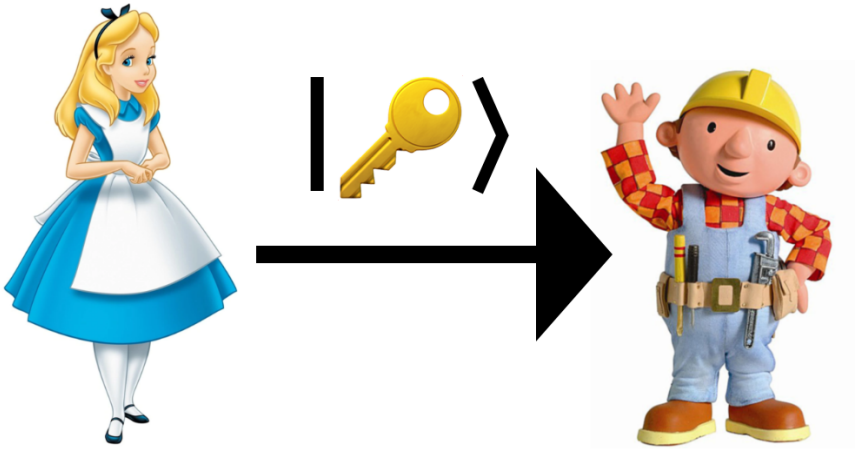
Comparison of Post-Quantum Cryptography and Quantum Key Distribution

Emiel Wiedijk

24 June 2022

Bachelor thesis Mathematics and Computer Science

Supervisor: Prof. Dr. Christian Schaffner, Dr. Jeroen Zuiddam



Informatics Institute &
Korteweg-de Vries Institute for Mathematics
Faculty of Sciences
University of Amsterdam

Abstract

The security of the current generation of cryptographic algorithms, called computational security, is based on hardness assumptions: the assumption that it is hard to solve a certain type of problem. Unfortunately, the current hardness assumptions will be broken by quantum computers, making the current algorithms insecure. Therefore, new cryptographic algorithms need to be developed that are secure even against quantum computers.

There are two approaches for these new algorithms. The first approach is post-quantum cryptography (PQC): a set of algorithms that switch to a new hardness assumption that holds against quantum computers. The second approach is quantum key distribution, an algorithm based on quantum communication that provides information-theoretical security. Information-theoretical security does not rely on any hardness assumptions, and instead provides unconditional security.

In this thesis we give an introduction to post-quantum cryptography and quantum key distribution. Classical algorithms that provide information-theoretical security are limited by the pre-shared mutual information of the honest parties. Quantum key distribution (QKD) can be used to simulate unlimited correlated pre-shared information. Thereby QKD can provide more information-theoretical security than classical algorithms.

Title: Comparison of Post-Quantum Cryptography and Quantum Key Distribution

Author: Emiel Wiedijk, emiel.wiedijk@student.uva.nl, 12699373
Supervisors: Prof. Dr. Christian Schaffner, Dr. Jeroen Zuiddam

End date: 24 June 2022

Informatics Institute &
University of Amsterdam
Science Park 904, 1098 XH Amsterdam
`http://www.ivi.uva.nl`

Korteweg-de Vries Institute for Mathematics
University of Amsterdam
Science Park 904, 1098 XH Amsterdam
`http://www.kdvi.uva.nl`

Contents

1 Introduction

Cryptography is a ubiquitous part of everyone's life. For example, cryptographic protocols protect the content of our messages in WhatsApp or Signal from eavesdroppers. It is also essential for our payment infrastructure as it allows us to securely communicate with our banks online. Cryptography is even used offline for payments by card to authorize payments on a bank account, while making it nearly impossible to copy the card.

The two main goals of cryptography are providing secrecy, and authenticity. Secrecy guarantees that two parties can communicate while the content of communication remains unknown for any third party eavesdropper. The sender achieves secrecy by *encrypting* a message. On the other hand authenticity guarantees that a message was sent by a specific party, and was not altered in any way. To achieve authenticity, the sender adds a *tag* or *signature* to the message, that can only be generated by a specific party.

A central part of cryptography is the notion of a key. A key is a (partially) secret bit string that gives an advantage to the legitimate party. For example, only the party that has the correct key can decrypt the message. Or only with the correct key is it possible to authorize a payment on a bank account.

There are two main types of cryptography: symmetric cryptography and asymmetric cryptography. In symmetric cryptography, both parties use the same key. For example, the same key is used to encrypt and decrypt a message. In asymmetric cryptography, different parties use different keys: the key for encryption can be different from the key for decryption. The advantage of using different keys is that the key used for encryption can be made public, while decryption requires a secret key, which remains secret.

Often asymmetric cryptography is used indirectly. Instead of encrypting a message directly, an asymmetric cryptographic protocol can be used to securely exchange a common key, that is unknown to any eavesdropper. This key can be used in a symmetric encryption algorithm.

In this thesis we will focus on the problem of key exchange, a form of asymmetric cryptography. In key exchange two honest parties, commonly called Alice and Bob, try to establish a common key k. To establish this k Alice and Bob exchange messages on a public channel that can be eavesdropped upon by a third party, commonly called Eve.

The cryptographic protocols in use today all have one caveat: they can be broken given enough time and resources. Current cryptography relies on computational security: on average, it takes an unreasonable amount of time to break the protocol. For encryption, computational security means that, while decrypting a message with access to a key is

easy, decrypting without access to the key is hard.

Computational security is based on hardness assumptions: some problem is assumed to be hard. Then it is proven that a cryptographic algorithm is at least as hard to break as the hardness assumption. A common hardness assumption is that factoring of (particular types of) large integers is hard.

This computational security works as long as the hardness assumptions hold. Unfortunately, factoring of integers may not be a hard problem anymore. Shor has developed an algorithm to factor integers in polynomial-time on a quantum computer [21]. Shor's algorithm also breaks the discrete logarithm problem, one of the other hardness assumptions that the current generation of asymmetric cryptographic algorithms use. On the other hand, symmetric cryptographic algorithms remain largely secure against quantum computers [11].

As of the time of writing, no one has yet built a quantum computer large enough to break cryptographic algorithms. A quantum computer that would break the current hardness assumptions and leave most of the current cryptography algorithms vulnerable, may be built already in the next decades.

To counter the threat of these quantum algorithms we need a new generation of cryptographical algorithms. There are two approaches: the first approach is *post-quantum cryptography* (PQC), a set of cryptography algorithms designed to be resistant against attackers with access to a quantum computer. Post-quantum cryptography algorithms are classical algorithms. Similar to the current algorithms, they provide computational security but based on a different hardness assumption.

The second approach to counter the quantum computer is *quantum key distribution* (QKD). Unlike post-quantum cryptography, quantum key distribution is based on quantum communication. Unlike post-quantum cryptography, quantum key distribution does not rely on any computational assumptions, but instead claims to provide information-theoretical security. QKD is provably secure without computational assumptions against any attacker. The key is nearly indistinguishable from a perfect uniformly random key, even with unlimited time and resources.

In this thesis we will compare post-quantum cryptography with quantum key distribution, with a focus on the security guarantees they provide. We will also compare the limits of information-theoretical security between the classical and quantum protocols. We will start describing the security guarantees of post-quantum cryptography, with a focus on the new hardness assumptions. In the third chapter we will investigate the limit of information-theoretical security for classical protocols. As a prerequisite for explaining quantum key distribution, we will describe how quantum systems work in the fourth chapter. Finally, in the fifth chapter, we describe the workings and security guarantees of quantum key distribution. In general the contribution of this thesis is to provide an introduction into the limits of information-theoretical security, and the additional benefit of quantum protocols. It has been written for people with a basic understanding of cryptography, but no previous knowledge of quantum information.

2 Post-Quantum Cryptography: security guarantees

Post-quantum cryptography is a set of cryptography schemes. Their goal is to achieve a type of security that is "secure" against attackers that have access to a quantum computer. The definition of "secure" depends on the precise goal of the scheme. These schemes are only allowed to use classical/non-quantum keys and communication. The word "quantum" in their name solely refers to the quantum capabilities of an attacker trying to break the scheme. More uncommon, but more descriptive names for post-quantum cryptography are quantum-resistant cryptography and quantum-safe cryptography. We contrast post-quantum cryptography with pre-quantum cryptography, the existing public-key infrastructure, designed without the currently known quantum attacks in mind.

Post-quantum cryptography usually refers to asymmetrical cryptography schemes. As pre-quantum symmetrical encryption schemes with an increased key size are considered secure against a quantum attacker [11], they do not require new algorithms.

There are three main types of asymmetrical cryptography schemes:

1. Public key encryption: the encryption of arbitrary messages.
2. Key encapsulation mechanism: facilitating the exchange of a key.
3. Signature schemes: allowing communicating parties to verify that messages were sent by a certain party, and that the messages arrived unaltered.

We use the definition of a public-key encryption scheme from Katz and Lindell.

Definition 2.1. A public-key encryption schemes is a triple of functions $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Let $n \geq 1$ a security parameter, m the plaintext. Let $s_g, s_e \in \{0,1\}^n$ be seeds for key generation, uniformly sampled on every function call. The functions Gen and Enc behave as

$$\begin{aligned}(sk, pk) &= \mathsf{Gen}(1^n; s_g)\\ c &= \mathsf{Enc}_{pk}(m; s_e)\\ m' &= \mathsf{Dec}_{sk}(c).\end{aligned}$$

The decryption function Dec_{sk} is the inverse of Enc_{pk}, but may fail with negligible probability. Formally, for S_g and S_e uniformly distributed over $\{0,1\}^n$. Let

$$(SK, PK) = \mathsf{Gen}(1^n; S_g)$$

be the distributions of the matching key pairs. Then *correctness* guarantees that with randomness over (SK, PK) and S_e that

$$\Pr[m \neq \mathsf{Dec}_{SK}(\mathsf{Enc}_{PK}(m; S_e))] \leq \mathsf{negl}(n). \tag{2.1}$$

Public-key encryption schemes have many advantages. For example, they can be easily implemented since they are more similar to the pre-quantum schemes. However, a public-key encryption scheme can never accomplish information-theoretical security, also known as unconditional security, since it is almost always possible to recover a plaintext message from the cipher text and the public key.

Theorem 2.2. *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *a public-key encryption scheme. An adversary with unbounded computational power, who has access to the public key* pk *and ciphertext* c *can recover the original message* m *up to negligible probability.*

Proof. Let m be a plaintext message, let c be the corresponding ciphertext, (sk, pk) the original key pair, n the security parameter and m a plaintext message.

The adversary samples $(pk_{s_g}, sk_{s_g}) = \mathsf{Gen}(s_g)$ and seeds s_e for all $s_g, s_e \in \{0,1\}^n$. The adversary picks a key pair (sk_{s_g}, pk_{s_g}) such that

$$pk_{s_g} = pk.$$

Since the public keys are the same there exists a seed s_e such that

$$c = \mathsf{Enc}_{pk_{s_g}}(m, s_e).$$

The attacker recovers the message by calculating

$$m' = \mathsf{Dec}_{sk_{s_g}}(c).$$

By correctness (Equation 2.1), we have

$$\begin{aligned}\Pr[m = m'] &= \Pr\Big[m = \mathsf{Dec}_{sk_{s_g}}(c)\Big] = \Pr\Big[m = \mathsf{Dec}_{sk_{s_g}}(\mathsf{Enc}_{pk_{s_g}}(m, s_e))\Big] \\ &\geq 1 - \mathsf{negl}(n).\end{aligned}$$

We conclude up that to negligible probability, it is possible to recover the plaintext m. □

Because information-theoretical security is impossible, post-quantum cryptography schemes rely on computational security instead. Computational security relies on hardness assumptions. Some problem P is assumed to be hard to solve for a polynomial time attacker, the hardness assumption. The security of a cryptography system C relies on that an attacker who can break a cryptography scheme C in polynomial time can also solve P in polynomial time, which is assumed to be hard.

Unfortunately, these hardness assumptions do not necessarily hold. Examples of common pre-quantum public key schemes are RSA and Diffie-Hellman. The security of RSA relies on the hardness of factoring certain large integers. Diffie-Hellman relies on the hardness of the discrete-logarithm problem. Both these problems can be solved efficiently by a quantum computer using Shor's algorithm [21]. As the hardness assumption fails to hold, RSA and Diffie-Hellman turn out to be insecure against attackers with access to quantum computers.

As factoring and the discrete-logarithm problem are no longer hard for computers, post-quantum cryptography algorithms switch to different computational hardness assumptions based on other problems. These problems are thought to have no efficient algorithms to solve them, even on quantum computers. The model of computational security remains unchanged: the security model only protects against polynomial-time attackers, for whom the hardness assumption holds.

There are several problems that can be used as hardness assumption in post-quantum cryptography. The NIST currently holds a competition for the best post-quantum cryptography systems. At the time of writing (June 2022), the competition is in its 3rd round [15]. Most submissions use one of the following hardness assumptions:

1. Lattice-based problems
2. Code problems

In the next sections we discuss the hardness assumptions for both of these problems.

