



CRYPTO READING GROUP

QUANTUM CRYPTOGRAPHY: QKD & QROM

Fall 2026



UNIVERSITY OF
WATERLOO

FACULTY OF
MATHEMATICS

Contents

About	3
Quantum Cryptography	3
Format	3
Useful Information	3
Organizers	3
Schedule	4
Fall 2026: QKD and QROM	4
Plan	5
Week 1: Quantum Computing	7
Classical Computing	7
Complexity Classes	8
Introduction to Quantum	8
Quantum Computing	13
Week 2: Quantum Information	14
Week 3: Quantum Circuits	15
Week 4: Quantum Key Distribution	16
Week 5: QKD Security	17
Week 6: Deploying QKD	18
Week 7: Classical Random Oracle Model	19
Week 8: Quantum Random Oracle Model	20
Week 9: QROM Lemmas	21
Week 10: QROM Security Proof	22
References	23

This document contains the plan and notes for the Fall 2026 quantum cryptography reading group.

Quantum Cryptography

This semester, the reading group focuses on *quantum cryptography*. We will build the background in quantum computing and quantum information needed to study its protocols and security proofs. The group is an opportunity to learn these topics together, including material that is new to many of us.

In the first half of the semester, we will study quantum key distribution (QKD), its security model, and the challenges of deploying it. The second half turns to the random-oracle model (ROM), the quantum random-oracle model (QROM), and a worked example of a QROM security proof.

Format

Each session will be prepared by the speaker or speakers listed in the schedule. The goal is to encourage collaboration and leave room for discussion during the presentation.

When two speakers are assigned, they may divide the talk or prepare it together with one person presenting. Some coordination with speakers from neighboring weeks may help connect the topics.

Useful Information

Talks. The talks will be held in the **Mathematics & Computer Science (MC)** building at the University of Waterloo, in room **MC 5501**, located on the fifth floor.

Lunches. We are also planning to organize group lunches every other week, alternating between the lounge and the plaza.

Organizers

Leonardo Colò University of Waterloo
Bruno Sterner University of Waterloo

Schedule

Fall 2026: QKD and QROM

	Fall 2026		
September 23, 2026	1	Maher Mamah Bruno Sterner	Introduction to Quantum Computing
September 30, 2026	2	Elnaz Hessami Pilehrood Owen Waldron	Quantum Information Theory
October 7, 2026	3	Mojtaba Fadavi Taha Hedayat	Quantum Circuits
October 14, 2026	Reading Week		
October 21, 2026	4	Maggie Simmons	Quantum Key Distribution
October 28, 2026	5	Douglas Stebila	QKD Security and Post-Quantum Cryptography
November 4, 2026	6	Aodhan Corrigan	Challenges in Deploying QKD
November 11, 2026	7	Huanhuan Chen Bruce Xu	Random Oracle Model and Classical Proofs
November 18, 2026	8	Jonas Janneck Camryn Steckel	QROM Definitions
November 25, 2026	9	Speaker to be confirmed	QROM Lemmas
December 2, 2026	10	Seunghoon Lee	Example Security Proof in the QROM
December 9, 2026	Ontario Crypto Days		

Introduction to Quantum Computing

Maher Mamah & Bruno Sterner

1

For the first week, we start with a motivation on quantum computing including the basic definitions and discussion on complexity classes. This will serve as a elementary foundation for the rest of the reading group.

References: [7] and [9]

Quantum Information Theory

Elnaz Hessami Pilehrood & Owen Waldron

2

Pure quantum states describe systems whose preparation is known exactly, but realistic quantum information often involves uncertainty, inaccessible subsystems, and interactions with the environment. In this talk, we introduce density operators as a general framework encompassing both pure and mixed states. We then discuss quantum channels, which model physical transformations and common forms of noise affecting quantum systems. Finally, we introduce classical and quantum entropy as measures of uncertainty and information, and explain how these ideas provide basic intuition for quantum cryptographic security.

