Quantum Cryptography and Quantum Key Distribution Notes

Gergely Lendvay

February 5, 2026

Contents

List of Acronyms

AES	Advanced Encryption Standard
OTP	One-Time Pad
PQC	Post-Quantum Cryptography
QKD	Quantum Key Distribution
RSA	Rivest–Shamir–Adleman

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. Solutions must be uploaded before each class.
2. Exercise sheets are discussed during the classes.
3. The four best exercise solutions count towards the overall course grade.

0.1.3 Exam

1. Assessment is based on four graded exercise sheets, and
2. Final oral exam consisting of:
   - 10–12 minutes presentation,
   - 13–15 minutes discussion and questions,
   - 5 possible topics, with 30 minutes of preparation time.
3. Tentative exam date: June 9, 2026

0.1.4 Books

Quantum Key Distribution – Ramona Wolf

0.2 Plan

The course begins with an introduction to classical cryptography and quantum key distribution (QKD), followed by a recap of quantum computing and quantum information theory. Subsequent topics include post-processing, security analysis, practical implementations of QKD, and post-quantum cryptography.

1 First Lecture

1.1 Classical Cryptography

The fundamental problem of cryptography is to enable secure communication between two parties, traditionally called Alice and Bob, while preventing an adversary Eve from gaining any information about the transmitted message.

Classical cryptography includes examples, such as the Caesar cipher, while more modern schemes can be categorized either by symmetric or asymmetric schemes. Well-known examples include Advanced Encryption Standard (AES) and Rivest–Shamir–Adleman (RSA).

Plain Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cipher Alphabet	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Table 1: Caesar Cipher Table (Right Shift by 3)

Impact of Quantum Computing on Classical Cryptography One of the major implications of quantum computing is its ability to efficiently solve certain mathematical problems that are believed to be hard for classical computers. In particular, efficient prime factorization through Shor's algorithm would render widely used public-key cryptographic schemes such as RSA insecure.

1.2 RSA Encryption Scheme

The RSA algorithm was introduced in 1977 by Rivest, Shamir, and Adleman and is still widely used, making up approximately 50% of all asymmetric encryption schemes. The goal is to find 3 integers n, e, and d such that for all messages

$$x \in \{0, \ldots, n-1\},$$

the following holds:

$$(x^e)^d \equiv x \pmod{n}$$

Key Generation Two large random prime numbers p and q are chosen, typically satisfying

$$2^{1023} < p, q < 2^{1024}$$

The public modulus is defined as

$$n = pq$$

Let $\lambda(n)$ denote Carmichael's function,

$$\lambda(n) = \mathrm{km}(p-1, q-1)$$

An integer e is chosen such that

$$1 < e < \lambda(n), \qquad \gcd(e, \lambda(n)) = 1$$

The private key d is defined as the modular inverse of e modulo $\lambda(n)$:

$$d \equiv e^{-1} \pmod{\lambda(n)}$$

Encryption and Decryption The public key (n, e) is made available to everyone, while the private key d is kept secret. To send a message x, Bob computes the ciphertext

$$y \equiv x^e \pmod{n}$$

and sends y to Alice. Using her private key, Alice decrypts the message by computing

$$y^d = (x^e)^d \equiv x \pmod{n}$$

An eavesdropper Eve, even knowing n, e, and the encrypted message y, cannot efficiently recover x without access to d. This is because computing d requires knowledge of $\lambda(n)$, which in turn requires factoring n into p and q.

Proof Since $n = pq$,

$$m^{ed} \equiv m \pmod{p} \quad \text{and} \quad m^{ed} \equiv m \pmod{q}$$

Because

$$ed \equiv 1 \pmod{\lambda(n)},$$

there exists an integer k such that

$$ed - 1 = k(p-1)$$

Then, for $x \not\equiv 0 \pmod{p}$,

$$m^{ed-1} = m^{k(p-1)} = (m^{p-1})^k = 1^k \equiv 1 \pmod{p},$$

by Fermat's Theorem, which states that if p is a prime and $\gcd(m, p) = 1$, then

$$m^{p-1} \equiv 1 \pmod{p}$$

The same argument holds for q.

Security Assumptions The security of RSA relies on the computational hardness of factoring large integers. The best known classical factoring algorithms run in sub-exponential time,

$$\exp(c(\log n)^{1/3}(\log \log n)^{2/3}),$$

which is infeasible for sufficiently large n.

Quantum Threat Shor's algorithm shows that a quantum computer can factor integers in polynomial time,

$$O((\log n)^3)$$

This would completely break RSA if large-scale, fault-tolerant quantum computers become available.

Currently, practical limitations such as qubit count, noise, and error correction prevent Shor's algorithm from being applied. In practice, breaking RSA would require millions of qubits.

Even if RSA remains secure in the near term, cryptographic systems must be developed that are resistant to quantum attacks. This includes both new classical schemes (post-quantum cryptography) and quantum approaches such as Quantum Key Distribution (QKD).