2.1 Lattice-based cryptography

Lattices are algebraic structure. In lattices certain hard problems can be defined that can be used in cryptography.

Definition 2.3. [16] We call any subset $L \subseteq \mathbb{R}^n$ a *lattice* when

$$L = \left\{ \sum_{i=1}^{n} k_i x_i \mid k_1, \ldots, k_n \in \mathbb{Z} \right\},$$

for some linearly independent $\{x_1, \ldots, x_k\}$. The set $\{x_1, \ldots, x_k\} \subseteq \mathbb{R}^n$ is called the *basis* of the lattice.

There are several computational hardness assumptions in lattices. The most foundational one is the Shortest-Independent-Vector Problem, or SIVP for short.

Definition 2.4. [16] For a lattice L let λ_1 be the shortest non-zero vector in a lattice,

$$\lambda_1(L) = \min_{\substack{\ell \in L \\ \ell \neq 0}} \|\ell\|.$$

Let λ_d be the shortest possible maximum length of a set of d linearly independent vectors

$$\lambda_d(L) = \min_{\substack{\ell_1, \ldots, \ell_n \in L \\ \ell_1 \neq 0, \ldots, \ell_d \neq 0 \\ \text{linearly independent}}} \max(\|\ell_1\|, \ldots, \|\ell_d\|)$$

Then the problem is to find some vector that finds a set of vectors that

Definition 2.5. [16] The SIVP problem is, given an n-dimensional lattice L, to find a basis $\ell_1, \ldots, \ell_n$ such that

$$\max(\|\ell_1\|, \ldots, \|\ell_d\|) \leq \gamma(n) \cdot \lambda_n(L)$$

for some approximation factor $\gamma(n)$.

The computational hardness of SIVP depends on the hardness factor $\gamma(n)$. Assuming that RP $\neq$ NP[1], for small $\gamma(n)$ it is impossible to solve SIVP in polynomial time, but for large $\gamma(n)$ polynomial-time algorithms are known, specifically

1. For $\gamma(n) = c$ for some fixed constant $c \geq 1$ SIVP is NP-hard. [3]
2. For $\gamma(n) = 2^{\log(n)^{1-\epsilon}}$ for some fixed constant $\epsilon > 0$ SIVP is NP-hard.

[1]The assumption RP $\neq$ NP is the probabilistic version of P $\neq$ NP where a negligible error probability is allowed.

3. For $\gamma(n) = \frac{n}{\sqrt{\log(n)}}$ SIVP is probably not NP-hard, but still no known polynomial-time algorithms exist.
4. For large $\gamma(n)$ more efficient algorithms are known. For example, the LLL-algorithm [8] (named after its inventors Lenstra, Lenstra and Lovasz) solves SIVP with $\gamma(n) = 2^{n-1}$ in polynomial-time. Schnorr has developed an algorithm with a tighter, but still exponential bound. [18].

Typically, SIVP is not used directly in cryptography systems. Instead, in a 2005 paper by Regev [16], he introduced the Learning With Errors (LWE) problem.

Definition 2.6. Let $p \geq 1$ be prime and let χ be a probability distribution on $\mathbb{Z}_p^n$ the inputs for the problem. Then let $s \in \mathbb{Z}_p^n$ be a uniformly random vector. Define the distribution $A_{s,\chi}$ that provides samples

$$(a, \langle a|s\rangle + e)$$

with $a \in \mathbb{Z}_p^n$ uniformly distributed, and $e \sim \chi$. The problem of $\mathrm{LWE}_{p,\chi}$ is to find s given the samples.

As LWE is a relatively new problem (2005), we would like to have some assurance of its hardness. Regev showed that if it is possible to solve LWE with negligible failure rate with a polynomial number of samples, it is possible to solve SIVP efficiently with an approximation factor of $\gamma(n) = \frac{n}{\alpha}$ for an arbitrary fixed $\alpha \in (0, 1)$.

There are other lattice-problems that can be reduced to SIVP with a certain approximation factor that can be used in cryptography. A slight variation of LWE is Ring-LWE [9]. In Ring-LWE the vectors in $\mathbb{Z}_p^n$ are replaced by elements of $\mathbb{Z}_p[x]/(x^n + 1)$ to increase efficiency. Another variation is Learning With Rounding (LWR), where the random error is replaced by rounding to avoid the required (pseudo-)randomness [1].

2.2 Code-based cryptography

Error-correcting codes can be used to add redundancy to data by encoding that data into *code words*. Even if a code word gets corrupted, it is possible to detect and correct the corruption, as long as the number of errors remains limited. Some operations on error-correcting codes are hard, which can be used as hardness assumption in cryptography. The usual metric for the number of errors is the Hamming distance.

Definition 2.7. Let V be an n-dimensional vector space. For $x, y \in V$ the Hamming distance is defined as

$$d(x, y) = \sum_{i=1}^{n} \delta_{x_i y_i}.$$

Here δ is the *Kronecker delta*

$$\delta_{ij} := \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

The code word can then be used to reconstruct the original data. We restrict ourselves to linear error-correcting code, where linear combinations of code words are also a code word. Thus, an error-correcting code generates a k-dimensional subspace of code words. The subspace has the property that distinct code words differ in at least t indices.

Definition 2.8. Let p be the size of a finite field $\mathbb{F}_p$. An $[n, k, t]$ *error-correcting code* is a right-invertible matrix $G \in (\mathbb{F}_p)^{k \times n}$ with the property that

$$\min_{\substack{x,y \in (\mathbb{F}_p)^k \\ x \neq y}} d(xG, yG) \geq t$$

The rows of G form a basis for a k-dimensional subspace $C \subseteq (\mathbb{F}_p)^n$.

An error-correcting code G can encode a message $m \in (\mathbb{F}_p)^k$ into a code word $c \in (\mathbb{F}_p)^n$ by calculating

$$c = mG.$$

Some errors may be introduced to c, for example due to noise during the transmission. So instead of c the code word is received as c_{err}. We can correct for these errors by finding the vector $c_{\text{cor}} \in (\mathbb{F}_p)^n$ that minimizes

$$d(c_{\text{err}}, c_{\text{cor}}).$$

Finding this c_{cor} is called solving or decoding the linear code. As long as $d(c, c_{\text{err}})$ is low, we have that $c = c_{\text{cor}}$.

Solving a general linear code G is an NP-complete problem [2]. As $P = NP$ is still an open question, no polynomial time algorithms for solving a general linear code are

known. It is of course possible to brute-force by trying all p^k code words in C, and finding the closest one, but this brute-forcing is computationally expensive.

However, there are specific classes of linear codes for which efficient error-correcting algorithms are known, for example the Goppa linear codes. These Goppa codes can be decoded significantly more efficiently in only $\mathcal{O}(n \log(n)^2)$ using Patterson's algorithm [17].

For computational security we want decryption with the (private) key to be easy, while decryption without the (private) key should be hard. In 1978 Robert McEliece introduced a cryptography system based on error-correcting codes as part of his work for NASA [12]. The McEliece system uses the fact that while solving a general linear code is hard, a Goppa code can be solved in polynomial time. McEliece starts with a randomly generated binary Goppa linear code. The generator matrix of this Goppa code is scrambled to form a more or less random general linear code. The Goppa code and the way it is scrambled form the private key, while the resulting scrambled generator matrix forms the public key.

To encrypt a message it is possible to encode a message using the scrambled generator matrix and add some errors to it. With access to the private key, it is possible to undo the scrambling and decode the Goppa code using Patterson's algorithm, which is efficient. Without the private key it becomes necessary to either solve the arbitrary linear code, which is NP-hard or undo some unknown scrambling. Both of these options are expected to be hard.

Remark 2.9. The assumption $P \neq NP$ is a necessary but not sufficient condition for the security of McEliece. The scrambled Goppa codes might form another category of "easy" linear codes, or it might be possible to undo the scrambling.

3 Classical Information Theoretic Security

There are cryptographic protocols that provide a form of security, called *information-theoretical* security. This security is provided even against unbounded attackers. As the attackers may be unbounded, it does not rely on any computational assumption.

For key-exchange protocols, this means that the generated key should be "secure". To quantify this security, we need some information-theoretical measure of key strength. From the perspective of an adversary, we model a key as a random variable. We will introduce two concepts in this chapter: entropy and the statistical distance. For classical key-exchange protocols we will find limits on the entropy of generated keys.

3.1 Entropy

When we design a key-exchange protocol, we want the adversary to have as much uncertainty of the key as possible. One possible measure of uncertainty is the *Shannon entropy*. Entropy was introduced by the American mathematician Claude Shannon in a seminal paper in 1948 [19]. With his paper, he was one of the founding fathers of the field of information theory. Another, equivalent interpretation of the entropy is how much information an observer gets when they observe the outcome of the random variable. We can use the entropy as measure of a strength of a generated key.

Let X be a discrete random variable, with $P_X = (p_1, p_2, \ldots, p_n)$ its discrete probability distribution, where the outcome i has probability p_i. Shannon showed that following three properties of a function $H(P_X)$, uniquely define the entropy.

Theorem 3.1. *[19] Let $H : \mathbb{R}^n \mapsto \mathbb{R}$ satisfy following properties*

1. *$H(P_X)$ is continuous.*
2. *The function $H(\frac{1}{n}, \ldots, \frac{1}{n})$ is monotonically increasing in n.*
3. *If one outcome i of the random variable $X = (p_1, \ldots, p_n)$ is replaced by the outcome of $Y = (q_1, \ldots, q_n)$ in X' then*

$$H(P_{X'}) = H(P_X) + p_i H(P_Y).$$

Then $H(p_1, p_2, \ldots, p_n) = -\sum_{i=1}^{n} p_i \log_2(p_i)$ up to a constant factor.

Remark 3.2. The base of the log in the formula for the entropy is a constant factor, which defines the unit of the entropy. When $\log_2$ is used the unit is called *bits*.

There are several possible log bases, which all behave the same up to a constant factor. Unless defined otherwise, we pick base 2 for our log function, which gives as unit bits. A random variable which has n bits entropy, has as much uncertainty as n independent, perfect coin tosses.

We use the shorthand $H(X) = H(P_X)$, even though the entropy is a function of the distribution, not the outcome of a random variable.

Example 3.3. *Let P_X be the uniform distribution over bit strings of length n. As every single bit string has probability 2^{-n}, the entropy is equal to*

$$\begin{aligned} H(X) &= -\sum_{i=1}^{2^n} 2^{-n} \log(2^{-n}) = -2^n \cdot 2^{-n} \log(2^{-n}) \\ &= n. \end{aligned}$$

Random variables X and Y with probability distributions P_X and P_Y can be combined, creating the joint distribution P_{XY}. The entropy of the joint distribution is defined just like the univariate case.

Definition 3.4. [5, p. 15] The definition of the entropy of a joint distribution P_{XY} with outcome $X = i$ and $Y = j$ with probability $p_{i,j}$ is

$$H(X, Y) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p_{i,j} \log(p_{i,j}).$$

Based on the joint distribution we can more precisely describe the interaction between X and Y

Definition 3.5. [5, p. 16] The *conditional entropy* is defined as

$$H(X \mid Y) = H(X, Y) - H(Y).$$

The conditional entropy $H(X \mid Y)$ can be interpreted as a measure of uncertainty on the outcome of X when the outcome of Y is known. Namely, if $H(X \mid Y)$ is small compared to $H(X)$, then knowing the outcome of Y lowers the uncertainty about the value of X. On the other hand, if $H(X \mid Y)$ is still large, then knowing the value of Y does not predict much about X.

The conditional entropies $H(X \mid Y)$ and $H(Y \mid X)$ give information about the overlapping information between X and Y. However, the size of the overlapping information is dependent on $H(X)$ and $H(Y)$. Furthermore, the conditional entropy is asymmetrical. As a measure for the information that is common between X and Y we define the mutual information

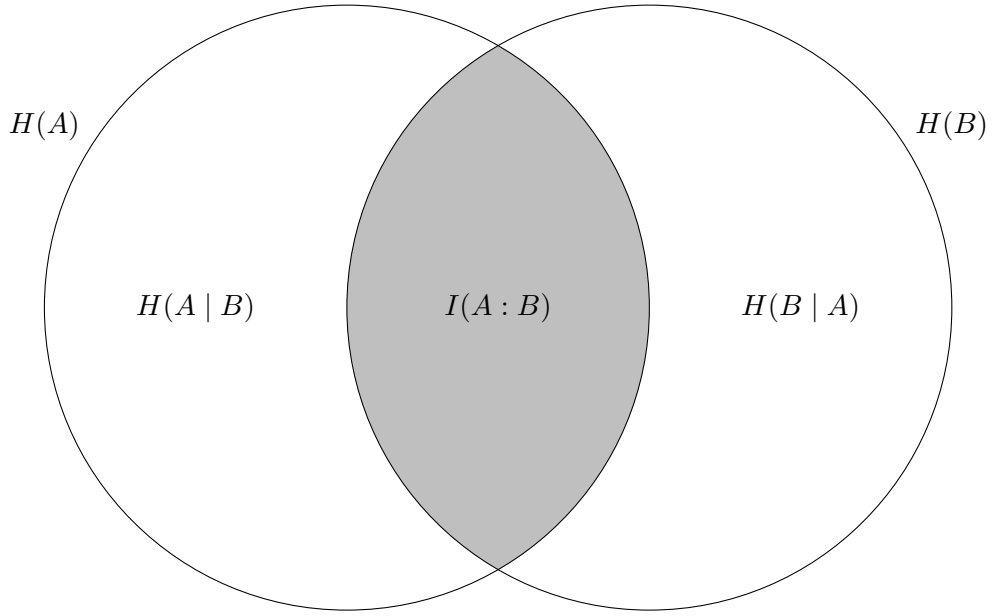


Figure 3.1: Information diagram of two variables [24]

Definition 3.6. [5, p. 19] For random variables X and Y, *the mutual information* between X and Y is defined as

$$I(X : Y) = H(X) + H(Y) - H(X, Y).$$

From these definitions, we can see that the entropy behaves similar to the classical set operations where

1. $H(X, Y) \approx X \cup Y$
2. $I(X : Y) \approx X \cap Y$
3. $H(X \mid Y) \approx X \setminus Y$.

