Quantum Cryptography and Quantum Key Distribution Notes

Gergely Lendvay

February 2, 2026

Contents

Introduction

0.1 Course Info

0.1.1 Teachers

Konstantin Wernli – kwernli@imada.sdu.dk
William Mistegard – wem@imada.sdu.dk
Jurek Bleischwitz – jurek@imada.sdu.dk

0.1.2 Exercise Classes

1. Upload solutions before class
2. In class discussion of the sheets
3. Four best solutions are counted in the overall lecture grade

0.1.3 Exam

1. Four graded exercise sheets
2. Final oral exam with 10-12 minutes of presentation and 13-15 minutes of questions. 5 topics with 30 minutes of preparation time. Most likely exam date: June 9, 2026

0.1.4 Books

Quantum Key Distribution – Ramona Wolf

0.2 Plan

Introduction to classical cryptography and quantum key distribution (QKD) followed by a recap in quantum computing and quantum information theory. Learn about post-processing, security analysis, QKD in practice as well as post-quantum cryptography.

1 First Lecture

1.1 Classical Cryptography

Problem: Send information securely from Alice to Bob while preventing Eve from eavesdropping on their communication.

Examples of cryptographic algorithms:

1. RSA
2. SHA256
3. Caesar Cipher

How will quantum computing change modern day cryptography? Efficient prime factorization can break RSA encryption.

Example for quantum attack on the RSA encryption scheme. Invented by Rivest–Shamir–Adleman in 1977, and over 50% of the asymmetric cryptography uses it.

Find 3 integers n, e, d such that for all $x \in \{0, \dots, n-1\}$ we have

$$(x^e)^d \equiv x(\mod n)$$

In the first step of the algorithm, Alice sends integers n, e to Bob. Bob gets x Bob uses efficient polynomial to $y = x^e \mod n$ In the third step, Bob sends the encrypted message y back to Alice. Alice receives the encrypted message y from Bob. Alice will be able to decrypt the message because she has d: $y^d = (x^e)^d = x(\mod n)$. Even if Eve has n, e, y, she cannot decrypt the message, because computing the e-th root is very difficult.

To find n, e, d, we need to pick two random primes such that $2^{1023} < p, q < 2^{1024}$. Public key n is the multiples of p and q, which is a one way function, meaning that factoring n into its products p and q with classical algorithms will take an exponentially long time: $\exp(const(\log n)^{\frac{1}{3}}(\log\log n)^{\frac{2}{3}})$ Choose e such that $1 < e < \lambda(n) = km(p-1, q-1)$. $d = e^{-1} \mod \lambda(n) \Rightarrow x^{ed} = x$ mod n

Proof $m^{ed} = m \mod n \Leftrightarrow m^{ed} = m \mod p, m^{ed} = m \mod q$

$$m^{ed-1} = 1 \mod p$$

$ed - 1 = 0 \mod \lambda n = km(p-1, q-1) \Rightarrow ed - 1 = km(p-1)$ for some $k \in \mathbb{Z}$

$m^{ed-1} = m^{k(p-1)} = (m^{p-1})^k = 1^k = 1 \mod p$

Theorem If p is a prime, then $x^{p-1} = 1 \mod p$ Fermat's theorem

Knowing p and q, or having the ability to efficiently factor $n = pq$, we can break the RSA encryption. Using Shor's algorithm, this can be done in $O(\log n^3)$ time on a quantum computer. However, since we need twice the number of the RSA key length in qubits, with current day quantum computers, the encryption is still unbreakable. The difficulty in precision and error correction further delays the actual use of the Shor's algorithm in breaking modern-day encryption.

Even if RSA is safe for now, new encryption methods are needed to be developed that resist quantum attacks. This includes new quantum encryption schemes, such as Quantum Key Distribution (QKD) or different classical encryption schemes (post-quantum cryptography).

2 Second Lecture