Quantum Computing: Password hacking Using Grover's Algorithm

Miyu Umemoto, Eli Kim, Junior Adoukonou, Keith Wallington

Motivation and Background

With the rapid growth of cryptocurrencies, managing complex passwords has become a serious challenge. Forgetting them can lock people out of valuable assets, and classical methods of searching passwords are too slow to be practical. This motivated me to explore the potential of quantum computing, especially Grover's algorithm, which can search more efficiently. Through this project, I aimed to deepen my understanding of quantum algorithms and improve my programming skills by implementing and simulating Grover's algorithm.

METHODS

We used Grover's algorithm to search for hidden passwords.

- **Steps in Grover's algorithm**

1. **Initialization**

   All qubits were placed into a superposition using Hadamard gates.

2. **Oracle Construction**

   We applied a phase inversion to the target state using the oracle.

3. **Diffuser Application**

   We amplified the amplitude of the target state using an inversion-about-average operator.

4. **Iteration**

   We repeated the oracle and diffuser steps to increase the likelihood of measuring the correct state.

5. **Simulation & Visualization**

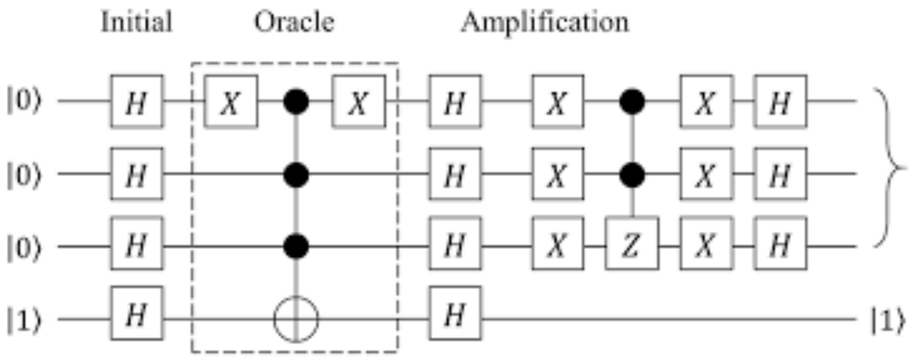
   We simulated the quantum circuit using Qiskit and visualized the measurement results.

6. **Generalization**

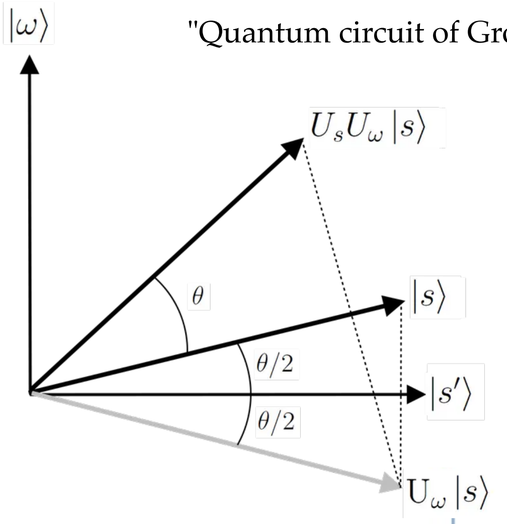
   We scaled the circuit from 2-bit (4 items) to 3-bit (8 items) search spaces to demonstrate Grover's algorithm can generalize to larger problems.

7. **Noise Testing**

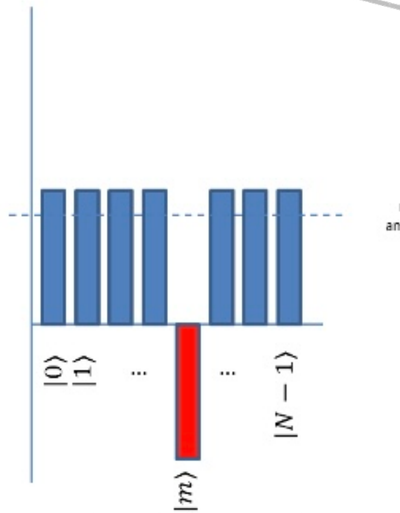
   We tested the circuit with noise using IBM's FakeVigo model to test its robustness in realistic conditions.