Therefore, these definitions can be summarized in the following *information diagram* as seen in Figure 3.1

Lemma 3.7. *[5] For random variables X and Y the following properties hold:*

1. $H(X, Y) \leq H(X) + H(Y)$ *with equality if only if X and Y are independent.*
2. $H(X \mid Y) \geq 0$*, with equality if only if X is a function of Y.*
3. $I(X : Y) \geq 0$*, with equality if and if only if X and Y are independent.*

We can extend these definitions to interactions with an arbitrary number of variables. For the (conditional) entropy this extension is relatively straightforward.

Definition 3.8. [5] Let $(X_1, \ldots, X_n)$ be a tuple of n discrete random variables. The entropy is defined as

$$H(X_1, \ldots, X_n) = -\sum_{\substack{i_1 \in \mathcal{X}_\infty \\ i_2 \in \mathcal{X}_\in \\ \cdots \\ i_n \in \mathcal{X}_\setminus}} p_{i_1,i_2,\ldots,i_n} \log_2(p_{i_1,i_2,\ldots,i_n})$$

Definition 3.9. [5] The conditional entropy is defined as

$$H(X_1, \ldots, X_n \mid Y_1, \ldots, Y_m) = H(X_1, \ldots, X_n, Y_1, \ldots, Y_m) - H(Y_1, \ldots, Y_m)$$

We can also extend the mutual information for an arbitrary number of variables, called information interaction.

Definition 3.10. Let $X_1, \ldots, X_n$ be random variables. Define the conditional *information interaction* as defined as

$$I(X_1 : \ldots : X_{n-1} \mid X_n) = \sum_{x_n} \Pr(X_n = X_n) I(X_1 : \ldots : X_{n-1} \mid X_n = x_n)$$

with the information interaction defined as

$$I(X_1 : \ldots : X_n) = I(X_1 : \ldots : X_{n-1} \mid X_n) - I(X_1 : \ldots : X_{n-1}).$$

Therefore, we can define entropy diagrams with an arbitrary number of random variables. [13] [24] We give the entropy diagram for 3 random variables here.

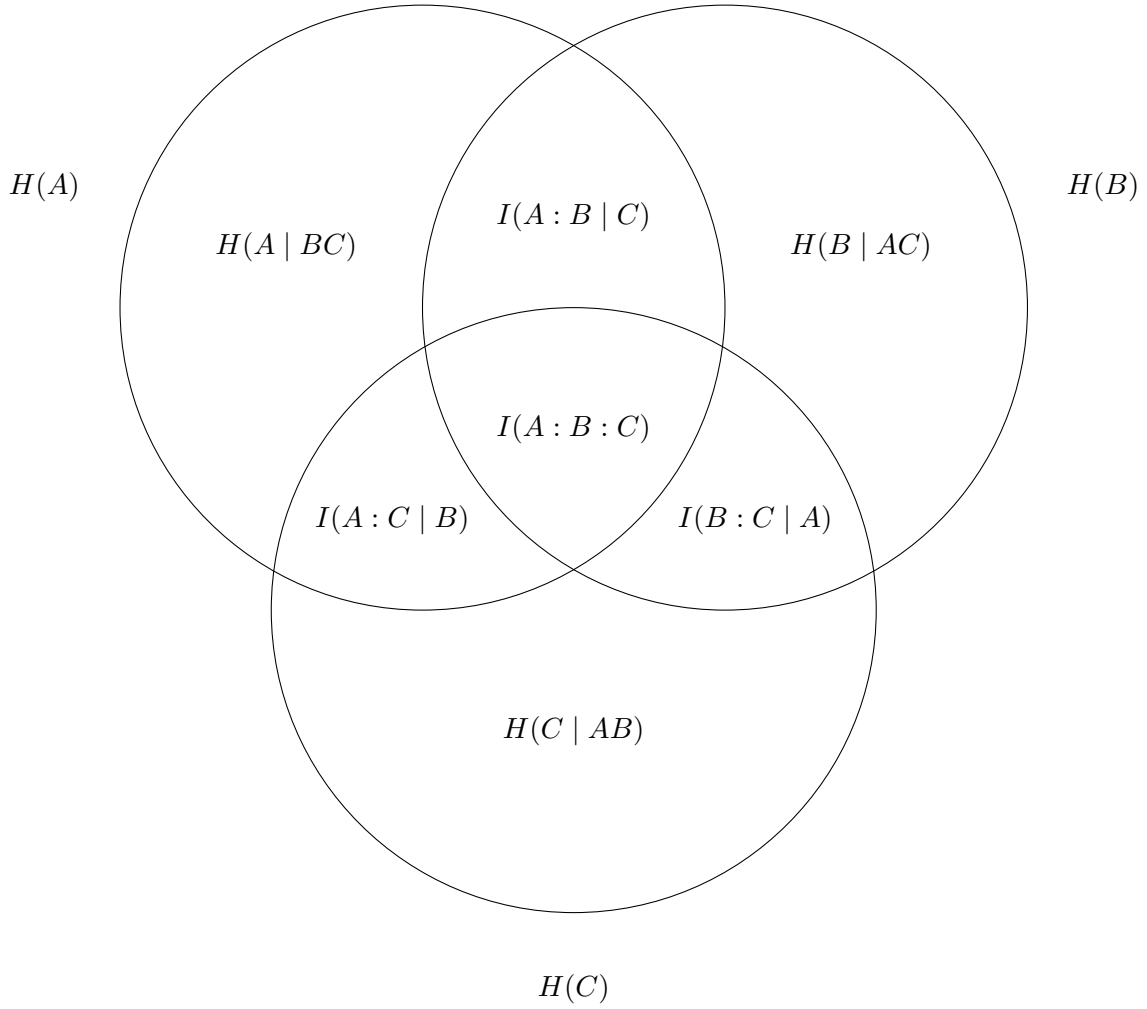


Figure 3.2: Information diagram of three variables
[24]

Finally, the entropy of a Bernoulli distribution with parameter p is called the *binary entropy*.

Definition 3.11. [5] Let $p \in [0, 1]$, then the binary entropy is defined as

$$H_2(p) = -p \log(p) - (1 - p) \log(1 - p),$$

with the convention that $0 \log(0) = 0$.

3.2 Security: statistical distance

While the entropy is a useful measure of uncertainty, it usually not directly used as measure of a key strength. Instead, we use the statistical distance

Definition 3.12. [14] We define the *statistical distance* between distributions $P = (p_1, \ldots, p_n)$ and $Q = (q_1, \ldots, q_n)$ as

$$d(P, Q) = \frac{1}{2} \sum_{i=1}^{n} |p_i - q_i|.$$

We want an adversary to know nothing about a key. Therefore, it should behave similar to a uniform key. So the statistical distance between the distribution key K and a uniform key U should be low. We want to find the relation between the statistical distance to a uniform key $d(H, U)$, and the entropy $H(K)$. The second definition is the entropy $H(K)$.

Intuitively, a key with a lower entropy should have a higher statistical distance to the uniform distribution. To make this relation more concrete, we give a lower bound for the entropy given the statistical distance δ. We do this by finding the distribution with the lowest entropy for the statistical distance δ. This is the distribution where one outcome is more likely than uniform, and one outcome is less likely than uniform. Any other distribution can be forced to have a lower entropy.

Theorem 3.13. *Let* $K = (p_1, \ldots, p_n)$ *and* $U = (\frac{1}{n}, \ldots, \frac{1}{n})$. *Let* $\delta := d(K, U)$, *and* $\epsilon = \lceil \delta n \rceil \cdot \frac{1}{n} - \delta$, *then*

$$H(K) \geq (1 - \epsilon) \cdot H_2\left(\frac{\frac{1}{n} + \delta}{1 - \epsilon}\right) + \left(1 - \epsilon - \frac{1}{n} - \delta\right) \cdot \log(\lfloor n(1 - \delta) \rfloor) + H_2(\epsilon)$$

Proof. Let $P = (\epsilon, 0, \ldots, 0, \frac{1}{n}, \ldots, \frac{1}{n}, \frac{1}{n} + \delta)$. Trivially $d(P, U) = \delta$. Then it follows that

$$\begin{aligned}
H(P) &= H\left(\epsilon, 0, \ldots, 0, \frac{1}{n}, \ldots, \frac{1}{n}, \frac{1}{n} + \delta\right) \\
&= H_2(\epsilon) + (1 - \epsilon) \cdot H\left(0, \ldots, 0, \frac{1}{1 - \epsilon} \cdot \frac{1}{n}, \ldots, \frac{1}{1 - \epsilon} \cdot \frac{1}{n}, \frac{\frac{1}{n} + \delta}{1 - \epsilon}\right) \\
&= H_2(\epsilon) + (1 - \epsilon) \cdot \left(H_2\left(\frac{\delta + \frac{1}{n}}{1 - \epsilon}\right) + \left(1 - \frac{\frac{1}{n} + \delta}{1 - \epsilon}\right) \cdot H\left(\frac{\frac{1}{n} + \delta}{1 - \epsilon} \cdot \frac{1}{n}, \ldots,\right)\right) \\
&= H_2(\epsilon) + (1 - \epsilon) \cdot \left(H_2\left(\frac{\delta + \frac{1}{n}}{1 - \epsilon}\right) + \left(1 - \frac{\frac{1}{n} + \delta}{1 - \epsilon}\right) \cdot \log(\lfloor n(1 - \delta) \rfloor)\right) \\
&= H_2(\epsilon) + (1 - \epsilon) \cdot H_2\left(\frac{\delta + \frac{1}{n}}{1 - \epsilon}\right) + \left(1 - \epsilon - \frac{1}{n} - \delta\right) \cdot \log(\lfloor n(1 - \delta) \rfloor)
\end{aligned}$$

Suppose that $Q = (q_1, q_2, q_3, \ldots, q_n)$, with $d(Q, U) = \delta$, then there are either two outcomes more probable than uniform, or less probable than uniform, but still possible. Therefore, without loss of generality either $\frac{1}{n} < q_1 \leq q_2$ or $0 < q_1 \leq q_2 < \frac{1}{n}$. The entropy is equal to

$$H(Q) = H_2(q_1 + q_2) + (q_1 + q_2)H_2\left(\frac{q_1}{q_1 + q_2}\right) + (1 - q_1 - q_2)H(q_3, \ldots, q_n).$$

If $\frac{1}{n} < q_1 \leq q_2$ then define $Q' = (\frac{1}{n}, q_1 + q_2 - \frac{1}{n}, q_3, \ldots, q_n)$. Then it still holds that $d(Q, U) = \delta$, and

$$\begin{aligned} H(Q') &= H_2(q_1 + q_2) + (q_1 + q_2)H_2\left(\frac{\frac{1}{n}}{q_1 + q_2}\right) + (1 - q_1 - q_2)H(q_3, \ldots, q_n) \\ &< H_2(q_1 + q_2) + (q_1 + q_2)H_2\left(\frac{q_1}{q_1 + q_2}\right) + (1 - q_1 - q_2)H(q_3, \ldots, q_n) \end{aligned}$$

In the case $0 < q_1 \leq q_2$ define $Q' = (0, q_1 + q_2, q_3, \ldots, q_n)$, then

$$H(Q') = H_2(q_1 + q_2) + (1 - q_1 - q_2)H(q_3, \ldots, q_n) < H(Q).$$

Then proceed inductively until Q' becomes equal to P. □

Most of these terms are negligible, as $\epsilon \leq \frac{1}{n}$, so the term $H_2(\epsilon)$ becomes small. Similarly, we have

$$(1 - \epsilon) \cdot H_2\left(\frac{\frac{1}{n} + \delta}{1 - \epsilon}\right) \leq 1.$$

Therefore, we can approximate that a key K that has statistical distance to the uniform key $d(K, U) = \delta$ that

$$H(K) \approx (1 - \delta) \cdot \log(\lfloor n(1 - \delta) \rfloor).$$

On the other hand, by the contrapositive if the entropy of the key is lower

$$H(K) < (1 - \lceil \delta n \rceil) \cdot \log(\lfloor n(1 - \delta) \rfloor),$$

then the statistical distance to a uniform key should be higher. Therefore, a key with low entropy, has a high statistical distance to the uniform distribution and be a bad key.

