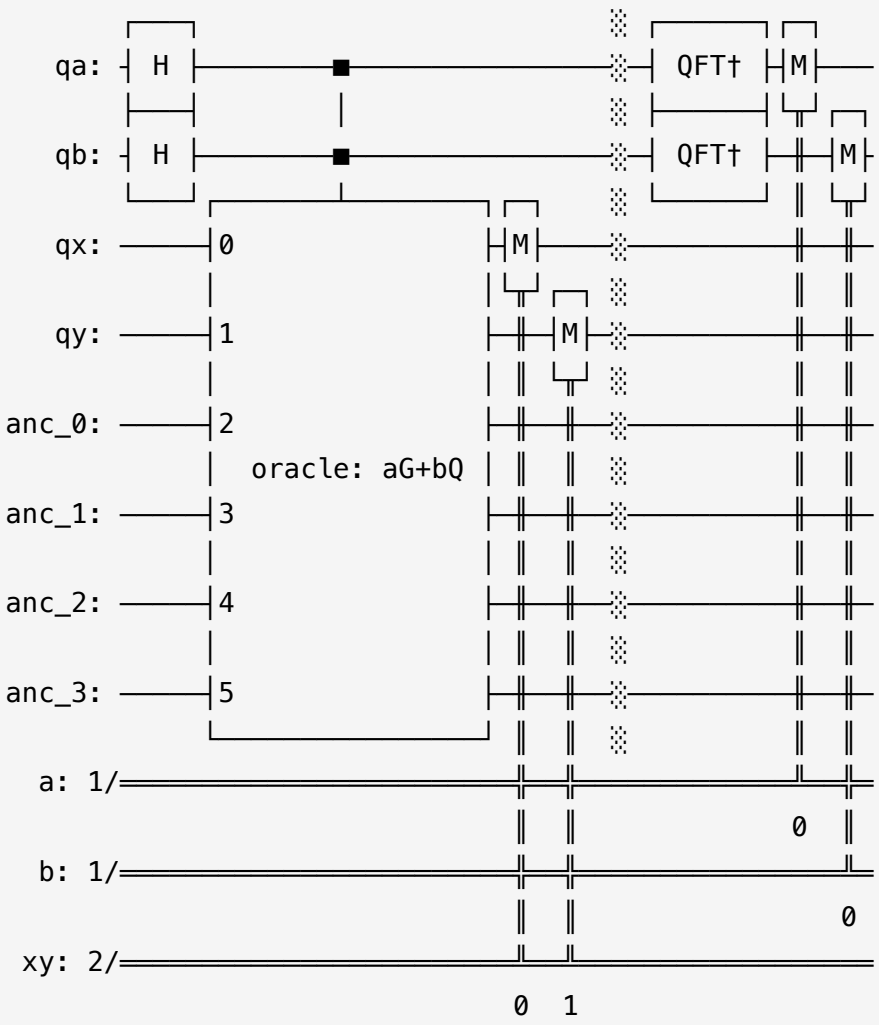
QDay Prize

Quantum Circuit Overview

The cryptography was broken by creating the following quantum circuit based on the Proos–Zalka algorithm.
Since the ECC parameter G and the public key $Q = dG$ are given, Shor's algorithm is used with $aG + bQ$ as the Oracle.



The contents of the quantum circuit are as follows:

1. qa and qb are prepared with the number of bits of the order n.
2. Apply H-gates to qa and qb to create a superposition.

3. Execute the quantum circuit that calculates $(qx, qy) = aG + bQ$.
4. Measure (qx, qy) to collapse it to a single coordinate.
5. Apply the inverse quantum Fourier transform ($QFT^{\dagger}$) to both qa and qb.
6. Finally, measure qa and qb.

The private key is obtained from the measured values of a and b using the following procedure.

$$\mathrm{len} = \lfloor \log_2 n \rfloor \tag{1}$$

$$x = \mathrm{floor}\left(\frac{a \cdot n}{2^{\mathrm{len}}} \right), \mathrm{ceil}\left(\frac{a \cdot n}{2^{\mathrm{len}}} \right) \tag{2}$$

$$y = \mathrm{floor}\left(\frac{b \cdot n}{2^{\mathrm{len}}} \right), \mathrm{ceil}\left(\frac{b \cdot n}{2^{\mathrm{len}}} \right) \tag{3}$$

$$d = x^{-1} \cdot y \pmod{n} \tag{4}$$

1. Let len be the number of bits of the order n.
2. Scale a to fit the range from 2^{len} to n.
3. Similarly, scale b.
4. Find the private key d by calculating the inverse of x, $x^{-1} = x^{n-2}$, and multiplying it by y.