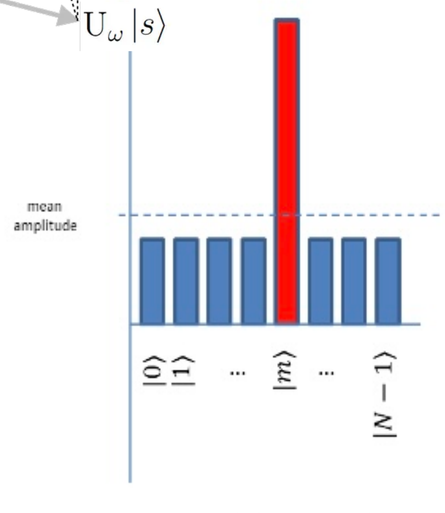
"Quantum circuit of Grover's algorithm"



"State rotation toward the target"



"Initial uniform distribution"



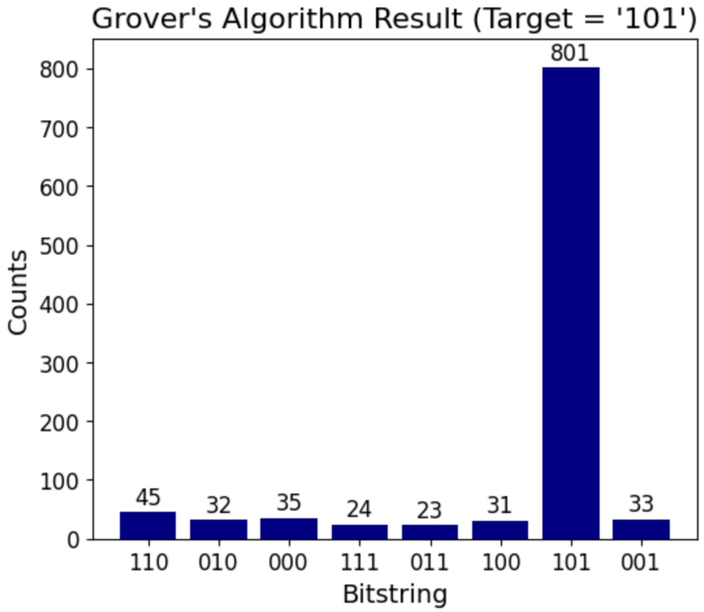
"Amplified marked state"