3.3 Required key strength: Shannon's theorem

We have described measures for the key strength. One of the main uses of cryptographic keys is in symmetric encryption protocols. How strong the key needs to be depends on the distribution of messages that needs to be encrypted. For perfect secrecy, symmetric encryption algorithms need guarantee that it is impossible to recover any information from the encrypted message, called the ciphertext. Shannon's theorem provides a lower bound on the required key strength.

For a symmetric encryption system, we encrypt the plaintext m using a key k such that any eavesdropper can extract as little information as possible from the ciphertext c, such that

$$c = \mathsf{Enc}_k(m)$$
$$m = \mathsf{Dec}_k(c)$$

For perfect security, we have two requirements:

1. Correctness: we can recover the message from the key and the ciphertext. Therefore, there should be no uncertainty about m given c and k. In entropy terms $H(M \mid CK) = 0$.
2. Security: without any knowledge of the value of key K, the variables M and C should be completely independent. Therefore, $I(M : C) = 0$.

Shannon's theorem shows that these are difficult requirements to fulfill together. It gives a rather strict requirement on the entropy of the key.

Theorem 3.14. *(Shannon's theorem) [20] For any perfectly secure cryptography system, we have $H(K) \geq H(M)$.*

Proof. We assume that the key and the message are independent, as we would like to combine arbitrary keys and messages that can be generated independently. Therefore, $I(M : K) = 0$. Since $H(M \mid CK) = 0$ we have $H(M) = H(M : CK)$. Let

$$\begin{aligned}
n &= H(M) \\
&= H(M) - H(M \mid CK) && H(M \mid CK) = 0 \text{ by correctness} \\
&= I(M : CK) && \text{definition of } I \\
&= I(M : C) + I(M : K) - I(M : C : K) \\
&= -I(M : C : K) && \text{assumption } I(M : C) = I(M : K) = 0.
\end{aligned}$$

Then

$$\begin{aligned}
I(K : CM) &= I(K : C) + I(K : M) - I(M : C : K) \\
&\geq I(K : M) - I(M : C : K) \\
&= -I(M : C : K) \\
&= n.
\end{aligned}$$

Finally, we have

$$H(K) = I(K : CM) + H(K \mid CM) \geq I(K : CM) \geq n.$$

□

On the other hand, Quantum Key Distribution claims to achieve statistical security, without any pre-shared key. Therefore, it could be an indication that Shannon's theorem does not hold in the quantum case.

We will not explicitly describe a quantum cryptography protocol that uses small key sizes. However, we will describe the Quantum Key Distribution protocol that can be used to increase the entropy of the key. Theoretically, QKD can then be used as part of a larger protocol, that then uses the key generated by QKD to achieve perfect security.

The requirements of Shannon's theorem may be much more strict than they look at a first glance. It assumes that Bob and Eve always receive exactly the same ciphertext. If Eve receives the bits with at least some error, then it is possible to find protocols that provide more security than Shannon's theorem would seem to imply. [10]

3.4 Key-exchange protocols: independent private randomness

Shannon's theorem is about symmetric encryption systems, where the honest parties share an existing key. On the other hand, as the name implies, Quantum Key Distribution is about exchanging information-theoretically secure keys over a public channel. Therefore, we want to find the limits of information-theoretical key-exchange using purely classical channels.

First we have to define a key-exchange protocol more precisely. Two honest parties. Alice and Bob send each other messages over a public channel, which can be eavesdropped on by Eve. Then Alice and Bob use these messages to derive a key.

Key-exchange protocols cannot be deterministic, since it should not yield the same key every time the protocol is run. Alice and Bob should both have access to some form of private randomness. To make it easier to reason about the protocol, we define the protocol as deterministic functions. These functions have the private randomness as explicit inputs. First, we look at the case where the local randomness of Alice and Bob is independent.

Definition 3.15. A key-exchange protocol is a tuple

$$(P_A, P_B, \mathsf{Key}_A, \mathsf{Key}_B, \mathsf{Mesg}_a, \mathsf{Mesg}_b),$$

with P_A and P_B probability distributions for the private randomness A and B. Alice and Bob sample their randomness P_A and P_B independently. Then Mesg_A, Mesg_B are functions that define the messages that Alice and Bob send each other such that

$$\begin{aligned}
M_{a_0} &= \mathsf{Mesg}_a(0, A) \\
M_{b_0} &= \mathsf{Mesg}_b(0, B, a_0) \\
M_{a_{i+1}} &= \mathsf{Mesg}_a(i+1, A, M_{a_0}, M_{b_0}, M_{a_1}, M_{b_1}, \ldots, M_{a_i}, M_{b_i}) \\
M_{b_{i+1}} &= \mathsf{Mesg}_b(i+1, B, M_{a_0}, M_{b_0}, M_{a_1}, M_{b_1}, \ldots, M_{a_i}, M_{b_i}, M_{a_{i+1}}).
\end{aligned}$$

Without loss of generality, Alice and Bob take turns sending messages. Every message M_{a_i}, M_{b_i} is either a bit string or the symbol $\perp$ to denote that the protocol has finished. If Alice sends a $\perp$ then without loss of generality Bob repeats the message. The messages from Alice, combined with the messages from Bob form a public transcript T. This T is only dependent on the private randomness A, B, and the message functions. For a given key-exchange protocol, we can interpret T as a function of the private randomness.

$$T = (M_{a_0}, M_{b_0}, \ldots, M_{a_n}, M_{b_n}) = \mathsf{Trans}(A, B)$$

Although we reason about the public transcript as a function of the private randomness, neither Alice nor Bob have access to both A and B. Alice and Bob can derive their key

based on their private randomness, and the public messages.

$$K_A = \mathsf{Key}_A(A, \mathsf{Trans}(A, B))$$
$$K_B = \mathsf{Key}_B(B, \mathsf{Trans}(A, B))$$

Of course, reasonable key-exchange protocols should conform to a few properties, similar to the symmetric case, namely

1. Correctness: Alice and Bob should end up with the same key, therefore we want the probability $\Pr(K_A = K_B)$ as high as possible.
2. Security: Based on the public transcript T, Eve should have as little knowledge about the key as possible. There are different security notions to define what "little knowledge" means.

For information-theoretical security, we want Eve to have as much uncertainty as possible about the key. Unlike computational security, we do not care about the complexity of the attack that Eve has to perform to deduce the key. Even an unbounded eavesdropper should not be able to get much information about the key from the transcript T.

3.4.1 Impossibility of classical security

Unfortunately, for classical protocols it turns out that information-theoretical security is more or less impossible. First we focus on the case that the key is always correct, that is $\Pr(K_A = K_B) = 1$.

Definition 3.16. A key-exchange protocol $(P_A, P_B, \mathsf{Key}_A, \mathsf{Key}_B, \mathsf{Mesg}_a, \mathsf{Mesg}_b)$ with transcription function Trans is called perfectly correct if

$$\mathsf{Key}(a, \mathsf{Trans}(a, b)) = \mathsf{Key}(b, \mathsf{Trans}(a, b))$$

for all a, b. For these protocols we can define $K := K_a = K_b$.

In that case, Eve can always perfectly recover the key based on the transcript. Intuitively, Eve can brute-force the private randomnesses of Alice and Bob. Based on the private randomness, Eve can simulate the protocol and check if the generated transcript matches the intercepted transcript. Then Alice can derive the key using the private randomness of Alice or Bob to derive the key.

Lemma 3.17. *Let* Trans *be the transcript function of a key-exchange protocol. If*

$$\mathsf{Trans}(a, b) = \mathsf{Trans}(a', b')$$

then

$$\mathsf{Trans}(a, b) = \mathsf{Trans}(a', b) = \mathsf{Trans}(a, b') = \mathsf{Trans}(a', b')$$

Proof. Suppose

$$t = (m_{a_0}, m_{b_0}, \ldots, m_{a_n}, m_{b_n}) = \mathsf{Trans}(a, b) = \mathsf{Trans}(a', b'),$$

Define

$$t' = (m'_{a_0}, m'_{b_0}, \ldots, m'_{a_m}, m'_{b_m}) = \mathsf{Trans}(a', b).$$

We prove by induction that $m = n$ and

$$\begin{aligned} m_{a_i} &= m'_{a_i} \\ m_{b_i} &= m'_{b_i} \end{aligned}$$

Since $\mathsf{Trans}(a, b) = \mathsf{Trans}(a', b')$, we know that

$$\begin{aligned} m_{a_i} &= \mathsf{Mesg}_a(i, a, m_{a_0}, m_{b_0}, \ldots, m_{a_{i-1}}, m_{b_{i-1}}) \\ &= \mathsf{Mesg}_a(i, a', m_{a_0}, m_{b_0}, \ldots, m_{a_{i-1}}, m_{b_{i-1}}) \\ m_{b_i} &= \mathsf{Mesg}_b(i, b, m_{a_0}, m_{b_0}, \ldots, m_{a_{i-1}}, m_{b_{i-1}}, a_i) \\ &= \mathsf{Mesg}_b(i, b', m_{a_0}, m_{b_0}, \ldots, m_{a_{i-1}}, m_{b_{i-1}}, a_i). \end{aligned}$$

Then we can easily see with induction that all messages are equal. For $i = 0$ we know that

$$m_{a_0} = \mathsf{Mesg}_a(0, a) = \mathsf{Mesg}_a(0, a') = m'_{a_0}$$

and

$$m_{b_0} = \mathsf{Mesg}_b(0, b, m_{a_0}) = \mathsf{Mesg}_b(0, b, m'_{a_0}).$$

Suppose for $i < k$ it holds that

$$\begin{aligned} m_{a_i} &= m'_{a_i} \\ m_{b_i} &= m'_{b_i} \end{aligned}$$

then it holds that

$$\begin{aligned} m_{a_k} &= \mathsf{Mesg}_a(k, a, m_{a_0}, m_{b_0}, \ldots, m_{a_{k-1}}, m_{b_{k-1}}) \\ &= \mathsf{Mesg}_a(k, a', m'_{a_0}, m'_{b_0}, \ldots, m'_{a_{k-1}}, m'_{b_{k-1}}) \\ &= m'_{a_k} \\ m_{b_k} &= \mathsf{Mesg}_b(k, b, m_{a_0}, m_{b_0}, \ldots, m_{a_{k-1}}, m_{b_{k-1}}, m_{a_k}) \\ &= \mathsf{Mesg}_b(k, b, m'_{a_0}, m'_{b_0}, \ldots, m'_{a_{k-1}}, m'_{b_{k-1}}, m'_{a_k}) \\ &= m'_{b_k} \end{aligned}$$

The protocol runs until $m_{b_k} = \perp$ for some k. Since all the messages are the same, we know that the length of the protocol also stays the same. Therefore, we conclude

$$\mathsf{Trans}(a, b) = \mathsf{Trans}(a', b).$$

The case

$$\mathsf{Trans}(a, b) = \mathsf{Trans}(a, b')$$

works similar. □

Now we can prove that perfectly correct key-exchange protocols are information theoretically insecure.

Theorem 3.18. *Let* $(P_A, P_B, \mathsf{Key}_A, \mathsf{Key}_B, \mathsf{Mesg}_a, \mathsf{Mesg}_b)$ *be a perfectly correct exchange protocol with transcription function* Trans*. Let* (A, B) *be private randomness used to generate a transcript* $T = \mathsf{Trans}(A, B)$ *and keys* $K_a = \mathsf{Key}_a(A, T)$ *and* $K_b = \mathsf{Key}_b(B, T)$*. Then Eve can generate a key* K' *with* $H(K' \mid T) = 0$ *and*

$$\Pr(K_a = K') = \Pr(K_a = K_b).$$

Proof. An attacker Eve with access to a transcript T can iterate over all $(A', B') \in P_a \times P_b$ until

$$\mathsf{Trans}(A', B') = T$$

Then, Eve can use the key derivation function to calculate

$$K' = \mathsf{Key}_b(B', \mathsf{Trans}(A', B'))$$

Then by applying Lemma 3.17 multiple times we have since B and B' are identically distributed

$$\begin{aligned}
\Pr(K_a = K') &= \Pr(\mathsf{Key}_a(A, \mathsf{Trans}(A, B)) = \mathsf{Key}_b(B', \mathsf{Trans}(A', B'))) \\
&= \Pr(\mathsf{Key}_a(A, \mathsf{Trans}(A, B')) = \mathsf{Key}_b(B', \mathsf{Trans}(A, B'))) \\
&= \Pr(\mathsf{Key}_a(A, \mathsf{Trans}(A, B)) = \mathsf{Key}_b(B, \mathsf{Trans}(A, B))) \\
&= \Pr(K_a = K_b)
\end{aligned}$$

□

Corollary 3.19. *For a perfectly correct key-exchange protocol* $(P_A, P_B, \mathsf{Key}_A, \mathsf{Key}_B, \mathsf{Mesg}_a, \mathsf{Mesg}_b)$ *with key* K *and transcript* T *we have*

$$H(K \mid T) = 0$$

Proof. In a perfectly correct key-exchange protocol we have

$$\Pr(K_a = K_b) = 1.$$

By Theorem 3.18 we know that Eve can generate a key K' with $H(K' \mid T) = 0$ with

$$\Pr(K_a = K') = \Pr(K_a = K_b) = 1$$

Therefore $H(K_a \mid T) = H(K' \mid T) = 0$. □

3.5 Key-exchange protocols: correlated private randomness

For key-exchange protocols we have made the assumption that the private randomness of Alice A and Bob B are completely independent. In practice, there might be some correlation between A and B. For example Alice and Bob might have access to some pre-shared secret string. Even some correlated bits can be used to extract a key. Therefore, we now extend our focus to the case where the private randomness of Alice and Bob are jointly distributed as P_{AB}.

