Universität Regensburg, Institut für Theoretische Physik
Prof. Dr. Christoph Lehner, Simon Bürger

Summer 2019

Exercise: Quantum Computing
Problem set 6 (to be discussed in week of June 10, 2019)

Problem 1 Breaking RSA encryption

You are given an RSA public key

$$K_{\text{pub}} = (277, 302446877) \tag{1}$$

and the encrypted message "kpsgl". Using the Shor algorithm, find the private key K_{priv} and decrypt the message. You can use the following hints and shortcuts:

a) The message is encoded using an alphabet of 26 characters, where we map the letter "a" to 0, "b" to 1, ..., and "z" to 25. The most significant letter is written to the left. Example:

$$\text{"test"} \equiv (26^3 c_t + 26^2 c_e + 26 c_s + c_t) = 337135 \tag{2}$$

with $c_t = 19$, $c_e = 4$, $c_s = 18$.

b) From the public key you can recover the prime factors p and q discussed in the lecture. Do this using Shor's algorithm. You may replace the period finding sub-routine by a classical version. With this you can then reconstruct the private key and decrypt the message.

c) You are encouraged to write a program to perform these steps, however, you may also do them with pen-and-paper assuming that you randomly drew $a = 23221710$ in Shor's algorithm and that $a^{1095696} = 1 \bmod 302446877$ solves the period for this a.

d) For the greatest common divisor, you may use Euclid's algorithm and for the modular inverse you may use the extended version which we have both discussed in the demonstrations given in the lecture. You may look at the RSA.ipynb and Shor.ipynb workbooks in the sqc simulator for inspiration but should be able to explain and reproduce all steps.

e) You are guaranteed that the public key was generated using the algorithm discussed in the lecture, therefore you can skip the primality tests. (Explain why.)

Problem 2 Deutsch-Jozsa on real quantum hardware

We have implemented the Deutsch-Jozsa algorithm for $f(x) = x$ and $f(x) = 1$ using two qubits in sqc in the lecture. Implement the respective circuits on the IBM Q Experience at `https://quantum-computing.ibm.com/` and run them on both the simulator and one of the 5-qubit machines. We will demonstrate the usage of this system in the lecture, please see me if you have technical problems getting started.
You may also implement the algorithm in sqc and then use the operator.toQASM() function to obtain the circuit representation for the IBM machines.