Cryptography and Quantum Computing[1]

Ivan Murashko

[1] St. Petersburg State Polytechnic University

Contents

Introduction

The purpose of this course is to introduce modern cryptography and how quantum mechanics can be used to solve complex cryptographic problems.

The course consists of 10 lectures, each lasting 1 hour.

The following topics are covered:

- Introduction to quantum mechanics (Lectures 1, 2)
- Description of basic principles of quantum computing (Lecture 3)
- Symmetric encryption algorithms and Grover's algorithm. (Lecture 4)
- The classical RSA algorithm and its connection to the problem of finding the period of a function. (Lecture 5)
- Discrete (classical) Fourier transform and its applications for finding the period of periodic functions. Implementation of the discrete Fourier transform on quantum elements is proposed (Lectures 6, 7)
- Shor's algorithm for breaking RSA (Lecture 8)
- Classical encryption algorithms based on the complexity of discrete logarithms. Modification of Shor's algorithm to solve the discrete logarithm problem (Lecture 9)
- The final lecture, Lecture 10, is devoted to encryption algorithms based on elliptic curves. The ECDH algorithm is considered. Modification of Shor's algorithm for solving the discrete logarithm problem on elliptic curves is described.

Throughout the lectures, necessary mathematical explanations will be provided, such as

- Discrete mathematics: Fermat's little theorem, Euclidean algorithm, etc.
- General algebra: concept of a group, Lagrange's theorem, cyclic group, concept of a field. Galois fields.

- Linear algebra and matrix operations: matrix multiplication, linear operators, eigenvalues and eigenfunctions of linear operators

- Classical probability theory: events, random variables, mean of a random variable

Chapter 1

Classical Cryptography

From the moment the importance of information was recognized, means of its protection began to appear.

New methods of encryption were invented, such as the Caesar cipher, in which each letter of the alphabet was replaced by another (for example, the one three positions later in the alphabet). Alongside new encryption methods, ways to break these ciphers appeared. For example, for the Caesar cipher, one can use the statistical properties of the language in which the original message was written.

Very often, the security of a cipher was ensured by keeping the algorithm that provided encryption secret, as in the Caesar cipher discussed above. In modern classical cryptography, algorithms are often published and accessible for study by anyone. Secrecy is ensured by mixing the message itself with a secret key according to a certain open algorithm.

Suppose we need to transmit a message from Alice to Bob over a secure communication channel. The message must be in a digital form. The protocol describing this transmission consists of several stages. In the first stage, Alice and Bob must obtain a common random sequence of numbers, which will be called a key. This procedure is called key distribution.

In the next stage, Alice must use a certain algorithm E to obtain an encrypted message C from the original message P and the key K. This procedure can be described by the following equation:

$$E_K\left(P\right) = C. \tag{1.1}$$

In the third stage, the encrypted message must be transmitted to Bob.

In the final stage, Bob, using the known algorithm D and the key K obtained in the first stage, must recover the original message P from the received encrypted message C. This procedure can be described by the following equation:

$$D_K\left(C\right) = P. \tag{1.2}$$

An analysis of this protocol raises the following questions. How to implement secure key distribution. Second, does an absolutely secure algorithm exist? And finally, is it possible to securely transmit an encrypted message in such a way that it cannot be intercepted or modified?

Classical cryptography provides a definitive answer only to the second question. An absolutely secure algorithm exists — it is called a one-time pad. Below is a detailed description of this algorithm.

1.1 One-Time Pad

The one-time pad scheme was proposed in 1917 by Major J. Maborn and G. Vernam. The classic one-time pad is a set of random keys, each of which is equal in size to the message being sent and is used only once.

Suppose we want to encrypt a message in a certain language (such as English). The number of characters (letters) used in the alphabet is denoted by X. For the English language (without punctuation and case distinction) $X = 26$. Then we assign a certain number c to each character of the language, such that $0 \leq c \leq X$. For example, for the English language, we can write

$$\begin{gathered} A \to 0 \\ B \to 1 \\ \dots \\ Z \to 25 \end{gathered}$$

The encryption procedure (1.1) is described by the following expression

$$E_{K_i}(P_i) = P_i + K_i \mod X = C_i, \tag{1.3}$$

where i is the number of the character being encrypted.

The decryption procedure (1.2) is described by the following expression

$$D_{K_i}(C_i) = C_i - K_i \mod X = P_i, \tag{1.4}$$

where i is the number of the character being encrypted.

This procedure easily generalizes to the case of binary data, using the XOR operation $(a \oplus b)$ instead of modulo addition for both encryption and decryption:

Claude Shannon showed [9] that if the key is truly random, has the same length as the original message, and is not reused, then the proposed one-time pad scheme is perfectly secure.

According to Shannon, perfect security can be defined as follows.

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Table 1.1: XOR $a \oplus b$

Definition 1.1.1. A cipher (E, D) is perfectly secure if for any two messages of equal length m_0 and m_1, some ciphertext c, and key $k \leftarrow_R K$, the probabilities that the original text is m_0 or m_1 are equal:

$$P\left(E\left(m_0, k\right) = c\right) = P\left(E\left(m_1, k\right) = c\right)$$

Rephrasing this definition, it can be said that based on the original statistics of the ciphertext, no information about the original message can be obtained.

Theorem 1.1.1 (Cryptographic Strength of the One-Time Pad)**.** *The one-time pad scheme is perfectly secure.*

Proof. Let $|K|$ denote the number of all possible keys of length l. Where l is also the length of the original messages: $|m_{0,1}| = l$. Given that the key used to encrypt the message is uniquely defined:

$$k_{0,1} = c \oplus m_{0,1},$$

we obtain for the probabilities

$$P\left(E\left(m_0, k\right) = c\right) = P\left(E\left(m_1, k\right) = c\right) = \frac{1}{|K|}.$$

□

1.2 Problems of Classical Cryptography

If there is a perfectly secure cryptographic system (the one-time pad), then what is wrong with classical cryptography? The problem lies in obtaining keys that meet the requirements of the one-time pad (the key length equals the message length, the key consists of random data and is not reused) and transmitting these keys to Bob and Alice.

Problems arise both at the stage of key generation, [1] and at the stage of transmitting these keys.

[1] obtaining large sequences of random numbers is a non-trivial mathematical problem

In classical cryptography, the so-called public key algorithms are used for key transmission. There are several key exchange protocols based on public key cryptographic systems. They are all based on the existence of two keys, the first of which, called the public key, is used only for encryption, and the second - the private key - for decryption. To obtain the private key from the public key, a complex mathematical operation must be performed. For example, the security of one of the most popular public key systems - RSA (see 5.1), is based on the difficulty of factoring[2] large numbers.

The key distribution protocol scheme based on public key cryptography can be described as follows. In the first step, Alice creates public and private keys and sends the first one to Bob. Bob, in turn, creates the key that both Alice and Bob would like to have (which needs to be distributed). This key is encrypted (for example, using RSA) with Alice's public key and sent to her. Upon receiving this encrypted key, Alice can decrypt it using her private key.

If an eavesdropper (Eve) wants to learn the transmitted key, she must solve a complex mathematical problem of factoring large numbers. It is believed, but not proven, that the difficulty of factoring grows exponentially with the number of digits in the number [14].[3] Thus, as the number of digits increases, the problem quickly becomes unsolvable.

There are several problems with this scheme. The first is that the complexity of factorization is not proven. Moreover, there are algorithms for quantum computers - Shor's algorithm (see 5.2) - that solve the factoring problem for a number N in time $O\left(logN\right)$, i.e., in time proportional to the number of digits in N. Therefore, at the moment when a quantum computer is built, all systems based on RSA, will become obsolete.

[2] decomposing into prime factors

[3] The fastest known algorithm solves the factoring problem of a number N in time on the order of $O\left(exp\left(log^{\frac{1}{3}}N\left(log\,logN\right)^{\frac{2}{3}}\right)\right)$.

Chapter 2

Fundamental Principles of Quantum Mechanics

2.1 Dirac Formulation of Quantum Mechanics

In the lecture course on quantum optics, we will consistently use the Dirac formalism [17]. In the usual formulation of quantum mechanics, we deal with wave functions, for example, $\psi\left(q,t\right)$ – the wave function in the coordinate representation. The same state of the system can be described by wave functions in different representations, related to each other by linear transformations. For example, the wave function in the momentum representation is related to the wave function in the coordinate representation by the equation:

$$\phi\left(p,t\right)=\frac{1}{2\pi\hbar}\int_{-\infty}^{+\infty}\psi\left(q,t\right)e^{-i\frac{pq}{\hbar}}dq \tag{2.1}$$

The main point here is that the same state can be described by wave functions expressed through different variables. From this, it follows that one can introduce a more general concept that characterizes the state of the system regardless of the representation. For such a concept, Dirac introduced the notion of a wave vector, or state vector, denoted by:

$$\left|\ldots\right\rangle \tag{2.2}$$

and called the ket vector.

2.1.1 Ket Vector

$\left|\ldots\right\rangle$ is the general designation of a ket vector; $\left|a\right\rangle$, $\left|x\right\rangle$, $\left|\psi\right\rangle$, etc., denote ket vectors describing certain particular states, the symbols of which are written inside the brackets.

2.1.2 Bra Vectors

Each ket vector corresponds to a conjugate bra vector. The bra vector is denoted by:

$$\langle \ldots | , \quad \langle a | , \quad \langle \psi | . \tag{2.3}$$

The names bra and ket vectors are derived from the first and second halves of the English word *bra-cket*.

Thus, the bra vectors $\langle a|$, $\langle x|$, $\langle \psi|$ correspond to their conjugate ket vectors $|a\rangle$, $|x\rangle$, $|\psi\rangle$ and vice versa. For state vectors, the same basic relationships hold that are true for wave functions:

$$|u\rangle = |a\rangle + |b\rangle , \quad \langle u| = \langle a| + \langle b| , \quad |v\rangle = l\,|a\rangle , \quad \langle v| = l\,\langle a| . \tag{2.4}$$

Bra and ket vectors are related to each other by the operation of Hermitian conjugation:

$$|u\rangle = (\langle u|)^\dagger , \quad \langle u| = (|u\rangle)^\dagger . \tag{2.5}$$

In well-known cases this reduces to the following relationships:

$$(\psi\,(q))^\dagger = \psi^*\,(q)$$

for the wave function in the coordinate representation;

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}^\dagger = (a_1^*, a_2^*, \cdots, a_n^*)$$

in the matrix representation.

With the help of bra and ket vectors, one can define the scalar product

$$\langle v|u\rangle = \langle u|v\rangle^* . \tag{2.6}$$

In specific cases, this means:

$$\langle \psi|\,\phi\rangle = \int \psi^* \phi dq$$

in the coordinate representation;

$$\langle a|b\rangle = (a_1^*, a_2^*, \cdots, a_n^*) \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = a_1^* b_1 + a_2^* b_2 + \cdots + a_n^* b_n$$

in the matrix representation.

From the relationship (2.6) it follows that the norm of the vector is real. Additionally, we assume that the norm of the vector is non-negative: $\langle a|a\rangle \geq 0$.

2.1.3 Operators

In quantum mechanics, linear operators are used. Operators connect one state vector with another:

$$|q\rangle = \hat{L}\,|p\rangle \tag{2.7}$$

The conjugate equality is of the form

$$\langle q| = \langle p|\,\hat{L}^{\dagger} \tag{2.8}$$

where $\hat{L}^{\dagger}$ is the operator conjugated to the operator $\hat{L}$.

We give some relationships that are valid for linear operators:

$$\begin{gathered}
\hat{L}^{++} = \hat{L}, \quad \left(l\hat{L}\,|a\rangle\right)^{\dagger} = l^{*}\,\langle a|\,\hat{L}^{\dagger}, \\
\left(\left(\hat{L}_1 + \hat{L}_2\right)|a\rangle\right)^{\dagger} = \langle a|\left(\hat{L_1}^{\dagger} + \hat{L_2}^{\dagger}\right), \\
\left(\left(\hat{L}_1\hat{L}_2\right)|a\rangle\right)^{\dagger} = \langle a|\left(\hat{L_2}^{\dagger}\hat{L_1}^{\dagger}\right), \\
\left(\left(\hat{L}_1\hat{L}_2\hat{L}_3\right)|a\rangle\right)^{\dagger} = \langle a|\left(\hat{L_3}^{\dagger}\hat{L_2}^{\dagger}\hat{L_1}^{\dagger}\right), \text{ etc.}
\end{gathered} \tag{2.9}$$

Note that the algebra of operators coincides with the algebra of square matrices. The matrix elements of operators are denoted as follows:

$$\langle a|\,\hat{L}\,|b\rangle = L_{ab} \tag{2.10}$$

For matrix elements, the equalities are valid

$$\langle a|\,\hat{L}\,|b\rangle^{*} = \langle b|\,\hat{L}^{\dagger}\,|a\rangle\,, \quad \langle a|\,\hat{L}_1\hat{L}_2\,|b\rangle^{*} = \langle b|\,\hat{L_2}^{\dagger}\hat{L_1}^{\dagger}\,|a\rangle \tag{2.11}$$

2.1.4 Eigenvalues and Eigenvectors of Operators

The eigenvalues and eigenvectors of operators are determined by the equality

$$\hat{L}\,|l_n\rangle = l_n\,|l_n\rangle\,, \tag{2.12}$$

where l_n is the eigenvalue; $|l_n\rangle$ is the eigenvector.

For bra vectors we have similar equalities:

$$\langle d_n|\,\hat{D} = d_n\,\langle d_n|\,. \tag{2.13}$$

If the operators correspond to observable quantities, they must be self-adjoint:

$$\hat{L} = \hat{L}^{\dagger}. \tag{2.14}$$

The eigenvalues of a self-adjoint (Hermitian) operator are real. Indeed, from

$$\hat{L}\,|l\rangle = l\,|l\rangle$$

it follows that

$$\langle l|\,\hat{L}\,|l\rangle = l\,\langle l|l\rangle\,.$$

On the other hand, recalling (2.9): $\langle l|\,\hat{L}^{\dagger} = l^{*}\,\langle l|$, from (2.14) we have

$$\langle l|\,\hat{L}\,|l\rangle = l^{*}\,\langle l|l\rangle\,.$$

Thus $l\,\langle l|l\rangle = l^{*}\,\langle l|l\rangle$, i.e., $l = l^{*}$

The eigenvectors of a self-adjoint operator are orthogonal. Indeed, consider two eigenvectors $|l_1\rangle$ and $|l_2\rangle$:

$$\hat{L}\,|l_1\rangle = l_1\,|l_1\rangle\,, \quad \hat{L}\,|l_2\rangle = l_2\,|l_2\rangle$$

From the second relationship we get

$$\langle l_1|\,\hat{L}\,|l_2\rangle = l_2\,\langle l_1|l_2\rangle$$

Considering the reality of the eigenvalues and the relationship (2.14) for the vector $|l_1\rangle$, we get:

$$\langle l_1|\,\hat{L} = l_1\,\langle l_1|\,.$$

From where

$$\langle l_1|\,\hat{L}\,|l_2\rangle = l_1\,\langle l_1|l_2\rangle\,.$$

Thus

$$(l_1 - l_2)\,\langle l_1|l_2\rangle = 0, \quad \text{i.e.,}\ \langle l_1|l_2\rangle = 0,\ \text{since}\ l_1 \neq l_2.$$

2.1.5 Observables. Decomposition by Eigenvectors. Completeness of the System of Eigenvectors

Operators corresponding to observable physical quantities are self-adjoint operators. This ensures the reality of the values of the observable physical quantity. We have a set of eigenstates of a Hermitian operator $|l_n\rangle$, $\hat{L}\,|l_n\rangle = l_n\,|l_n\rangle$. If the set of eigenstates is complete, according to the principles of quantum mechanics, any state can be represented as a superposition of states $|l_n\rangle$:

$$|\psi\rangle = \sum_{(n)} c_n\,|l_n\rangle\,. \tag{2.15}$$

Thus for the decomposition coefficients we have: $c_n = \langle l_n | \, \psi \rangle$, and therefore the equality holds

$$|\psi\rangle = \sum_{(n)} \langle l_n | \, \psi\rangle \, |l_n\rangle = \sum_{(n)} |l_n\rangle \, \langle l_n | \, \psi\rangle \, . \tag{2.16}$$

From the equality 2.16 follows the important relation:

$$\sum_{(n)} |l_n\rangle \, \langle l_n| = \hat{I}. \tag{2.17}$$

where $\hat{I}$ is the unit operator. This equality is the condition of completeness of the system of eigenvectors (the condition for decomposability).

2.1.6 Projection Operator

Consider the operator $\hat{P}_n = |l_n\rangle \, \langle l_n|$. The result of the action of this operator on the state $|\psi\rangle$ will be

$$\hat{P}_n \, |\psi\rangle = \sum_{(k)} |l_n\rangle \, \langle l_n| \, c_k \, |l_k\rangle = c_n \, |l_n\rangle \, . \tag{2.18}$$

The operator $\hat{P}_n = |l_n\rangle \, \langle l_n|$ is called the projection operator.

One can write the following properties of this operator

$$\sum_{(n)} \hat{P}_n = \hat{I}. \tag{2.19}$$

$$\hat{P}_n^2 = \hat{P}_n. \tag{2.20}$$

The action of the projection operator has a simple geometric interpretation (see Figure 2.1):

$$\hat{P}_n \, |\psi\rangle = \cos\theta \, |l_n\rangle \, ,$$

where $\cos\theta = \langle\psi|l_n\rangle = c_n$.

2.1.7 Trace of the Operator

In an orthonormal basis $\{|l_n\rangle\}$ the quantity

$$Sp\hat{L} = \sum_{n} \langle l_n| \, \hat{L} \, |l_n\rangle \tag{2.21}$$

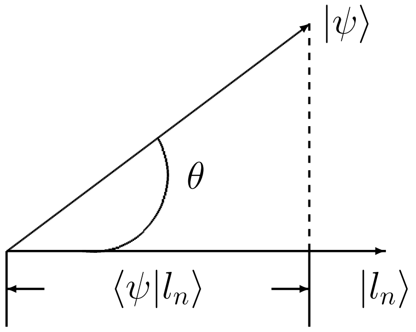

Figure 2.1: Projection operator. The action of the operator can be interpreted as the projection of vector $|\psi\rangle$ onto the axis $|l_n\rangle$

is called the trace of the operator $\hat{L}$. Under certain conditions [15] the series 2.21 converges absolutely and does not depend on the choice of basis.

If using the matrix representation

$$L_{kn} = \langle l_k| \hat{L} |l_n\rangle ,$$

then the trace of the operator is the sum of the diagonal elements of the matrix representation

$$Sp\hat{L} = \sum_n L_{nn}$$

The following properties of the trace of the operator can be written:

$$\begin{aligned} Sp\left(l\hat{L} + m\hat{M}\right) &= lSp\hat{L} + mSp\hat{M}, \\ Sp\left(\hat{L}\hat{M}\right) &= Sp\left(\hat{M}\hat{L}\right). \end{aligned} \tag{2.22}$$

2.1.8 Expectation Values of Operators

The expectation value of an operator $\hat{L}$ in the state $|\psi\rangle$ is given by the equation

$$\left\langle \hat{L} \right\rangle_\psi = \langle\psi| \hat{L} |\psi\rangle \tag{2.23}$$

under the condition

$$\langle\psi|\psi\rangle = 1.$$

Indeed, if we assume that $|\psi\rangle$ can be expanded into a series by the eigenfunctions of the operator $\hat{L}$ as follows:

$$|\psi\rangle = \sum_n c_n |l_n\rangle ,$$

then $\hat{L}\left|\psi\right\rangle$ can be written as

$$\hat{L}\left|\psi\right\rangle = \sum_n l_n c_n \left|l_n\right\rangle ,$$

where l_n is the eigenvalue corresponding to the eigenstate $\left|l_n\right\rangle$. If we now substitute the last two expressions into (2.23) we get:

$$\left\langle\psi\right|\hat{L}\left|\psi\right\rangle = \sum_{n,m} l_n c_n c_m^* \left\langle l_m|l_n\right\rangle = \sum_n l_n c_n c_n^* = \sum_n l_n \left|c_n\right|^2 ,$$

which (under the condition $\left\langle\psi\right|\psi\rangle = 1$) proves, that the expression (2.23) indeed represents the expression for the expectation value of the operator $\hat{L}$ in the state $\left|\psi\right\rangle$. [1]

If we take some orthonormal basis $\{\left|n\right\rangle\}$, forming a complete set, i.e., satisfying the condition (2.17): $\sum_n \left|n\right\rangle\left\langle n\right| = \hat{I}$, then expression (2.23) can be rewritten as follows:

$$\left\langle\hat{L}\right\rangle_\psi = \left\langle\psi\right|\hat{L}\left|\psi\right\rangle = \left\langle\psi\right|\hat{I}\hat{L}\left|\psi\right\rangle =$$

$$= \sum_n \left\langle\psi|n\right\rangle\left\langle n\right|\hat{L}\left|\psi\right\rangle = \sum_n \left\langle n\right|\hat{L}\left|\psi\right\rangle\left\langle\psi|n\right\rangle = Sp\left(\hat{L}\hat{\rho}\right),$$

where $\hat{\rho} = \left|\psi\right\rangle\left\langle\psi\right| = \hat{P}_\psi$ is the projection operator on the state $\left|\psi\right\rangle$. Considering (2.22) one can write

$$\left\langle\hat{L}\right\rangle_\psi = Sp\left(\hat{\rho}\hat{L}\right). \tag{2.24}$$

2.1.9 Representation of Operators Using Outer Products of Eigenvectors

Using the completeness condition (2.16) twice, we get:

$$\hat{A} = \hat{I}\hat{A}\hat{I} = \sum_{(l)}\sum_{(l')} \left|l\right\rangle\left\langle l\right|\hat{A}\left|l'\right\rangle\left\langle l'\right| = \sum_{(l)}\sum_{(l')} \left|l\right\rangle\left\langle l'\right| A_{ll'}, \tag{2.25}$$

where $A_{ll'} = \left\langle l\right|\hat{A}\left|l'\right\rangle$ is the matrix element of the operator $\hat{A}$ in the representation $\left|l\right\rangle$.

An operator expressed through its own eigenvectors can be represented by the decomposition [2]

$$\hat{L} = \sum_{(l)} l \left|l\right\rangle\left\langle l\right| . \tag{2.26}$$

[1] To do this, it is enough to recall that $\left|c_n\right|^2$ gives the probability of finding the system in the state $\left|l_n\right\rangle$, i.e., to get the measurement value in l_n

[2] Provided that the normalization of the eigenvectors: $\left\langle l|l\right\rangle = 1$

The generalization of this equality for an operator function has the form

$$F\left(\hat{L}\right)=\sum_{(l)}F\left(l\right)\left|l\right\rangle\left\langle l\right|. \tag{2.27}$$

2.1.10 Wave Functions in Coordinate and Momentum Representations

The transition from a state vector to a wave function is carried out by scalar multiplication of this state vector by the state vector of the corresponding observable quantity. For example, for the wave function in the coordinate representation

$$\phi\left(q\right)=\left\langle q\right|\left.\psi\right\rangle. \tag{2.28}$$

where $\langle q|$ is the eigenvector of the coordinate operator. In the momentum representation, we get:

$$\phi\left(p\right)=\left\langle p\right|\left.\psi\right\rangle. \tag{2.29}$$

where $\langle p|$ is the eigenvector of the momentum operator.

2.2 Dynamics of the Wave Function Change

The wave function $|\phi\rangle$ can change via two mechanisms:

- Wave function reduction during measurement
- Schrödinger equation between two successive measurements

2.2.1 Schrödinger Equation

The change in the state of a pure quantum system between two successive measurements is described by the following equation (Schrödinger)

$$i\hbar\frac{\partial\left|\phi\right\rangle}{\partial t}=\hat{\mathcal{H}}\left|\phi\right\rangle. \tag{2.30}$$

Equation (2.30) is reversible and, accordingly, not applicable to describing the change in the wave function at the moment of measurement.

It is worth noting the connection of the Schrödinger equation with Stone's theorem (**??**) TBD.