Note that this correlation between A and B breaks the proof of Lemma 3.18. If Alice has sampled private randomness a, the distribution of Bob from Alice's perspective is $P_{B|A=a}$ while Eve only has access to P_B.

However, there is still a limit on the entropy that can be extracted from P_{AB}. We can extract this limit from an entropy diagram.

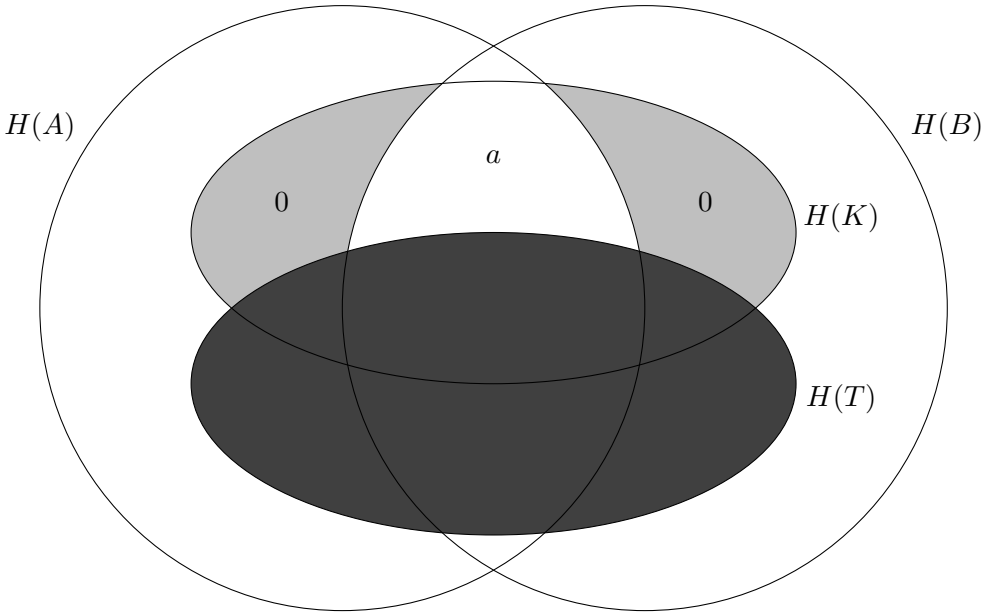
Theorem 3.20. *For a key purification protocol we have*

$$H(K \mid T) \leq I(A : B \mid T).$$

Proof. In this proof we use an information diagram of four variables: A, B, K, T. First we simplify the diagram by noting that, since the entire protocol is dependent on the private randomness A and B, we have

$$H(T \mid AB) = H(K \mid AB) = 0.$$

Therefore, we can write the circles of T and K inside AB.



Since Alice can calculate the key based on the transcript T and her private randomness A we have

$$H(K \mid AT) = 0$$

□

3.5.1 Special case: pre-shared key

A very simple case would be where A and B are perfectly correlated, that is $\Pr(A = B) = 1$. This perfect correlation happens when Alice and Bob have access to a pre-shared randomly sampled bit string. Then Alice and Bob could directly use their private randomness as key without any communication. Then it holds that $H(AB) = H(A) = H(B)$.

$$I(A : B) = H(A) + H(B) - H(AB) = H(A) + H(B) - H(B) = H(A).$$

Entropy-wise, any generated key can only have as much entropy as the pre-shared secret.

We could extend this case by adding private randomness to a pre-shared key. We can describe this combination as $A = (A_P, S)$ and $B = (B_P, S)$, with A_P, B_P private randomness and S a pre-shared secret. Then A_P, B_P and S are pairwise independent. Then we have

$$\begin{aligned} I(A : B) &= H(A) + H(B) - H(AB) = H(A_P S) + H(B_P S) - H(A_P B_P S) \\ &= H(A_P) + H(S) + H(B_P) + H(S) - (H(A_P) + H(B_P) + H(S)) \\ &= H(S). \end{aligned}$$

If the pre-shared key S has entropy $H(S) = n$, then we know by Theorem 3.20 that the entropy of any generated key K must be less than the entropy of the pre-shared secret S. Therefore, entropy-wise it is optimal to just use the pre-shared secret as key.

In this chapter we have limited the entropy of the key based on the amount of pre-shared information available to Alice and Bob. We also conclude that if the entropy of the key is low, the statistical distance to a uniform key is high, and the quality of the key is not good.

4 Quantum states

4.1 Qubits

Quantum Cryptography is based on communicating using quantum systems. In this section, we introduce the necessary axioms and notation to reason about quantum systems. We represent quantum systems using linear algebra, which is the standard in theoretical computer science. A quantum state is an element of a vector space. In general, this vector space can be an arbitrary Hilbert space, which is potentially infinite-dimensional. However, as all the states are finite-dimensional, we limit ourselves to elements of $\mathbb{C}^n$.

Remark 4.1. Unless otherwise specified, for vectors $x, y \in \mathbb{C}^n$ we use the standard inner product $\langle \cdot, \cdot \rangle$ with the induced $\|_2$ norm.

Defined as

$$\langle v, w \rangle = \sum_{i=1}^{n} \overline{v_i} w_i$$

and the induced norm

$$\|v\|_2 = \sqrt{\sum_{i=1}^{n} |v_i|^2} = \sqrt{\langle v, v \rangle}$$

For communication, we need to store data in a quantum state. We can define a quantum state as follows:

Definition 4.2. A *quantum state* is a vector $v \in \mathbb{C}^n$ with $\| |v\rangle \| = 1$.

When we manipulate quantum states, we often write down products of row vectors, column vectors, and matrices. To make it easy to see the type of the object, we use the bra-ket notation for these vectors.

Definition 4.3. We notate a column vector $\phi \in \mathbb{C}^n$ as $|\phi\rangle$ called a *ket* vector. We notate the *bra* vector as $\langle \phi | = \phi^* = \overline{\phi}^\top$.

The quantum state is equivalent of a classical bit string, a sequence of bits. A classical bit can have two discrete values, usually represented by the numbers 0 and 1. The quantum equivalent of a bit is called a *qubit*. A single qubit is a relatively simple instance of a quantum system.

Definition 4.4. A *qubit* is a vector $v \in \mathbb{C}^2$ with $\|v\| = 1$.

The simplest way to use a qubit is to let it store a classical bit either 0 or 1.

Definition 4.5. The quantum representations of the classical bits are defined as

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

But qubits can do more than just representing a classical bit. Qubits can also be in superposition, simultaneously being 0 and 1. Let a qubit

$$|\phi\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = a_0 |0\rangle + a_1 |1\rangle ,$$

can be interpreted as a state that is 0 with probability $|a_0|^2$ and is 1 with probabilities $|a_1|^2$.

Remark 4.6. For vectors $\phi, \psi \in \mathbb{C}^n$,

$$\langle \phi | \psi \rangle \in \mathbb{C}$$

represents the standard inner product, while

$$|\phi\rangle\langle\psi| \in \mathbb{C}^{n \times n}$$

represents the outer product, generating a rank-1 matrix. Unlike the inner product, the outer product is well-defined on vectors of different sizes.

4.2 Measurements

In theory, qubits exist in a continuous domain, and it is possible to manipulate them as such. However, to get any information from the qubit, it is necessary to perform a measurement. There are many possible measurements. But compared to measuring classical bits, there are three counter-intuitive differences. These differences are fundamental when designing quantum cryptography systems.

- A measurement is probabilistic: the same measurement performed on identical qubits may yield different results.
- A measurement is destructive: the state of the qubit is changed after the measurement.
- A measurement does not fully quantify the qubit: it is impossible to get the exact state of an unknown qubit that can have arbitrary value using a measurement.

By quantum mechanics, measurements, like operations, have to be linear. Therefore, operations can be represented as matrices. We have seen that operations in a closed quantum system are represented by unitary matrices, which are by definition invertible. Measurements do not happen in closed quantum systems, therefore they can be destructive. However, there are some other constraints to ensure that the measurement results are valid.

For a measurement, measurements results can occur, which are all represented by a matrix. Therefore, a measurement is represented by a set of matrices. We have to define two things: the state of the quantum system after the measurement, and the probability that we measure a certain result. Since the probabilities should sum up to one, there are constraints on the set of matrices that define a valid measurement.

Definition 4.7. A matrix A in $\mathbb{C}^{n\times n}$ is positive semi-definite if for all $|\phi\rangle \in \mathbb{C}^n$ it holds that

$$\langle\phi| A |\phi\rangle \geq 0.$$

Definition 4.8. A positive-operator-valued measure (POVM) is a set positive semi-definite operators/matrices $\{M_i\} \subseteq C^{n\times n}$ such that

$$\sum_i M_i^* M_i = I$$

Definition 4.9. Let $\{M_i\}_i$ be a POVM, and let $|\phi\rangle \in \mathbb{C}^n$ a (normalized) quantum state. Then the probability of the measurement $\{M_i\}$ is

$$p(i) = \|M_i |\phi\rangle\|^2.$$

The outcome probabilities sum up to one as

$$\begin{aligned}
\sum_i p(i) = \|M_i |\phi\rangle\|^2 &= \sum_i \langle\phi| M_i^* M_i |\phi\rangle \\
&= \langle\phi| \left(\sum_i M_i^* M_i\right) |\phi\rangle \\
&= \langle\phi| I |\phi\rangle && \text{by definition of a POVM} \\
&= \||\phi\rangle\|^2 = 1. && \text{quantum states are normalized}
\end{aligned}$$

The resulting state of the quantum system after the measurement is

$$\frac{M_i |\phi\rangle}{\|M_i |\phi\rangle\|}.$$

The post-measurement state is by definition normalized and therefore a quantum state.

The simplest form of measurement is a measurement in an orthogonal basis. We can define a POVM measurement based on an orthogonal basis.

Lemma 4.10. *Let $|\phi\rangle \in \mathbb{C}^n$ a qubit and $\{|\psi_1\rangle, \ldots, |\psi_n\rangle\}$ an orthogonal basis in $\mathbb{C}^n$. Let $M_i = |\psi_i\rangle\langle\psi_i|$. Then $\{M_1, \ldots, M_n\}$ is a POVM. The result of the measurement in the basis $\{|\psi_1\rangle, \ldots, |\psi_n\rangle\}$ is*

$$|\psi_i\rangle$$

with probability

$$|\langle\phi|\psi_i\rangle|$$

The post-measurement state of the qubit after the measurement outcome i is

$$|\psi_i\rangle$$

Proof. First we prove that $\{|\psi_1\rangle\langle\psi_1|, \ldots, |\psi_n\rangle\langle\psi_n|\}$ is a POVM. Every $|\psi_i\rangle$ is trivially Hermitian and positive semi-definite since

$$\langle\phi| \left(|\psi_i\rangle\langle\psi_i|\right) |\phi\rangle = \left(\langle\phi|\psi_i\rangle\right)^2 \geq 0.$$

Now we aim to show that

$$\sum_{i=1}^{n} M_i^* M_i = I,$$

Equivalently, take an arbitrary $|\phi\rangle \in \mathbb{C}^n$, and we aim to show

$$\left(\sum_{i=1}^{n} M_i^* M_i\right) |\phi\rangle = |\phi\rangle$$

First note that

$$\begin{aligned}\sum_{i=1}^{n} M_i^* M_i &= \sum_{i=1}^{n} (|\psi_i\rangle\langle\psi_i|)^* |\psi_i\rangle\langle\psi_i| = \sum_{i=1}^{n} |\psi_i\rangle \langle\psi_i|\psi_i\rangle \langle\psi_i| \\ &= \sum_{i=1}^{n} |\psi_i\rangle\langle\psi_i|\end{aligned}$$

Now let $|\phi\rangle$ be an arbitrary vector, and decompose it the basis as $|\phi\rangle = \sum_j \alpha_j |\psi_j\rangle$. Then

$$\begin{aligned}\left(\sum_{i=1}^{n} M_i^* M_i\right) |\phi\rangle &= \left(\sum_{i=1}^{n} |\psi_i\rangle\langle\psi_i|\right) \left(\sum_{j=1}^{n} \alpha_j |\psi_j\rangle\right) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_j |\psi_i\rangle \langle\psi_i|\psi_j\rangle\end{aligned}$$

since $|\psi_i\rangle$ forms an orthogonal basis, we know that $\langle\psi_i|\psi_j\rangle = 0$ if $i \neq j$ and $\langle\psi_i|\psi_j\rangle = 1$ if $i = j$. Thus,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_j |\psi_i\rangle \langle\psi_i|\psi_j\rangle = \sum_{i} \alpha_i |\psi_i\rangle = |\phi\rangle$$

Therefore

$$\left(\sum_{i=1}^{n} M_i^* M_i\right) |\phi\rangle = |\phi\rangle$$

Since $|\phi\rangle$ is arbitrary we conclude that $\sum_i M_i^* M_i = I$.

