Post-quantum cryptography

Introduction

Nowadays, with the rapid development of quantum computing technologies, traditional cryptographic systems that secure modern communications are facing an unprecedented threat. Some of the most reliable encryption methods that were previously intractable for classical computers (such as RSA and ECC (Elliptic Curve Cryptography)), which ensure the security of current digital infrastructures including online banking, data transmission, and digital signatures, are now at risk.

These systems rely on the computational hardness of mathematical problems like integer factorization and discrete logarithms, which could be efficiently solved by powerful quantum computers. Indeed, the emergence of quantum algorithms, such as Shor's algorithm, demonstrates that a sufficiently powerful quantum computer could efficiently factor large integers, effectively breaking RSA encryption. This potential capability threatens the confidentiality and integrity of nearly all existing communication systems.

Project Objectives

The objective of this project is to explore post-quantum cryptography. In particular, this project focuses on:

- Understanding how RSA works and why it is vulnerable to quantum attacks.
- Exploring classical and quantum approaches to breaking RSA (e.g., brute force, GNFS, and Shor's algorithm).
- Introducing the fundamental principles of quantum physics and how they relate to quantum computation.
- Implementing and testing simple quantum algorithms using Python and Qiskit, IBM's open-source quantum computing framework.
- Analyzing the implications of quantum computing on current data security and identifying alternative cryptographic methods for the post-quantum era.

Ultimately, this project aims to provide both theoretical and practical insights into how quantum computing challenges modern cryptography and to evaluate the transition toward quantum-resistant security mechanisms from the basics (prime numbers) to some

quantum programs using Qiskit library. The goal was to give you an insight into the state of the art in post-quantum cryptography and guide you through implementing some algorithms, but in a way that focuses more on learning and understanding than just completing tasks.

Project Execution

To run the project, first, read the README to set up the environnement correctly and from there you can start the lab just by following was it asked by the jupyter notebook. Really do it step by step (maybe 1 task a day) trying to understand what you do rather than just doing. You should complete the assignment in the notebook (.ipynb) called `assignment`. Enjoy !!

Verifying solutions

To verify your solutions, we have provided you a `correction` file. Please use this file whenever you are stuck or to check your work but do not copy paste the solutions without thinking about the problem and trying. Finally ensure that your implementation meets the lab requirements and that the `test` cells in the jupyter notebook compile and give you a correct result.

References

Website References

- Wikipedia. (n.d.) RSA cryptosystem. Available at: `https://en.wikipedia.org/wiki/RSA_cryptosystem`.
- Wikipedia. (n.d.) General number field sieve. Available at: `https://en.wikipedia.org/wiki/General_number_field_sieve`.
- Wikipedia. (n.d.) History of quantum mechanics. Available at: `https://en.wikipedia.org/wiki/History_of_quantum_mechanics`.
- Wikipedia. (n.d.) Quantum mechanics. Available at: `https://en.wikipedia.org/wiki/Quantum_mechanics`.
- Quantum Inspire. (n.d.) Superposition and entanglement. Available at: `https://www.quantum-inspire.com/kbase/superposition-and-entanglement/`.
- Wikipedia. (n.d.) Quantum computing. Available at: `https://en.wikipedia.org/wiki/Quantum_computing`.
- Wikipedia. (n.d.) Qubit. Available at: `https://en.wikipedia.org/wiki/Qubit`.