Schrödinger Equation in the Interaction Representation

Let's assume that the Hamiltonian can be divided into two parts:

$$\hat{\mathcal{H}} = \hat{\mathcal{H}}_0 + \hat{\mathcal{V}}.$$

Let's introduce the following wave function transformation:

$$\left|\phi\right\rangle_I = \exp\left\{\left(\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\}\left|\phi\right\rangle$$

and examine the following expression:

$$i\hbar\frac{\partial\left|\phi\right\rangle_I}{\partial t} = i\hbar\frac{i\hat{\mathcal{H}}_0}{\hbar}\exp\left\{\left(\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\}\left|\phi\right\rangle + \exp\left\{\left(\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\} i\hbar\frac{\partial\left|\phi\right\rangle}{\partial t} =$$

$$= -\hat{\mathcal{H}}_0\exp\left\{\left(\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\}\left|\phi\right\rangle + \exp\left\{\left(\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\}\left(\hat{\mathcal{H}}_0 + \hat{\mathcal{V}}\right)\left|\phi\right\rangle =$$

$$-\hat{\mathcal{H}}_0\exp\left\{\left(\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\}\left|\phi\right\rangle + \hat{\mathcal{H}}_0\exp\left\{\left(\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\}\left|\phi\right\rangle + \exp\left\{\left(\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\}\hat{\mathcal{V}}\left|\phi\right\rangle =$$

$$= \exp\left\{\left(\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\}\hat{\mathcal{V}}\left|\phi\right\rangle =$$

$$= \exp\left\{\left(\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\}\hat{\mathcal{V}}\exp\left\{\left(-\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\}\exp\left\{\left(\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\}\left|\phi\right\rangle =$$

$$= \hat{\mathcal{V}}_I\left|\phi\right\rangle_I,$$

where

$$\hat{\mathcal{V}}_I = \exp\left\{\left(\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\}\hat{\mathcal{V}}\exp\left\{\left(-\frac{i\hat{\mathcal{H}}_0 t}{\hbar}\right)\right\} \tag{2.31}$$

is the interaction Hamiltonian in the interaction representation.

Thus, we obtain the Schrödinger equation in the interaction representation:

$$i\hbar\frac{\partial\left|\phi\right\rangle_I}{\partial t} = \hat{\mathcal{V}}_I\left|\phi\right\rangle_I. \tag{2.32}$$

Density Matrix Equation of Motion

From the relation (2.30) we have

$$i\hbar\frac{\partial\left|\phi\right\rangle}{\partial t} = \hat{\mathcal{H}}\left|\phi\right\rangle,$$

$$-i\hbar\frac{\partial\left\langle\phi\right|}{\partial t} = \hat{\mathcal{H}}\left\langle\phi\right|,$$

thus for the density matrix $\hat{\rho} = |\phi\rangle \langle\phi|$ we obtain

$$i\hbar\frac{\partial\hat{\rho}}{\partial t} = i\hbar\frac{\partial |\phi\rangle \langle\phi|}{\partial t} = i\hbar\left(\frac{\partial |\phi\rangle}{\partial t} \langle\phi| + |\phi\rangle \frac{\partial \langle\phi|}{\partial t}\right) =$$
$$= \hat{\mathcal{H}} |\phi\rangle \langle\phi| - |\phi\rangle \langle\phi| \hat{\mathcal{H}} = \left[\hat{\mathcal{H}}, \hat{\rho}\right] \tag{2.33}$$

Equation (2.33) is often called the quantum Liouville equation and the von Neumann equation.

Evolution Operator. Heisenberg and Schrödinger Representations

The change in the wave function governed by (2.30) can also be described using some operator (evolution) $\hat{U}\left(t, t_0\right)$:

$$|\phi\left(t\right)\rangle = \hat{U}\left(t, t_0\right) |\phi\left(t_0\right)\rangle\,. \tag{2.34}$$

Equation (2.30) can be rewritten in the form

$$|\phi\left(t\right)\rangle = \exp\left(-\frac{i}{\hbar}\hat{\mathcal{H}}\left(t - t_0\right)\right) |\phi\left(t_0\right)\rangle\,,$$

from which we have for the evolution operator

$$\hat{U}\left(t, t_0\right) = \exp\left(-\frac{i}{\hbar}\hat{\mathcal{H}}\left(t - t_0\right)\right) \tag{2.35}$$

The evolution operator is unitary. Indeed:

$$\hat{U}\left(t, t_0\right) \hat{U}^{\dagger}\left(t, t_0\right) =$$
$$= \exp\left(-\frac{i}{\hbar}\hat{\mathcal{H}}\left(t - t_0\right)\right) \exp\left(+\frac{i}{\hbar}\hat{\mathcal{H}}\left(t - t_0\right)\right) = \hat{I}$$

Along with the Schrödinger representation where operators do not depend on time and wave functions change, there exists the Heisenberg representation where operators change with time.

Obviously, the average values of operators should not depend on the representation:

$$\langle\phi_H\left(t_0\right)| \hat{A}_H\left(t\right) |\phi_H\left(t_0\right)\rangle = \langle\phi_S\left(t\right)| \hat{A}_S |\phi_S\left(t\right)\rangle =$$
$$= \langle\phi_H\left(t_0\right)| \hat{U}^{\dagger}\left(t, t_0\right) \hat{A}_S \hat{U}\left(t, t_0\right) |\phi_H\left(t_0\right)\rangle\,,$$

from which, taking into account $\hat{A}_H\left(t_0\right) = \hat{A}_S\left(t_0\right)$, we obtain the evolution law of operators in the Heisenberg representation:

$$\hat{A}_H\left(t\right) = \hat{U}^{\dagger}\left(t, t_0\right) \hat{A}_H\left(t_0\right) \hat{U}\left(t, t_0\right) \tag{2.36}$$

At the same time, the equation for the operator $\hat{A}_H$ will look as follows:

$$\frac{\partial \hat{A}_H}{\partial t} = \frac{i}{\hbar}\hat{\mathcal{H}}\hat{U}^\dagger\left(t,t_0\right)\hat{A}_H\left(t_0\right)\hat{U}\left(t,t_0\right) - \\ -\frac{i}{\hbar}\hat{U}^\dagger\left(t,t_0\right)\hat{A}_H\left(t_0\right)\hat{U}\left(t,t_0\right)\hat{\mathcal{H}} = \frac{i}{\hbar}\left[\hat{\mathcal{H}},\hat{A}_H\right] \tag{2.37}$$

2.2.2 Differences Between Pure and Mixed States. Decoherence

Definition 2.2.1 (Pure State)**.** If the state of a system is described by a density matrix $\hat{\rho}$ that can be represented as

$$\hat{\rho} = \left|\psi\right\rangle\left\langle\psi\right| \tag{2.38}$$

then the state is called pure.

Definition 2.2.2 (Mixed State)**.** If the state of a system is described by a density matrix $\hat{\rho}$ that **cannot** be represented as in (2.38), i.e.

$$\hat{\rho} \neq \left|\psi\right\rangle\left\langle\psi\right|$$

then the state is called mixed.

A particular interest is the difference between pure and mixed states, in particular - how the transition from pure states to mixed states occurs.

Consider a two-level state (see Figure 2.2). In a pure state, it is described by the following wave function:

$$\left|\phi\right\rangle = c_a\left|a\right\rangle + c_b\left|b\right\rangle,$$

the corresponding density matrix appears as

$$\hat{\rho} = \left|\phi\right\rangle\left\langle\phi\right| = \\ = \left|c_a\right|^2\left|a\right\rangle\left\langle a\right| + \left|c_b\right|^2\left|b\right\rangle\left\langle b\right| + \\ + c_a c_b^*\left|a\right\rangle\left\langle b\right| + c_b c_a^*\left|b\right\rangle\left\langle a\right|, \tag{2.39}$$

or in matrix form

$$\hat{\rho} = \begin{pmatrix} \left|c_a\right|^2 & c_a c_b^* \\ c_b c_a^* & \left|c_b\right|^2 \end{pmatrix}.$$

The density matrix for a mixed state has only diagonal elements:

$$\hat{\rho} = \begin{pmatrix} \left|c_a\right|^2 & 0 \\ 0 & \left|c_b\right|^2 \end{pmatrix} = \\ = \left|c_a\right|^2\left|a\right\rangle\left\langle a\right| + \left|c_b\right|^2\left|b\right\rangle\left\langle b\right|. \tag{2.40}$$

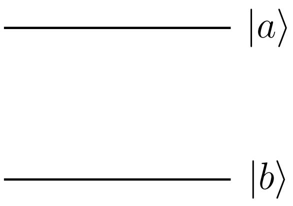

Figure 2.2: Model of the two-level atom used to describe decoherence.

The transition from (2.39) to (2.40) is called decoherence. In describing the decoherence process, we will follow [16].

The distinction between mixed and pure states manifests in the effect of the environment $\mathcal{E}$. In the case of pure states, the system considered and its environment are independent, i.e.

$$|\phi\rangle_{pure} = |\phi\rangle_{at} \otimes |\mathcal{E}\rangle \,. \tag{2.41}$$

In the case of mixed states, the atom and its environment form a so-called entangled state where states $|a\rangle$ and $|b\rangle$ correspond to distinguishable states of the environment $|\mathcal{E}_a\rangle$ and $|\mathcal{E}_b\rangle$.

$$|\phi\rangle_{mix} = c_a |a\rangle |\mathcal{E}_a\rangle + c_b |b\rangle |\mathcal{E}_b\rangle \,. \tag{2.42}$$

The density matrix corresponding to (2.42) appears as

$$\begin{aligned} \hat{\rho}_{mix} = |\phi\rangle_{mix} \langle\phi|_{mix} = \\ = |c_a|^2 |a\rangle \langle a| \otimes |\mathcal{E}_a\rangle \langle\mathcal{E}_a| + |c_b|^2 |b\rangle \langle b| \otimes |\mathcal{E}_b\rangle \langle\mathcal{E}_b| + \\ + c_a c_b^* |a\rangle \langle b| \otimes |\mathcal{E}_a\rangle \langle\mathcal{E}_b| + c_b c_a^* |b\rangle \langle a| \otimes |\mathcal{E}_b\rangle \langle\mathcal{E}_a| \,. \end{aligned} \tag{2.43}$$

If we now apply averaging over the environment variables to expression (2.43), we obtain

$$\begin{aligned} \langle\hat{\rho}_{mix}\rangle_{\mathcal{E}} = Sp_{\mathcal{E}}\left(\hat{\rho}\right) = \\ = \langle\mathcal{E}_a| \hat{\rho}_{mix} |\mathcal{E}_a\rangle + \langle\mathcal{E}_b| \hat{\rho}_{mix} |\mathcal{E}_b\rangle = \\ = |c_a|^2 |a\rangle \langle a| + |c_b|^2 |b\rangle \langle b| \,. \end{aligned} \tag{2.44}$$

Expression (2.44) is obtained under the assumption of an orthonormal basis $\{|\mathcal{E}_a\rangle, |\mathcal{E}_b\rangle\}$:

$$\begin{aligned} \langle\mathcal{E}_a |\mathcal{E}_a\rangle = \langle\mathcal{E}_b |\mathcal{E}_b\rangle = 1, \\ \langle\mathcal{E}_a |\mathcal{E}_b\rangle = \langle\mathcal{E}_b |\mathcal{E}_a\rangle = 0. \end{aligned} \tag{2.45}$$

The conditions (2.45) are key to understanding why the considered atomic system basis is distinguished and why, for example, other bases such as the Hadamard transform basis relative to the original are not considered for mixed states :

$$|\mathcal{A}\rangle = \frac{|a\rangle + |b\rangle}{\sqrt{2}},$$
$$|\mathcal{B}\rangle = \frac{|a\rangle - |b\rangle}{\sqrt{2}}. \tag{2.46}$$

The environmental states corresponding to the basis (2.46) are not orthogonal, hence the impossibility of using (2.46) as basis vectors for mixed states.

The process of decoherence, i.e., the transition from (2.41) to (2.42), can be described using the Schrödinger equation, and thus is theoretically reversible. The only requirement is the orthogonality of distinguishable environmental states: $\langle \mathcal{E}_a | \mathcal{E}_b \rangle = 0$. This requirement is always fulfilled for macroscopic systems, where the state depends on a very large number of variables. At the same time, in the case of macroscopic systems, it should be noted that there are many possible variants of final states $|\mathcal{E}_{a,b}\rangle$ such that the reverse process becomes practically unrealisable, as it is necessary to control a large number of possible variables described by the state of the environment. In this sense, the decoherence process has the same nature as the second law of thermodynamics (increasing entropy), which describes irreversible processes. [3]

The decoherence process is very fast, in particular [13] provides the following estimate: for systems with a mass of 1 g at a separation $\Delta x = 1$ cm. and a temperature $T = 300$ K , with relaxation time equal to the lifetime of the Universe $\tau_R = 10^{17}$ s. , the decoherence process takes 10^{-23} s.

2.2.3 Wave Function Reduction. Measurement in Quantum Mechanics

The process of choice (measurement result) is one of the most complex in quantum mechanics. Unlike the deterministic change of the wave function described by the Schrödinger equation (2.30), the measurement process is random in nature and requires different equations for its description.

Let's first consider pure states and assume that a measurement of a physical observable is being made, described by the operator $\hat{L}$. The eigenvalues and eigenfunctions of this operator are $\{l_k\}$ and $\{|l_k\rangle\}$ respectively. At the moment of measurement, instrument readings can take values corresponding to the eigenvalues of the measured operator (see Figure 2.3). Suppose the instrument reading is

[3] One should be a bit careful here since the second law of thermodynamics applies to closed systems, and decoherence processes themselves occur in open systems.

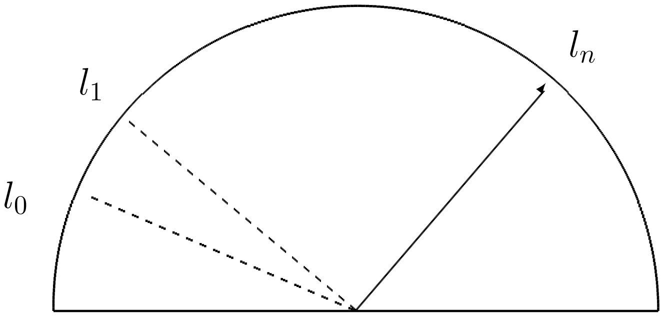

Figure 2.3: Measurement process. The instrument reading corresponds to one of the eigenvalues of the operator $\hat{L}$: $\{l_k\}$

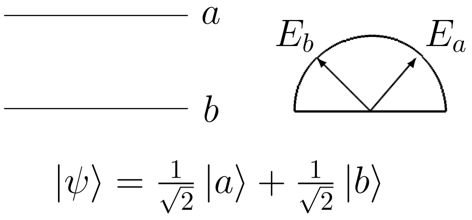
a
b
E_b
E_a
$|\psi\rangle = \frac{1}{\sqrt{2}}|a\rangle + \frac{1}{\sqrt{2}}|b\rangle$

Figure 2.4: The process of measuring the energy of a two-level atom in a pure state $|\psi\rangle = \frac{1}{\sqrt{2}}|a\rangle + \frac{1}{\sqrt{2}}|b\rangle$. The device registers the energy value E_a or E_b.

l_n, in this case the wave function should be $|l_n\rangle$, thus the following change in the wave function occurred:

$$|\phi\rangle \rightarrow |l_n\rangle ,$$

which can be described by the action of the projection operator $\hat{P}_n = |l_n\rangle\langle l_n|$ (2.18):

$$\hat{P}_n |\phi\rangle = c_n |l_n\rangle .$$

Example 2.2.1 (Measurement of the Energy of a Two-Level Atom)**.** *Consider a two-level atom in a pure state (see Figure 2.4)* $|\psi\rangle = \frac{1}{\sqrt{2}}|a\rangle + \frac{1}{\sqrt{2}}|b\rangle$.

Our instrument measures the energy of this atom and the Hamiltonian operator has 2 eigenfunctions $|a,b\rangle$*, corresponding to eigenvalues* E_a, E_b*. Thus, possible instrument readings belong to the set* $\{E_a, E_b\}$*.*

In the case where the instrument's needle shows E_a*, the following reduction*

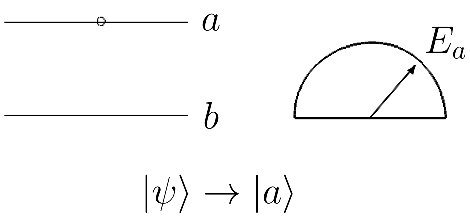

Figure 2.5: The process of measuring the energy of a two-level atom in a pure state $|\psi\rangle = \frac{1}{\sqrt{2}}|a\rangle + \frac{1}{\sqrt{2}}|b\rangle$. The device registers the energy value E_a. During the measurement, the following reduction occurs $|\psi\rangle \to |a\rangle$

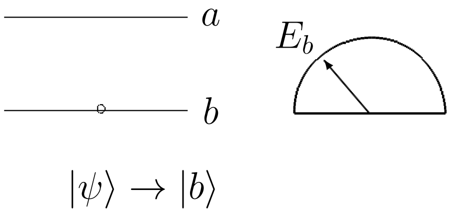

Figure 2.6: The process of measuring the energy of a two-level atom in a pure state $|\psi\rangle = \frac{1}{\sqrt{2}}|a\rangle + \frac{1}{\sqrt{2}}|b\rangle$. The device records the energy value E_b. During measurement, the following reduction occurs $|\psi\rangle \to |b\rangle$

occurs (see Figure 2.5)

$$|\psi\rangle \to |a\rangle .$$

Similarly, in the case of E_b, the following reduction occurs (see Figure 2.6)

$$|\psi\rangle \to |b\rangle .$$

There is no way to predict the result that will be obtained from a single measurement. However, one can say with what probability a particular result will be obtained.

Indeed, in the case of a mixed state

$$\hat{\rho} = \sum_n |c_n|^2 |l_n\rangle \langle l_n|$$

the coefficients $P_n = |c_n|^2$ define the probabilities of finding the system in the state $|l_n\rangle$.

For a pure state

$$|\phi\rangle = \sum_n c_n |l_n\rangle$$

we also have that the probability of finding the system in state $|l_n\rangle$ is given by the number $P_n = |c_n|^2$.

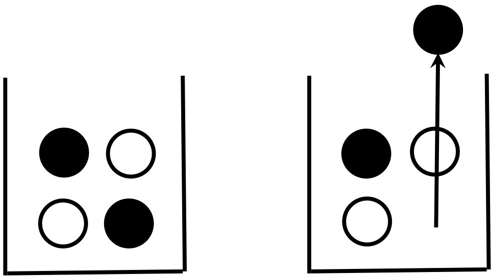

Figure 2.7: Example of a mixed state. The color of the ball does not change as a result of the "measurement"

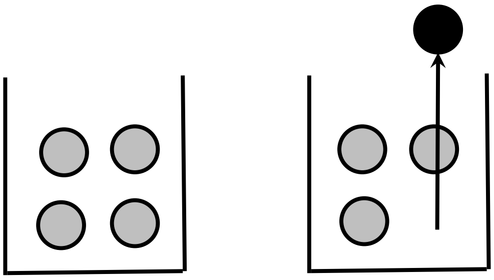

Figure 2.8: Example of a pure state. The color of the sphere changes as a result of "measurement"

The main difference between pure and mixed states in terms of measurement is that in the first case (pure state), the measurement changes the wave function, i.e., the state itself. At the same time, if during the measurement a certain final state $|l_i\rangle$ was obtained, one can't say it was the same before the measurement. Mixed states behave like classical objects, i.e., if during the measurement a state $|l_i\rangle$ was obtained, it can be asserted it was the same before the measurement, and the measurement represents a selection of one state from many possible ones.

Example 2.2.2. Choosing from an urn with balls of two colors *Suppose we have an urn with 4 balls. With a probability of $\frac{1}{2}$, either a white or black ball will be drawn. Suppose that as a result of the experiment, a black ball is obtained. If the considered system is quantum and in a mixed state (see Figure 2.7), the state of the drawn ball (color) did not change as a result of the experiment.*

If the considered system is pure (see Figure 2.8), the state of each ball is described by a superposition of two colors - black and white. Thus, as a result of the experiment, this superposition is destroyed, and the ball acquires a definite color (black in our case), i.e., it can be said that the ball's color changes.

2.3 Composite Systems

Systems consisting of multiple parts behave fundamentally differently in the cases of classical and quantum systems.

As an example, consider a system of two particles. The position of the first is described by 3 coordinates, which represent a point in some linear space L_1: $(x_1, y_1, z_1) \in L_1$. For the second, let's assume we have 2 coordinates in space L_2 completely determining the position: $(x_2, y_2) \in L_2$. It is obvious that to fully describe the position of the two particles, we need 5 coordinates: $(x_1, y_1, z_1, x_2, y_2) \in L_1 \times L_2$. Thus, composite systems, in the classical case, are described by points in space $L^{classical}$, which is the Cartesian product of the original spaces: $L = L_1 \times L_2$. A distinctive feature of the Cartesian product is that the dimensions of the corresponding spaces add up:

$$\dim L^{classical} = \dim L_1 + \dim L_2.$$

In the quantum case, a composite system is described by vectors (points) in a space that is the tensor product of the original spaces:

$$L^{quantum} = L_1 \otimes L_2.$$

In this space, there are 6 basis vectors, and consequently, 6 numbers are required to describe the position of the system: $(x_1 \cdot x_2, y_1 \cdot x_2, z_1 \cdot x_2, x_1 \cdot y_2, y_1 \cdot y_2, z_1 \cdot y_2) \in L^{quantum}$. Accordingly, the dimensions multiply:

$$\dim L^{quantum} = \dim L_1 \cdot \dim L_2.$$

Thus, if we have two independent quantum systems with wave functions $|\psi_1\rangle$ and $|\psi_2\rangle$, the composite system will have the following wave function

$$|\psi_{12}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle ,$$

and obviously the following equalities hold

$$\begin{aligned} |\psi_1\rangle &= Sp_2 \, |\psi_{12}\rangle , \\ |\psi_2\rangle &= Sp_1 \, |\psi_{12}\rangle \end{aligned} \tag{2.47}$$

i.e., to obtain the state of subsystem 1, one must take the trace over the states of system 2 from the total wave function.

Composite systems have an interesting property where even though the parts of this system represent mixed states, the whole system is pure. As an example, consider so-called entangled states :

$$|\psi\rangle = \frac{|0_1\rangle \, |1_2\rangle - |1_1\rangle \, |0_2\rangle}{\sqrt{2}}$$

which is pure. For the first particle, we have

$$\begin{aligned}\rho_1 = Sp_2\,|\psi\rangle \otimes \langle\psi| =\\ = \langle 0_2|\psi\rangle\,\langle\psi|0_2\rangle + \langle 1_2|\psi\rangle\,\langle\psi|1_2\rangle =\\ = \frac{|1_1\rangle\,\langle 1_1| + |0_1\rangle\,\langle 0_1|}{2}\end{aligned}$$

i.e., the state of the first particle is mixed.

Chapter 3

Quantum Computing

Algorithms play a significant role in computer science. An algorithm is a sequence of steps necessary to obtain an answer to a given problem. Each problem is characterized by a certain number, which determines its size. The complexity of an algorithm is assessed as the number of simple operations required to solve the given problem. Obviously, in most cases (but not always), this number grows with the size of the problem.

Example 3.0.1. Searching for an array element *The task is to find an element of an array that meets certain conditions. The size of the problem is the number of elements in the array N.*

In the general case (an unstructured array of data), the search is conducted by simple enumeration. This search requires a number of operations (comparisons) that grows linearly with the size of the array $O\left(N\right)$.

In the case of structured data, the number of operations required for the search can be reduced. For example, in the case of a sorted array, the complexity of the problem grows as $O\left(logN\right)$.

However, the existence of an algorithm does not yet guarantee its practical implementability. In particular, algorithms that require an exponential number of steps relative to the size of the original problem are considered practically unfeasible, despite the theoretical possibility of a solution.

One example is the problem of factoring a natural number, that is, the task of decomposing it into prime factors (see example 3.0.2).

Example 3.0.2. Factorization of natural numbers *The task is to find the decomposition of a number into prime factors. The size of the problem is the digit length of the original number. For example, for the case of digit length $r = 4$: $1 \leq N = 15 \leq 2^r = 2^4 = 16$). The result can be found easily and quickly: $15 = 3 \cdot 5$.*

As the number of digits r increases, the number of operations required for factorization in classical algorithms grows as $O\left(2^r\right)$, which for the case $r = 1000$–2000 means the practical impossibility of factorizing such numbers.

Quantum objects have properties that differ from classical objects; accordingly, algorithms based on quantum objects can, in some cases, have characteristics inaccessible to classical algorithms. For example, Grover's quantum algorithm [3] solves the problem of searching an unstructured data array (see example 3.0.1) with $O\left(\sqrt{N}\right)$ operations. Shor's algorithm [10] allows solving the problem of factoring a number (see example 3.0.2) using a linear number of operations $O\left(r\right)$.