We get measurement result i with probability

$$\begin{aligned} p(i) &= \| |\phi_i\rangle \langle\phi_i|\phi\rangle \|^2 = \langle\phi_i|\phi_i\rangle \langle\phi|\phi\rangle \langle\phi_i|\phi_i\rangle \\ &= \langle\phi_i|\phi\rangle^2 . \end{aligned}$$

The resulting state of the qubit after measurement result i becomes

$$\frac{|\phi_i\rangle \langle\phi_i|\phi\rangle}{\| |\phi_i\rangle \langle\phi_i|\phi\rangle \|} = \frac{|\phi_i\rangle \langle\phi_i|\phi\rangle}{\langle\phi_i|\phi\rangle} = |\phi_i\rangle ,$$

which finishes the proof. □

Quantum states can be measures in arbitrary bases. Some bases are commonly used and have their own name.

Definition 4.11. The computational basis is

$$\{|0\rangle , |1\rangle\}$$

Definition 4.12. The Hadamard basis is

$$\{|+\rangle , |-\rangle\}$$

where

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

4.3 Density matrix

We have described how one can reason about qubits as vectors. However, sometimes the vector representation of qubits is unsatisfactory. For example, only know the probability distribution over a set of qubits, and want to reason about it. For example, a qubit $|\psi\rangle$ can be in state $|0\rangle$ with probability 0.5, and in state $|1\rangle$ with probability 0.5. It may be tempting to model this state as

$$|\psi'\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

Though at a first glance similar-looking, $|\psi\rangle$ and $|\psi\rangle'$ are completely different states. We can always perfectly recognize the state $|\psi'\rangle$ by a measurement in the Hadamard basis. The result of this measurement is always $|\psi'\rangle$. On the other hand, if we measure $|\psi\rangle$, the result will always be unpredictable no matter the choice of measurement basis. We introduce the density matrix to make it easier to reason about probability distributions of qubits.

Definition 4.13. [14] A density matrix is a matrix $M \in \mathbb{C}^{n\times n}$ such that

1. M is Hermitian, $M^* = M$.
2. M is positive semi-definite.
3. $\mathrm{tr}(M) = 1$.

A density matrix can be interpreted as the weighted average of multiple quantum states using the spectral theorem. First, we introduce some more convenient notation for the spectral decomposition.

Lemma 4.14. *[14] Let $M \in \mathbb{C}^{n\times n}$ be a Hermitian matrix. Then we can write*

$$M = \sum_i \lambda_i \left|\phi_i\right\rangle\!\left\langle\phi_i\right| .$$

For some scalars λ_i and $\{|\phi_i\rangle\}_i \in \mathbb{H}$ an orthogonal basis, with $\sum_i \lambda_i = \mathrm{tr}(M)$.

Proof. Since M is Hermitian, we can use the spectral theorem to calculate the eigenvalue decomposition.

$$M = UDU^*$$

where U is unitary and D is diagonal. Let $\lambda_i := D_{ii}$ be the diagonal entries. Then since the trace is invariant under chance of basis $\sum_i \lambda_i = \mathrm{tr}(D) = \mathrm{tr}(M)$. Let $|\phi\rangle_i = U_i$ the i'th column of U. Finally, let $|i\rangle = e_i$ the unit vector. Then we write can write

$$U = \sum_i \left|\phi_i\right\rangle\!\left\langle i\right| , \; D = \sum_j \lambda_j \left|j\right\rangle\!\left\langle j\right|$$

Since $\langle i|j\rangle = 0$ if $i \neq j$, and $\langle i|i\rangle = 1$. This gives desired expression for M, namely

$$\begin{aligned} M &= \left(\sum_i \left|\phi_i\right\rangle\!\left\langle i\right|\right)\left(\sum_j \lambda_j \left|j\right\rangle\!\left\langle j\right|\right)\left(\sum_k \left|k\right\rangle\!\left\langle\phi_k\right|\right) = \sum_{i,j,k} \lambda_j \left|\phi_i\right\rangle \left\langle i|j\right\rangle \left\langle j|k\right\rangle \phi_k \\ &= \sum_i \lambda_i \left|\phi_i\right\rangle \left\langle i|i\right\rangle \left\langle i|i\right\rangle \left\langle\phi_i\right| = \sum_i \lambda_i \left|\phi_i\right\rangle\!\left\langle\phi_i\right| . \end{aligned}$$

□

This motivates why the trace of the density matrix should be one. It represents a probability distribution, over multiple qubits. Hence, we can define the density matrix of a distribution of qubits

Example 4.15. *We want to describe a qubit that is in state $|\psi_k\rangle \in \mathbb{C}^n$ with probability $p_k \in [0, 1]$, such that $\sum_k p_k = 1$. Then, the corresponding density matrix $\rho \in \mathbb{C}^{n\times n}$ is*

$$\rho := \sum_k p_k \left|\psi_k\right\rangle\!\left\langle\psi_k\right| .$$

This is indeed a density matrix since $\rho^* = \rho$, $\operatorname{tr}(\rho) = \sum_k p_k = 1$, *and it is positive semi-definite since for arbitrary* $|\phi\rangle \in H$

$$\langle\phi| \rho |\phi\rangle = \sum_k p_k \langle\phi|\psi_k\rangle \langle\psi_k|\phi\rangle = \sum_k p_k (\langle\phi|\psi_k\rangle)^2 \geq 0.$$

Remark 4.16. Note that the decomposition of resulting qubits need not be unique since

$$\frac{1}{2} I_2 = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) = \frac{1}{2}(|+\rangle\langle +| + |-\rangle\langle -|).$$

We now introduce the distinction between a pure state and a mixed state. Some density matrices consist of one vector, while others are a combination of different vectors.

Definition 4.17. [14] A state with density matrix ρ is called pure if

$$\rho = |\phi\rangle\langle\phi|$$

for some vector $|\phi\rangle$. Otherwise, the density matrix is called mixed.

On density matrices, we can also apply any POVM measurement. To do this we rewrite the definitions of POVM measurements for vectors.

Lemma 4.18. *The trace operation is cyclic, it holds* $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ *for all* $A, B \in \mathbb{C}^{n\times n}$

Proof. We see that with A_i the i'th column of A

$$\operatorname{tr}(AB) = \sum_i (AB)_{ii} = \sum_i (A^\top)_i B_i = \sum_i (B^\top)_i A_i = \sum_i (BA)_{ii} = \operatorname{tr}(BA)$$

□

Lemma 4.19. *[14] Let* $\{M_i\}_i \subseteq \mathbb{C}^{n\times n}$ *be a POVM, and let* $|\phi\rangle \in \mathbb{C}^n$. *Then we get the measurement result* i *with probability*

$$p(i) = \operatorname{tr}(M_i |\phi\rangle\langle\phi| M_i^*)$$

and the post-measurement state is

$$\frac{M_i |\phi\rangle}{\sqrt{\operatorname{tr}(M_i^* |\phi\rangle\langle\phi| M_i)}}$$

Proof. We see, since

$$p(i) = \|M_i |\phi\rangle\| = \langle\phi| M_i^* M_i |\phi\rangle$$

This is number, which is its own trace, here we can use the cyclic property

$$p(i) = \operatorname{tr}(\langle\phi| M_i^* M_i |\phi\rangle) = \operatorname{tr}(M_i |\phi\rangle\langle\phi| M_i^*)$$

Then the measurement result is

$$\frac{M_i |\phi\rangle}{\sqrt{p_i}} = \frac{M_i |\phi\rangle}{\sqrt{\operatorname{tr}(M_i^* |\phi\rangle\langle\phi| M_i)}}$$

□

Then using the linearity of matrix multiplication and the trace we see that

Definition 4.20. [14] Let $\{M_i\}_i \subseteq \mathbb{C}^{n\times n}$ be a POVM, and let $\rho \in \mathbb{C}^{n\times n}$ be a density matrix. Then we get measurement result i with probability

$$p(i) = \operatorname{tr}(M_i \rho M_i^*)$$

with the resulting state

$$\frac{M_i \rho M_i^*}{\operatorname{tr}(M_i \rho M_i^*)}.$$

4.4 Tensor product

We now know how a single qubit works. Classically we can combine multiple bits simply by concatenating them. We want to also work with multiple qubits and reason about how they interact. However, in the quantum case, the interaction between qubits is more subtle. To model this interaction, we introduce the tensor product

Definition 4.21. Let $|\phi\rangle \in \mathbb{C}^n$ and $|\psi\rangle \in \mathbb{C}^m$. We define $|\phi\rangle \otimes |\psi\rangle \in \mathbb{C}^{nm}$ as

$$|\phi\rangle \otimes |\psi\rangle = \begin{bmatrix} |\phi\rangle_1 |\psi\rangle_1 \\ |\phi\rangle_1 |\psi\rangle_2 \\ \cdots \\ |\phi\rangle_1 |\psi\rangle_m \\ |\phi\rangle_2 |\psi\rangle_1 \\ |\phi\rangle_2 |\psi\rangle_2 \\ \cdots \\ |\phi\rangle_n |\psi\rangle_m . \end{bmatrix}$$

Notice that the tensor product is not commutative. A simple case of this is

$$|0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

while

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

We can reason about the resulting vector $|\phi\rangle \otimes |\psi\rangle$ in the usual ways. Performing measurements, operations in the usual ways.

4.5 Operations

We want to model how to operate on quantum states. It turns out that we can model all such operation that do not interact with the outside environment, as unitary matrices. Therefore, the result of an operation U on qubit $|\psi\rangle$ can be described as

$$U |\phi\rangle$$

Note that every unitary matrix is invertible. Therefore, unlike quantum measurements, every quantum operation U can be undone, since

$$U^* U |\phi\rangle = |\phi\rangle$$

4.6 Partial trace

We have seen that we can combine smaller systems into larger systems using the tensor product. It is also possible to go the other way around: find the impact of an operation on a big system on a smaller system. This "inverse" is called the partial trace. The partial trace only works on density matrices, not on vectors.

Let $A = \mathbb{C}^{n \times n}$ and $B = \mathbb{C}^{m \times m}$ are vector spaces of density matrices describing quantum systems. The space $A \otimes B$ contains vector the combined system. If we want to "ignore" the B part of the state we have an operation of type:

$$\mathrm{tr}_B : A \otimes B \mapsto A.$$

As the partial trace ignores the B part, for $\rho^A \in \mathbb{A}$ and $\rho^B \in \mathbb{B}$ it should conform to

$$\mathrm{tr}_B(\rho^A \otimes \rho^B) = \rho^A$$

Definition 4.22. [14] The partial trace is defined for rank-1 matrices as

$$\mathrm{tr}_B(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|) = |a_1\rangle\langle a_2| \,\mathrm{tr}(|b_1\rangle\langle b_2|)$$

extend this by linearity to arbitrary matrices.

The partial trace is a linear operation, so it is only the inverse of the tensor product up to a constant factor. However, in the case of density matrices it is an inverse as $\mathrm{tr}(\rho) = 1$ per definition of a density matrix.

5 Quantum Key Distribution

Quantum Key Distribution (QKD) is a family of cryptographic protocols that perform key-exchange using communication of quantum states. The promise of QKD is that it generates keys that are provably information-theoretically secure. In other words, it does not rely on any computational assumptions. But the security of Quantum Key Distribution does rely on quantum mechanics working as is currently expected. In this chapter we first describe the working of Quantum Key Distribution, and then describe the formal security guarantees that QKD aims to provide.

In the limits on information-theoretical security in Chapter 3, we have assumed that Eve and Bob would receive the exact same messages. Because QKD is based on the transfer of quantum states this assumption does not hold, because the measurements taken by Bob and Eve are probably not the same. Therefore, Bob can gain an advantage over Eve and QKD can provide information-theoretical security.

5.1 Protocol

Quantum Key Distribution is not a single protocol, but rather a family of Key Exchange Protocols based on quantum communication. In this section we explain the working of one of these protocols: BB84.

On a high level, BB84 works as follows:

1. Alice generates a uniformly random string, encodes it as a quantum state and sends it to Bob via an insecure quantum channel. Eve can perform any measurement or operation on this state. Bob then measures the quantum state, and receives a string that is correlated with Alice's original.
2. Similar to classical key-exchange protocols, using error correcting codes Alice and Bob ensure they have the same string.
3. The resulting string is hashed to a shorter string, to "remove" any correlation with the measurements Eve may have made. This resulting hash is the key.