References: [7] and [9]

Quantum Circuits

Mojtaba Fadavi & Taha Hedayat

3

Abstract

References: [7] and [9]

Quantum Key Distribution

Maggie Simmons

4

Abstract

References: [2], [3], and [5]

QKD Security and Post-Quantum Cryptography in QKD

Douglas Stebila

5

Abstract

References: [3], [5], and [10]

Challenges in Deploying QKD

Aodhan Corrigan

6

Abstract

References: [8] and [11]

The Random Oracle Model and Classical Security Proofs

Huanhuan Chen & Bruce Xu

7

Abstract

References: [6]

Definitions for the Quantum Random Oracle Model

Jonas Janneck & Camryn Steckel

8

Abstract

References: [12] and [13]

QROM Lemmas

Speaker to be confirmed

9

Abstract

References: [1], [4], and [13]

An Example Security Proof in the QROM

Seunghoon Lee

10

Abstract

References: [4]

Quantum Computing

Talk by Bruno and Maher.

Classical Computing

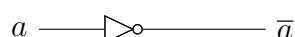
Our modern understanding of computers dates back to the idea of Turing machines. At an informal level a Turing machine consists of 4 main elements:

- Finite state control;
- Tape;
- Read-write tape head; and
- Program.

The finite state control consists of a finite set of internal states $q_1, \dots, q_m$ including the starting q_s and halting q_h states and acts as the coordinator for the Turing machines operations. The Tape is an infinite one dimensional object consisting of tape squares. Each square contains a symbol drawn from some alphabet (e.g. a bit). The read-write tape head points the position on the tape which is currently readable or writeable. In other words it is the current square being read in the tape. Finally the program is a finite ordered list of program lines of the form $\langle q, x, q', x', s \rangle$. One looks for $\langle q, x, \cdot, \cdot, \cdot \rangle$ such that q is the current state and x is the symbol being read on the tape. If this exists then the program line is executed and the internal state is changed to q and the symbol on the the tape is overwritten by x' . One moves the tape head either left or right or stays in place depending on the value s . If such $\langle q, x, \cdot, \cdot, \cdot \rangle$ does not exist then the internal state is changed to the halting state q_h and the machine halts the operation.

A natural question one can ask is what can we do on Turing machines? This was conclusively answered in the Church-Turing these where in a very broad sense they showed that the class of computable functions that can be made by a Turing machine is equal to the class of functions that can be computed via an algorithm. There are many variants of this statement depending on the specific problem that is being solved but they all have this sort of flavour.

This Turing machine model is a little too idealised and theoretical. Modern computers are finite in size, whereas Turing machines may have unbounded memory. An alternative model for computation that better captures this is the circuit model. This model for computation is equivalent to the Turing machine model in terms of computational power. In a circuit one has wires and gates. These take the role of carrying information around and performing simple computational tasks (respectively). For example, the NOT gate on 1 wire takes a bit and simply flips the bit.



Also, given the equivalence to Turing machines, any circuit within this model comes from an algorithmic description of the procedure.

Complexity Classes

An important question when it comes to problems to be solved in either of these models is computational complexity. Concretely this is the study of time and space resources to solve computational problems on a Turing machine or through a circuit. One initial challenge in capturing this complexity is that different models may require different resources to solve the same problem. For instance a multi-tape Turing machine can solve a problem much quicker than a single-tape machine. The common fix to this is to fix the number of bits, n , as input to the problem and measure resources as bit-operations. An algorithm is said to run in polynomial time if it runs in $O(n^k)$ for some fixed constant k .

As a result there are a broad range of complexity classes which classifies a range of algorithms depending on their computational complexity. The two most well-known complexity classes are P which is the set of problems, with n -bits as input, that can be solved on a Turing machine in polynomial time; and NP which is the set of problems for which you have a candidate solution and you want to check whether that solution is valid in polynomial time. For example, factoring $\in$ NP but it is not known whether factoring $\in$ P. This is encompassed in the well-known P vs NP conjecture. There are many other complexity classes: PSPACE, NP-complete, NP-hard, BPP and more.