3.1 Basic Principles of Quantum Computing

3.1.1 Information Representation. Classical and Quantum States

The main difference between quantum and classical computers lies in how they store information.

In the classical case, information is stored in certain memory cells. The state of each memory cell is described by a single number that can take the value 0 or 1. If m memory cells are combined, the overall state of the classical system (which it can take at a particular moment in time) is described by m numbers.

In the quantum case, a memory cell is represented by a qubit, which requires two complex numbers α_0 and α_1 for its description [1]:

$$|\psi\rangle_1 = \alpha_0 \left|0\right\rangle + \alpha_1 \left|1\right\rangle .$$

To describe a composite system consisting of m qubits, 2^m complex numbers are required. In other words, a quantum state contains all possible classical states as a superposition. As an example, consider a system consisting of 3 qubits:

$$\begin{aligned} |\psi\rangle_3 = \alpha_0 \left|000\right\rangle + \alpha_1 \left|001\right\rangle + \alpha_2 \left|010\right\rangle + \alpha_3 \left|011\right\rangle + \\ +\alpha_4 \left|100\right\rangle + \alpha_5 \left|101\right\rangle + \alpha_6 \left|110\right\rangle + \alpha_7 \left|111\right\rangle . \end{aligned} \tag{3.1}$$

As can be seen, any classical state of a system of 3 bits is represented as one of the terms in the superposition (3.1). For example, the number $5_{10} = 101_2$ appears in (3.1) with the coefficient α_5.

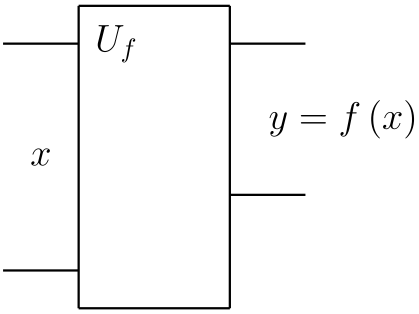

Figure 3.1: Classical computations. An input number x consisting of n bits is given, and the output is the result $y = f(x)$ described by m bits

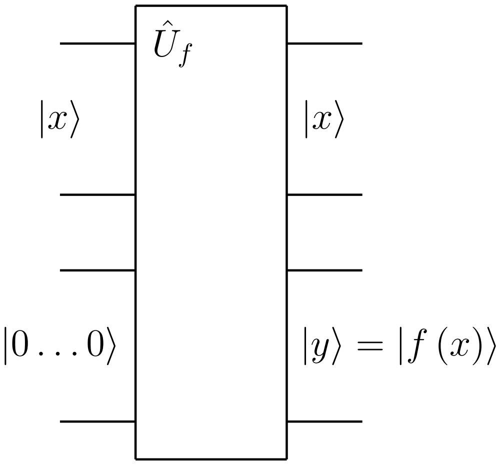

Figure 3.2: Quantum reversible computation. The input is the number $|x\rangle$ consisting of n qubits and an auxiliary input of zero states (m qubits), and the output is the result $|y\rangle = |f(x)\rangle$ described by m qubits and the original state $|x\rangle$.

3.1.2 Reversible Computations

In the classical case, computation involves transforming the original n bits into a result described by m bits (see Figure 3.1). The transformation is defined by a certain function $f(x)$. A typical example is addition modulo 2 (see Table 1.1) where the input consists of 2 bits ($n = 2$), and the output is 1 bit ($m = 1$).

Such a scheme will not work in the quantum case, primarily because the change of pure quantum states over time must occur via a unitary evolution operator (2.35), i.e., it must be reversible, which is impossible for our classical example [2]. Therefore, in quantum computing, a different scheme is used (see Figure 3.2) that allows reversible computations.

The input consists of the initial data x described by n qubits, along with m qubits in the state $|0\rangle$, so that the total number of inputs and outputs matches

[1] More precisely, it is described by three real numbers, because $\alpha_{0,1}$ are subject to the constraint $|\alpha_0|^2+|\alpha_1|^2 = 1$, from which, considering $\alpha_{0,1} = r_{0,1}e^{i\theta_{0,1}}$, we get that $r_0^2 = 1 - r_1^2$

[2] it is impossible to derive two bits of source information from one bit (result)

each other. Consequently, the relationship between input and output can be described as [3]

$$\underbrace{|x\rangle}_{n}\underbrace{|f(x)\rangle}_{m} = \hat{U}_f \underbrace{|x\rangle}_{n}\underbrace{|0\ldots0\rangle}_{m}. \tag{3.2}$$

3.2 Quantum Logic Elements

How can an element be constructed that performs the transformation $\hat{U}_f$ (3.2). There is a set of elements from which an element can be constructed, with a given accuracy, that performs the required transformation $\hat{U}_f$. Such sets are called universal.

3.2.1 Universal Set of Quantum Gates

Identity Transformation

$$\hat{\sigma}_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Negation

$$\hat{\sigma}_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Phase Shift

$$\hat{\sigma}_2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

[3] More precisely, it is generally written as $\underbrace{|x\rangle}_{n}\underbrace{|f(x)\rangle}_{m}\underbrace{|r\rangle}_{k} = \hat{U}_f \underbrace{|x\rangle}_{n}\underbrace{|0\ldots0\rangle}_{m+k}$, where $|r\rangle$ is a remainder of size k qubits not used in the computations, serving to ensure the unitarity of the operator $\hat{U}_f$

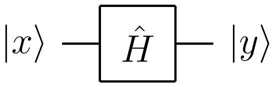

Figure 3.3: Hadamard Transformation on a Single Qubit

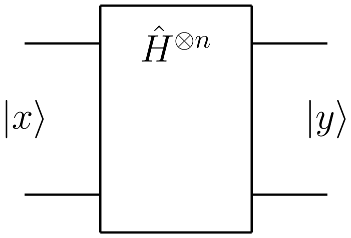

Figure 3.4: Hadamard transformation $\hat{H}^{\otimes n}$ on multiple qubits

Hadamard Transformation

One of the basic quantum logic elements is the Hadamard transformation (see Figure 3.3), which is defined by the following relations

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

In matrix form, this transformation can be written as

$$\hat{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{3.3}$$

where the basis vectors chosen are

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

and

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

From (3.3), the following property of the operator $\hat{H}$ can be obtained:

$$\hat{H}\hat{H} = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{3.4}$$

This transformation is used to obtain a superposition of states containing all possible values of the argument of the calculated function (see Figure 3.4).

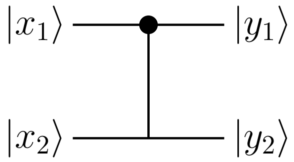

Figure 3.5: Controlled CNOT Gate

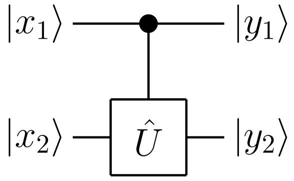

Figure 3.6: Control Element

CNOT

The transformation matrix is

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

This gate (see Figure 3.5) is applied to two qubits and inverts the state of the second qubit only if the first qubit is equal to one.

Thus, if our initial state of two qubits was

$$|\psi_i\rangle = a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + d\,|11\rangle$$

it transforms into

$$CNOT\,|\psi_i\rangle = |\psi_f\rangle = a\,|00\rangle + b\,|01\rangle + c\,|11\rangle + d\,|10\rangle$$

Universal Set

Definition 3.2.1 (Universal Set of Quantum Gates)**.** A set of quantum gates is called universal if any unitary transformation can be approximated with a given accuracy by a finite sequence of gates from this set.

Theorem 3.2.1 (Kitaev)**.** *The set $\hat{\sigma}_0, \hat{\sigma}_1, \hat{\sigma}_2, \hat{H}, CNOT$ is universal.*

Proof. TBD □

3.2.2 Control Elements

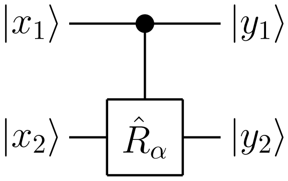

Figure 3.7: Controlled phase shift $\hat{R}_\alpha$

Chapter 4

Grover's Algorithm

4.1 Grover's Algorithm

Consider the following problem. Suppose we have a large dataset consisting of N elements in which we need to find an element satisfying certain conditions (see Figure 4.1). If the data is sorted, then using divide-and-conquer algorithms, the desired element can be found in time on the order of $O\left(logN\right)$ (see section 8.9). In some cases, the original dataset cannot be prepared for fast search, in which case the classical search is carried out in time on the order of $O\left(N\right)$.

One example is symmetric encryption algorithms where the task is to determine the key based on known encrypted text and its corresponding original text. In this case, preprocessing the data is not feasible, and the "brute-force" solution is a simple enumeration of all possible values.

Grover's algorithm [3] solves the problem of unstructured search in time on the order of $O\left(\sqrt{N}\right)$.

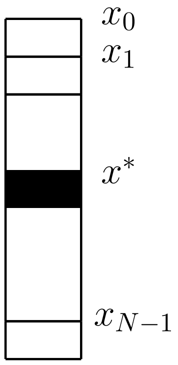

Figure 4.1: Search in an unstructured data set (search for a "needle in a haystack")

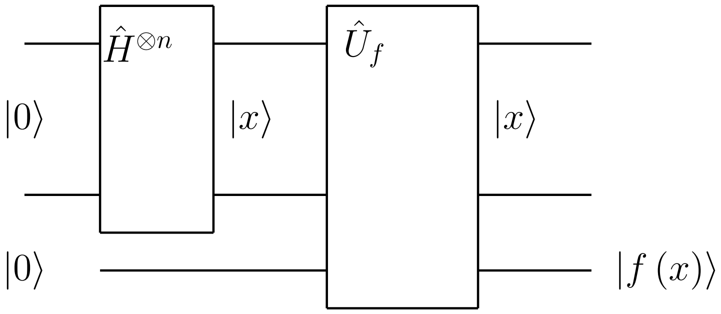

Figure 4.2: Computation of the function $f(x)$. The output of the circuit is a superposition of states of the form $\frac{1}{\sqrt{N}}\left(\sum_{x\neq x^*}|x\rangle\otimes|0\rangle+|x^*\rangle\otimes|1\rangle\right)$

4.1.1 Algorithm Description

Suppose we have a quantum circuit that computes the value of a function $f(x)$ which can take only two values: 0 and 1. The value 1 holds only for the desired element:

$$\begin{aligned} f(x)|_{x=x^*} &= 1, \\ f(x)|_{x\neq x^*} &= 0. \end{aligned} \tag{4.1}$$

Figure Figure 4.2 shows a diagram for calculating the desired function. The output is a state of the form

$$|out\rangle = \frac{1}{\sqrt{N}}\left(\sum_{x\neq x^*}|x\rangle\otimes|0\rangle+|x^*\rangle\otimes|1\rangle\right), \tag{4.2}$$

where N is the total number of elements in the sequence being searched.

Looking at expression (4.2), we can see that the proposed scheme, despite computing the function at the desired point, does not allow us to select the desired element because all elements of the resulting sequence are equally likely, i.e., each element can be chosen (as a result of measurement) with equal probability: $\frac{1}{N}$.

Grover proposed an algorithm that would increase the probability of detecting the desired element in the resulting superposition (4.2).

The circuit implementing Grover's algorithm is a certain block described by the operator $\hat{U}_G$, which is repeated several times (see Figure 4.3). With each iteration step, the probability of detecting the desired element increases.

The basic element $\hat{U}_G$ consists of the sequential action of two operators (see Figure 4.4):

$$\hat{U}_G = \hat{U}_s\hat{U}_{x^*},$$

where $\hat{U}_{x^*}$ is a phase inversion operator, and $\hat{U}_s$ is an inversion about the mean operator.

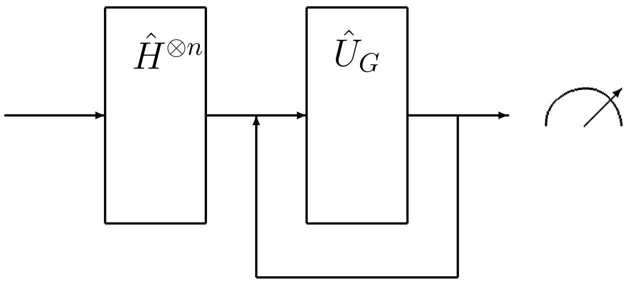

Figure 4.3: Grover's Algorithm

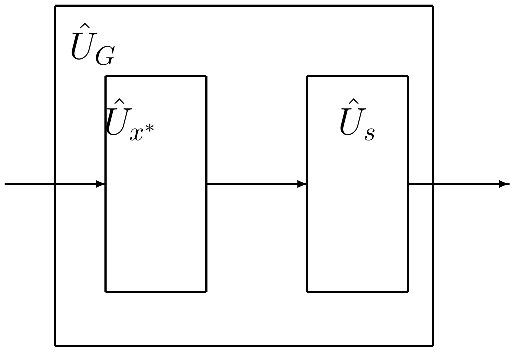

Figure 4.4: Grover's Algorithm. Basic Element

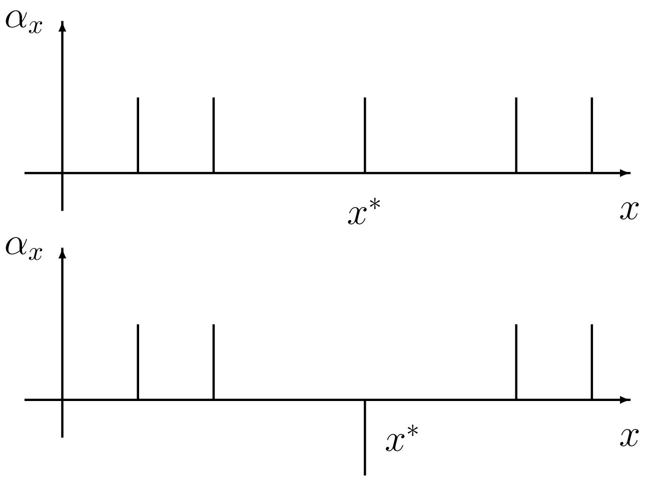

Figure 4.5: Grover's Algorithm. Phase Inversion. Described by the following relation $\hat{U}_{x^*} \left(\sum_x \alpha_x \left|x\right\rangle\right) = \sum_x \alpha_x \left(-1\right)^{f(x)} \left|x\right\rangle$

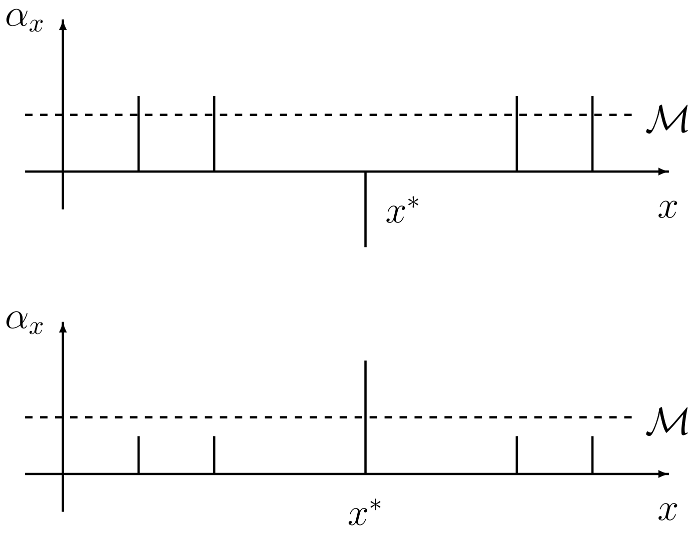

Figure 4.6: Grover's Algorithm. Inversion about the mean. Described by the following relation $\hat{U}_s\left(\sum_x \alpha_x |x\rangle\right) = \sum_x (2\mathcal{M} - \alpha_x)|x\rangle$

The action of the operator $\hat{U}_{x^*}$ is described by the following relation (see Figure 4.5):

$$\hat{U}_{x^*}\left(\sum_x \alpha_x |x\rangle\right) = \sum_x \alpha_x (-1)^{f(x)} |x\rangle . \tag{4.3}$$

The operator $\hat{U}_{x^*}$ can be rewritten as

$$\hat{U}_{x^*} = \hat{I} - 2|x^*\rangle\langle x^*| .$$

Indeed,

$$\begin{aligned} \left(\hat{I} - 2|x^*\rangle\langle x^*|\right)\left(\sum_x \alpha_x |x\rangle\right) = \\ = \sum_x \alpha_x |x\rangle - 2\alpha_{x^*}|x^*\rangle = \sum_{x \neq x^*} \alpha_x |x\rangle - \alpha_{x^*}|x^*\rangle = \\ = \sum_x \alpha_x (-1)^{f(x)} |x\rangle , \end{aligned}$$

which matches (4.3).

The action of the operator $\hat{U}_s$ is described by the following relation (see Figure 4.6):

$$\hat{U}_s\left(\sum_x \alpha_x |x\rangle\right) = \sum_x (2\mathcal{M} - \alpha_x)|x\rangle , \tag{4.4}$$

where $\mathcal{M} = \sum_x \frac{\alpha_x}{N}$.

The operator $\hat{U}_s$ can be rewritten as

$$\hat{U}_s = 2\left|s\right\rangle\left\langle s\right| - \hat{I},$$

where $\left|s\right\rangle = \frac{1}{\sqrt{N}}\sum_x \left|x\right\rangle$ is the initial state in Grover's algorithm. Indeed,

$$\begin{aligned}\left(2\left|s\right\rangle\left\langle s\right| - \hat{I}\right)\left(\sum_x \alpha_x \left|x\right\rangle\right) &= \\ = 2\sum_x \alpha_x \left\langle s|x\right\rangle \left|s\right\rangle - \sum_x \alpha_x \left|x\right\rangle &= \\ = \frac{2}{N}\sum_x \alpha_x \sum_x \left|x\right\rangle - \sum_x \alpha_x \left|x\right\rangle &= \\ = \sum_x \left(2\mathcal{M} - \alpha_x\right)\left|x\right\rangle&,\end{aligned}$$

which matches (4.4).

4.1.2 Analysis of Grover's Algorithm

The schematic form of Grover's algorithm is given in Alg. 1.

Algorithm 1 Grover's Algorithm

```
$\left|\psi\right\rangle_0 \Leftarrow \frac{1}{\sqrt{N}}\sum_x \left|x\right\rangle$
$t \Leftarrow 1$
**repeat**
    $\left|\psi\right\rangle_t \Leftarrow \hat{U}_s\hat{U}_{x^*}\left|\psi\right\rangle_{t-1}$
    $t \Leftarrow t + 1$
**until** $(t < \frac{\pi}{4}\sqrt{N})$
**return** result of measuring the state $\left|\psi\right\rangle_t$
```

We will be interested in two questions: what is the algorithmic complexity of Grover's algorithm and are there algorithms that can perform the search task in an unstructured data set more efficiently than Grover's algorithm.

The criterion of algorithm efficiency is the following fact: a good algorithm should find the desired value with a minimum number of function calls (4.1).

Let us consider the very first iteration. The initial state $\left|\psi\right\rangle_0$ has the following form

$$\left|\psi\right\rangle_0 = \sum_x \alpha_x \left|x\right\rangle = \left|s\right\rangle = \frac{1}{\sqrt{N}}\sum_x \left|x\right\rangle = \frac{1}{\sqrt{N}}\sum_{x\neq x^*} \left|x\right\rangle + \frac{1}{\sqrt{N}}\left|x^*\right\rangle .$$

Thus, the coefficient before the target element has the form $\alpha_x^* = \frac{1}{\sqrt{N}}$.

After applying the phase inversion operator U_{x^*} from (4.3) we get

$$\hat{U}_{x^*} |\psi\rangle_0 = \frac{1}{\sqrt{N}} \sum_{x \neq x^*} |x\rangle - \frac{1}{\sqrt{N}} |x^*\rangle = \sum_x \beta_x |x\rangle ,$$

where $\beta_{x^*} = -\frac{1}{\sqrt{N}}$ and $\beta_{x \neq x^*} = \frac{1}{\sqrt{N}}$.

After applying the inversion about the mean operator $\hat{U}_s$ from (4.4) we get

$$\hat{U}_G |\psi\rangle_0 = \hat{U}_s \hat{U}_{x^*} |\psi\rangle_0 = \hat{U}_s \sum_x \beta_x |x\rangle =$$
$$= \sum_x (2M - \beta_x) |x\rangle \approx \sum_{x \neq x^*} \left(2\frac{1}{\sqrt{N}} - \frac{1}{\sqrt{N}} \right) |x\rangle +$$
$$+ \left(2\frac{1}{\sqrt{N}} + \frac{1}{\sqrt{N}} \right) |x^*\rangle = \frac{1}{\sqrt{N}} \sum_{x \neq x^*} |x\rangle + \frac{3}{\sqrt{N}} |x^*\rangle . \tag{4.5}$$

In deriving (4.5), it was assumed that

$$\mathcal{M} = \frac{\sum_x \alpha_x}{N} \approx \frac{N}{N\sqrt{N}} = \frac{1}{\sqrt{N}}.$$

Thus after the first iteration of Grover's algorithm, the amplitude α_{x^*} increased by $\frac{2}{\sqrt{N}}$. If we approximate this result for an arbitrary iteration, we can conclude that a 50% probability of finding $|x^*\rangle$ will be achievable after the following number of iterations:

$$\frac{1}{\sqrt{2}} / \frac{2}{\sqrt{N}} = \frac{\sqrt{N}}{2\sqrt{2}} = O\left(\sqrt{N}\right).$$

More accurate calculations [5] give for the number of iterations $\frac{\pi}{4}\sqrt{N}$.

We may ask about the optimality of Grover's algorithm: is there a quantum algorithm that performs a search in an unstructured data set faster than $O\left(\sqrt{N}\right)$ function calls (4.1). The article [1] shows that no such algorithm exists.

4.1.3 Implementation of Basic Elements of Grover's Algorithm

Phase Inversion

How can phase inversion be implemented: what does the quantum logic element look like that performs the transformation (4.3), i.e., how can $f(x)$ be "sent" into the phase?

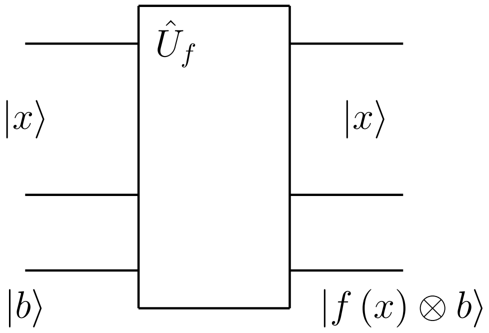

Figure 4.7: Grover's Algorithm. Implementation of Phase Inversion (operator $\hat{U}_{x^*}$). Assuming $b = |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$, for the depicted circuit we obtain $\hat{U}_f \left(\sum_x \alpha_x |x\rangle\right) \otimes |-\rangle = \sum_x \alpha_x (-1)^{f(x)} |x\rangle \otimes |-\rangle$

Consider the circuit shown in Figure 4.7. The proposed scheme performs the following transformation:

$$|x\rangle \otimes |b\rangle \to |x\rangle \otimes |b \oplus f(x)\rangle ,$$

where the following notation is introduced: $a \oplus b = a + b \mod 2$.