5.1.1 Quantum channel

First Alice generates a (classical) secret string $a' \leftarrow \{0,1\}^n$. Simply encoding this in a certain basis does not accomplish anything, since then the bits can be perfectly recovered

by a simple measurement. Therefore, Alice encodes every bit randomly in either the computational basis, or the Hadamard basis. Formally, Alice picks a basis to encode it in as $s^{(a)} \leftarrow \{0,1\}^n$. We then define

$$\sigma_0^0 = |0\rangle\,,\ \sigma_1^0 = |1\rangle\,,\ \sigma_0^1 = |+\rangle\,,\ \sigma_1^1.$$

Then a bit string can be represented as

$$\rho_i = \bigotimes_{i=1}^{n} \rho_{a_i}^{(s_i^{(a)})}$$

Alice sends ρ_i over an insecure Quantum channel. Eve may perform measurements, or simply due to noise the state may slightly change. Therefore, Bob receives the (potentially different) state ρ_i'. Ideally Bob would store this ρ_i'. However, it is difficult to correctly store quantum states. Therefore, in practice Bob immediately measures ρ_i' and throw away the remaining quantum state. Bob picks a random basis, independently of Alice, as $s^{(b)} \leftarrow \{0,1\}^n$ (completely independent of Alice). Then for

$$M_0^{(0)} = |0\rangle\langle 0|\,,\ M_1^{(0)} = |1\rangle\langle 1|\,,\ M_0^{(1)} = |+\rangle\langle +|\,,\ M_1^{(1)} = |-\rangle\langle -|$$

Bob measures every ρ_i in the basis $M^{(s_i^b)}$, stores it in a string $b' \in \{0,1\}^n$ and acknowledges the reception.

Of course, there is a probability 0.5 that Bob measures in the wrong basis. Therefore, Alice and Bob publically announce $s^{(a)}$ respectively $s^{(b)}$, and only keep the bits where $s^{(a)} = s^(b)$. Alice and Bob then store their pruned string as $a, b \in \{0,1\}^m$, where

$$m = \sum_{i=1}^{n} \mathbf{1}_{s_i^{(a)} = s_i^{(b)}}.$$

5.1.2 Parameter estimation

As far as Alice and Bob are concerned, this ends the quantum part of the protocol. They have two correlated classical strings and have to reduce them to a secret key.

First they need to find the strength of their correlation. Alice and Bob publically pick a sample of their bit strings a and b to see how similar they are. Formally they publically pick a random permutation $\sigma \leftarrow S_m$ such that

$$a^{(p)} = (a_{\sigma(1)} a_{\sigma(2)} \ldots, a_{\sigma(\lfloor \frac{m}{2} \rfloor)}),\ a^{(k)} = (a_{\sigma(\lfloor \frac{m}{2} \rfloor + 1)} \ldots, a_{\sigma(m)})$$
$$b^{(p)} = (b_{\sigma(1)} b_{\sigma(2)} \ldots, b_{\sigma(\lfloor \frac{m}{2} \rfloor)}),\ b^{(k)} = (b_{\sigma(\lfloor \frac{m}{2} \rfloor + 1)} \ldots, b_{\sigma(m)})$$

Then they publically announce $a^{(p)}$ and $b^{(p)}$, and calculate the difference

$$\hat{e} = \sum_{i=1}^{\lfloor \frac{m}{2} \rfloor} a_i^{(p)} \oplus b_i^{(p)}.$$

Then if $\hat{e} > e$ for some security parameter e the correlation between a and b is found to be too weak, and the protocol has failed. Otherwise, with a high probability, $a^(k))$ and $b^{(k)}$ are also similar.

5.1.3 Error correction

Although $a^{(k)}$ and $b^{(k)}$ are correlated they are not exactly the same. Therefore, Alice and Bob agree on a $[n, k, t]$ linear code C (see section 2.2). They find the nearest code word of $a^{(k)}$ and $b^{(k)}$. Since there were not many errors with a very high probability

$$P[d(a^{(k)}, b^{(k)}) > t] \leq \epsilon$$

for some security parameters e_1 and ϵ.

Therefore, we can assume that Alice and Bob now share an identical bit string

$$k = \mathsf{Corr}(a^{(k)}) = \mathsf{Corr}(b^{(k)}) \in \{0, 1\}^{\cdot}$$

5.1.4 Privacy amplification

Although the k key is now correct, shared between Alice and Bob with very high probability, Eve might still have some information about k. For example, she happened to measure one of the qubits in the same basis as Alice and Bob. While the probability that this happens for all the qubits is negligibly small, it still may give a significant amount of information away. On the other hand, Alice and Bob should now share the exact same bit string.

Therefore, Alice and Bob have to perform privacy amplification. In practice this is performed by hashing the resulting key with a hash function. This is not a single hash function, but a family of universal-2 hash functions H with every $h \in H$ a function $h : A \to B$ that follow the following property

$$\Pr_{h \in H}(h(x) = h(y)) = \frac{1}{|B|}$$

Carter and Wegman have shown that such family of functions exist when $|A|$ and $|B|$ are powers of two [4].

Then Alice and Bob publically agree on a hash function $h \in H$, based on the amount of errors and take as final key

$$h(k).$$

5.2 Security definition

While we have described the protocol, the security guarantees it provides are not immediately clear. Quantum Key Distribution promises an information theoretical secure

key. In the classical case a key K_p generated by a protocol is ϵ-secure if there exists a uniformly distributed random variable K_u such that

$$P[K_p = K_u] \geq 1 - \epsilon.$$

Equivalently, the statistical distance between the distribution of K_p and a uniform distribution is less than ϵ.

In the quantum case, an eavesdropper gets quantum information about the key ρ_k. Ideally this should not give any information so the information about ρ_k should be similar to that of a uniform key ρ_u. Therefore, we need some distance metric between ρ_k and ρ_u.

Then QKD guarantees that for this norm

$$\|\rho_k - \rho_u\| < \epsilon$$

for some arbitrary small ϵ dependent on the security parameters. This proof is out-of-scope but can be found in [23].

For this to work we need to extend the measure of statistical distance to quantum states. The quantum case is a bit more subtle because of two reasons.

1. To get any information from the quantum state, an observer needs to perform a measurement. We do not know in advance which measurement the observer is going to perform.
2. The measurement changes the quantum state. The post-measurement state should not increase the distance between states. As a small distance means it is difficult to distinguish two states, it should not be possible to increase the distance between states (on average).

5.2.1 Trace distance

The trace distance functions as a distance metric for quantum states with desirable properties. For this we have to extend some familiar functions on $\mathbb{R}/\mathbb{C}$ to matrices.

Definition 5.1. Let $A \in \mathbb{C}^{n\times n}$ be Hermitian, for $U^*DU =: A$ the spectral decomposition. Define the positive square root as

$$\sqrt{A} = U^*\sqrt{D}U$$

where $\sqrt{D}$ is a diagonal matrix with $\sqrt{D}_{ii} = \sqrt{D_{ii}}$ the positive square root.

It is obvious from the definition that $\sqrt{A}^2 = A$.

Definition 5.2. For $A \in \mathbb{C}^{n\times n}$ we define

$$|A| = \sqrt{A^*A}$$

Remark 5.3. Let $A \in \mathbb{C}^{n\times n}$ be Hermitian. Write the spectral decomposition $U^*DU =: A$. Then

$$|A| = \sqrt{A^*A} = \sqrt{A^2} = \sqrt{U^*D^2U} = U^*\sqrt{D^2}U = U^*|D|U$$

where $|D|$ is a positive diagonal matrix with $|D|_{ii} = |D_{ii}|$.

Definition 5.4. For $A, B \in \mathbb{C}^{n\times n}$ we write $A \leq B$ if and only if

$$B - A$$

is positive semi-definite.

Lemma 5.5. *Let $A, B \in \mathbb{C}^{n\times n}$ Hermitian. If $A \leq B$ then $\operatorname{tr} A \leq \operatorname{tr} B$.*

Proof. We see that

$$B - A$$

is positive semi-definite, which means

$$0 \leq \operatorname{tr}(B - A) \Rightarrow \operatorname{tr}(A) \leq \operatorname{tr}(B).$$

□

Definition 5.6. [14] For density matrices ρ and σ, the trace metric is defined as

$$\|\rho\|_{\mathrm{tr}} = \frac{1}{2}\operatorname{tr}|\rho|,$$

with the implied trace distance

$$\mathrm{d}_{\mathrm{tr}}(\rho, \sigma) = \|\rho - \sigma\|_{\mathrm{tr}}.$$

Remark 5.7. If ρ is a density matrix then we can compute the spectral decomposition ρ as U^*DU with D diagonal.

$$\|\rho\|_{\mathrm{tr}} = \|D\|_{\mathrm{tr}} = \sum_i |\lambda_i|$$

with λ_i the eigenvalues of ρ or the diagonal entries of D.

Lemma 5.8. *[22] The trace distance $\mathrm{d}_{tr}(\rho, \sigma) = \|\rho - \sigma\|_{tr}$ is a distance function.*

Proof. We show that the implied distance function $\mathrm{d}_{\mathrm{tr}}(\rho, \sigma) = \|\rho - \sigma\|_{\mathrm{tr}}$ for density matrices $\rho, \sigma \in \mathbb{C}^{n\times n}$ follows all axioms.

1. We see trivially that $\mathrm{d}_{\mathrm{tr}}(\rho, \rho) = 0$, and $\mathrm{d}_{\mathrm{tr}}(\rho, \sigma) \geq 0$ from 5.7.

2. We see that for λ_i the eigenvalues of $\rho - \sigma$ that

$$\begin{aligned} \mathrm{d_{tr}}(\rho, \sigma) &= \|\rho - \sigma\|_{\mathrm{tr}} \\ &= \sum_{i=1}^{n} |\lambda_i| \\ &= \sum_{i=1}^{n} |-\lambda_i| \\ &= \|\sigma - \rho\|_{\mathrm{tr}} \\ &= \mathrm{d_{tr}}(\sigma, \rho). \end{aligned}$$

3. Let $\rho_1, \rho_2, \rho_3 \in \mathbb{C}^{n \times n}$ density matrices, then

$$\mathrm{d_{tr}}(\rho_1, \rho_3) \leq \mathrm{d_{tr}}(\rho_1, \rho_2) + \mathrm{d_{tr}}(\rho_2, \rho_3).$$

We will proof this as the corollary of the next theorem.

□

The trace distance has a very nice property, namely that it corresponds to the observers advantage. If it is known that a quantum system is either in state ρ or in state σ, then the best measurement can only tell ρ and σ with probability $\mathrm{tr}\{\rho - \sigma\}$. Recall that for a POVM $\{M_i\}_i$ on ρ we have the measurement result i with probability $\mathrm{tr}(M_i \rho M_i^*)$. As every M_i is positive semi-definite, we also have $0 \leq M_i \leq I$.

Theorem 5.9. *[14] Let ρ, σ be density matrices. Then we have*

$$\|\rho - \sigma\|_{tr} = \max_{0 \leq M \leq I} \mathrm{tr}((\rho - \sigma)M).$$

Proof. Since ρ and σ are density matrices, the difference is Hermitian, so we can compute the spectral decomposition $\rho - \sigma = UDU^*$. Since all eigenvalues are real, we can split the diagonal matrix D into the two positive semi-definite matrices $D = D_P - D_Q$, where D_P contains the positive entries and D_Q the negative entries in absolute value. Then let $P = UD_PU^*$ and $Q = UD_QU^*$. Since $\rho - \sigma = P - Q$ we have

$$\|\rho - \sigma\|_{\mathrm{tr}} = \|P - Q\|_{\mathrm{tr}} = \frac{1}{2}(\mathrm{tr}\, P + \mathrm{tr}\, Q).$$

And since $\mathrm{tr}(\rho) = \mathrm{tr}(\sigma) = 1$ (as a property of density matrices)

$$\mathrm{tr}(P) - \mathrm{tr}(Q) = \mathrm{tr}(P - Q) = \mathrm{tr}(\rho - \sigma) = \mathrm{tr}(\rho) - \mathrm{tr}(\sigma) = 1 - 1 = 0$$

Therefore $\|\rho - \sigma\|_{\mathrm{tr}} = \mathrm{tr}\, P$. We then get for arbitrary $M \leq I$

$$\mathrm{tr}((\rho - \sigma)M) = \mathrm{tr}((P - Q)M) \leq \mathrm{tr}(PM) \leq \mathrm{tr}(P) = \|\rho - \sigma\|_{\mathrm{tr}}.$$

Now let D' a diagonal matrix with $D_{ii} = 0$ if $(D_P)_{ii} = 0$, and $D_{ii} = 1$ if $(D_P)_{ii} > 1$. Then let $M = UD'U^*$, which projects a vector onto the span of P. Note that $PM = P$ and $QM = 0$. Then we have

$$\mathrm{tr}((\rho - \sigma)M) = \mathrm{tr}((P - Q)M) = \mathrm{tr}(P) = \|\rho - \sigma\|_{\mathrm{tr}}.$$