- IBM Quantum. (n.d.) Bits, gates and circuits. Available at: https://quantum.cloud.ibm.com/learning/en/courses/utility-scale-quantum-computing/bits-gates-and-circuits.
- Quantum Computing Stack Exchange. (n.d.) What is the difference between relaxation, dephasing, and decoherence? Available at: https://quantumcomputing.stackexchange.com/questions/8352/what-is-the-difference-between-relaxation-dephasing-and-decoherence.
- Quantum Computing Stack Exchange. (n.d.) Understanding theoretical computing power of quantum computers. Available at: https://quantumcomputing.stackexchange.com/questions/4652/understanding-theoretical-computing-power-of-quantum-computers.
- Quantum Untangled. (2019) Tensor products: linear algebra for quantum computing. Medium, 15 October. Available at: https://medium.com/quantum-untangled/tensor-products-linear-algebra-for-qc-8f7bb5020c6c.
- BlueQubit. (n.d.) Quantum computing hardware. Available at: https://www.bluequbit.io/quantum-computing-hardware.
- BlueQubit. (n.d.) Quantum computing simulators. Available at: https://www.bluequbit.io/quantum-computing-simulators.
- KTH Canvas. (n.d.) Assignment 346573. Available at: https://canvas.kth.se/courses/56029/assignments/346573.
- KTH Canvas. (n.d.) Tutorial: IBM Quantum Platform – Upgraded login, API key and your first run (Bell pair). Available at: https://canvas.kth.se/courses/56029/pages/tutorial-ibm-quantum-platform-upgraded-login-api-key-and-your-first-run-bell-pair?module_item_id=1256103.
- Wikipedia. (n.d.) Shor's algorithm. Available at: https://en.wikipedia.org/wiki/Shor%27s_algorithm.
- IBM Quantum. (n.d.) Shor's algorithm tutorial. Available at: https://quantum.cloud.ibm.com/docs/en/tutorials/shors-algorithm.
- IBM Quantum. (n.d.) Qiskit Circuit Library: Quantum Fourier Transform (QFT). Available at: https://quantum.cloud.ibm.com/docs/en/api/qiskit/qiskit.circuit.library.QFT.
- Google Security Blog. (2025) Tracking the cost of quantum factoring. 15 May. Available at: https://security.googleblog.com/2025/05/tracking-cost-of-quantum-factori.html (Accessed: 19 October 2025).
- Wikipedia. (n.d.) IBM Condor. Available at: https://en.wikipedia.org/wiki/IBM_Condor.
- Wikipedia. (n.d.) Elliptic-curve cryptography. Available at: https://en.wikipedia.org/wiki/Elliptic-curve_cryptography.

- Abhiveerhome, A. (2020) Building elliptic curve cryptography (ECC) from scratch. Medium, 25 June. Available at: https://medium.com/@abhiveerhome/building-elliptic-curve-cryptography-ecc-from-scratch-7b28e3b27531.
- Wikipedia. (n.d.) Bell state. Available at: https://en.wikipedia.org/wiki/Bell_state.

Image References

- Nobel Prize Educational. (n.d.) Quantised world figure 16 (GIF). Available at: https://educationalgames.nobelprize.org/educational/physics/quantised_world/w-p-images/fig16.gif.
- MIT Technology Review. (2019) Quantum explainer image. Available at: https://wp.technologyreview.com/wp-content/uploads/2019/07/quantumexplainer3.2-01-10.jpg?fit=1616,908.
- Qader, I. (2018) Hydrogen atom and superposition states figure. ResearchGate. Available at: https://www.researchgate.net/profile/Ibrahim-Qader/publication/323676617/figure/fig6/AS:602570158518280@1520675329231/a-Hydrogen-atom-in-0-state-b-The-state-1-c-Superposition-of-states-1-2.png.
- Medium.com. (n.d.). Hydrogen atom superposition states figure. Medium. Available at: https://miro.medium.com/v2/resize:fit:1200/1*Xjtz7XgDNSqTJz6rb7jwWg.jpeg.

PDF References

- Gasarch, W. (n.d.) The Number Field Sieve made easy. University of Maryland. Available at: https://www.cs.umd.edu/~gasarch/TOPICS/factoring/NFSmadeeasy.pdf.

Contributions

This lab has been created as a project of course DD2391 by Yann Lazrek, Thomas Muel, Mathis Ledouit and Vadim El Guedj.

