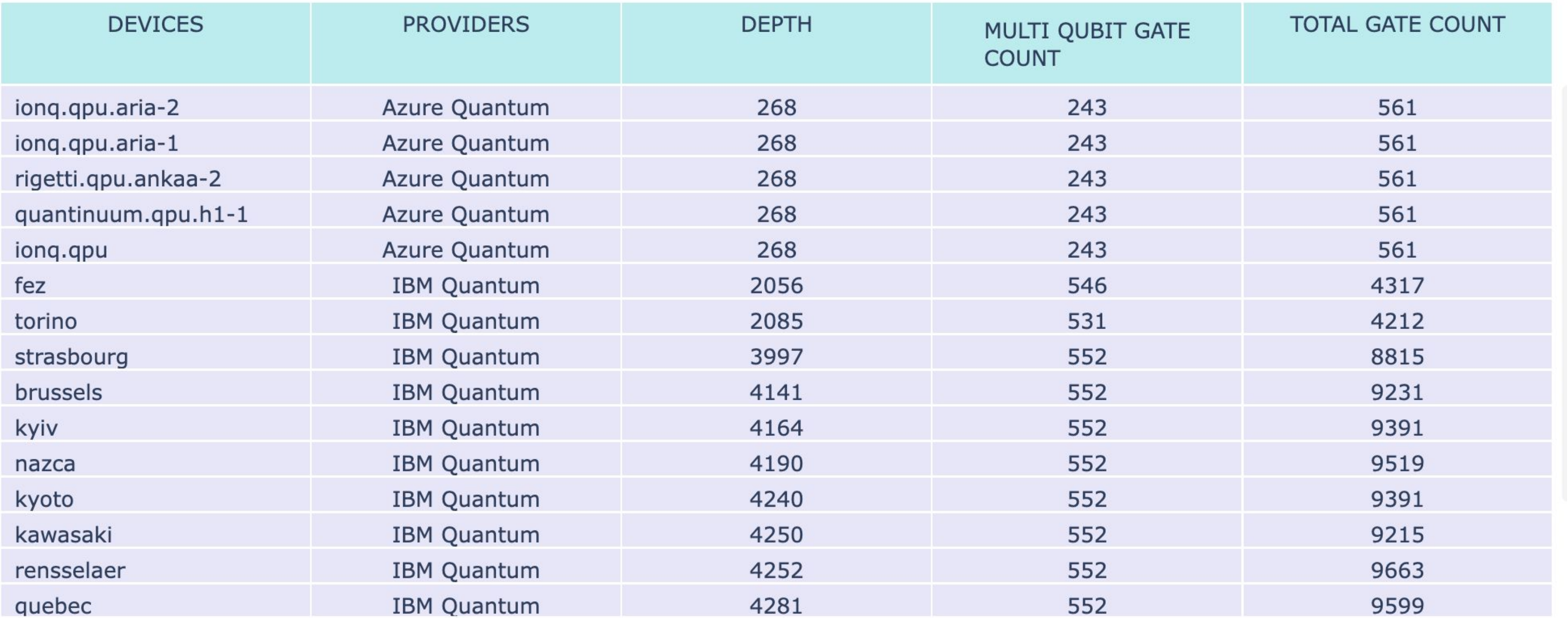
TABLE OF CIRCUIT SIZES*



Circuit	Qubits	Gates (CCR_ϕ/ Toffoli allowed)	Gates (Clifford + T)	T Gates	Depth	Qubit measmts.
$n = 128$ (takes seconds on a desktop [3])						
Qubit-optimized phase	128	1.1×10^6	—	—	1.1×10^6	128
Gate-optimized phase	264	4.3×10^5	—	—	6.3×10^4	0
Schoolbook	515	1.4×10^5	9.1×10^5	3.9×10^5	1.9×10^4	3.5×10^4
Karatsuba	942	1.3×10^5	7.7×10^5	3.3×10^5	2.0×10^3	3.4×10^4
$n = 400$ (takes hours on a desktop [3])						
Qubit-optimized phase	400	$3.3 \times 10^{7*}$	—	—	$3.3 \times 10^{7*}$	400
Gate-optimized phase	812	$4.2 \times 10^{6*}$	—	—	$6.2 \times 10^{5*}$	0
Schoolbook	1603	1.3×10^6	8.7×10^6	3.6×10^6	5.9×10^4	3.3×10^5
Karatsuba	3051	8.8×10^5	5.4×10^6	2.3×10^6	5.3×10^4	2.4×10^5
$n = 829$ (record for factoring [4])						
Qubit-optimized phase	829	$3.0 \times 10^{8*}$	—	—	$2.9 \times 10^{8*}$	829
Gate-optimized phase	1671	$1.8 \times 10^{7*}$	—	—	$2.6 \times 10^{6*}$	0
Schoolbook	3319	5.6×10^6	3.8×10^7	1.6×10^7	$1.2 \times 10^{5*}$	1.4×10^6
Karatsuba	5522	3.0×10^6	1.8×10^7	7.7×10^6	$1.1 \times 10^{5*}$	8.0×10^5
$n = 1024$ (exceeds factoring record)						
Qubit-optimized phase	1024	$5.6 \times 10^{8*}$	—	—	$5.5 \times 10^{8*}$	1024
Gate-optimized phase	2061	$2.7 \times 10^{7*}$	—	—	$4.0 \times 10^{6*}$	0
Schoolbook	4097	8.3×10^6	5.7×10^7	2.4×10^7	$1.5 \times 10^{5*}$	2.1×10^6
Karatsuba	6801	4.3×10^6	2.6×10^7	1.1×10^7	$1.4 \times 10^{5*}$	1.1×10^6
Other algs. at $n = 1024$						
Rev. schoolbook †	8192	—	6.4×10^8	2.2×10^8	1.1×10^8	0
Rev. Karatsuba †	12544	—	5.7×10^8	1.9×10^8	2.4×10^7	0
Shor's alg. ‡	3100	—	—	$1.9 \times 10^{9*}$	—	—

*From the paper

Gates and devices - Our Implementation



DEVICES	PROVIDERS	DEPTH	MULTI QUBIT GATE COUNT	TOTAL GATE COUNT
ionq.qpu.aria-2	Azure Quantum	268	243	561
ionq.qpu.aria-1	Azure Quantum	268	243	561
rigetti.qpu.ankaa-2	Azure Quantum	268	243	561
quantinuum.qpu.h1-1	Azure Quantum	268	243	561
ionq.qpu	Azure Quantum	268	243	561
fez	IBM Quantum	2056	546	4317
torino	IBM Quantum	2085	531	4212
strasbourg	IBM Quantum	3997	552	8815
brussels	IBM Quantum	4141	552	9231
kyiv	IBM Quantum	4164	552	9391
nazca	IBM Quantum	4190	552	9519
kyoto	IBM Quantum	4240	552	9391
kawasaki	IBM Quantum	4250	552	9215
rensselaer	IBM Quantum	4252	552	9663
quebec	IBM Quantum	4281	552	9599

Note : We didn’t implement any code for optimizing qubit number or number of gates.

Future direction (for our project)

- For better Estimation from QPE, we propose using QFT_N (QFT with respect to N != 2^n.)

- Implement code for optimizing qubit number and number of gates.

- Implement the protocol for Decisional Diffie-Hellman.

References

1. Kahanamoku-Meyer, G. D., Choi, S., Vazirani, U. V., & Yao, N. Y. (2022). Classically verifiable quantum advantage from a computational Bell test. *Nature Physics*, *18*(8), 918-924. (Link).

2. Code Implementation of the paper - (Link)

THANK YOU