□

Corollary 5.10. *[22] The triangle inequality of the trace distance follows from 5.9.*

Proof. Let $\rho_1, \rho_2, \rho_3 \in \mathbb{C}^{n\times n}$ density matrices. Then

$$\begin{aligned}
\mathrm{d}_{\mathrm{tr}}(\rho_1, \rho_3) &= \max_{0\le M\le I} \mathrm{tr}((\rho_1 - \rho_3)M) \\
&= \max_{0\le M\le I} (\mathrm{tr}((\rho_1 - \rho_2)M) + \mathrm{tr}((\rho_2 - \rho_3)M)) \\
&\le \max_{0\le M\le I} \mathrm{tr}((\rho_1 - \rho_2)M) + \max_{0\le M\le I} \mathrm{tr}((\rho_2 - \rho_3)M) \\
&= \mathrm{d}_{\mathrm{tr}}(\rho_1 - \rho_2) + \mathrm{d}_{\mathrm{tr}}(\rho_2 - \rho_3).
\end{aligned}$$

□

Note that we do not know the measurement result in advance. But if $\|\rho - \sigma\|_{\mathrm{tr}}$ is low, whatever measurement happened, the probability that it happened is more or less similar for ρ and σ. It seems that this property suffices, but it is important to remember that quantum measurements are a destructive operation. And we want the post-measurement state of ρ and σ to be similar as well. Note that this is not necessarily the case. As a simple example take the pure qubits with a small $\epsilon > 0$

$$\rho = |0\rangle\langle 0|, \quad \sigma = (\sqrt{(1-\epsilon^2)}\,|0\rangle + \epsilon\,|1\rangle)(\sqrt{(1-\epsilon^2)}\,\langle 0| + \epsilon\,\langle 1|)$$

Then the trace distance is as expected small

$$\mathrm{tr}(\rho - \sigma) = \epsilon + 1 - \sqrt{1-\epsilon^2}$$

We also have that the trace distance functions as a contraction, the trace distance does not increase by a trace preserving operation. The trace distance, on average, can only become smaller after a measurement.

Theorem 5.11. *[14] Let $\{M_i\}_i$ be a POVM, and let ρ, σ quantum states. Then the average post-measurement state is*

$$\mathrm{d}_{tr}\left(\sum_i M_i \rho M_i^*, \sum_i M_i \sigma M_i^*\right) \le \mathrm{d}_{tr}(\rho, \sigma)$$

Proof. Let $\rho - \sigma = P - Q$, and let N be the projection that achieves the maximum as in the proof of Theorem 5.9. We have

$$\begin{aligned}
\mathrm{d_{tr}}\left(\sum_i M_i \rho M_i^*, \sum_i M_i \sigma M_i^*\right) &= \mathrm{tr}\left(N\left(\sum_i M_i \rho M_i^* - \sum_i M_i \sigma M_i^*\right)\right) \\
&\leq \mathrm{tr}\left(N \sum_i M_i (P - Q) M_i^*\right) \\
&\leq \mathrm{tr}\left(N \sum_i M_i P M_i^*\right) \\
&\leq \mathrm{tr}\left(\sum_i M_i P M_i^*\right) \\
&= \mathrm{tr}(P) \\
&= \mathrm{d_{tr}}(\rho, \sigma).
\end{aligned}$$

□

Now we have shown that the trace distance conforms to the two required properties: the trace distance represents the observer advantage, and it is a contraction after measurements. Therefore, if a key provided by QKD is similar to a uniform key in the trace distance. An adversary will not be able to get more information from the received key, than if the adversary would uniformly sample key themselves.

6 Conclusion

In this thesis we have given an introduction to post-quantum cryptography, to classical protocols that provide information-theoretical security, and to quantum key distribution. Post-quantum cryptography provides computational security on one of several hardness assumptions. The protocols that will probably be standardized rely either on lattice-based cryptography or on code-based cryptography.

For information-theoretical security, we have seen that classical protocols are limited in what they can do. Classical protocols require some form of pre-shared correlated data, which is often limited. This pre-shared data limits the entropy of any resulting key. If the entropy of a key is limited, then we also see that it has a significant statistical distance to a uniformly distributed bit string.

On the other hand, quantum key distribution does not have this limit on the entropy of the key, given a channel for quantum communication. Quantum key distribution is able to exchange this key, because Bob and Eve make different measurements of the quantum states, with very high probability. Therefore, as there is no classical transcript, Eve is more limited in what she can do.

For this thesis, we have focussed on key exchange protocols against a passive eavesdropper. We have not considered an active adversary that can modify messages that are sent between Alice and Bob. Additional security definitions and considerations would be required to protect against this kind of adversary. To compare QKD and PQC in this scenario, further work would be required.

Ethical considerations. Cryptography allows people to communicate in secret. Unfortunately, criminals could also use cryptography to plan for, or hide evidence of a crime. Some governments have proposed to introduce backdoors in encryption algorithms, so law enforcement can decrypt communication if deemed necessary. On the other hand, privacy is a fundamental right in a democratic society. Even if the government would only use the backdoor for legitimate purposes, a backdoor would structurally weaken cryptography, and limit privacy for ordinary citizens. The consensus of a report written for the US government is that the advantages of encryption outweigh the danger of possible illegitimate use [6]. Therefore, research on cryptography, while potentially dangerous, can happen in good conscience.

Bibliography

[1] Abhishek Banerjee, Chris Peikert, and Alon Rosen. "Pseudorandom Functions and Lattices". In: *Advances in Cryptology – EUROCRYPT 2012*. Springer Berlin Heidelberg, 2012, pp. 719–737. DOI: 10.1007/978-3-642-29011-4_42.

[2] E. Berlekamp, R. McEliece, and H. van Tilborg. "On the inherent intractability of certain coding problems (Corresp.)" In: *IEEE Transactions on Information Theory* 24.3 (May 1978), pp. 384–386. DOI: 10.1109/tit.1978.1055873.

[3] Johannes Blömer and Jean-Pierre Seifert. "On the complexity of computing short linearly independent vectors and short bases in a lattice". In: *Proceedings of the thirty-first annual ACM symposium on Theory of computing - STOC '99*. ACM Press, 1999, pp. 711–720. DOI: 10.1145/301250.301441.

[4] J.Lawrence Carter and Mark N. Wegman. "Universal classes of hash functions". In: *Journal of Computer and System Sciences* 18.2 (Apr. 1979), pp. 143–154. DOI: 10.1016/0022-0000(79)90044-8.

[5] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991. DOI: 10.1002/0471200611.

[6] Kenneth W. Dam and Herbert S. Lin. *Cryptography's Role in Securing the Information Society*. National Academies Press, Oct. 1996. ISBN: 9780309054751. DOI: 10.17226/5131.

[7] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC Cryptography and Network Security Series. Chapman and Hall/CRC, Dec. 2020, p. 628. ISBN: 9781351133012. DOI: 10.1201/9781351133036. URL: https://books.google.nl/books?id=RsoOEAAAQBAJ.

[8] A. K. Lenstra, H. W. Lenstra, and L. Lovász. "Factoring polynomials with rational coefficients". In: *Mathematische Annalen* 261.4 (Dec. 1982), pp. 515–534. DOI: 10.1007/bf01457454.

[9] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *Advances in Cryptology – EUROCRYPT 2010*. Springer Berlin Heidelberg, 2010, pp. 1–23. DOI: 10.1007/978-3-642-13190-5_1.

[10] U.M. Maurer. "Secret key agreement by public discussion from common information". In: *IEEE Transactions on Information Theory* 39.3 (May 1993), pp. 733–742. DOI: 10.1109/18.256484.

[11] Vasileios Mavroeidis et al. “The Impact of Quantum Computing on Present Cryptography”. In: *International Journal of Advanced Computer Science and Applications* 9.3 (2018). DOI: 10.14569/ijacsa.2018.090354.

[12] Robert J McEliece. “A Public-Key Cryptosystem Based On Algebraic Coding Theory”. In: *Deep Space Network Progress Report* 4244 (Jan. 1978), pp. 114–116. URL: https://www.semanticscholar.org/paper/A-public-key-cryptosystem-based-on-algebraic-coding-McEliece/14a22cae27878549c1dbcf74bec7d6f39dfcbe2a.

[13] W. McGill. “Multivariate information transmission”. In: *Transactions of the IRE Professional Group on Information Theory* 4.4 (Sept. 1954), pp. 93–111. DOI: 10.1109/tit.1954.1057469.

[14] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. June 2012. DOI: 10.1017/cbo9780511976667.

[15] NIST. *Post-Quantum Cryptography Round 3 submissions*. Accessed: 2022-06-09. July 2020. URL: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions.

[16] Oded Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *Journal of the ACM* 56.6 (Sept. 2009), pp. 1–40. DOI: 10.1145/1568318.1568324.

[17] D. Sarwate. “On the complexity of decoding Goppa codes (Corresp.)” In: *IEEE Transactions on Information Theory* 23.4 (July 1977), pp. 515–516. DOI: 10.1109/tit.1977.1055732.

[18] C.P. Schnorr. “A hierarchy of polynomial time lattice basis reduction algorithms”. In: *Theoretical Computer Science* 53.2-3 (1987), pp. 201–224. DOI: 10.1016/0304-3975(87)90064-8.

[19] C. E. Shannon. “A Mathematical Theory of Communication”. In: *Bell System Technical Journal* 27.3 (July 1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[20] C. E. Shannon. “Communication Theory of Secrecy Systems”. In: *Bell System Technical Journal* 28.4 (Oct. 1949), pp. 656–715. DOI: 10.1002/j.1538-7305.1949.tb00928.x.

[21] P.W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. Ieee. IEEE Comput. Soc. Press, 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[22] Mark Tame. “AQI: Advanced Quantum InformationLecture 8 (Module 2): Distance Measures”. In: (2013).

[23] Marco Tomamichel and Anthony Leverrier. “A largely self-contained and complete security proof for quantum key distribution”. In: *Quantum* 1 (July 2017), p. 14. DOI: 10.22331/q-2017-07-14-14.

[24] R.W. Yeung. “A new outlook on Shannon’s information measures”. In: *IEEE Transactions on Information Theory* 37.3 (May 1991), pp. 466–474. DOI: 10.1109/18.79902.

Popular summary

Suppose Alice and Bob want to communicate, but a third party, called Eve, is listening to their conversation. Alice and Bob do not want Eve to know what their conservation is about, so Alice *encrypts* her messages before she sends them to Bob. Encryption means that Alice and Bob scramble their messages such that Alice and Bob can recover the messages, but Eve cannot. For encryption, Alice and Bob require a key, a bit string shared between Alice and Bob that Eve does not know about.

Ideally Alice and Bob can share their key without Eve listening. However, this is not always possible. Luckily, there are ways that Alice and Bob can establish a secret key, even with Eve listening. These are called *key exchange* algorithms. In a key exchange algorithm, Alice and Bob send messages $e_1, \ldots, e_n$ to each other in order to establish a shared key k. On the other hand, Eve who is listening to their communication $e_1, \ldots, e_n$, should not be able to get information about k.

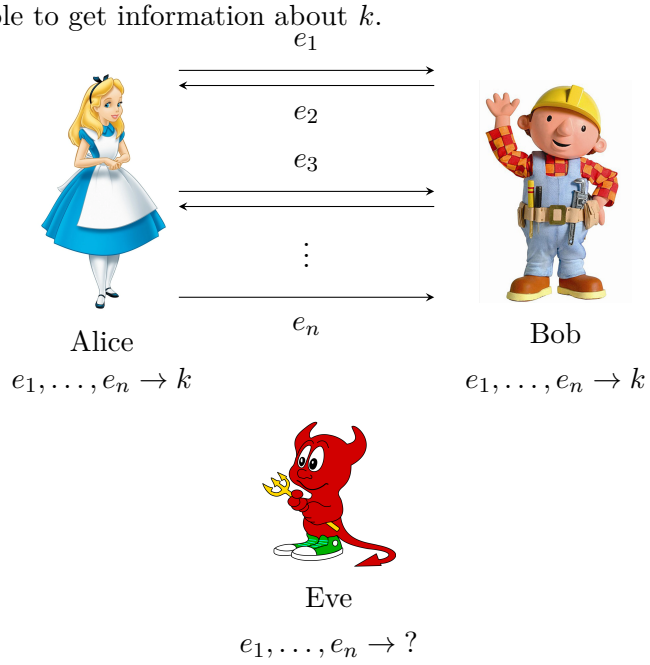


Figure 6.1: Alice and Bob can derive the key k, but Eve cannot know k.

There are two categories of key exchange protocols, that provide different kinds of security. The algorithms of the first category, most commonly used today, provide *computational security*. Computational security means that it is expected to be very hard for Eve

to derive the key k, but not necessarily impossible. The downside is, that new smarter algorithms, for example using a quantum computer, may be developed in the future, so that Eve then can derive k more easily than expected. Algorithms of the second category provide *information-theoretical security*, which means that Eve, even with unlimited resources, cannot have any information about k. In this thesis we have compared these two types of algorithms, which provide computational security and information-theoretical security.