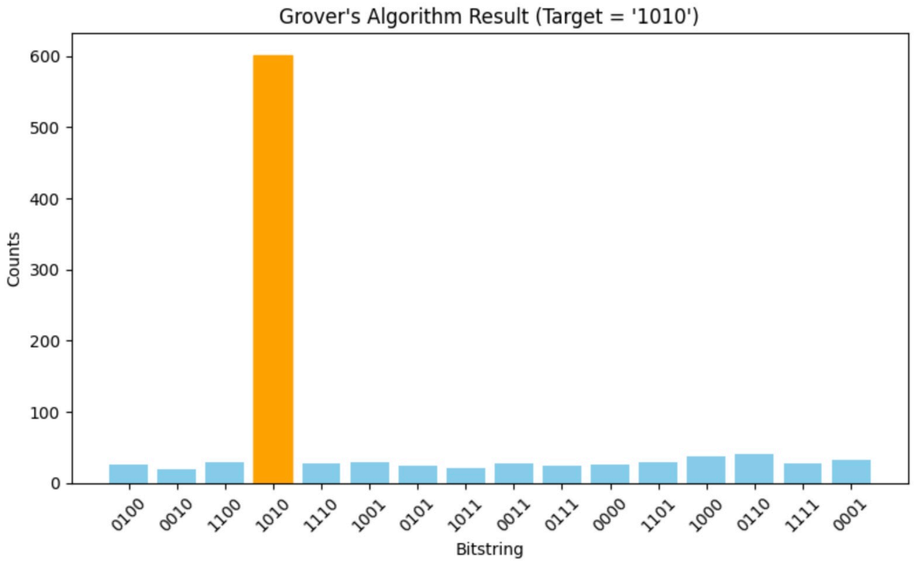
SIMULATIONS

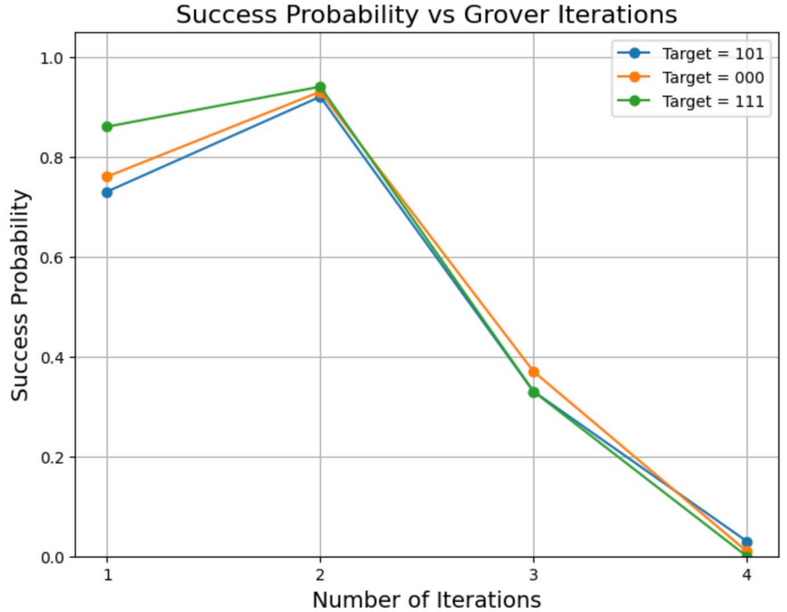
We built and tested Grover's circuits for 2-bit, 3-bit, and 4-bit password searches.

- The circuits were generalized to work for any password.
- Simulations were run using Qiskit on Google Colab.
- The number of iterations significantly affects the success rate.
- Different bit lengths require different numbers of steps.

(Each graph shows how probability of correct result increases after applying Grover's algorithm.)