For the case $|b\rangle = |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$ we have

$$\begin{aligned}
|x\rangle \otimes |-\rangle \to |x\rangle \otimes \left(\frac{|0 \oplus 0\rangle - |1 \otimes 0\rangle}{\sqrt{2}}\right) = \\
= |x\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = |x\rangle \otimes |-\rangle , x \neq x^*, \\
|x\rangle \otimes |-\rangle \to |x\rangle \otimes \left(\frac{|0 \oplus 1\rangle - |1 \oplus 1\rangle}{\sqrt{2}}\right) = \\
= |x\rangle \otimes \left(\frac{|1\rangle - |0\rangle}{\sqrt{2}}\right) = - |x\rangle \otimes |-\rangle , x = x^*,
\end{aligned}$$

thus, we have the following transformation

$$|x\rangle \otimes |-\rangle \to (-1)^{f(x)} |x\rangle \otimes |-\rangle . \tag{4.6}$$

Inversion About the Mean

Consider the circuit shown in Figure 4.8. The element $\hat{U}_{x\neq 0}$ performs a transformation similar to (4.6) with the function $f(x = 0) = 0$, and $f(x \neq 0) = 1$, thus

$$\begin{aligned}
\hat{U}_{x\neq 0} |x\rangle \otimes |-\rangle = |x\rangle \otimes |-\rangle , x = 0, \\
\hat{U}_{x\neq 0} |x\rangle \otimes |-\rangle = - |x\rangle \otimes |-\rangle , x \neq 0,
\end{aligned}$$

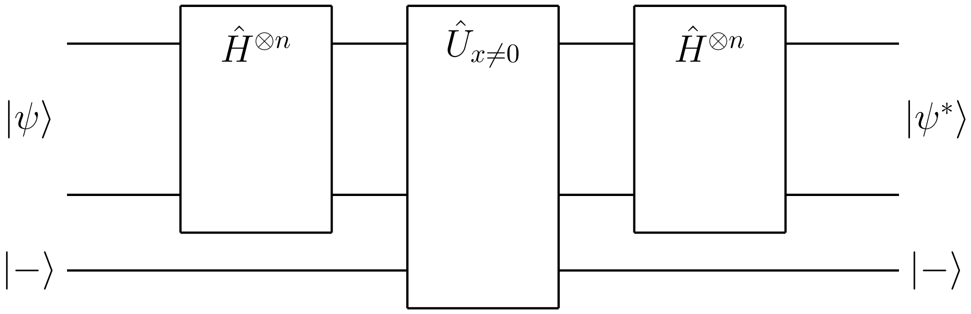

Figure 4.8: Grover's Algorithm. Implementation of the inversion about the mean (operator $\hat{U}_s$): $|\psi\rangle \otimes |-\rangle \rightarrow |\psi^*\rangle \otimes |-\rangle$, where $|\psi\rangle = \sum_x \alpha_x |x\rangle$, $|\psi^*\rangle = \sum_x (2\mathcal{M} - \alpha_x) |x\rangle$. The proposed scheme implements the following transformation: $\hat{H}^{\otimes n}\hat{U}_{x\neq 0}\hat{H}^{\otimes n} \sum_x \alpha_x |x\rangle \otimes |-\rangle = \sum_x (2\mathcal{M} - \alpha_x) |x\rangle \otimes |-\rangle$

i.e., the transformation matrix looks like this

$$\hat{U}_{x\neq 0} = \begin{pmatrix} 1\otimes|-\rangle & 0 & 0 & \cdots & 0 \\ 0 & -1\otimes|-\rangle & 0 & \cdots & 0 \\ 0 & 0 & -1\otimes|-\rangle & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1\otimes|-\rangle \end{pmatrix} =$$

$$= \left\{ \begin{pmatrix} 2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \right\} \otimes |-\rangle \,.$$

Combining this result with two Hadamard transformations , and using (3.4), we get:

$$\hat{H}^{\otimes n}\left\{\begin{pmatrix}2&0&0&\cdots&0\\0&0&0&\cdots&0\\0&0&0&\cdots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&0\end{pmatrix}-\begin{pmatrix}1&0&0&\cdots&0\\0&1&0&\cdots&0\\0&0&1&\cdots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&1\end{pmatrix}\right\}\otimes\left|-\right\rangle\hat{H}^{\otimes n}=$$

$$=\left\{\hat{H}^{\otimes n}\begin{pmatrix}2&0&0&\cdots&0\\0&0&0&\cdots&0\\0&0&0&\cdots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&0\end{pmatrix}\hat{H}^{\otimes n}-\hat{H}^{\otimes n}\hat{I}\hat{H}^{\otimes n}\right\}\otimes\left|-\right\rangle=$$

$$=\left\{\begin{pmatrix}\frac{2}{N}&\frac{2}{N}&\frac{2}{N}&\cdots&\frac{2}{N}\\\frac{2}{N}&\frac{2}{N}&\frac{2}{N}&\cdots&\frac{2}{N}\\\frac{2}{N}&\frac{2}{N}&\frac{2}{N}&\cdots&\frac{2}{N}\\\vdots&\vdots&\vdots&\ddots&\vdots\\\frac{2}{N}&\frac{2}{N}&\frac{2}{N}&\cdots&\frac{2}{N}\end{pmatrix}-\hat{I}\right\}\otimes\left|-\right\rangle=$$

$$=\left\{\begin{pmatrix}\frac{2}{N}-1&\frac{2}{N}&\frac{2}{N}&\cdots&\frac{2}{N}\\\frac{2}{N}&\frac{2}{N}-1&\frac{2}{N}&\cdots&\frac{2}{N}\\\frac{2}{N}&\frac{2}{N}&\frac{2}{N}-1&\cdots&\frac{2}{N}\\\vdots&\vdots&\vdots&\ddots&\vdots\\\frac{2}{N}&\frac{2}{N}&\frac{2}{N}&\cdots&\frac{2}{N}-1\end{pmatrix}\right\}\otimes\left|-\right\rangle. \qquad (4.7)$$

If we apply the operator $\hat{H}^{\otimes n}\hat{U}_{x\neq0}\hat{H}^{\otimes n}$, using the result from (4.7) we get:

$$\hat{H}^{\otimes n}\hat{U}_{x\neq0}\hat{H}^{\otimes n}\sum_x\alpha_x\left|x\right\rangle=$$

$$=\begin{pmatrix}\frac{2}{N}-1&\frac{2}{N}&\frac{2}{N}&\cdots&\frac{2}{N}\\\frac{2}{N}&\frac{2}{N}-1&\frac{2}{N}&\cdots&\frac{2}{N}\\\frac{2}{N}&\frac{2}{N}&\frac{2}{N}-1&\cdots&\frac{2}{N}\\\vdots&\vdots&\vdots&\ddots&\vdots\\\frac{2}{N}&\frac{2}{N}&\frac{2}{N}&\cdots&\frac{2}{N}-1\end{pmatrix}\begin{pmatrix}\alpha_0\\\alpha_1\\\alpha_2\\\vdots\\\alpha_{N-1}\end{pmatrix}=$$

$$=\begin{pmatrix}\frac{2}{N}\sum_x\alpha_x-\alpha_0\\\frac{2}{N}\sum_x\alpha_x-\alpha_1\\\frac{2}{N}\sum_x\alpha_x-\alpha_2\\\vdots\\\frac{2}{N}\sum_x\alpha_x-\alpha_{N-1}\end{pmatrix}=\sum_x\left(2\mathcal{M}-\alpha_x\right)\left|x\right\rangle.$$

Thus, the scheme proposed in Figure 4.8 indeed performs inversion about the mean.

Chapter 5

Public Key Cryptosystems. RSA Algorithm

Public key cryptosystems are currently the main ones in use. Moreover, quantum computers can effectively break them. We will consider these algorithms in more detail later.

5.1 RSA Algorithm

The RSA algorithm (abbreviation from the last names Rivest, Shamir, and Adleman) is an asymmetric encryption algorithm [1], based on the complexity of factorizing a number into prime factors.

5.1.1 Key Generation

Consists of several steps

- Two prime numbers p and q are chosen
- The product of the chosen prime numbers is calculated $n = p \cdot q$
- The Euler's function (8.3.1) is calculated (see properties 8.3.1 and 8.3.2) $\phi(n) = (p-1)(q-1)$
- An integer e is chosen such that $1 < e < \phi(n)$ and e and $\phi(n)$ are coprime, i.e., $\text{GCD}(e, \phi(n)) = 1$.
- Calculate $d \equiv e^{-1} \mod \phi(n)$, i.e., $d \cdot e \equiv 1 \mod \phi(n)$.

[1] An asymmetric (public key) encryption algorithm uses two different keys: one for encryption and the other for decryption

The public key consists of two numbers: the modulus n and the public exponent e. These two numbers are used to encrypt the original message.

The private key also consists of two numbers: the modulus n and the private exponent d.

The original numbers p and q are kept secret, as with their help, the calculation of $\phi(n)$ becomes trivial.

It should be noted that to obtain the private key from the public key, one must calculate $\phi(n)$ given n. This task is difficult (if p and q are unknown), as noted in comment 8.3.1.

Example 5.1.1. (RSA. Key Generation) *Choose two prime numbers* $p = 3$ *and* $q = 7$. *The product of these numbers is* $n = 21$. *Euler's function (8.3.1)* $\phi(n) = (p-1)(q-1) = 2 \cdot 6 = 12$.

Choose a number e *(public exponent) such that* $1 < e < 12$ *and* $GCD(e, 12) = 1$. *Obviously,* $e = 5$ *satisfies the stated conditions.*

Calculate the private exponent $d \equiv 5^{-1} \mod 12$, *i.e.,* $d = 5$. *Indeed,* $5 \cdot 5 = 25 = 2 \cdot 12 + 1$, *i.e.,* $5 \cdot 5 \equiv 1 \mod 12$.

Thus, we get

- *Public key* $(n = 21, e = 5)$
- *Private key* $(n = 21, d = 5)$

5.1.2 Encryption

Suppose we need to encrypt a certain message M. Initially, it is converted into an integer (or integers) m such that $0 < m < \phi(n)$. The encrypted text c is then calculated:

$$c \equiv m^e \mod n \tag{5.1}$$

Example 5.1.2. (RSA. Encryption) *Suppose we have a public key* $(n = 21, e = 5)$ *(see Ex. 5.1.1) and we want to encrypt the following message* $m = 1101_2 = 11_{10}$. *The ciphertext is calculated using formula* (5.1) $c \equiv 11^5 \mod 21 = 2$.

5.1.3 Decryption

m can be recovered from c using the following formula:

$$m \equiv c^d \mod n. \tag{5.2}$$

Having m, one can recover the original message M.

Example 5.1.3. (RSA. Decryption) *Suppose we have a private key* $(n = 21, d = 5)$ *(see Example 5.1.1) and ciphertext* $c = 2$ *from Example 5.1.2.*

The original text is computed using the formula (5.2) $m \equiv 2^5 \mod 21 = 11 = 1101_2$.

5.1.4 Proof

We want to prove that

$$\left(m^e\right)^d \equiv m \mod p \cdot q$$

for any positive number m when p and q are prime numbers, and e and d satisfy the expression

$$d \cdot e \equiv 1 \mod \phi\left(p \cdot q\right),$$

which we can rewrite as

$$d \cdot e - 1 = h\left(p-1\right)\left(q-1\right).$$

Thus,

$$m^{e \cdot d} = mm^{h(p-1)(q-1)}.$$

Then there are two cases: when m is divisible by p and when m and p are coprime.

In the first case

$$m^{e \cdot d} \equiv m \equiv 0 \mod p$$

In the second case, we use Fermat's Little Theorem (Theorem 8.4.1) :

$$mm^{h(p-1)(q-1)} = m\left(m^{p-1}\right)^{h(q-1)} \equiv m \cdot 1^{h(q-1)} \equiv m \mod p.$$

Similarly, we have either

$$m^{e \cdot d} \equiv m \equiv 0 \mod q$$

or due to Fermat's Little Theorem

$$mm^{h(p-1)(q-1)} = m\left(m^{q-1}\right)^{h(p-1)} \equiv m \cdot 1^{h(p-1)} \equiv m \mod q.$$

Thus, we have the following two types of relationships: the trivial equality

$$x_1 = m \equiv m \mod p,$$
$$x_1 = m \equiv m \mod q,$$

and the newly obtained relationships

$$x_2 = m^{ed} \equiv m \mod p,$$
$$x_2 = m^{ed} \equiv m \mod q,$$

from which, by virtue of the Chinese Remainder Theorem (Theorem 8.5.1)

$$x_1 \equiv x_2 \mod p \cdot q$$

i.e.

$$m^{e \cdot d} \equiv m \mod n$$

5.2 Shor's Algorithm

One of the most popular RSA encryption algorithms (see section 5.1) is based on the assumption of the complexity of factorization (factoring into prime numbers) of large numbers. Therefore, algorithms that allow performing factorization into prime numbers are of particular interest. Below is the description of such an algorithm proposed by Shor [10].

5.2.1 Number Factorization and Period Finding for Functions

The problem of factorizing a certain number N is closely related to finding the period of functions. Consider the following, which is called the modular exponentiation function

$$f(x, a) = a^x \mod N. \tag{5.3}$$

The function (5.3) depends on the analyzed number N and two arguments x and a. The argument a is chosen from the following conditions

$$\begin{aligned} 0 < a < N, \\ \gcd(N, a) = 1. \end{aligned} \tag{5.4}$$

A typical graph of the function (5.3) is presented in **??**.

The conditions for selecting the coefficient a (5.4) are such that a and N do not have common divisors. If such divisors exist, they are the desired solution to the factorization problem and can be easily found using the Euclidean algorithm (see section 8.1).

The function (5.3) is periodic, i.e., there exists a number r such that $f(x + r, a) = f(x, a)$. The smallest possible number r is called the period of the function (5.3).

To prove the periodicity, note that $f(x, a)$ cannot be equal to zero. Indeed, if the condition $f(x, a) = 0$ is met, then

$$\exists x \in \{0, 1, \dots\} : a^x = k \cdot N,$$

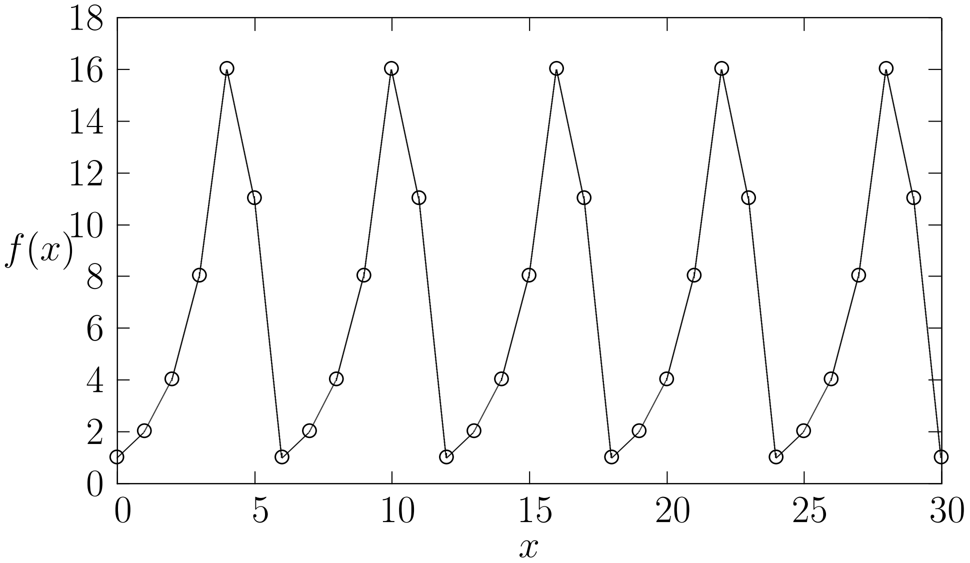

Figure 5.1: Graph of the function $f(x, a) = a^x \mod N$ with $a = 2$, $N = 21$. The period of the function is $r = 6$.

where k is an integer, which is not possible due to the coprimeness of a and N (5.4) [2].

Thus, the range of values of function (5.3) is limited to the set

$$f(x, a) \in \{1, \dots, N-1\},$$

hence

$$\exists k, j : k > j, k, j \in \{0, 1, \dots, N\}, f(k, a) = f(j, a),$$

which proves the periodicity of the function (5.3).

Let $k = j + r$, then

$$a^k \mod N = a^{j+r} \mod N = a^j a^r \mod N = a^j \mod N,$$

since a and N are coprime, we can write

$$a^r \equiv 1 \mod N. \tag{5.5}$$

The period of the function (5.3) can be either even or odd. In Shor's algorithm, we are interested in the former case: the period is an even number. Otherwise, a new number a is chosen and the period finding is repeated. Thus, considering $r = 2 \cdot l$ we can rewrite (5.5) as

$$a^{2 \cdot l} \equiv 1 \mod N,$$

[2] Here it is obviously assumed that $N > 1$

and since r is the smallest number satisfying the periodicity condition, then

$$a^l \not\equiv 1 \mod N.$$

If, at the same time, a number a is chosen such that

$$a^l \not\equiv -1 \mod N,$$

then we have

$$\left(a^l - 1\right)\left(a^l + 1\right) = k \cdot N, \tag{5.6}$$

where k is some integer. From (5.6) we find that $a^l \pm 1$ have common non-trivial (different from 1) divisors with N.

Example 5.2.1. Finding the Divisors of the Number $N = 21$ *As an example, consider the problem of finding the divisors of $N = 21$. Choosing $a = 2$, we obtain the period of the function* (5.3) $r = 6$ *(see* **??***). Obviously,*

$$2^3 \equiv 8 \mod 21 \not\equiv -1 \mod 21.$$

Thus, by finding the corresponding greatest common divisors, we solve the problem

$$\begin{aligned} \gcd\left(2^3 - 1, 21\right) &= \gcd(7, 21) = 7, \\ \gcd\left(2^3 + 1, 21\right) &= \gcd(9, 21) = 3, \\ 21 &= 7 \cdot 3. \end{aligned}$$

Thus, the problem of factorizing the number N can be reduced to the problem of finding the period of a certain function using the following algorithm:

Algorithm 2 Shor's Algorithm

```
$a \Leftarrow 0$
repeat
    Choose a new number $a$ such that $0 < a < N$
    if GCD$(a, N) \neq 1$ then
        return $a$
    end if
    Find the period $r$ of the function $f(x, a) = a^x \mod N$
until $(r \not\equiv 0 \mod 2)$ or $(a^{\frac{r}{2}} \equiv -1 \mod N)$
return GCD$\left(a^{\frac{r}{2}} \pm 1, N\right)$
```

5.3 Discrete Fourier Transform

The Fourier transform plays an important role in digital signal processing, particularly for analyzing periodic sequences.

5.3.1 Definition

Definition 5.3.1. Assume there are M samples $x_0, x_1, \dots, x_{M-1}$, then the discrete Fourier transform is given by the following relation

$$\tilde{X}_k = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} x_m e^{-\frac{2\pi i}{M} k \cdot m}, \tag{5.7}$$

which is also written as

$$\{x_m\} \longleftrightarrow \left\{\tilde{X}_k\right\}.$$

The inverse Fourier transform can be obtained using a similar relation

$$x_k = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} \tilde{X}_m e^{\frac{2\pi i}{M} k \cdot m},$$

In Figure 5.2, a graph of a certain periodic function and its Fourier transform is shown.

Expression (5.7) can also be rewritten in matrix form

$$\vec{\tilde{X}} = \hat{F}\vec{x},$$

where

$$\vec{x} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{M-1} \end{pmatrix}, \vec{\tilde{X}} = \begin{pmatrix} \tilde{X}_0 \\ \tilde{X}_1 \\ \vdots \\ \tilde{X}_{M-1} \end{pmatrix},$$

and the matrix $\hat{F}$ is given by

$$\hat{F} = \frac{1}{\sqrt{M}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-i\omega} & e^{-2i\omega} & \cdots & e^{-(M-1)i\omega} \\ 1 & e^{-2i\omega} & e^{-4i\omega} & \cdots & e^{-2(M-1)i\omega} \\ 1 & e^{-3i\omega} & e^{-6i\omega} & \cdots & e^{-3(M-1)i\omega} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-(M-1)i\omega} & e^{-2(M-1)i\omega} & \cdots & e^{-(M-1)(M-1)i\omega} \end{pmatrix}, \tag{5.8}$$

where

$$\omega = \frac{2\pi}{M}.$$

For the matrix element of the matrix (5.8), we can write

$$F_{nm} = \frac{1}{\sqrt{M}} e^{-i\omega nm}, \tag{5.9}$$

where $n, m \in \{0, 1, \dots, M-1\}$.

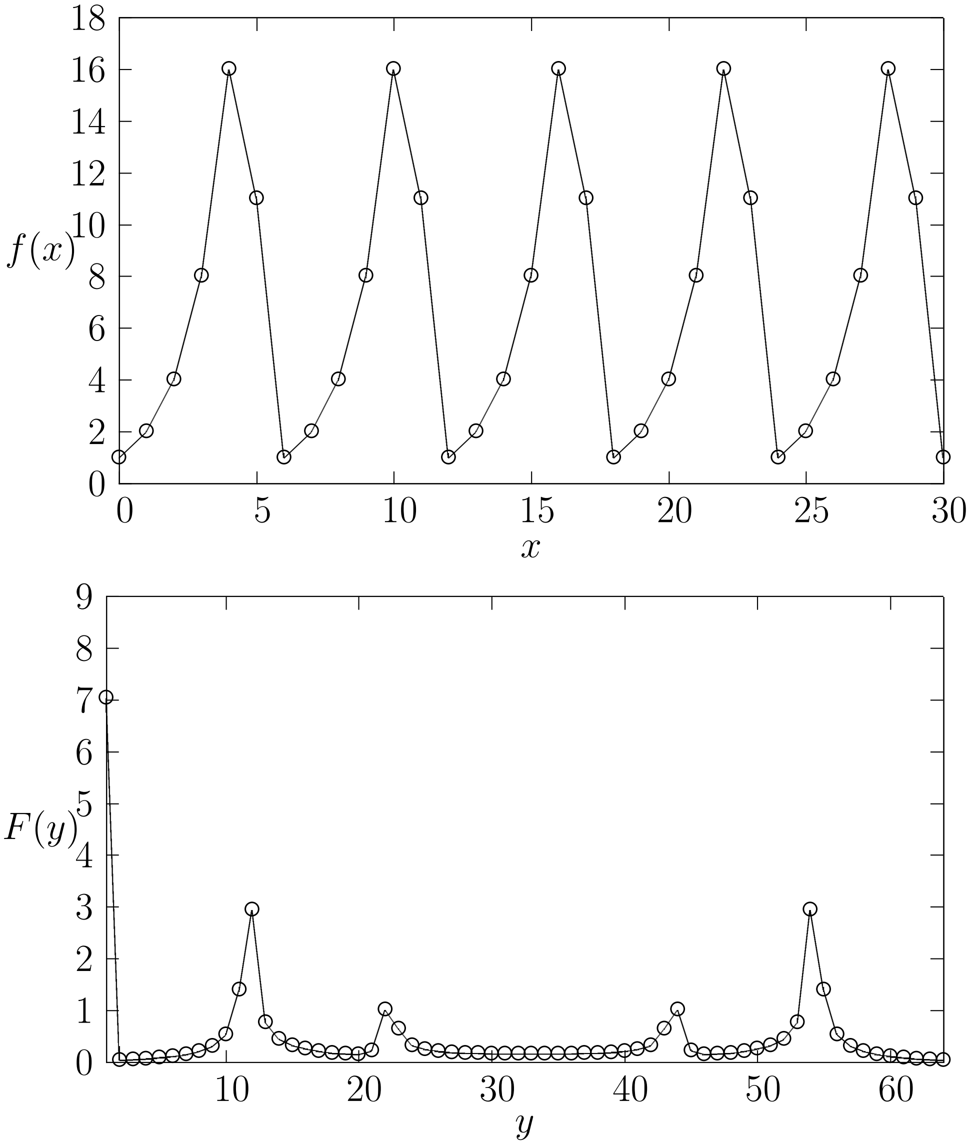

Figure 5.2: Periodic function $f\left(x,a\right) = a^x \mod N$ for $a = 2$, $N = 21$ (top graph) and its discrete Fourier transform (bottom graph). Period of the original function $r = 6$. Number of samples $M = 64$, local maxima with period $T \approx \frac{M}{r} \approx 10.67$ are visible.