Personal contributions

Yann Lazrek:

- *In the first part of the project, I explored the historical foundations of modern cryptography through the example of RSA. Before the 1970s, encryption relied mainly on symmetric keys, where both sender and receiver needed to share the same secret key. This raised a huge problem: how could two parties exchange the key securely without interception? It's in 1976 with Diffie and Hellman that an answerwas found. They introduced the idea of public-key cryptography which is a system using a public key for encryption and a private key for decryption. Building on this, in 1977, Rivest, Shamir, and Adleman proposed the RSA algorithm, which relies on the mathematical difficulty of factoring large prime numbers. This concept became one of the fundamental of modern secure communication, enabling online banking, digital signatures, and secure data exchange.*

  The second part of my work involved discovering quantum computers by setting up and running simple quantum algorithms using Qiskit, IBM's open-source framework for quantum programming. I implemented and tested the Quantum Random Number Generator (QRNG), one of the simplest examples of quantum computation. Unlike classical random number generators that rely on deterministic algorithms, the QRNG exploits the superposition principle of qubits to produce genuinely random results. This experiment allowed me to understand how quantum systems process information in fundamentally different ways from classical systems, offering a first step toward simulating and implementing more complex quantum algorithms in further parts.

  Finally, I introduced the question of whether our data is truly secure in the age of quantum computing. Although algorithms like Shor's algorithm theoretically threaten RSA by enabling efficient factorization, current quantum computers are not yet powerful enough to carry out such large-scale computations. However, their rapid progress poses an undeniable risk for future data security, motivating the need to develop post-quantum cryptography which are algorithms designed to resist both classical and quantum attacks.

Thomas Muel:

- *During this period, I attended a class on quantum computing and we used the Qiskit library. So I was really familiar with these concepts. This allowed me to assist others by explaining the principles of quantum computers, quantum mechanics, and how to use Qiskit. Even before this project, I was already interested in these topics, so I was able to provide guidance whenever needed and, in a way, supervise the overall progress of the project. Individually, I conducted research on the GNFS (General*

  Number Field Sieve) algorithm to understand how it works. I aimed to make this complex algorithm as easy to understand as possible for both my teammates and future students who might work on this lab. I then implemented the different steps of the algorithm in Python. I also wrote the section about quantum computers, explain-

  ing what a quantum computer is, what a qubit represents, and how quantum states are measured. In addition, I described quantum circuits and gates used to build algorithms, provided useful links for further reading, and highlighted key aspects of qubits, such as relaxation and decoherence times. Furthermore, I learned about post-

  quantum cryptography and aimed to implement a simple example using Bell pairs, which I had studied in the quantum computing course. I also wrote the README file, which explains how to set up a virtual environment, install all required dependencies, and start the project. Finally, in the same README, I included a short introduction on how to use Jupyter Notebook.

Mathis Ledouit:

- *Firstly I did the RSA Implementation part. I used the official algorithm description I found online to explain using my own words the steps for the students and I included the mathematical proof. Then I wrote a simple python implementation to give a more practical approach and permits students to experiment themselves this crypto algorithm.*

  Then I did the step by step part of the shor algorithm. It's explaining everything in the simplest way possible but without removing any important part or assumption needed for the shor algo steps so the students can understand as much as possible. I provided some functions in the assignement to help them because it would be too much to think about in order to implement it. Also, I provided some images to represent quantum circuits using gates to help them understand better this very theoritical part.

  Finally I did the elliptic curves part so we can explain a bit more what other crypto algorithms are used today and how they work in a simple way. There is also 1 question to give the students a bit more to do so they can practice as much as possible. Since it's more a "further reading" part, it's a bit short but I included a link "read more here" if anyone would be interested to learn more about that.

Vadim El Guedj:

- *I wrote the "Can we crack RSA" introduction, outlining the context and the importance of integer factoring for RSA security. I then studied several brute-force approaches applicable to RSA, comparing their principles and constraints. After selecting the approaches, I implemented the chosen methods and carried out practical tests to observe their behavior. These experiments allowed me to evaluate the relative performance of the approaches and identify their limitations under realistic conditions. Finally, I summarized and documented the results and observations.*

  I also contributed by explaining the key concepts of superposition, the no-cloning theorem, and entanglement. I described how superposition allows particles to exist in multiple states at once, how the no-cloning theorem prevents perfect copying of unknown quantum states, and how entanglement links the states of particles regardless of distance.

  I carefully studied Shor's algorithm and explained its principle with a clear and rigorous mathematical approach, showing how it can be used to factor large integers and, in principle, break RSA encryption with of course the help of a quantum computer.