Considering rounding errors during scaling, combinations of floor and ceiling (four types) are calculated for both x and y to determine if any of them satisfy the value for the private key.

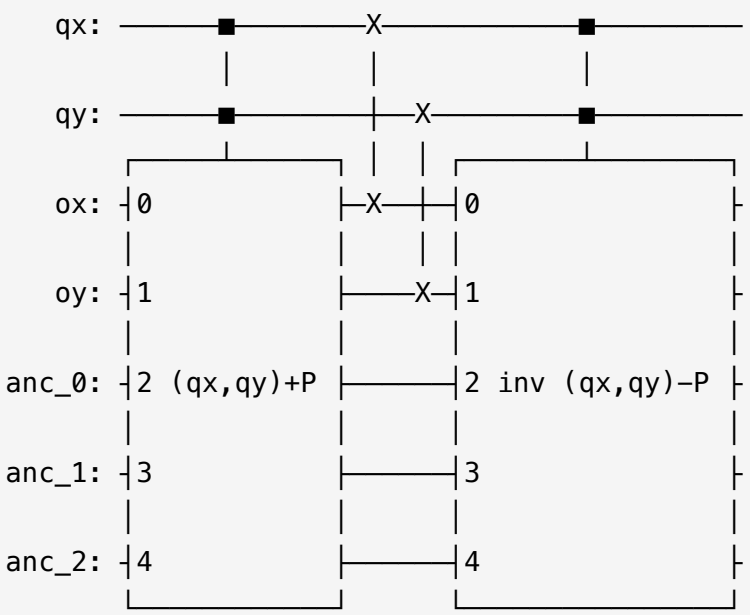
It was found that the probability of obtaining a measurement result that can lead to the correct answer is about 60% to 70% when the number of bits of the order is small, but it has been found that the correct measurement result can be obtained with a probability of 80% or more if the number of bits of the order is large.

Oracle Details

As shown below, the addition of coordinates in ECC is repeated with each bit of a and b as control bits.

$$|a_{n-1}...a_2a_1a_0\rangle\, G + |b_{n-1}...b_2b_1b_0\rangle\, Q = \sum_{i=0}^{n-1} a_i \cdot G^{2^i} + \sum_{i=0}^{n-1} b_i \cdot Q^{2^i}$$

The results of classical calculations for G^{2^i} and Q^{2^i} are applied to the quantum circuit.



- Calculate $(qx, qy) + P$ based on the input (qx, qy) and store the result in (ox, oy).
- Swap (qx, qy) and (ox, oy).
- Since $(ox, oy) = (qx, qy) - P$, executing the inverse circuit of $(qx, qy) - P$ cancels out (ox, oy), returning it to $|0...00\rangle$.
- For the quantum circuit, $(qx, qy) - P$ reuses the addition circuit as $(qx, qy) + (-P)$.

This addition circuit uses the following ancilla bits:

- Three flags (1 bit each) indicating the following states:
  - $(qx, qy) = O$
  - $(qx, qy) = -P$
  - $(qx, qy) = P$
- $dx := (qx - Px) \mod n$
- $dy := (qy - Py) \mod n$
- $dx^{-1} := dx^{n-2} \mod n$
  - Additional qubits are prepared to hold powers like $dx^2, dx^4, dx^8 \ldots$ up to the most significant bit of $n - 2$.

    - A temporary quantum register for calculations is also prepared.
- $\lambda := dx^{-1} \cdot dy \mod n$
- $\lambda \cdot ox$ (for calculating oy)
- A 1-bit carry bit (used in various `addr mod` calculations)

Required Resources

In this work, since it is configured with a general-purpose quantum circuit that works correctly for any ECC parameters, it requires a relatively large number of quantum gates.
For the ECC Curves and Keys presented this time, the required number of qubits and quantum gates[1] are as follows.
For reference, we also include the circuit creation time and simulation time measured on an iMac (Apple M1, 16GB memory).

ecc bits	prime	order	quantum bits	gate count[1]	circuit creation	simulation
4	13	7	58	107164	4.2s	1.5s
6	43	31	98	894838	43.4s	19.1s
7	67	79	130	1772688	88.6s	1m 22s
8	163	139	156	4213330	230s	10m 23s
9	349	313	184	12567301	756s	2h 22m 51s
10	547	547	214	13165145	856s	14h 19m 23s
11	1051	1093	246	28303785	2095s	112h 5m 23s

It can be estimated that breaking secp256k1, which is used in Bitcoin, would require more than 68,000 qubits and over a billion quantum gates.

---

[1] Since gates with multiple control bits are also counted as one quantum gate, the number of quantum gates will be much larger when transpiled for an actual quantum computer.