5.3.2 Properties of the Discrete Fourier Transform

The following properties of the Fourier transform should be noted, as they play a key role in Shor's algorithm:

Lemma 5.3.1. (Shift) *If* $\{x_n\} \longleftrightarrow \left\{\tilde{X}_k\right\}$ *then* $\left\{x_{(n-m) \mod M}\right\} \longleftrightarrow \left\{e^{-i\omega mk}\tilde{X}_k\right\}$

Proof. For the sequence $\left\{x_{(n-m) \mod M}\right\}$, we can write

$$\left\{x_{(n-m) \mod M}\right\} = \begin{pmatrix} x_{-m \mod M} \\ x_{-m+1 \mod M} \\ \vdots \\ x_{-1 \mod M} \\ x_0 \\ x_1 \\ \vdots \\ x_{M-m-1} \end{pmatrix} = \begin{pmatrix} x_{M-m} \\ x_{M-m+1} \\ \vdots \\ x_{M-1} \\ x_0 \\ x_1 \\ \vdots \\ x_{M-m-1} \end{pmatrix},$$

and thus

$$\hat{F}\left\{x_{(n-m) \mod M}\right\} =$$

$$= \frac{1}{\sqrt{M}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-i\omega} & e^{-2i\omega} & \cdots & e^{-(M-1)i\omega} \\ 1 & e^{-2i\omega} & e^{-4i\omega} & \cdots & e^{-2(M-1)i\omega} \\ 1 & e^{-3i\omega} & e^{-6i\omega} & \cdots & e^{-3(M-1)i\omega} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-(M-1)i\omega} & e^{-2(M-1)i\omega} & \cdots & e^{-(M-1)(M-1)i\omega} \end{pmatrix} \begin{pmatrix} x_{M-m} \\ x_{M-m+1} \\ \vdots \\ x_0 \\ \vdots \\ x_{M-m-1} \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{x_{M-m}}{\sqrt{M}} + \frac{x_{M-m+1}}{\sqrt{M}} + \cdots + \frac{x_0}{\sqrt{M}} + \cdots + \frac{x_{M-m-1}}{\sqrt{M}} \\ \frac{x_{M-m}}{\sqrt{M}} + \frac{e^{-i\omega}x_{M-m+1}}{\sqrt{M}} + \cdots + \frac{e^{-i\omega m}x_0}{\sqrt{M}} + \cdots + \frac{e^{-i\omega(M-1)}x_{M-m-1}}{\sqrt{M}} \\ \frac{x_{M-m}}{\sqrt{M}} + \frac{e^{-2i\omega}x_{M-m+1}}{\sqrt{M}} + \cdots + \frac{e^{-2i\omega m}x_0}{\sqrt{M}} + \cdots + \frac{e^{-2i\omega(M-1)}x_{M-m-1}}{\sqrt{M}} \\ \vdots \\ \frac{x_{M-m}}{\sqrt{M}} + \frac{e^{-mi\omega}x_{M-m+1}}{\sqrt{M}} + \cdots + \frac{e^{-mi\omega m}x_0}{\sqrt{M}} + \cdots + \frac{e^{-mi\omega(M-1)}x_{M-m-1}}{\sqrt{M}} \\ \vdots \end{pmatrix}. \tag{5.10}$$

Considering the relation

$$e^{-i\omega kM} = 1, k \in \{0, 1, \dots\},$$

the expression (5.10) can be rewritten as

$$\hat{F}\left\{x_{(n-m) \mod M}\right\} =$$

$$= \frac{1}{\sqrt{M}} \begin{pmatrix} x_{M-m} + \dots + x_{M-m-1} \\ e^{-i\omega m} e^{-i\omega(M-m)} x_{M-m} + \dots + e^{-i\omega 2m} e^{-i2\omega(M-m-1)} \\ e^{-i\omega 2m} e^{-i2\omega(M-m)} x_{M-m} + \dots + e^{-i\omega 2m} e^{-i2\omega(M-m-1)} \\ \vdots \end{pmatrix} =$$

$$= \begin{pmatrix} \tilde{X}_0 \\ e^{-i\omega m} \tilde{X}_1 \\ e^{-i\omega 2m} \tilde{X}_2 \\ \vdots \end{pmatrix}.$$

□

Lemma 5.3.2. (Periodicity) *If the sequence $\{x_n\}$ has a period r: $x_n = x_{n+r}$, and the number of samples M is a multiple of r, then the non-zero terms of the Fourier transform follow with a period of $\frac{M}{r}$.*

Proof. Indeed, if $M \mod r = 0$ and $kr \mod M \neq 0$, then

$$1 - e^{-i\omega kr} \neq 0,$$

hence

$$\tilde{X}_k = \frac{1}{\sqrt{M}} \sum_{n=0}^{M-1} e^{-i\omega kn} x_n =$$

$$= \frac{1}{\sqrt{M}} \left(\sum_{n=0}^{r-1} e^{-i\omega kn} x_n + \sum_{n=0}^{r-1} e^{-i\omega k(n+r)} x_{n+r} + \right.$$

$$\left. + \sum_{n=0}^{r-1} e^{-i\omega k(n+2r)} x_{n+2r} + \dots \right) =$$

$$= \frac{1}{\sqrt{M}} \left(\sum_{n=0}^{r-1} e^{-i\omega kn} x_n + \sum_{n=0}^{r-1} e^{-i\omega k(n+r)} x_n + \sum_{n=0}^{r-1} e^{-i\omega k(n+2r)} x_n + \dots \right) =$$

$$= \frac{1}{\sqrt{M}} \left(\sum_{n=0}^{r-1} x_n e^{-i\omega kn} \sum_{l=0}^{\frac{M}{r}-1} e^{-i\omega klr} = \sum_{n=0}^{r-1} x_n e^{-i\omega kn} \frac{1 - e^{-i\omega k \frac{M}{r} r}}{1 - e^{-i\omega kr}} \right) =$$

$$= \frac{1}{\sqrt{M}} \frac{1 - e^{-i\omega kM}}{1 - e^{-i\omega kr}} \sum_{n=0}^{r-1} x_n e^{-i\omega kn} = 0. \quad (5.11)$$

If $M \mod r = 0$ and $kr \mod M = 0$, then

$$e^{-i\omega kr} = e^{-i\frac{2\pi}{M}kr} = 1,$$

hence

$$\begin{aligned} \tilde{X}_k = \frac{1}{\sqrt{M}} \sum_{n=0}^{M-1} e^{-i\omega kn} x_n = \\ = \frac{1}{\sqrt{M}} \left(\sum_{n=0}^{r-1} e^{-i\omega kn} x_n + \sum_{n=0}^{r-1} e^{-i\omega kn} x_{n+r} + \sum_{n=0}^{r-1} e^{-i\omega kn} x_{n+2r} + \dots \right) = \\ = \frac{1}{\sqrt{M}} \left(\sum_{n=0}^{r-1} e^{-i\omega kn} x_n + \sum_{n=0}^{r-1} e^{-i\omega kn} x_n + \sum_{n=0}^{r-1} e^{-i\omega kn} x_n + \dots \right) = \\ = \frac{1}{\sqrt{M}} \frac{M}{r} \sum_{n=0}^{r-1} e^{-i\omega kn} x_n \neq 0. \end{aligned} \tag{5.12}$$

Thus, from expressions (5.11) and (5.12) it follows that non-zero terms follow with a period of $T = \frac{M}{r}$. □

Remark 5.3.1. (Periodicity of Maxima) *It should be noted that expression (5.11) (in the case when the period is not a multiple of the number of samples: $M \mod r \neq 0$) will be approximately equal to 0 for those values of k that differ significantly from values multiples of $\frac{M}{r}$, i.e., local maxima of the Fourier transform will repeat with a period of $\frac{M}{r}$.*

5.3.3 Fast Fourier Transform

Calculations using formula (5.7) have a complexity of order $O\left(M^2\right)$, where M is the number of elements (samples) [3].

There is an algorithm for fast calculation using formula (5.7) which has a complexity of $O\left(M \log M\right)$.

Using the divide and conquer paradigm (see section 8.9), one can notice the form of the recording (5.9) and observe that the expression (5.7) can be rewritten as

$$\tilde{X}_k = \sum_{m=0}^{M-1} F^M_{km} x_m,$$

[3] Indeed, it is necessary to obtain M elements, for each of which M multiplication operations are required

where the notation F_{km}^{M} indicates that the matrix (5.9) of size $M \times M$ is used. If M is even, then

$$\tilde{X}_k = \sum_{m=0}^{M-1} F_{k,m}^{M} x_m = \sum_{m=0}^{\frac{M}{2}-1} F_{k,2m}^{M} x_{2m} + \sum_{m=0}^{\frac{M}{2}-1} F_{k,2m+1}^{M} x_{2m+1},$$

where

$$F_{k,2m}^{M} = e^{-i\omega k 2m} = e^{-ikm\frac{2\pi}{\frac{M}{2}}} = F_{k,m}^{\frac{M}{2}},$$
$$F_{k,2m+1}^{M} = e^{-i\omega k(2m+1)} = e^{-i\omega k} e^{-ikm\frac{2\pi}{\frac{M}{2}}} = e^{-2\pi i\frac{k}{M}} F_{k,m}^{\frac{M}{2}},$$

i.e.,

$$\tilde{X}_k = \sum_{m=0}^{\frac{M}{2}-1} F_{k,m}^{\frac{M}{2}} x_{2m} + \exp\left\{\left(-2\pi i\frac{k}{M}\right)\right\} \sum_{m=0}^{\frac{M}{2}-1} F_{k,m}^{\frac{M}{2}} x_{2m+1}. \tag{5.13}$$

The complexity of calculations using formula (5.13) is determined by the following relation

$$T(M) = 2T\left(\frac{M}{2}\right) + O(M). \tag{5.14}$$

The validity of (5.14) can be verified by noticing that the calculations of complexity $T(M)$ in (5.13) break down into two sub-problems of complexity $T\left(\frac{M}{2}\right)$.

Using the master theorem for recurrence relations (case 2) (Theorem 8.8.1) , we obtain

$$T(M) = O(M \log M).$$

5.4 Quantum Fourier Transform

For the analysis of periodic sequences (functions), the discrete Fourier transform may be used (see section 5.3), which is defined by the following relation (5.7):

$$\tilde{X}_k = \sum_{m=0}^{M-1} x_m e^{-\frac{2\pi}{M} k\cdot m}, \tag{5.15}$$

where the original sequence of numbers $\{x_m\}$ has M terms.

5.4.1 Quantum Fourier Transform Scheme

The quantum Fourier transform [4] deals with states of the form

$$|x\rangle = \sum_{k=0}^{M-1} x_k \, |k\rangle \,, \tag{5.16}$$

where there is a sequence of amplitudes $\{x_k\}$ that defines the original sequence for the Fourier transform (5.15). The basis vector $|k\rangle$ records the number of the term in this sequence.

Obviously, the terms of the sequence (5.16) must meet the normalization condition

$$\sum_k |x_k|^2 = 1.$$

Assume that some operator $\hat{F}^M$ (quantum Fourier transform operator) transforms the basis vector $|k\rangle$ according to the rule defined by relation (5.7):

$$\hat{F}^M \, |k\rangle = \frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} e^{-i\omega kj} \, |j\rangle_{inv} \tag{5.17}$$

The systems of basis vectors $\{|k\rangle\}$ and $\{|k\rangle_{inv}\}$ represent the same set of vectors numbered differently.

From (5.16) and (5.17), we obtain

$$\begin{aligned} \hat{F}^M \, |x\rangle = \sum_{j=0}^{M-1} x_k \hat{F}^M \, |k\rangle = \\ = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} \sum_{j=0}^{M-1} e^{-i\omega kj} x_k \, |j\rangle_{inv} = \\ = \sum_{j=0}^{M-1} \left\{ \frac{1}{\sqrt{M}} \left(\sum_{k=0}^{M-1} e^{-i\omega kj} x_k \right) \right\} |j\rangle_{inv} = \\ = \sum_{j=0}^{M-1} \tilde{X}_j \, |j\rangle_{inv} = \left| \tilde{X} \right\rangle_{inv} , \end{aligned}$$

where

$$\tilde{X}_j = \tilde{X}_j^M = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} e^{-i\omega kj} x_k . \tag{5.18}$$

[4] The work [4] was used for analyzing the quantum Fourier transform scheme

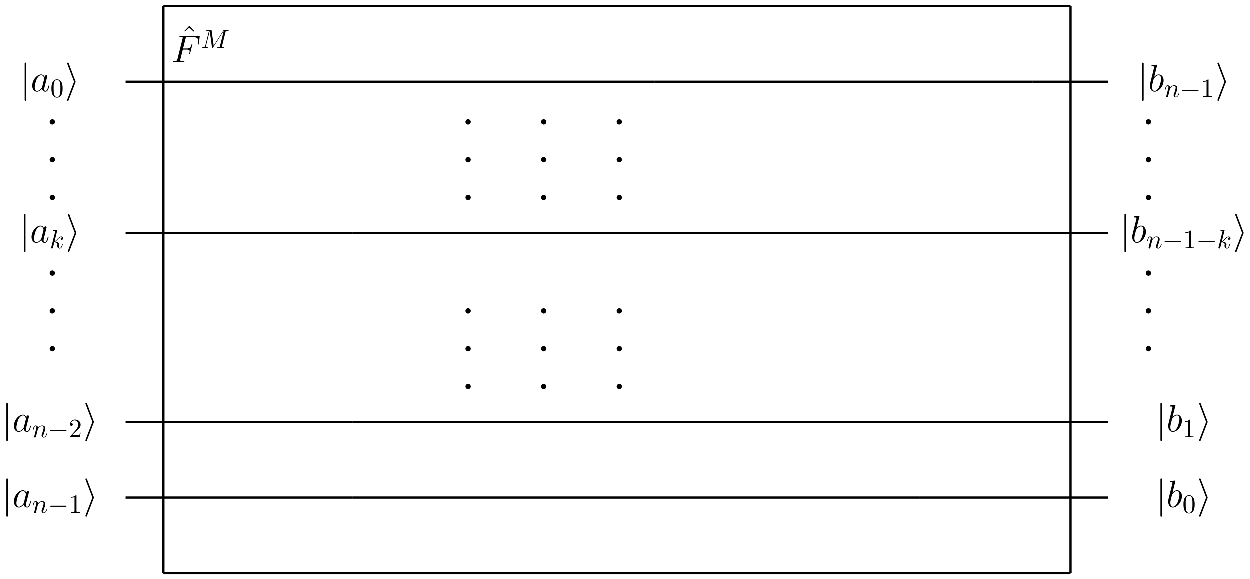

Figure 5.3: Quantum Fourier transform scheme based on the fast Fourier transform algorithm. Input and output data

Expression (5.18) repeats the classical analog (5.7), i.e., it can be written as:

$$|x\rangle \longleftrightarrow \left|\tilde{X}\right\rangle_{inv}.$$

Now, let's assume that the input to our system is a state of the form (5.16) which represents a superposition of M basis states $\{|k\rangle\}$ (see Figure 5.3). Let's assume that the number of basis states is a power of two, i.e., the basis state can be represented as the tensor product of $n = \log_2 M$ qubits:

$$|k\rangle = \left|a_0^{(k)}\right\rangle \otimes \left|a_1^{(k)}\right\rangle \otimes \cdots \otimes \left|a_{n-1}^{(k)}\right\rangle,$$

where

$$k = a_0^{(k)} + 2^1 a_1^{(k)} + \cdots + 2^{n-1} a_{n-1}^{(k)},$$
$$a_i^{(k)} \in \{0, 1\}.$$

At the output (see Figure 5.3), we have a superposition of M basis states $\{|j\rangle_{inv}\}$, where the state $|j\rangle_{inv}$ is derived as:

$$|j\rangle_{inv} = \left|b_{n-1}^{(j)}\right\rangle \otimes \left|b_{n-2}^{(j)}\right\rangle \otimes \cdots \otimes \left|b_0^{(j)}\right\rangle,$$

where

$$j = b_0^{(j)} + 2^1 b_1^{(j)} + \cdots + 2^{n-1} b_{n-1}^{(j)},$$
$$b_i^{(j)} \in \{0, 1\}.$$

From formula (5.13), it can be observed that if there is an input signal x consisting of $n = \log_2 M$ bits, then the bit $a_0^{(k)}$ can be used to select even (the first term of the sum (5.13)) or odd (the second term of the sum (5.13)) components.

In fact, excluding $a_0^{(k)}$, the state (5.16) can be represented as the sum of even and odd components:

$$\begin{aligned}
|x\rangle = \sum_{k=0}^{M-1} x_k |k\rangle = \sum_{k=0}^{M-1} x_k \left|a_0^{(k)}\right\rangle \otimes \left|a_1^{(k)}\right\rangle \otimes \cdots \otimes \left|a_{n-1}^{(k)}\right\rangle = \\
= \sum_{m=0}^{\frac{M}{2}-1} x_{k=2m} |0\rangle \otimes \left|a_1^{(k)}\right\rangle \otimes \cdots \otimes \left|a_{n-1}^{(k)}\right\rangle + \\
+ \sum_{m=0}^{\frac{M}{2}-1} x_{k=2m+1} |1\rangle \otimes \left|a_1^{(k)}\right\rangle \otimes \cdots \otimes \left|a_{n-1}^{(k)}\right\rangle = \\
= \sum_{m=0}^{\frac{M}{2}-1} x_{2m} |0\rangle \otimes |m\rangle + \sum_{m=0}^{\frac{M}{2}-1} x_{2m+1} |1\rangle \otimes |m\rangle = \\
= \sum_{m=0}^{\frac{M}{2}-1} x_{2m} |2m\rangle + \sum_{m=0}^{\frac{M}{2}-1} x_{2m+1} |2m+1\rangle \,,
\end{aligned}$$

where

$$m = a_1^{(k)} + 2^1 a_2^{(k)} + \cdots + 2^{n-2} a_{n-1}^{(k)}.$$

Applying the Fourier transform to only the higher bits $\hat{F}^{\frac{M}{2}}$, i.e., excluding $a_0^{(k)}$, we get (see Figure 5.4):

$$\begin{aligned}
|x\rangle \to \hat{F}^{\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}-1} x_{2m} |2m\rangle + \hat{F}^{\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}-1} x_{2m+1} |2m+1\rangle = \\
= \hat{F}^{\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}-1} x_{2m} |0\rangle \otimes |m\rangle + \hat{F}^{\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}-1} x_{2m+1} |1\rangle \otimes |m\rangle = \\
= \sum_{m=0}^{\frac{M}{2}-1} x_{2m} |0\rangle \otimes \hat{F}^{\frac{M}{2}} |m\rangle + \sum_{m=0}^{\frac{M}{2}-1} x_{2m+1} |1\rangle \otimes \hat{F}^{\frac{M}{2}} |m\rangle \,. \qquad (5.19)
\end{aligned}$$

Considering expression (5.17), we have

$$\hat{F}^{\frac{M}{2}} |m\rangle = \sqrt{\frac{2}{M}} \sum_{j=0}^{\frac{M}{2}-1} e^{-i\frac{4\pi}{M}mj} |j\rangle_{inv} \,.$$

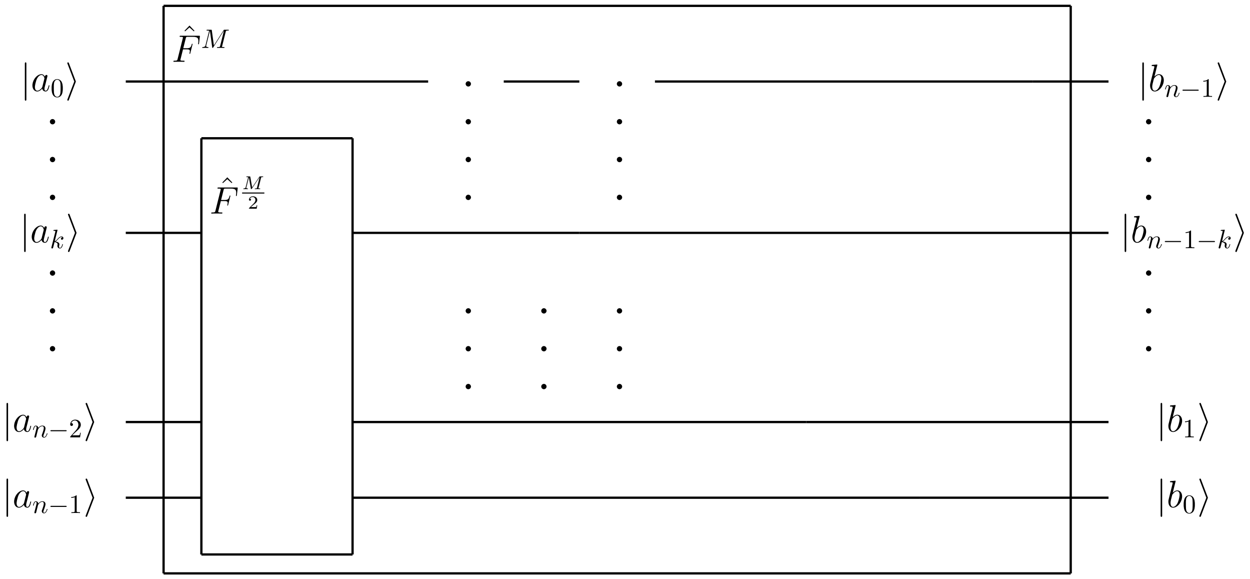

Figure 5.4: Scheme of quantum Fourier transformation based on the fast Fourier transform algorithm. Step 1: $|x\rangle \rightarrow \hat{F}^{\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}-1} x_{2m} |2m\rangle + \hat{F}^{\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}-1} x_{2m+1} |2m+1\rangle$

Thus, for (5.19), we have

$$
\begin{aligned}
|x\rangle \rightarrow \sum_{m=0}^{\frac{M}{2}-1} x_{2m} |0\rangle \otimes \hat{F}^{\frac{M}{2}} |m\rangle + \sum_{m=0}^{\frac{M}{2}-1} x_{2m+1} |1\rangle \otimes \hat{F}^{\frac{M}{2}} |m\rangle = \\
= \sqrt{\frac{2}{M}} \sum_{j=0}^{\frac{M}{2}-1} e^{-i\frac{4\pi}{M}mj} \sum_{m=0}^{\frac{M}{2}-1} x_{2m} |0\rangle \otimes |j\rangle_{inv} + \\
+ \sqrt{\frac{2}{M}} \sum_{j=0}^{\frac{M}{2}-1} e^{-i\frac{4\pi}{M}mj} \sum_{m=0}^{\frac{M}{2}-1} x_{2m+1} |1\rangle \otimes |j\rangle_{inv} = \\
= \sum_{j=0}^{\frac{M}{2}-1} \left(\sqrt{\frac{2}{M}} \sum_{m=0}^{\frac{M}{2}-1} e^{-i\frac{4\pi}{M}mj} x_{2m} \right) |j\rangle_{inv} + \\
+ \sum_{j=0}^{\frac{M}{2}-1} \left(\sqrt{\frac{2}{M}} \sum_{m=0}^{\frac{M}{2}-1} e^{-i\frac{4\pi}{M}mj} x_{2m+1} \right) \left| \frac{M}{2} + j \right\rangle_{inv} = \\
= \sum_{j=0}^{\frac{M}{2}-1} \tilde{A}_j |j\rangle_{inv} + \sum_{j=0}^{\frac{M}{2}-1} \tilde{B}_j \left| \frac{M}{2} + j \right\rangle_{inv},
\end{aligned}
$$

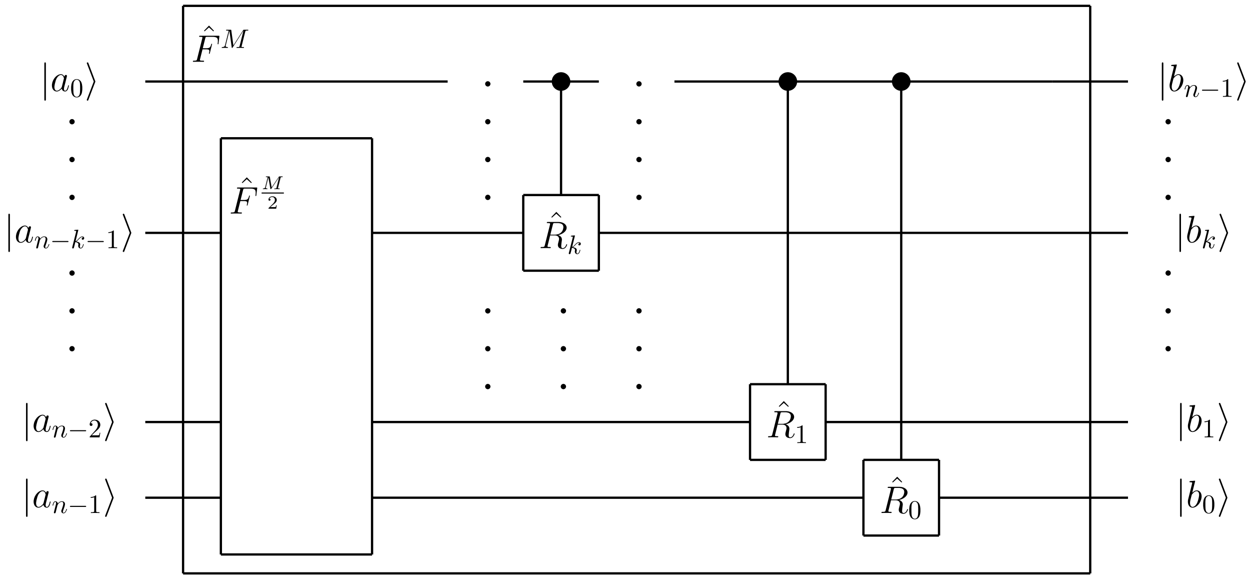

Figure 5.5: Scheme of quantum Fourier transform based on the fast Fourier transform algorithm. Step 2: $|x\rangle \to \hat{F}^{\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}-1} x_{2m} |2m\rangle + \hat{R}\hat{F}^{\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}-1} x_{2m+1}$

where

$$\tilde{A}_j = \sqrt{\frac{2}{M}} \sum_{m=0}^{\frac{M}{2}-1} e^{-i\frac{4\pi}{M}mj} x_{2m}$$

$$\tilde{B}_j = \sqrt{\frac{2}{M}} \sum_{m=0}^{\frac{M}{2}-1} e^{-i\frac{4\pi}{M}mj} x_{2m+1} \tag{5.20}$$

If we now add a phase shift for odd elements, i.e., for those with $a_0^k = 1$, we get the scheme shown in Figure 5.5:

$$\begin{aligned} |x\rangle \to \hat{F}^{\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}-1} x_{2m} |2m\rangle + \hat{R}\hat{F}^{\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}-1} x_{2m+1} |2m+1\rangle = \\ = \sum_{j=0}^{\frac{M}{2}-1} \tilde{A}_j |j\rangle_{inv} + \sum_{j=0}^{\frac{M}{2}-1} \tilde{B}_j \hat{R} \left|\frac{M}{2} + j\right\rangle_{inv}, \\ = \sum_{j=0}^{\frac{M}{2}-1} \tilde{A}_j |j\rangle_{inv} + \sum_{j=0}^{\frac{M}{2}-1} \tilde{C}_j \left|\frac{M}{2} + j\right\rangle_{inv}. \end{aligned} \tag{5.21}$$

Using the expression

$$\hat{R}_l \left| b_l^{(j)} \right\rangle = exp\left(-2\pi i \frac{b_l^{(j)}}{2^{n-l}} \right) \left| b_l^{(j)} \right\rangle$$

we obtain that the operator $\hat{R}$ acts on the state $\left|\frac{M}{2} + j\right\rangle_{inv}$ as follows:

$$\begin{aligned} \hat{R} \left| \frac{M}{2} + j \right\rangle_{inv} &= \hat{R} \left|1\right\rangle \otimes \left|j\right\rangle_{inv} = \\ = \left|1\right\rangle \otimes \hat{R}_0 \left| b_0^{(j)} \right\rangle \otimes \cdots \otimes \hat{R}_{n-2} \left| b_{n-2}^{(j)} \right\rangle &= \\ = \prod_{l=0}^{n-2} exp\left(-2\pi i \frac{2^l b_l^{(j)}}{2^n} \right) \left|1\right\rangle \otimes \left|j\right\rangle_{inv} &= \\ = exp\left(-2\pi i \frac{j}{M} \right) \left| \frac{M}{2} + j \right\rangle_{inv} \end{aligned} \tag{5.22}$$

In deriving (5.22), it was taken into account that $j = b_0^{(j)} + 2^1 b_1^{(j)} + \cdots + 2^{n-2} b_{n-2}^{(j)}$.