2 Second Lecture

2.1 Provably Secure Cryptography

An encryption scheme is considered secure if its security can be mathematically proven.

Classical public-key schemes such as RSA are not proven secure against all efficient (polynomial-time) adversaries. Their security relies on unproven computational hardness assumptions, such as the difficulty of factoring large numbers.

One-Time Pad A fundamental example of provably secure encryption is the One-Time Pad (Vernam cipher, 1926).
Alice and Bob share a secret key

$$S_A = S_B \in \{0,1\}^n,$$

unknown to Eve, and encrypt a message $m \in \{0,1\}^n$ using bitwise XOR:

$$c_i = m_i \oplus (S_A)_i$$

Example:

$$S_A = S_B = 101, \qquad m = 011, \qquad c = 110$$

Bob decrypts the message using his key:

$$m_i' = c_i \oplus (S_B)_i = (m_i \oplus (S_A)_i) \oplus (S_B)_i = m_i \oplus \big((S_A)_i \oplus (S_B)_i\big)$$

If $S_A = S_B$, then

$$m_i' = m_i \oplus 0 = m_i$$

The One-Time Pad (OTP) satisfies

$$P[M = m \mid C = c] = P[M = m],$$

meaning that the ciphertext c reveals no information about the message m. This holds if the following conditions are satisfied:

1. The key is uniformly random
2. The key length equals the message length

3. The key is never reused

4. The key remains completely secret

Ideally, the shared key satisfies:

1. **Correctness:** $S_A = S_B$

2. **Uniform Randomness:** $P[S = S] = 2^{-n}$

3. **Secrecy:** Eve has no information about the keys S_A and S_B

In practice, small imperfections may occur:

$$P[S_A \neq S_B] < \varepsilon,$$

the key is statistically close to uniform, and the protocol aborts if eavesdropping is detected.

2.2 Quantum Key Distribution

Quantum Key Distribution enables two parties to generate a shared secret key with information-theoretic security, guaranteed by the laws of quantum mechanics. The most well-known protocol is BB84, introduced by Bennett and Brassard in 1984.

The adversary Eve may listen to the classical channel and interact with the quantum channel, but cannot violate the principles of quantum mechanics.

Quantum Transmission Phase Alice selects two random bit strings

$$a, b \in \{0,1\}^n,$$

where a is the raw key and b determines the encoding basis. Each bit a_i is encoded into a qubit $|\psi_i\rangle$ as follows:

1. Computational basis ($b_i = 0$):

$$0 \mapsto |0\rangle\,, \qquad 1 \mapsto |1\rangle$$

2. Diagonal/Hadamard basis ($b_i = 1$):

$$0 \mapsto |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \qquad 1 \mapsto |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Alice sends the qubits to Bob over the quantum channel.

Measurement Phase Bob independently chooses a random basis string

$$b' \in \{0,1\}^n,$$

and measures each received qubit in basis b'_i, obtaining outcomes a'_i.

- If $b'_i = b_i (0 \to |0\rangle)$, then $a'_i = a_i$ with probability 1.
- If $b'_i \neq b_i$, the measurement outcome is uniformly random and uncorrelated with a_i.

For example, if Alice sends $|0\rangle$ and Bob measures in the computational basis,

$$P_{|0\rangle}(0) = |\langle 0 \mid 0\rangle|^2 = 1, \qquad P_{|0\rangle}(1) = 0,$$

and therefore Bob obtains the correct bit with probability 1.
However, if Alice sends $|+\rangle$ but Bob measures in the computational basis,

$$P_{|+\rangle}(0) = |\langle 0 \mid +\rangle|^2 = \frac{1}{2}, \qquad P_{|+\rangle}(1) = \frac{1}{2},$$

Bob may only obtain the correct bit with probability $\frac{1}{2}$.

Sifting Alice and Bob publically compare the basis strings b and b' over the classical channel and discard all indices where $b_i \neq b'_i$. The remaining bits form the sifted key, for which ideally (in the absence of noise and eavesdropping)

$$a_i = a'_i$$

Example The two tables below illustrate one run of the BB84 protocol. Alice chooses random bits a and random bases b to prepare the quantum state ψ. Bob independently chooses random measurement bases b' and obtains outcomes a'.

a	0	1	1	1	0	1
b	1	0	1	1	0	1
$\boldsymbol{\psi}$	$\|+\rangle$	$\|1\rangle$	$\|-\rangle$	$\|-\rangle$	$\|0\rangle$	$\|-\rangle$

Table 2: Alice's side

b'	1	0	0	0	1	1
a'	0	1	1	0	1	1
Basis Match	Yes	Yes	No	No	No	Yes
Sifted Key	0	1	-	-	-	1

Table 3: Bob's side

In this example, the sifted key is 011.

Security Principle The security of QKD comes from fundamental quantum properties:

- **Measurement Disturbs the State:** Any attempt by Eve to measure the quantum states introduces detectable errors.
- **No-Cloning Theorem:** Unknown quantum states cannot be copied, preventing Eve from perfectly duplicating transmitted qubits.

3 Third Lecture