Introduction to Quantum

1. From Classical Bits to Qubits

Classical computation is based on the notion of a *bit*, which represents information using one of two possible states:

$$0 \quad \text{or} \quad 1.$$

Mathematically, a classical bit can be viewed as an element of the discrete set $\{0, 1\}$. A classical register containing n bits can therefore store one element from a space of size

$$|\{0, 1\}^n| = 2^n.$$

A quantum computer replaces the classical bit with a *qubit*. Unlike a classical bit, whose state is restricted to two possibilities, a qubit is a vector in a two-dimensional complex Hilbert space:

$$\mathcal{H} = \mathbb{C}^2.$$

The computational basis of this space is given by the orthonormal vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The inner product on this space is defined by

$$\langle\phi|\psi\rangle = \langle\phi||\psi\rangle,$$

which determines the geometry of quantum states. The computational basis satisfies

$$\langle 0|0\rangle = \langle 1|1\rangle = 1, \quad \langle 0|1\rangle = \langle 1|0\rangle = 0.$$

A general pure quantum state is a normalized vector in this Hilbert space:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where $\alpha, \beta \in \mathbb{C}$. Normalization requires

$$\langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1.$$

The complex numbers α and β are called *probability amplitudes*. Their magnitudes determine measurement probabilities, while their relative phases influence interference effects.

A useful parametrization of a single qubit is

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle,$$

where $0 \leq \theta \leq \pi$ and $0 \leq \varphi < 2\pi$. This representation corresponds geometrically to a point on the Bloch sphere.

A physical qubit can be implemented using different quantum systems. A particularly intuitive physical realization of a qubit is provided by the polarization of a single photon. Light is an electromagnetic wave whose electric field oscillates in a direction perpendicular to its propagation direction. When the possible polarization directions are restricted to two orthogonal modes, the photon behaves as a two-level quantum system.

For example, consider horizontal and vertical polarization states. These two states form an orthonormal basis for the polarization Hilbert space:

$$|0\rangle = |H\rangle, \quad |1\rangle = |V\rangle.$$

Here, $|H\rangle$ represents a photon whose electric field oscillates horizontally, while $|V\rangle$ represents a photon whose electric field oscillates vertically. Mathematically, these states behave exactly like the computational basis vectors:

$$|H\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |V\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The important quantum feature is that a photon is not restricted to being only horizontally or vertically polarized. It can exist in a coherent superposition of both polarization modes:

$$|\psi\rangle = \alpha|H\rangle + \beta|V\rangle,$$

where

$$|\alpha|^2 + |\beta|^2 = 1.$$

For instance, diagonal polarization corresponds to the state

$$|+\rangle = \frac{|H\rangle + |V\rangle}{\sqrt{2}}.$$

This state does not mean that the photon is half horizontally polarized and half vertically polarized in a classical sense. Rather, the photon exists in a quantum superposition of the two possible polarization modes.

Therefore, the polarization of a single photon provides a direct physical model of a qubit: the two polarization directions define the basis states, while superpositions correspond to arbitrary points on the quantum state space.

2. Measurement of a Qubit

Quantum states evolve according to linear transformations, but measurement produces classical information. The mathematical description of measurement is given by a set of operators satisfying the completeness condition.

For a measurement in the computational basis, the projectors are

$$P_0 = |0\rangle\langle 0|, \quad P_1 = |1\rangle\langle 1|.$$

The probability of obtaining outcome i is given by the Born rule:

$$P(i) = \langle \psi | P_i | \psi \rangle.$$

For

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

we obtain

$$P(0) = |\alpha|^2, \quad P(1) = |\beta|^2.$$

After measurement, the state collapses to the corresponding basis state:

$$|\psi\rangle \rightarrow |0\rangle \quad \text{or} \quad |1\rangle.$$

For the state

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}},$$

both measurement outcomes occur with equal probability:

$$P(0) = P(1) = \frac{1}{2}.$$

Coming back to the photon realization of a qubit, when measured in the horizontal-vertical basis, the photon is detected as either horizontal or vertical polarization with probabilities

$$P(H) = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2}, \quad P(V) = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2}.$$

The relative phase between polarization components is also physically meaningful. For example,

$$|-\rangle = \frac{|H\rangle - |V\rangle}{\sqrt{2}}$$

has the same measurement probabilities as $|+\rangle$ in the horizontal-vertical basis, but differs in phase. This difference becomes observable when the photon is measured in another basis, illustrating the role of quantum interference.