Therefore, for $\tilde{C}_j$ in (5.21), we have:

$$\begin{aligned} \tilde{C}_j = \sqrt{\frac{2}{M}} \sum_{m=0}^{\frac{M}{2}-1} e^{-2\pi i \frac{j}{M}} e^{-i \frac{4\pi}{M} mj} x_{2m+1} &= \\ = \sqrt{\frac{2}{M}} \sum_{m=0}^{\frac{M}{2}-1} e^{-i \frac{2\pi}{M} (2m+1) j} x_{2m+1} \end{aligned} \tag{5.23}$$

If we now apply the Hadamard transform to the qubit $|a_0\rangle$, we obtain the scheme shown in Figure 5.6. The initial state then transforms according to the

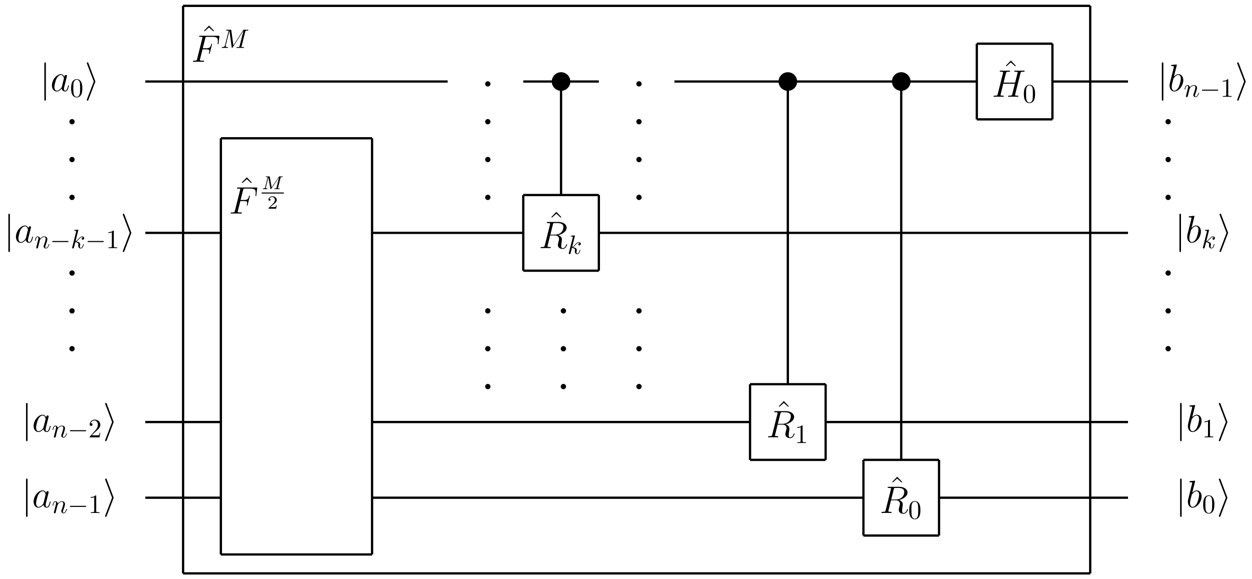

Figure 5.6: Quantum Fourier transform circuit based on the fast Fourier transform algorithm

following law:

$$\begin{aligned}
|x\rangle \to \hat{H}_0 \hat{F}^{\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}-1} x_{2m} |2m\rangle + \hat{H}_0 \hat{R} \hat{F}^{\frac{M}{2}} \sum_{m=0}^{\frac{M}{2}-1} x_{2m+1} = \\
= \sum_{j=0}^{\frac{M}{2}-1} \tilde{A}_j \hat{H} |0\rangle \otimes |j\rangle_{inv} + \sum_{j=0}^{\frac{M}{2}-1} \tilde{C}_j \hat{H} |1\rangle \otimes |j\rangle_{inv} = \\
= \frac{1}{\sqrt{2}} \sum_{j=0}^{\frac{M}{2}-1} \tilde{A}_j \left(|0\rangle + |1\rangle\right) \otimes |j\rangle_{inv} + \frac{1}{\sqrt{2}} \sum_{j=0}^{\frac{M}{2}-1} \tilde{C}_j \left(|0\rangle - |1\rangle\right) \otimes |j\rangle_{inv} = \\
= \sum_{j=0}^{\frac{M}{2}-1} \frac{\tilde{A}_j + \tilde{C}_j}{\sqrt{2}} |0\rangle \otimes |j\rangle_{inv} + \sum_{j=0}^{\frac{M}{2}-1} \frac{\tilde{A}_j - \tilde{C}_j}{\sqrt{2}} |1\rangle \otimes |j\rangle_{inv} = \\
= \sum_{j=0}^{\frac{M}{2}-1} \frac{\tilde{A}_j + \tilde{C}_j}{\sqrt{2}} |j\rangle_{inv} + \sum_{j=0}^{\frac{M}{2}-1} \frac{\tilde{A}_j - \tilde{C}_j}{\sqrt{2}} \left|\frac{M}{2} + j\right\rangle_{inv} . \quad (5.24)
\end{aligned}$$

For the terms in (5.24), considering the equalities (5.20) and (5.23), we have:

$$\frac{\tilde{A}_j + \tilde{C}_j}{\sqrt{2}} = \sqrt{\frac{1}{M}} \sum_{m=0}^{\frac{M}{2}-1} e^{-i\frac{4\pi}{M}mj} x_{2m} + \sqrt{\frac{1}{M}} \sum_{m=0}^{\frac{M}{2}-1} e^{-i\frac{2\pi}{M}(2m+1)j} x_{2m+1} = \\ = \sqrt{\frac{1}{M}} \sum_{m=0}^{M-1} e^{-i\frac{2\pi}{M}mj} x_m \tag{5.25}$$

and

$$\frac{\tilde{A}_j - \tilde{C}_j}{\sqrt{2}} = \sqrt{\frac{1}{M}} \sum_{m=0}^{\frac{M}{2}-1} e^{-i\frac{4\pi}{M}mj} x_{2m} - \sqrt{\frac{1}{M}} \sum_{m=0}^{\frac{M}{2}-1} e^{-i\frac{2\pi}{M}(2m+1)j} x_{2m+1} = \\ = \sqrt{\frac{1}{M}} \sum_{m=0}^{M-1} e^{-i\frac{2\pi}{M}mj} x_m \frac{1+e^{-i\pi m}}{2} - \sqrt{\frac{1}{M}} \sum_{m=0}^{M-1} e^{-i\frac{2\pi}{M}mj} x_m \frac{1-e^{-i\pi m}}{2} = \\ = \sqrt{\frac{1}{M}} \sum_{m=0}^{M-1} e^{-i\frac{2\pi}{M}mj} e^{-i\pi m} x_m = \sqrt{\frac{1}{M}} \sum_{m=0}^{M-1} e^{-i\frac{2\pi}{M}mj} e^{-i\frac{2\pi}{M}m\frac{M}{2}} x_m = \\ = \sqrt{\frac{1}{M}} \sum_{m=0}^{M-1} e^{-i\frac{2\pi}{M}m\left(\frac{M}{2}+j\right)} x_m \tag{5.26}$$

Combining (5.24), (5.25), and (5.26), we finally obtain

$$|x\rangle \rightarrow \sum_{j=0}^{\frac{M}{2}-1} \sqrt{\frac{1}{M}} \sum_{m=0}^{M-1} e^{-i\frac{2\pi}{M}mj} x_m \left|j\right\rangle_{inv} + \\ + \sum_{j=0}^{\frac{M}{2}-1} \sqrt{\frac{1}{M}} \sum_{m=0}^{M-1} e^{-i\frac{2\pi}{M}m\left(\frac{M}{2}+j\right)} x_m \left|\frac{M}{2}+j\right\rangle_{inv} = \\ = \sum_{j=0}^{M-1} \tilde{X}_j^M \left|j\right\rangle_{inv}$$

5.4.2 Finding the Period of Functions Using Quantum Fourier Transform

To determine the period of the function (**??**), we use the scheme presented in Figure 5.7.

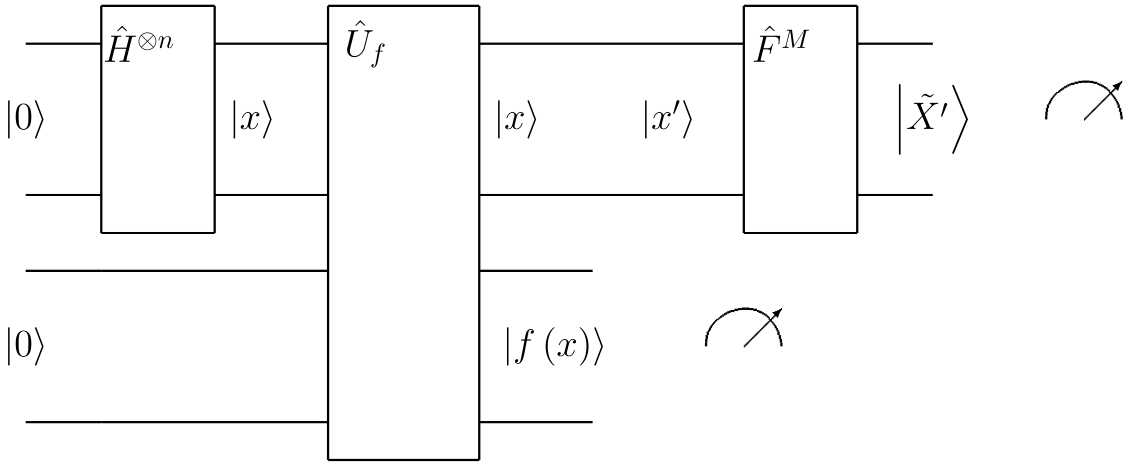

Figure 5.7: Period finding of functions using quantum Fourier transform

The first element is the Hadamard transformation on n qubits, which prepares the initial state in the form:

$$|in\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |0\rangle .$$

After the function computing element $\hat{U}_f$, the state is

$$\hat{U}_f |in\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |f(x)\rangle .$$

After measuring the function value, only those elements for which the function value is equal to the measured value will remain in the list of coordinates. As a result, the input of the element measuring the Fourier transformation is in the state

$$|in'\rangle = \sum_{x'} |x'\rangle ,$$

where all non-zero elements have the same amplitude and follow with a period equal to the period of the studied function. The initial value will have a shift that depends on the experiment (different experiments will have different shifts). Due to lemma 5.3.1, the Fourier image will be the same for different function measurements.

Further, according to lemma 5.3.2 (on periodicity) (see also comment 5.3.1), it follows that the most probable samples (probability maxima) follow with a period related to the original period of the function. Thus, as a result of several experiments, the period of the desired function can be found with the required level of probability (see Figure 5.8).

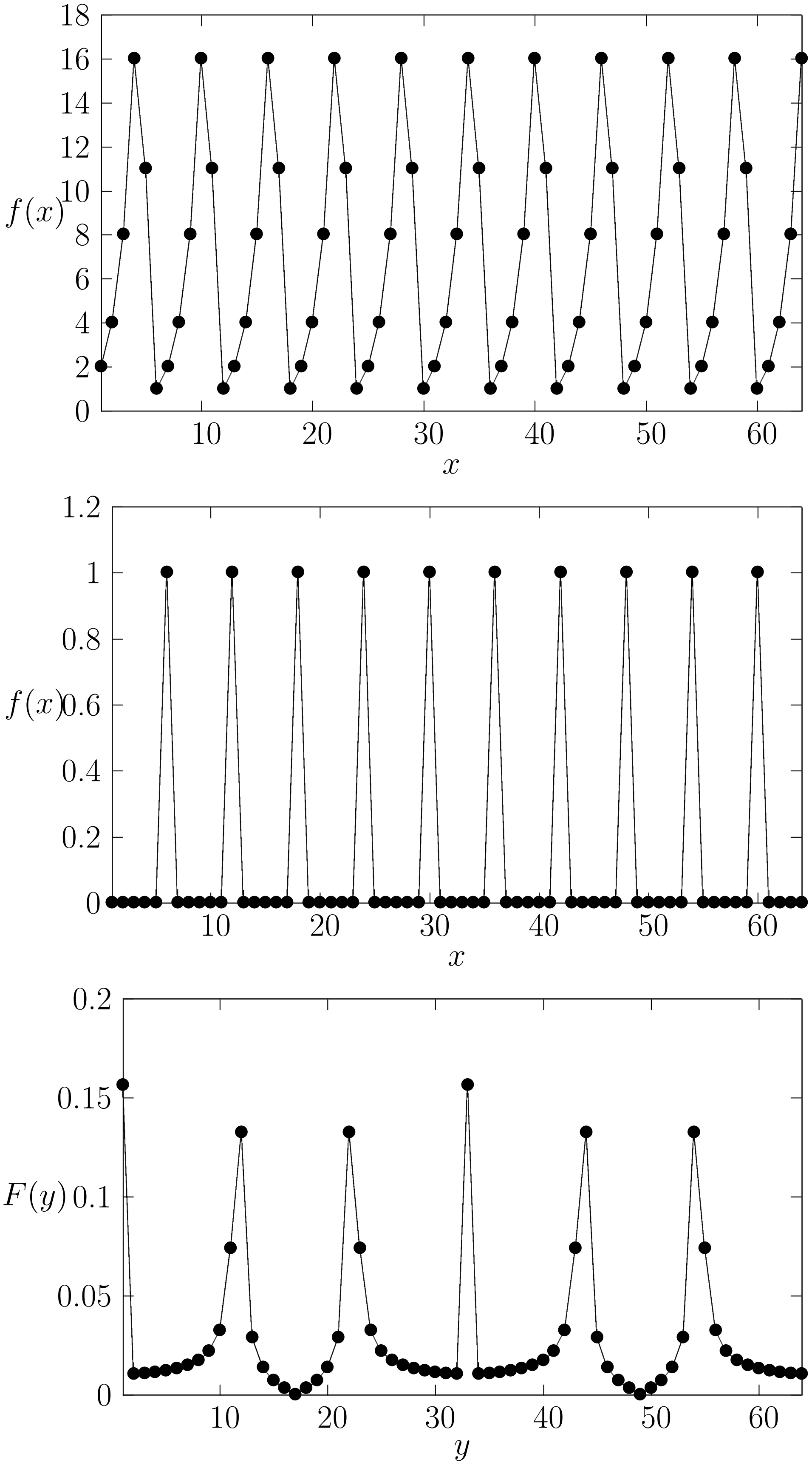

Figure 5.8: Shor's Algorithm. Finding the period of the function $f(x, a) = a^x$ mod N for $a = 2$, $N = 21$ (top graph). The function value 1 repeats with period $r = 6$ (middle graph). Local maxima of the Fourier transform from the middle graph appear with period $\frac{M}{r} \approx 10.67$ (bottom graph). See example 5.4.1

Example 5.4.1. Finding the period of the function $f\left(x\right)=2^{x} \mod 21$ *As an example, consider the problem of finding the period of the function* $f\left(x,a\right)=a^{x} \mod N$ *for* $a=2$*,* $N=21$*, see Figure 5.8.*

The number of samples M *must be a power of two. In our example, we choose* $M=2^{6}=64$ *as the number of samples. Thus, 6 qubits are required for our example.*

The initial state after the Hadamard transformation has the form:

$$\left|in\right\rangle=\frac{1}{8}\sum_{x=0}^{63}\left|x\right\rangle\otimes\left|0\right\rangle,$$

where $\left|x\right\rangle$ *represents the tensor product of 6 qubits that encode the binary representation of the argument of the studied function. For example, for* $x=5_{10}=000101_{2}$ *we have*

$$\left|x\right\rangle=\left|0\right\rangle\otimes\left|0\right\rangle\otimes\left|0\right\rangle\otimes\left|1\right\rangle\otimes\left|0\right\rangle\otimes\left|1\right\rangle$$

After computing the function, we have a state of the form (see the top graph in Figure 5.8)

$$\hat{U}_{f}\left|in\right\rangle=\frac{1}{8}\sum_{x=0}^{63}\left|x\right\rangle\otimes\left|f\left(x\right)\right\rangle=$$
$$=\frac{1}{8}\left(\left|0\right\rangle\otimes\left|2\right\rangle+\left|1\right\rangle\otimes\left|4\right\rangle+\left|2\right\rangle\otimes\left|8\right\rangle+\cdots+\right.$$
$$\left.+\left|62\right\rangle\otimes\left|8\right\rangle+\left|63\right\rangle\otimes\left|16\right\rangle\right). \tag{5.27}$$

If the result of measuring the function was equal to 1*, then from the entire sum (5.27) the terms for which the function value is equal to* 1 *will remain (see the middle graph in Figure 5.8):*

$$\left|in'\right\rangle=\frac{1}{\sqrt{10}}\left(\left|5\right\rangle\otimes\left|1\right\rangle+\left|11\right\rangle\otimes\left|1\right\rangle+\left|17\right\rangle\otimes\left|1\right\rangle+\cdots+\left|60\right\rangle\otimes\left|1\right\rangle\right). \tag{5.28}$$

Expression (5.28) contains 10 terms of the same amplitude, so the normalization factor is $\frac{1}{\sqrt{10}}$*.*

The Fourier transform for the sequence (5.28) is shown in the lower graph Figure 5.8. The most probable values of the Fourier transform result will be the values corresponding to local maxima repeating with the period $\frac{M}{r}\approx 10.67$ *from which the period of the desired function can be found* $r=6$*.*

Chapter 6

Public Key Cryptosystems. Algorithms Using the Discrete Logarithm.

Public key cryptosystems are the main ones currently in use. Moreover, quantum computers can effectively break them. We will further examine these algorithms in more detail.

6.1 Discrete Logarithm

Definition 6.1.1 (Discrete Logarithm)**.** Consider some abelian multiplicative group G and the equation

$$g^x = a \tag{6.1}$$

The solution to this equation, i.e., a non-negative integer x satisfying the equation (6.1), is called a discrete logarithm.

Equation (6.1) does not generally have a solution for any values of a. However, if g is a generating element of G, i.e., $G =< g >$, then a solution always exists and is unique. When discussing discrete logarithms, we will assume that g is chosen such that $G =< g >$.

In applied cryptography, we often deal with a special kind of discrete logarithm in the ring of residues modulo p:

Definition 6.1.2 (Discrete Logarithm in the Ring of Residues Modulo p)**.** The discrete logarithm $ind_g(a) \mod p$ [1] is defined as the smallest number x that satisfies the following equation (if such a number exists):

$$g^x \equiv a \mod p \tag{6.2}$$

[1] From the word **ind**ex - an alternative name for the discrete logarithm

Example 6.1.1. ($ind_3 14 \mod 17$) *Let's solve the problem by method of enumeration [12]:*

$$3^1 \equiv 3 \mod 17,\ 3^2 \equiv 9 \mod 17,\ 3^3 \equiv 10 \mod 17,\ 3^4 \equiv 13 \mod 17,$$
$$3^5 \equiv 5 \mod 17,\ 3^6 \equiv 15 \mod 17,\ 3^7 \equiv 11 \mod 17,\ 3^8 \equiv 16 \mod 17,$$
$$3^9 \equiv 14 \mod 17,\ 3^{10} \equiv 8 \mod 17,\ 3^{11} \equiv 7 \mod 17,\ 3^{12} \equiv 4 \mod 17,$$
$$3^{13} \equiv 12 \mod 17,\ 3^{14} \equiv 2 \mod 17,\ 3^{15} \equiv 6 \mod 17,\ 3^{16} \equiv 1 \mod 17,$$

Thus, it can be seen that $ind_3 14 \mod 17 = 9$, *since* $3^9 \equiv 14 \mod 17$.

The problem of finding the discrete logarithm is a complex task. The fastest known algorithm [2] solves it in time on the order of $O\left(c \cdot exp\left(\log p^{\frac{1}{3}} \log\log p^{\frac{2}{3}}\right)\right)$, where c is some constant, which is why algorithms using discrete logarithms are widely applied in cryptography.

6.2 Diffie-Hellman Protocol (DH)

Assume there are two subscribers, Alice and Bob. They are aware of two numbers g and p, which are not secret.

Alice chooses a random number a and sends Bob the following value

$$A \equiv g^a \mod p. \tag{6.3}$$

Bob computes the following number (using a secret random value b)

$$B \equiv g^b \mod p. \tag{6.4}$$

Alice calculates the key using the number a known only to her

$$K \equiv B^a \mod p \equiv g^{ab} \mod p. \tag{6.5}$$

Bob can obtain the same key value using his secret number b:

$$K \equiv A^b \mod p \equiv g^{ab} \mod p. \tag{6.6}$$

Thus, Alice and Bob obtain the same key, which can later be used for message transmission using symmetric encryption algorithms (such as AES).

Example 6.2.1. (Diffie-Hellman) *Initial data (public information):* $g = 2$, $p = 23$. *Alice selects a random number* $a = 6$ *and calculates using the formula* (6.3) *the number* $A = 18$ *and sends it to Bob. Bob chooses a random number* $b = 9$ *and, using the formula* (6.4), *calculates* $B = 6$ *and sends this number to Alice.*

Alice calculates the key $K = 12$ *using the formula* (6.5). *Bob can obtain the same key value* $K = 12$ *using* (6.6)

The eavesdropper Eve knows the numbers g, p, A, and B. To obtain the key K, Eve would need to obtain one of the secret numbers a or b:

$$a \equiv ind_g\,(A) \mod p,$$

from which the desired value K can be obtained using (6.5).

6.3 ElGamal Scheme

One of the variations of the Diffie-Hellman protocol is the ElGamal scheme. It is important to distinguish between the ElGamal encryption algorithm and the ElGamal digital signature algorithm. The ElGamal digital signature forms the basis of the digital signature standards of the USA (DSA) and Russia (GOST R 34.10-94).