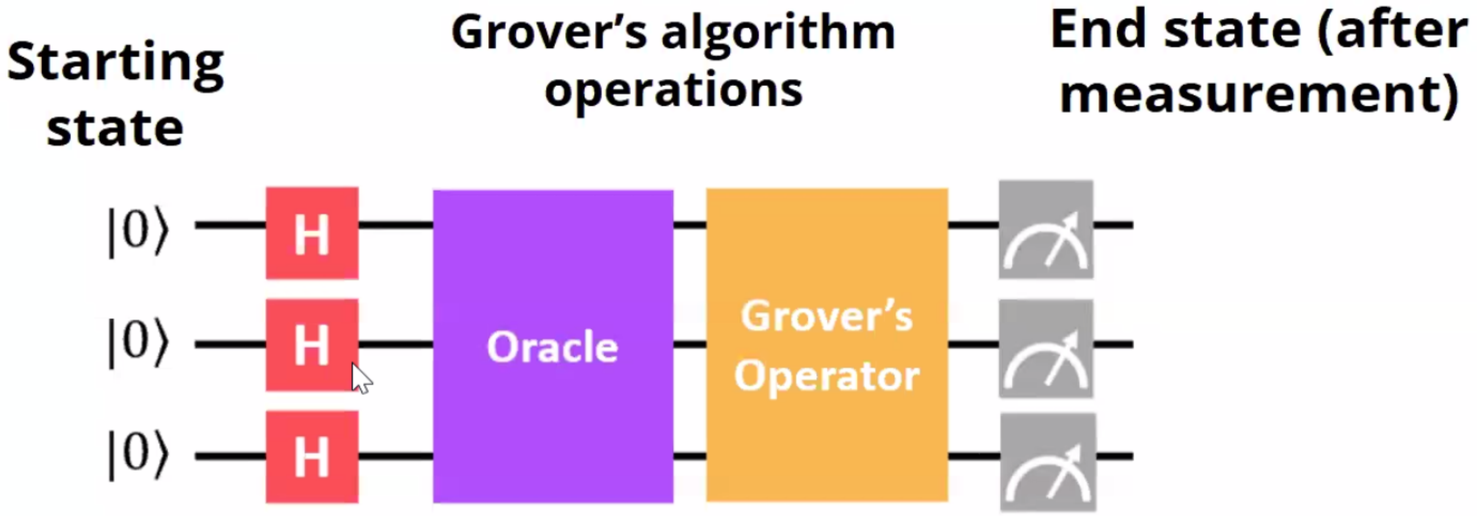


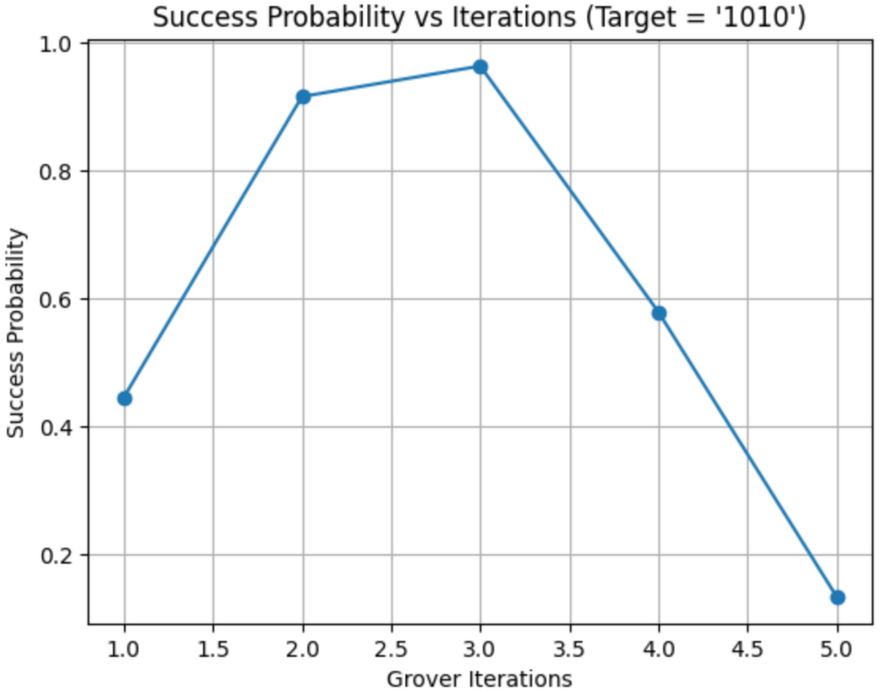


Key Findings & Conclusions

- ✓ Grover's algorithm successfully finds the password with high probability.
- ✓ It is more efficient than classical brute-force search.
- ✓ The number of iterations depends on the size of the search space.
- ✓ As the number of bits increases, fewer iterations are relatively needed (due to square-root scaling).

(Graph: Probability of success increases with number of iterations)





Reflections

In this project, I built quantum circuits to implement Grover's algorithm and ran simulations.

Through this, I deepened my understanding by experiencing quantum-specific phenomena such as superposition and interference.

I also observed how noise and the number of operations affect the success rate, realizing there is a significant gap between theory and practical implementation.

Potential Next Steps

- Extend Grover's algorithm to 4-bit passwords and analyze how the number of iterations affects success.
- Add noise using Qiskit to test how it changes the results.
- Compare with classical brute-force search to quantify quantum speedup.
- Build a simple UI to input a password and show results clearly.