3. Tensor Products and Multiple Qubits

The state space of multiple quantum systems is constructed using the tensor product. For two qubits,

$$\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B.$$

Since

$$\dim(\mathbb{C}^2) = 2,$$

the dimension of the two-qubit space is

$$\dim(\mathcal{H}_{AB}) = 2^2 = 4.$$

The computational basis becomes

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle.$$

A general two-qubit state is therefore

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle,$$

with

$$\sum_{x \in \{0,1\}^2} |\alpha_x|^2 = 1.$$

For n qubits, the state space is

$$\mathcal{H}_n = (\mathbb{C}^2)^{\otimes n},$$

whose dimension grows exponentially:

$$\dim(\mathcal{H}_n) = 2^n.$$

Therefore, a general quantum state requires 2^n complex amplitudes:

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle.$$

The exponential dimension of the Hilbert space does not mean that a quantum computer directly stores 2^n classical values. Instead, quantum algorithms exploit linear transformations, interference, and entanglement to manipulate the amplitudes in a way that produces computational advantages.

Quantum Computing

We translate this quantum knowledge back to the computing setting. One can define quantum Turing machines as an analogue of the classical Turing machine mentioned at the start (see [Benioff80] for more details on this). More commonly, one uses the quantum circuit model for computation. We will see more of this model in Week 3 but many of the same properties that the classical circuit model has is translated to the quantum analogue.

As a result, we can reason about quantum algorithms as a quantum circuit or a program on a quantum Turing machine. So we can have the same computational complexity discussion as we had before. I.e. measure the amount of resources needed to solve computational problems on these quantum computers given a predescribed amount of input.

There are several complexity classes in this remit, but the most well known is BQP which consists of problems that can be solved in quantum polynomial time on a circuit/Turing machine, where bounded probability error is allowed. One can simulate an classical algorithm as a quantum algorithm, so one clearly has $P \subseteq BQP$ and BQP is the analogy of P in the quantum setting (more precisely BPP is the true analogy here).

Open question: Where does BQP fit within P vs NP? In particular it is unclear where BQP fits within NP where a weird overlap predicted.

Week 2

Quantum Information

Talk by .

Quantum Circuits

Talk by .

Week 4

Quantum Key Distribution

Talk by .

Quantum Security

Talk by .

Week 6

Deploying QKD

Invited talk by .

Classical Random Oracle Model

Talk by .

Quantum Random Oracle Model

Talk by .

QROM Lemmas

Talk by .

QROM Security Proof

Talk by .

Bibliography:

- [1] A. Ambainis, M. Hamburg, and D. Unruh. Quantum security proofs using semi-classical oracles. *Cryptology ePrint Archive*, 2018/904. <https://eprint.iacr.org/2018/904>.
- [2] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. 1984; reprinted in *Theoretical Computer Science* 560, pp. 7–11, 2014. <https://arxiv.org/abs/2003.06557>.
- [3] S. Bruckner, S. Ramacher, and C. Striecks. Muckle+: End-to-End Hybrid Authenticated Key Exchanges. *PQCrypto*, 2023. <https://eprint.iacr.org/2023/653>.
- [4] J. Don, S. Fehr, C. Majenz, and C. Schaffner. Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model. 2019. <https://arxiv.org/abs/1902.07556>.
- [5] B. Dowling, T. B. Hansen, and K. G. Paterson. Many a Mickle Makes a Muckle: A Framework for Provably Quantum-Secure Hybrid Key Exchange. *PQCrypto*, 2020. <https://eprint.iacr.org/2020/099>.
- [6] J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. See the random-oracle model and proof techniques.
- [7] D. Leung. *Introduction to Quantum Information Processing*, University of Waterloo course materials, Winter 2025. <https://www.math.uwaterloo.ca/~wcleung/intro-qinfo-w25.html>.
- [8] H.-K. Lo, M. Curty, and K. Tamaki. Secure Quantum Key Distribution. 2015. <https://arxiv.org/abs/1505.05303>.
- [9] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
- [10] M. Tomamichel and A. Leverrier. A largely self-contained and complete security proof for quantum key distribution. 2015. <https://arxiv.org/abs/1506.08458>.
- [11] V. Zapatero, Á. Navarrete, and M. Curty. Implementation security in quantum key distribution. 2023. <https://arxiv.org/abs/2310.20377>.
- [12] M. Zhandry. How to Construct Quantum Random Functions. *FOCS*, 2012. <https://eprint.iacr.org/2012/182>.
- [13] M. Zhandry. How to Record Quantum Queries, and Applications to Quantum Indifferentiability. 2018. <https://eprint.iacr.org/2018/276>.