Below we will consider the algorithm in encryption mode.

6.3.1 Key Generation

- A prime number p is generated.
- An integer g is chosen.
- A random integer $x : 1 < x < p$ is chosen.
- $y = g^x \mod p$ is computed.

The public key is the triplet p, g, y. The private key is the number x.

Example 6.3.1 (Key Generation (Elgamal))**.** *Let's choose* $p = 21, g = 10, x = 3$. $y = 10^3 \mod 21 = 13$.

6.3.2 Encryption

The message to be encrypted M must satisfy the condition $0 < M < p$.

- A session key is chosen - a random number $k : 1 < k < p - 1$.
- $a = g^k \mod p$ is computed.
- $b = y^k M \mod p$ is computed.

The pair of numbers (a, b) is considered the ciphertext.

Example 6.3.2 (Encryption (Elgamal))**.** *Suppose we want to encrypt* $M = 6$. *Choose* $k = 7$. $a = 10^7 \mod 21 = 10$, $b = 13^7 \cdot 6 \mod 21 = 15$.

6.3.3 Decryption

Knowing the private key x, we can restore the original message using

$$M = b \cdot \left(a^x\right)^{-1} \mod p, \tag{6.7}$$

indeed because

$$\left(a^x\right)^{-1} = g^{-kx} \mod p$$

we have

$$b \cdot \left(a^x\right)^{-1} = y^k M g^{-kx} = g^{kx} M g^{-kx} = M \mod p.$$

Example 6.3.3 (Decryption (Elgamal))**.** *The encrypted message from ex. 6.3.3* $C = (a = 10, b = 15)$*. Using* (6.7) *we have*

$$M = 15 \cdot 13 \mod 21 = 6$$

where it was used

$$\left(10^3\right)^{-1} \equiv 13 \mod 21,$$

since $13 \cdot 10^3 = 1 \mod 21$*. Thus, we restored the message* $M = 6$ *encrypted in ex. 6.3.3.*

6.4 Two-Dimensional Fourier Transform

6.4.1 Definition

Definition 6.4.1 (Two-Dimensional Fourier Transform)**.** Suppose we have a two-dimensional signal $x\left(k_1, k_2\right)$, where $k_1, k_2 \in \{0, \ldots, M-1\}$. The two-dimensional Fourier transform is called a two-dimensional function

$$\tilde{X}\left(j_1, j_2\right), j_1, j_2 \in \{0, \ldots, M-1\}$$

such that

$$\tilde{X}\left(j_1, j_2\right) = \frac{1}{M} \sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} x\left(k_1, k_2\right) e^{-i\omega\left(k_1 j_1 + k_2 j_2\right)},$$

where

$$\omega = \frac{2\pi}{M}.$$

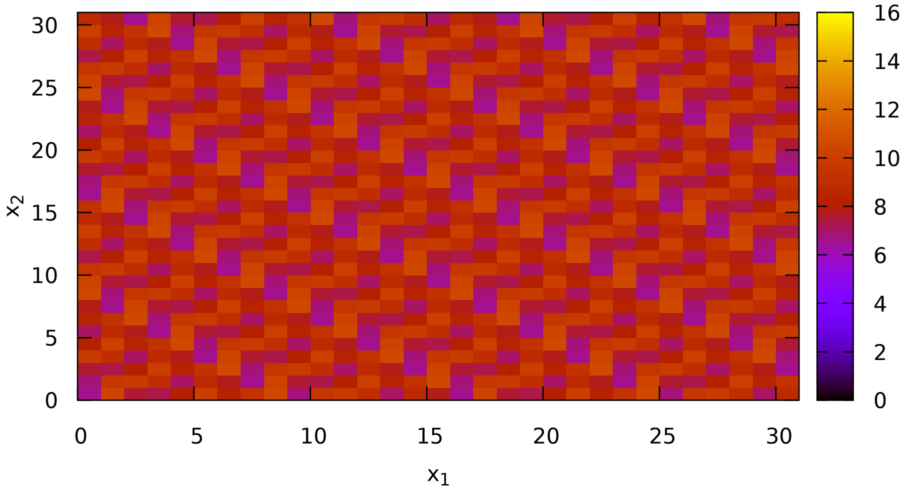

Figure 6.1: Investigated function $f(x_1, x_2) = 14^{x_1} 3^{x_2}$

6.5 Quantum Fourier Transform and Discrete Logarithm

The discrete logarithm (see section 6.1) forms the basis for many modern cryptographic algorithms (see section 6.3, section 6.2). The method proposed by Shor for factorizing integers can also be applied for computing discrete logarithms, potentially breaking these cryptographic algorithms.

Let's set the problem as follows: given the expression

$$b = a^x \mod p,$$

where the numbers a, b and p are given, the number x is unknown and needs to be determined. Analogous to the application of the quantum Fourier transform for factorization (see subsection 5.2.1), we should construct a periodic function whose period will enable us to determine the unknown number x. We'll choose as the function to analyze

$$f(x_1, x_2) = b^{x_1} a^{x_2} = a^{x \cdot x_1} a^{x_2} \mod p \tag{6.8}$$

As an example, we will consider the quantum analogue of solving the problem from example 6.1.1:

Example 6.5.1. ($ind_3 14 \mod 17$) *In our example $p = 17$, $b = 14$ and $a = 3$. The function* (6.8) *looks like*

$$f(x_1, x_2) = b^{x_1} a^{x_2} = 14^{x_1} 3^{x_2}.$$

and is depicted in Figure 6.1.

Both $b = 14$ and $a = 3$ are generators of $(\mathbb{Z}/17\mathbb{Z})^{\times}$. Moreover, $3 \equiv 14^9 \mod 17$. The periods of the depicted function, as can be seen from Figure 6.1, are the following numbers

$$\begin{aligned} t_1 &\equiv 1 \mod 16, \\ t_2 &\equiv 9 \mod 16 \end{aligned} \tag{6.9}$$

Analogous to the solution for factorization, we measure this function. Suppose as a result of measurement we obtained the number $c \in (\mathbb{Z}/p\mathbb{Z})^{\times}$. Given that a is a generator (see definition 8.6.4) of the multiplicative group $(\mathbb{Z}/p\mathbb{Z})^{\times}$ (see definition 8.6.5), $\exists x_0 : c = a^{x_0}$. Thus, considering the small Fermat theorem $a^{p-1} \equiv 1 \mod p$ (Theorem 8.4.1) , we have

$$x_0 \equiv x x_1 + x_2 \mod q,$$

where $q = p - 1$. From this expression it follows that

$$x_2 \equiv x_0 - x x_1 \mod q.$$

Thus, if the function is periodic with respect to the first argument:

$$f(x_1 + t_1, x_2) = f(x_1, x_2),$$

then it will be periodic with respect to the second argument

$$f(x_1, x_2 + t_2) = f(x_1, x_2),$$

and

$$t_2 \equiv x t_1 \mod q. \tag{6.10}$$

6.5.1 Two-Dimensional Fourier Transform and the Period of the Function $f(x_1, x_2)$

Our function of interest is the following:

$$f'(x_1, x_2) = \begin{cases} 1, x x_1 + x_2 \equiv x_0 \mod q \\ 0, x x_1 + x_2 \not\equiv x_0 \mod q \end{cases} \tag{6.11}$$

Example 6.5.2. ($ind_3 14 \mod 17$) *Continuing example 6.5.1 assume that as a result of the function f measurement, we obtained the value $f = 3$. Thus, $f = a^{x_0} = 3^{x_0} \equiv 3 \mod 17$. Therefore, for x_1, x_2, only values corresponding to the observed function value are retained (see Figure 6.2), i.e., $x x_1 + x_2 = x_0 \equiv 1 \mod 16$. With fixed values of x, x_1 and the number of samples $M = q = 16$.*

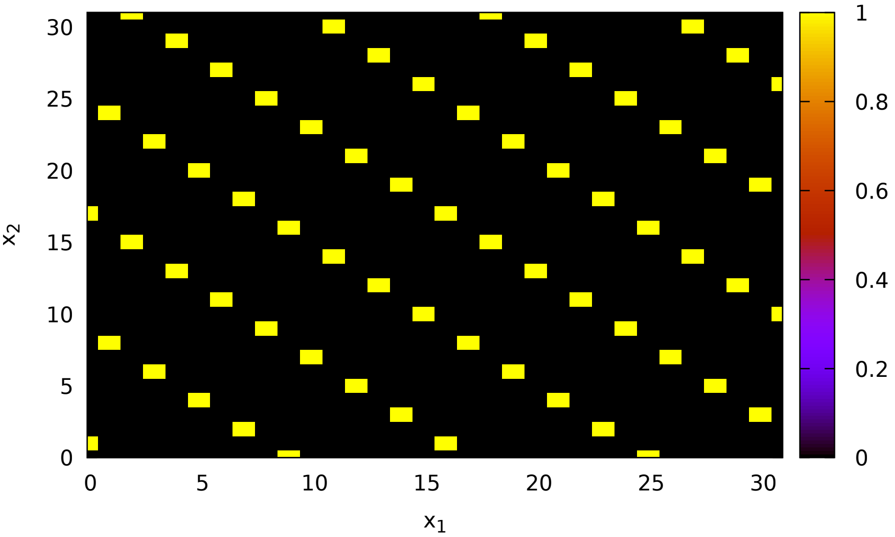

Figure 6.2: The function under study $f(x_1, x_2) = 14^{x_1}3^{x_2}$. Only those pairs x_1, x_2 where $f(x_1, x_2) = 3, x_0 = 1$ are marked.

For the Fourier image $\tilde{f}'$ we have

$$\tilde{f}'(j_1, j_2) = \frac{1}{M}\sum_{x_1=0}^{M-1}\sum_{x_2=0}^{M-1} f'(x_1, x_2)\, e^{-i\omega(x_1 j_1 + x_2 j_2)}, \tag{6.12}$$

where $\omega = \frac{2\pi}{M}$, M is the number of samples.

First of all, consider the case when $M = q$. In this case, there are two options for x_2:

1. $x_2 = x_0 - xx_1$, if $x_0 \geq xx_1$

2. $x_2 = x_0 + q - xx_1$, if $x_0 < xx_1$

Thus,

$$e^{-i\omega x_2 j_2} = e^{-i\omega(x_0 - xx_1 + q)j_2} =$$
$$= e^{-i\omega(x_0 - xx_1)j_2 - i\omega q j_2} = e^{-i\omega(x_0 - xx_1)j_2},$$

i.e., both options coincide and can be reduced to the first: $x_2 = x_0 - xx_1$.

Therefore, continuing (6.12), we get

$$\tilde{f}'\left(j_1, j_2\right) = \frac{1}{M} \sum_{x_1=0}^{M-1} \sum_{x_2=0}^{M-1} f'\left(x_1, x_2\right) e^{-i\omega\left(x_1 j_1 + x_2 j_2\right)} =$$

$$= \frac{1}{M} \sum_{x_1=0}^{M-1} e^{-i\omega\left(x_1 j_1 + \left(x_0 - x x_1\right) j_2\right)} =$$

$$= e^{-i\omega x_0 j_2} \frac{1}{M} \sum_{x_1=0}^{M-1} e^{-i\omega x_1 \left(j_1 - x j_2\right)} = \frac{1}{M} e^{-i\omega x_0 j_2} \sum_{x_1=0}^{M-1} e^{-i\omega x_1 \left(j_1 - x j_2\right)}. \tag{6.13}$$

In the expression (6.13), $\tilde{f}'(j_1, j_2) = e^{-i\omega x_0 j_2} \neq 0$, if condition

$$j_1 \equiv x j_2 \mod M. \tag{6.14}$$

is satisfied. Otherwise, according to the geometric progression formula,

$$\tilde{f}'\left(j_1 \neq x j_2, j_2\right) = e^{-i\omega x_0 j_2} \frac{1}{M} \sum_{x_1=0}^{M-1} e^{-i\omega x_1 \left(j_1 - x j_2\right)} =$$

$$= \frac{e^{-i\omega x_0 j_2}}{M} \frac{e^{-i\omega M \left(j_1 - x j_2\right)} - 1}{e^{-i\omega \left(j_1 - x j_2\right)} - 1} =$$

$$= \frac{e^{-i\omega x_0 j_2}}{M} \frac{e^{-i\frac{2\pi}{M} M \left(j_1 - x j_2\right)} - 1}{e^{-i\omega \left(j_1 - x j_2\right)} - 1} = 0.$$

Thus, to determine the period, we need to find the coordinates (j_1, j_2) of some maximum of the Fourier transform and use the expression

$$x \equiv j_1 j_2^{-1} \mod M, \tag{6.15}$$

which follows from (6.14).

Remark 6.5.1 (On Zero Divisors in $\mathbb{Z}/M\mathbb{Z}$). *If there exists a number y such that*

$$j_2 y \equiv 0 \mod M,$$

then j_2 is called a zero divisor. Obviously,

$$GCD\left(j_2, M\right) \neq 1,$$

therefore by subsection 8.2.2 j_2^{-1} does not exist. In this case, other coordinates (j_1, j_2) should be used.

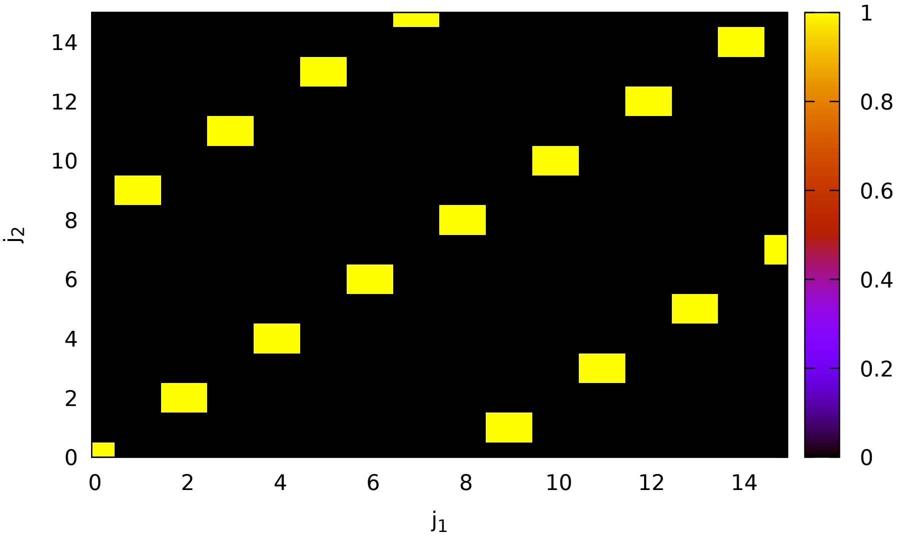

Figure 6.3: Fourier transform of the function from Figure 6.2. Number of samples $M = 16$. Coordinates of the maximum $j_1 = 9$, $j_2 = 1$. The solution of the equation $3^x \equiv 14 \mod 17$ is $x \equiv 9 \cdot 1^{-1} \equiv 9 \mod 16$. Considering that the period is less than the number of samples, we can conclude that $x = 9$

Example 6.5.3. $(ind_3 14 \mod 17)$

The Fourier transform of the function from Figure 6.2 is shown in Figure 6.3. It can be seen that counts at intervals $T_{j_1} = 9$ for j_1 and $T_{j_2} = 1$ for j_2 will be registered with the highest probability. Considering that the number of counts $M = 16$, we can derive the coordinates of the Fourier transform maximum $j_1 = 9$ and $j_2 = 1$. The solution to the equation $3^x \equiv 14 \mod 17$ is, according to (6.15), $x = 9 \cdot 1^{-1} = 9$, which corresponds to the result in example 6.1.1.

The same result can be obtained if we take the point with coordinates $j_1 = 11, j_2 = 3$. Since $3 \cdot 11 = 33 \equiv 1 \mod 16$, we have $j_2^{-1} \equiv 11 \mod 16$, i.e., $x \equiv 11 \cdot 11 \equiv 121 \equiv 9 \mod 16$, which again matches the result in example 6.1.1.

Note that points lying on the diagonal, such as $j_1 = 6, j_2 = 6$, will not yield a correct result because $GCD(6, 16) = 2 \neq 1$.

It is worth noting that the obtained result (6.15) is in direct accordance with lemma 5.3.2 for one-dimensional Fourier transform. The same analog as in comment 5.3.1, which states that even when the number of Fourier transform counts is not equal to q: $M \neq q$, but $M \approx q$, one can still approximately consider (6.15) as valid [7].

Example 6.5.4. $(ind_2 14 \mod 59)$

For example, consider $p = 59$ with the number of counts $M = 64 \approx q = p - 1 = 58$. The generator of the group $\mathbb{F}_{59}$ (see subsection 8.2.3) is $g = 2$, i.e.,

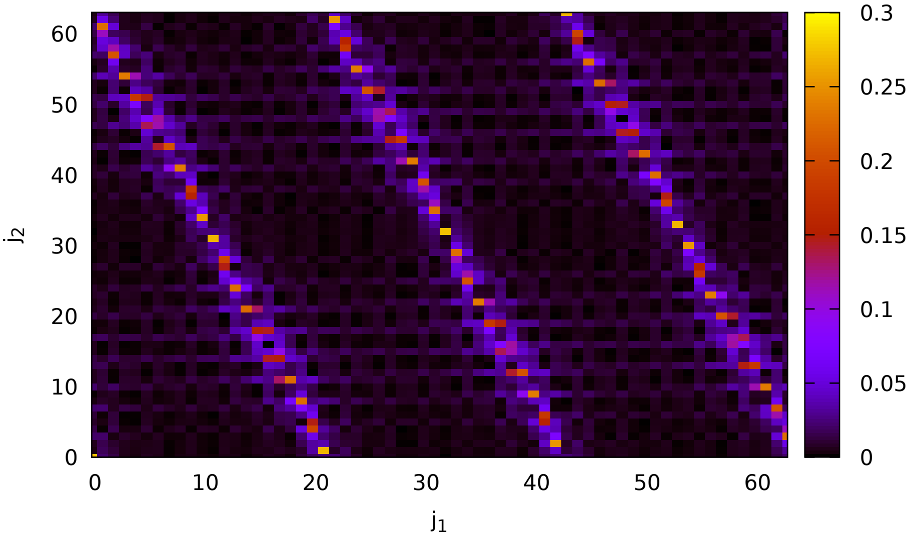

Figure 6.4: Fourier transform of the samples of the function $f'(x_1, x_2)$ Number of samples $M = 64$. The three lower maxima have coordinates $\approx (20, 1), (41, 2.2), (62, 3)$, which give the following estimates for x: $x \approx 20, 18.6, 20.6$, which is close to the actual value $x = 19$

$\mathbb{F}_{59} = \langle 2 \rangle$. *This means that the equation* $2^x \equiv b \mod 59$ *will have a solution for any* b*, in particular* $x = 19$ *is a solution for*

$$2^x \equiv 14 \mod 59.$$

The examined function has the form

$$f(x_1, x_2) = 14^{x_1} 2^{x_2} \mod 59,$$

Suppose that $x_0 = 50$*, i.e., a function value* $f(x_1, x_2) = 2^{x_0} = 2^{50} \equiv 3 \mod 59$ *was registered.*

As mentioned earlier, the number of Fourier transform counts is $M = 64$*. Note that* $q = p - 1 = 58 \not\equiv 0 \mod 64$.

The Fourier image of function

$$f'(x_1, x_2) = \begin{cases} 1, & \textit{if } 14^{x_1} 2^{x_2} \equiv 3 \mod 59 \\ 0, & \textit{if } 14^{x_1} 2^{x_2} \not\equiv 3 \mod 59 \end{cases}$$

is shown on Figure 6.4. Three lower maxima have coordinates

$$(j_1, j_2) \approx (20, 1), (41, 2.2), (62, 3),$$

which gives estimates for x*:* $x \approx 20, 18.6, 20.6$*, close to the actual value* $x = 19$.

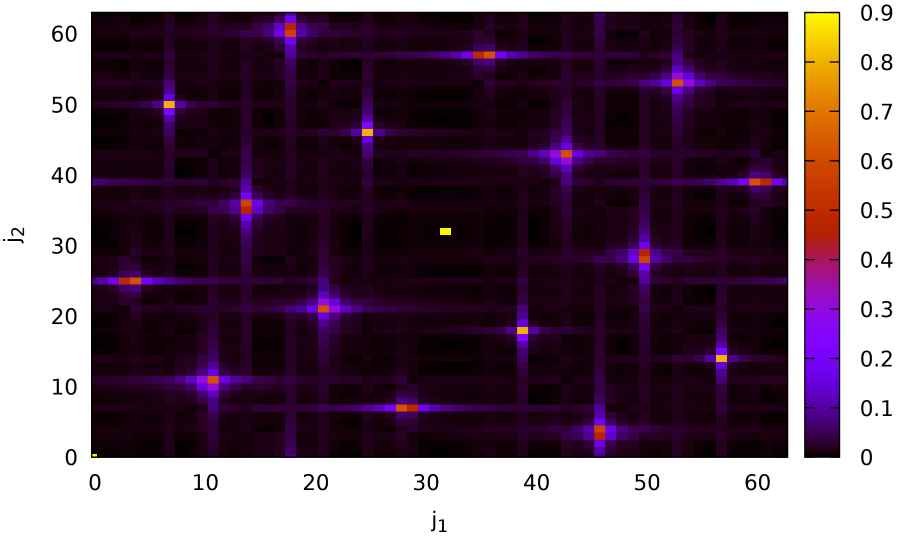

Figure 6.5: Fourier transform of the samples of the function $f'(x_1, x_2)$ Number of samples $M = 64$. Coordinates of the maximum $j_1 \approx 46$, $j_2 \approx 3.5$. The solution of the equation $3^x \equiv 14 \mod 19$ is $x = 13 \approx \frac{46}{3.5} \approx 13.14$

Example 6.5.5. ($ind_3 14 \mod 19$)

As an example consider determining the x such that

$$3^x \equiv 14 \mod 19.$$

The function to analyze is

$$f(x_1, x_2) = 14^{x_1} 3^{x_2} \mod 19,$$

Assuming $x_0 = 1$, i.e., the function value $f(x_1, x_2) = 3$ is registered.

The number of Fourier transform samples is $M = 64$. Note that $18 \not\equiv 0 \mod 64$.

The Fourier image of function

$$f'(x_1, x_2) = \begin{cases} 1, & \textit{if } 14^{x_1} 3^{x_2} \equiv 3 \mod 19 \\ 0, & \textit{if } 14^{x_1} 3^{x_2} \not\equiv 3 \mod 19 \end{cases}$$

is shown on Figure 6.5. The lowest maximum has coordinates $j_1 = 46, j_2 = 3.5$ hence we have an estimate

$$x \approx \frac{46}{3.5} \approx 13.14.$$

It's noteworthy that the solution to the desired equation $x = 13$ matches the found approximate solution.

