| Quantum Cryptography Challenge | | | |
|---|---|---|---|
| Sprawozdanie | | | |
| *Prowadzący:* <br> ***mgr inż. Jakub Hamerliński*** | *Autorzy:* <br> **Katarzyna Badio 145306** <br> **Julia Chabora 145218** | *Grupa dziekańska:* | ***L1*** |

# 1. Circuit design.

## 1.1 How the circuit is designed.

H applies a Hadamard transform to a single qubit, this gate puts qubits into a superposition. The problem is that it is hard to apply this gate to a control qubit, because the gate is part of 2 qubit state.

The solution is to expand the Hadamard matrix, by multiplying Hadamard gate and identity gate, denoted by I.

$$\mathrm{H} \otimes I = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

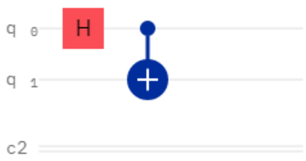
$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

$$|\Phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

$$|\Psi^+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}$$

$$|\Psi^-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

There are four bell states. The first one can be created by using H gate and CNOT, and it is shown below (Drawing 1.).




*Drawing 1. First bell state.*

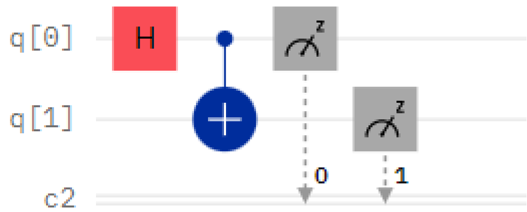
## 1.2 Code

```
// Alice's encoding function
function aliceEncoding() {
    const encodedBits = [];
    const measurementBases = [];
```

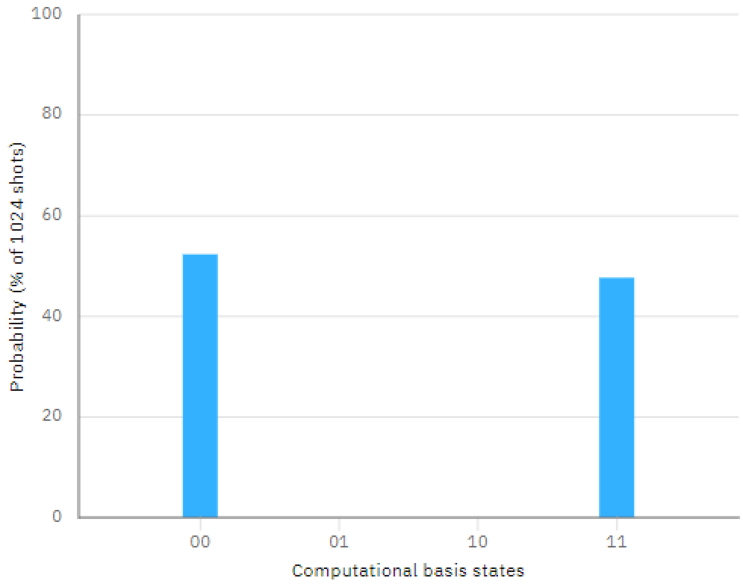
```
    for (let i = 0; i < numQubits; i++) {
        // Generate a random bit
        const bit = Math.round(Math.random());

        // Choose a random basis for encoding
        const basis = Math.round(Math.random()) ? 'Standard':'Hadamard';

        // Apply the chosen basis to the qubit
        const circuit = new Q.Circuit(1);
        if (basis === 'Standard') {
            circuit.x(0, bit); // Apply X gate (NOT gate) if bit is 1
        } else if (basis === 'Hadamard') {
            circuit.h(0);
            circuit.x(0, bit);
        }

        // Measure the qubit and record the result
        const measurement = circuit.measure(0);

        // Store the encoded bit and measurement basis
        encodedBits.push(measurement);
        measurementBases.push(basis);
    }

    return { encodedBits, measurementBases };
}

// Bob's decoding function
function bobDecoding(encodedBits, measurementBases) {
    const decodedBits = [];

    for (let i = 0; i < numQubits; i++) {
       // Choose a measurement basis based on Alice's communicated basis
        const basis = measurementBases[i];

        // Measure the received qubit using the chosen basis
        const circuit = new Q.Circuit(1);
        if (basis === 'Standard') {
            // No gate needed for standard basis
        } else if (basis === 'Hadamard') {
            circuit.h(0);
        }

        // Measure the qubit and record the result
        const measurement = circuit.measure(0, encodedBits[i]);
```

```
        // Store the decoded bit
        decodedBits.push(measurement);
    }

    return decodedBits;
}

// Main function to simulate BB84 protocol
function bb84Protocol(message) {
    // Alice's encoding
    const { encodedBits, measurementBases } = aliceEncoding();

    // Bob's decoding
    const decodedBits = bobDecoding(encodedBits, measurementBases);
```

1.3 Circuit




*Drawing 2. Hadamard gate at moment 1 on register 1 and Controlled-Not gate at moment 2, with its control component on register 1 and its target component on register 2.*




*Drawing 3. Probability results.*

## 2. How the encryption and decryption process works.

There are two bases to choose from, the Computational basis and Hadamard basis, where the base is a polarization state of single photons. In Computational basis, to encode one in a qubit, the X gate is applied, which is applying NOT. To encode 0, no action is needed.

In Hadamard basis, the Hadamard gate is applied and then the X gate is applied if a bit is equal to one. The qubits are sent from Alice to Bob and the base is saved for later.

To decrypt a message, first the base is applied to a message, and then there is measurement applied on a Computational basis.

## 3. Strategy for attempting to crack the encryption.

- Intercept - resend strategy

  Let assume that Eve is an eavesdropper and reads from the insecure channel the quantum state and the basis. If she measures the quantum state in a correct basis and sends it to Bob, he gets correct data. Otherwise, if she measures in another basis, there is a random outcome.

- Photon number splitting attack

  Eve can split the photons, and keep one to her and send one to Bob.

## 4. Discussion on quantum attacks on blockchain technology.

Reverse engineering for cryptographic hashing in classical computers is too costly, because it takes a lot of time and computational power. Quantum computers can overcome this. For example, there is Shor's algorithm where cryptographic keys associated with any public wallet can be figured out. There is also Grover's algorithm, which executes hash collision attacks and it finds two identical inputs that make the same hashes.