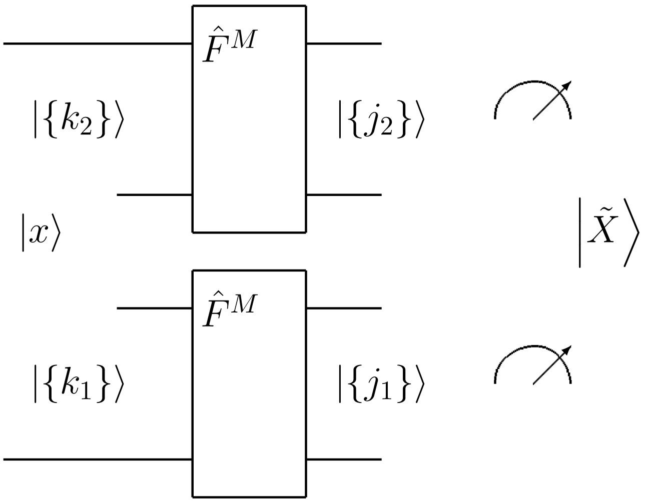

Figure 6.6: Scheme of two-dimensional quantum Fourier transform. The input signal is $|x\rangle = \sum_{k_1,k_2=0}^{M-1} x\left(k_1,k_2\right)|k\rangle_1 \otimes |k\rangle_2$. The output signal $\left|\tilde{X}\right\rangle = \sum_{k_1,k_2=0}^{M-1} \tilde{X}\left(j_1,j_2\right)|j_1\rangle \otimes |j_2\rangle$ is the two-dimensional Fourier transform of the original $|x\rangle$

6.5.2 Two-Dimensional Quantum Fourier Transform

To determine the periods of a two-argument function, one can use a two-dimensional Fourier transform, which can be constructed from blocks implementing one-dimensional Fourier transform, as shown in Figure 6.6. For analyzing this scheme, let's consider the trivial case when the input is (see also (5.16))

$$|x\rangle = |x\rangle_1 \otimes |x\rangle_2\,,$$

$$|x\rangle_{1,2} = \sum_{k_{1,2}=0}^{M-1} x_{k_{1,2}}^{(1,2)}\left|k_{1,2}\right\rangle.$$

Given that the output is

$$\left|\tilde{X}\right\rangle = \left|\tilde{X}_1\right\rangle \otimes \left|\tilde{X}_2\right\rangle,$$

where

$$\left|\tilde{X}_{1,2}\right\rangle = \sum_{j_{1,2}=0}^{M-1} \tilde{X}_{j_{1,2}}^{(1,2)}\left|j_{1,2}\right\rangle$$

and according to (5.18)

$$\tilde{X}_{j_{1,2}}^{(1,2)} = \frac{1}{\sqrt{M}} \sum_{k_{1,2}=0}^{M-1} e^{-i\omega_{1,2}k_{1,2}j_{1,2}} x_{k_{1,2}}^{(1,2)}.$$

we obtain

$$\left|\tilde{X}\right\rangle = \left|\tilde{X}_1\right\rangle \otimes \left|\tilde{X}_2\right\rangle =$$

$$= \sum_{j_1=0}^{M-1} \sum_{j_2=0}^{M-1} \tilde{X}_{j_1}^{(1)} \tilde{X}_{j_2}^{(2)} \left|j_1\right\rangle \otimes \left|j_2\right\rangle =$$

$$= \sum_{j_1=0}^{M-1} \sum_{j_2=0}^{M-1} \tilde{X}_{j_1,j_2} \left|j_1\right\rangle \otimes \left|j_2\right\rangle ,$$

where

$$\tilde{X}_{j_1,j_2} = \frac{1}{\left(\sqrt{M}\right)^2} \sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} e^{-i\omega(k_1 j_1 + k_2 j_2)} x_{k_1}^{(1)} x_{k_2}^{(2)} =$$

$$= \frac{1}{\left(\sqrt{M}\right)^2} \sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} e^{-i\omega(k_1 j_1 + k_2 j_2)} x_{k_1,k_2}$$

which, according to definition 6.4.1, *two-dimensional Fourier transform* is a two-dimensional Fourier transform of the original two-dimensional signal

$$\left|x\right\rangle = \sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} x_{k_1}^{(1)} x_{k_2}^{(2)} \left|k_1\right\rangle \otimes \left|k_2\right\rangle = \sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} x_{k_1,k_2} \left|k_1\right\rangle \otimes \left|k_2\right\rangle .$$

Thus, using the scheme shown in Figure 6.7, one can determine the coordinates of the maxima of the two-dimensional Fourier transform j_1, j_2 and then use (6.15) to determine the desired x.

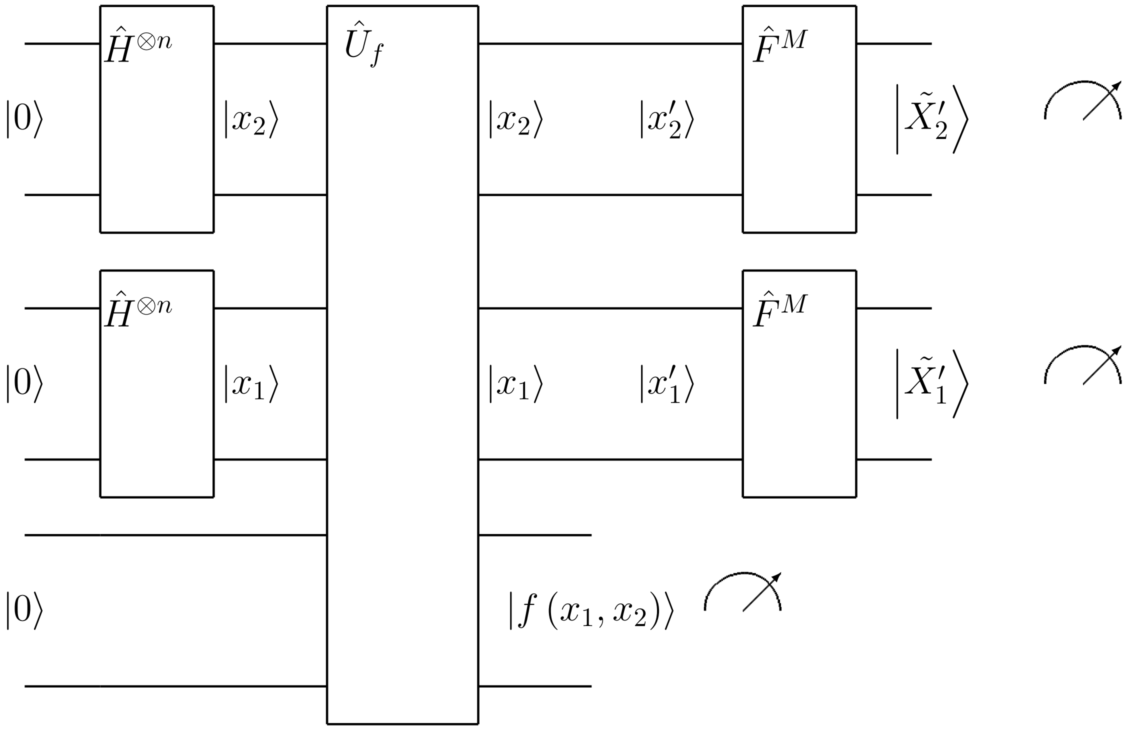

Figure 6.7: Period determination of two-argument functions using quantum Fourier transform

Chapter 7

Elliptic Cryptography

Currently, a special type of discrete logarithmization known as elliptic curve cryptography is particularly popular. The reason for this popularity is the fact that there are currently no effective algorithms for breaking elliptic curve cryptography. The most effective classical algorithm, Pollard's ρ-method, has an efficiency of $O\left(\sqrt{N}\right)$ [6]. This advantageously distinguishes these cryptographic algorithms from RSA and conventional discrete logarithmization, for which sufficiently effective classical algorithms for breaking exist, necessitating the use of longer numerical sequences for these algorithms, for example, 4096 bits for RSA.

7.1 Elliptic Cryptography

7.1.1 Elliptic Curves over the Field $\mathbb{R}$

In elliptic cryptography, certain sets of objects that form a group are considered (see section 8.6). As such a set, we will consider points belonging to a certain curve (see Figure 7.1):

$$E : y^2 = x^3 + ax + b,$$

where the coefficients a, b must satisfy the following relation

$$4a^3 + 27b^2 \neq 0,$$

in this case, the cubic equation $x^3 + ax + b = 0$ will have 3 distinct real roots [11].

According to definition 8.6.1, there is a binary operation defined for the points on the curve that maps two points p, q with coordinates (p_x, p_y) and (q_x, q_y), respectively, to a third point r with coordinates (r_x, r_y). We will call this operation addition:

$$p + q = r.$$

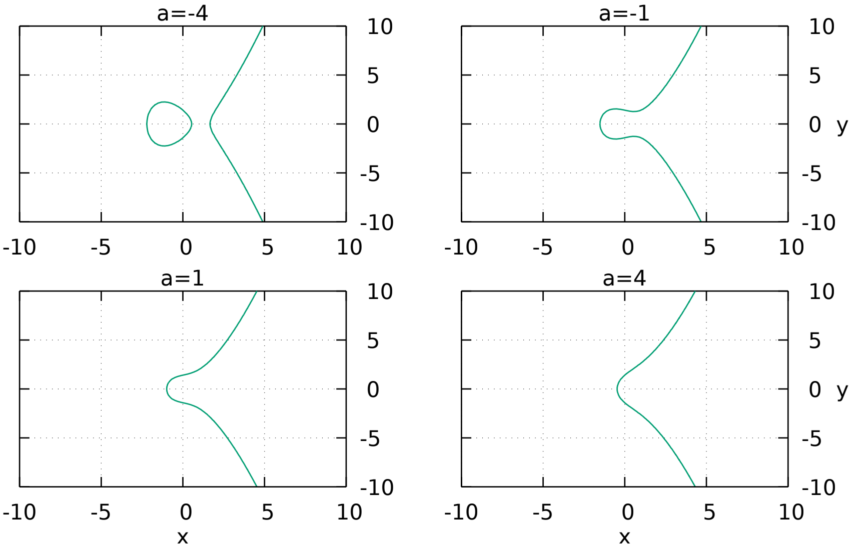

Figure 7.1: Elliptic curve $y^2 = x^3 + ax + 2$ over the field of real numbers $\mathbb{R}$ for various values of parameter a

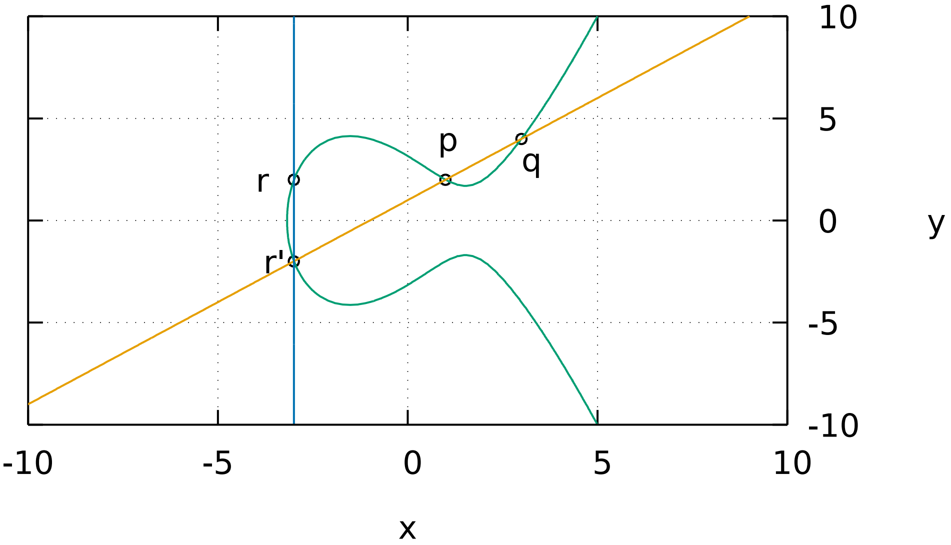

Figure 7.2: Elliptic curve $y^2 = x^3 - 7x + 10$ over the field of real numbers $\mathbb{R}$. Addition of two points $p(1, 2)$ and $q(3, 4)$. The line passing through these points intersects the curve at a third point $r'(-3, -2)$. The point symmetric to r' with respect to the curve, $r(-3, 2)$, is the sum of the original two: $p + q = r$

There is a simple geometric interpretation of the addition operation (see Figure 7.2). Suppose there are 2 points on the curve that we want to add: p and q with coordinates $(x_p, y_p), (x_q, y_q)$ respectively. If $x_p \neq x_q$, a line through these points can be drawn, which has a slope

$$m = \frac{y_p - y_q}{x_p - x_q}.$$

and intersects the curve at point r'. If this point has coordinates $(x_{r'}, y_{r'})$, then due to the fact that it lies on a line with slope m:

$$m = \frac{y_{r'} - y_p}{x_{r'} - x_p},$$

consequently

$$y_{r'} = y_p + m\left(x_{r'} - x_p\right).$$

This point must belong to the curve, i.e.

$$y_{r'}^2 = \left(y_p + m\left(x_{r'} - x_p\right)\right)^2 = x_{r'}^3 + ax_{r'} + b$$

which can be rewritten

$$x_{r'}^3 - m^2 x_{r'}^2 + \cdots = 0.$$

The equation $x^3 - m^2x^2 + \cdots = 0$ has 3 roots: $x_p, x_q, x_{r'}$, i.e. it can also be rewritten as

$$(x - x_{r'})(x - x_p)(x - x_q) = x^3 - (x_{r'} + x_p + x_q)x^2 + \cdots = 0.$$

Thus

$$x_{r'} + x_p + x_q = m^2.$$

Consequently

$$\begin{aligned} x_{r'} &= m^2 - x_p - x_q, \\ y_{r'} &= y_p + m\left(x_{r'} - x_p\right), \end{aligned}$$

Reflecting this point with respect to the X-axis, we obtain the final point r, which we will call the sum of the original points (see Figure 7.2). The coordinates of this point x_r, y_r can be obtained using the following formulas

$$\begin{aligned} x_r &= m^2 - x_p - x_q, \\ y_r &= -y_p + m\left(x_p - x_r\right). \end{aligned} \tag{7.1}$$

In the case of $x_p = x_q$, there are two possibilities:

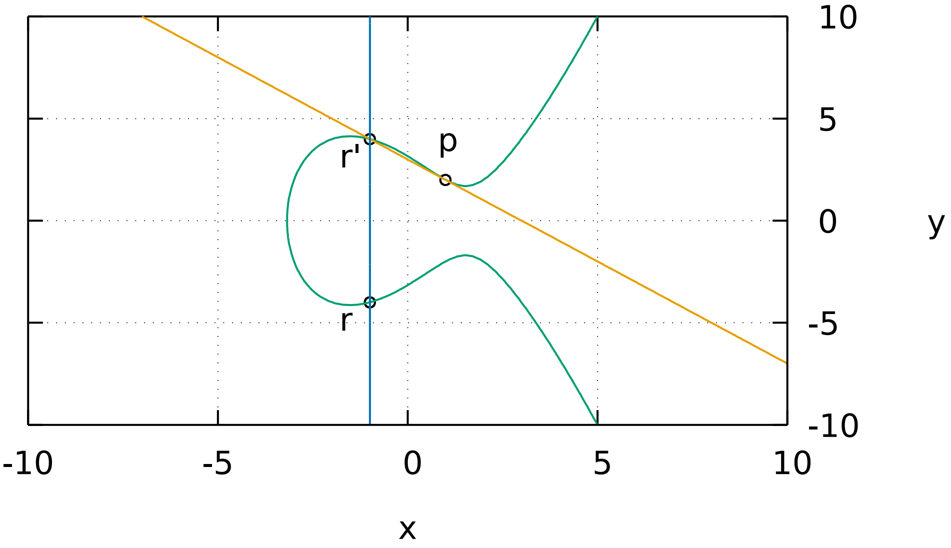

Figure 7.3: Elliptic curve $y^2 = x^3 - 7x + 10$ over the field of real numbers $\mathbb{R}$. Addition of two points with the same coordinates $p(1, 2)$: $2p = r(-1, -4)$

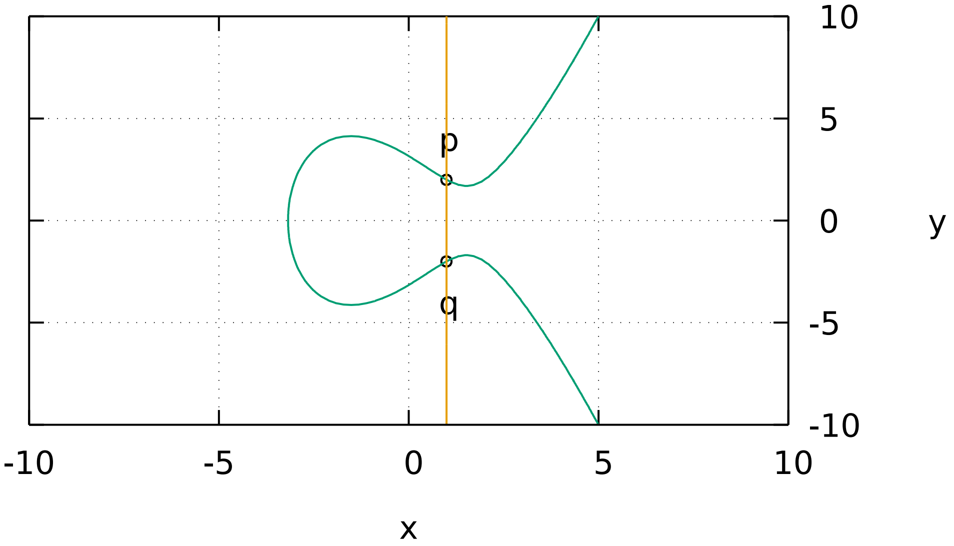

Figure 7.4: Elliptic curve $y^2 = x^3 - 7x + 10$ over the field of real numbers $\mathbb{R}$. Addition of two points $p(1, 2)$ and $q(1, -2)$. The line passing through these points does not intersect the curve. The result of the addition is the zero element: $p + q = 0$, i.e., $q = -p$

1. $y_p = y_q$ (see Figure 7.3). In the case when the points on the curve approach each other, the line through them tends to the tangent. The coefficient m can be found using the formula: $m = \frac{dy}{dx}$. Given $2ydy = 3x^2dx + adx$, we have $m = \frac{dy}{dx} = \frac{3x^2+a}{2y}$. Further calculation is carried out using formulas (7.1).

2. $y_p \neq y_q$ (see Figure 7.4). In this case, due to the symmetry of the curve with respect to the X-axis, only one option is possible: $y_p = y_q$, and the line passing through these two points does not intersect the curve at a third point. For this case, an additional point 0 is introduced where the line through the two points goes. Thus, in this case, we have $p + q = 0, q = -p$.

Thus, we can define an elliptic curve over the field of real numbers $\mathbb{R}$ as the following set of points

$$E(\mathbb{R}) = \left\{(x, y) \in \mathbb{R} \times \mathbb{R} : y^2 = x^3 + ax + b, \right\} \cup \{0\}. \tag{7.2}$$

where $a, b \in \mathbb{R}$.

For these points, a binary operation is defined, which we called addition. The set has a zero element, and for each element, an inverse element can be defined. It can be proven that the introduced operation is associative: $(a+b)+c = a+(b+c)$ [11]. Thus, the set $E(\mathbb{R})$ forms a group with respect to the addition operation. Considering the obvious equality $p + q = q + p$, this group will be commutative, i.e., Abelian (see definition 8.6.3).

Remark 7.1.1 (About the Addition Operation)**.** *The introduced addition operation for points on an elliptic curve may initially seem unnatural; why not use more obvious point addition operations on the plane, for example, vector addition rules. In this case, if $p, q \in E(\mathbb{R})$, it may well be that $p+q \notin E(\mathbb{R})$, which would violate the main group property - closure. Meanwhile, for the binary operation we defined (7.1), all group properties are valid, and accordingly, all resulting properties, such as Lagrange's theorem (see theorem 8.6.2) , which can be applied to the objects we introduced.*

7.1.2 Elliptic Curves over the Field $\mathbb{F}_p$

The set (7.2) along with the addition operation (7.1) can be defined over an arbitrary field (see definition 8.7.1), i.e. not only over $\mathbb{R}$. From the cryptography perspective, the field $\mathbb{F}_p$ is of special interest (see subsection 8.2.3). One can define a set of elements of an elliptic curve over the field $\mathbb{F}_p$ analogous to the expression (8.2.3):

$$E(\mathbb{F}_p) = \left\{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p : y^2 \equiv x^3 + ax + b, \right\} \cup \{0\}. \tag{7.3}$$

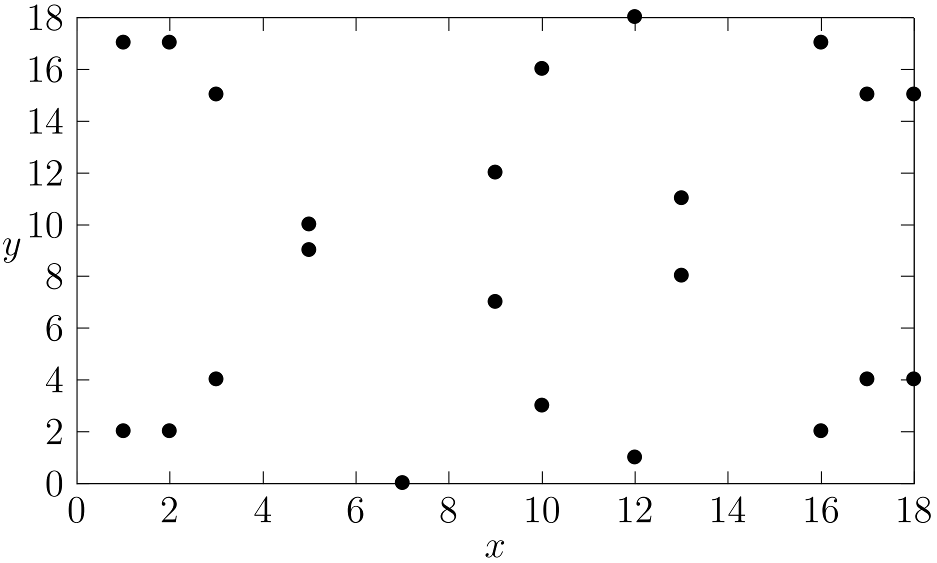

Figure 7.5: Elliptic curve $y^2 = x^3 - 7x + 10$ over the field $\mathbb{F}_{19}$

where $a, b \in \mathbb{F}_p$.

Definition 7.1.1 (Order of an Elliptic Curve)**.** The number of points on an elliptic curve E is called its order and is denoted as $|E|$.

In Figure 7.5, such a set is shown for the field $\mathbb{F}_{19}$, i.e., $p = 19$. The equation of the curve is $y^2 \equiv x^3 - 7x + 10 \mod 19$.

For each point a with coordinates x_a, y_a, an inverse element $-a$ is defined with coordinates $x_{-a} = x_a, y_{-a} \equiv -y_a \mod p$.

For points on this curve, the following addition rule is defined $a+b = c, b \neq -a$

$$\begin{aligned} x_c &\equiv m^2 - x_a - x_b \mod p, \\ y_c &\equiv -y_a + m\left(x_a - x_c\right) \mod p, \end{aligned} \tag{7.4}$$

where $(x_{a,b,c}, y_{a,b,c})$ are the coordinates of the points a, b, and c respectively. For the coefficient m, the following relations are used:

$$\begin{aligned} m &= \left(y_a - y_b\right)\left(x_a - x_b\right)^{-1} \mod p, \text{ if } x_a \neq x_b \\ m &= \left(3x^2 + a\right)\left(2y\right)^{-1} \mod p, \text{ if } x_a = x_b. \end{aligned}$$

Obviously, if $b = -a$, then $a + b = a + (-a) = 0$.

7.1.3 Scalar Multiplication and Discrete Logarithm

Suppose n is a natural number $n \in \mathbb{N}$ and $a \in E(\mathbb{F}_p)$. Define the scalar multiplication as follows

$$n \cdot a = a + a + \ldots a = \sum_{k=1}^{n} a$$

A naive implementation requires $O(n)$ addition operations, but using the "divide and conquer" paradigm (see section 8.9) can reduce it to $O(\log n)$ addition operations.

Example 7.1.1 (Scalar Multiplication)**.** *Consider the elliptic curve*

$$E(\mathbb{F}_{19}) = \{(x, y) \in \mathbb{F}_{19} \times \mathbb{F}_{19} : y^2 \equiv x^3 - 7x + 10\} \cup \{0\},$$

depicted in Figure 7.5. Choose the point $p = (13, 8)$*, then*

$$\begin{aligned}
0 \cdot p &= 0, \\
1 \cdot p = p &= (13, 8), \\
2 \cdot p = p + p &= (16, 7), \\
3 \cdot p = 2 \cdot p + p &= (18, 5), \\
4 \cdot p = 3 \cdot p + p &= (12, 1), \\
5 \cdot p = 4 \cdot p + p &= (5, 10), \\
6 \cdot p = 5 \cdot p + p &= (7, 0), \\
7 \cdot p = 6 \cdot p + p &= (5, 9), \\
8 \cdot p = 7 \cdot p + p &= (12, 18), \\
9 \cdot p = 8 \cdot p + p &= (18, 4), \\
10 \cdot p = 9 \cdot p + p &= (16, 2), \\
11 \cdot p = 10 \cdot p + p &= (13, 11), \\
12 \cdot p = 11 \cdot p + p &= 0
\end{aligned}$$

As can be seen from example 7.1.1, each element of the elliptic curve is a generator of some cyclic subgroup. Meanwhile, the entire group of points on the elliptic curve is not necessarily cyclic (TBD). From the other side, to form a discrete logarithm problem, we need exactly a cyclic group. Thus, for a given elliptic curve, one first computes its order (see def. 7.1.1), for this there is an efficient Schoof's algorithm [8]. Then a prime divisor of the found order is determined and

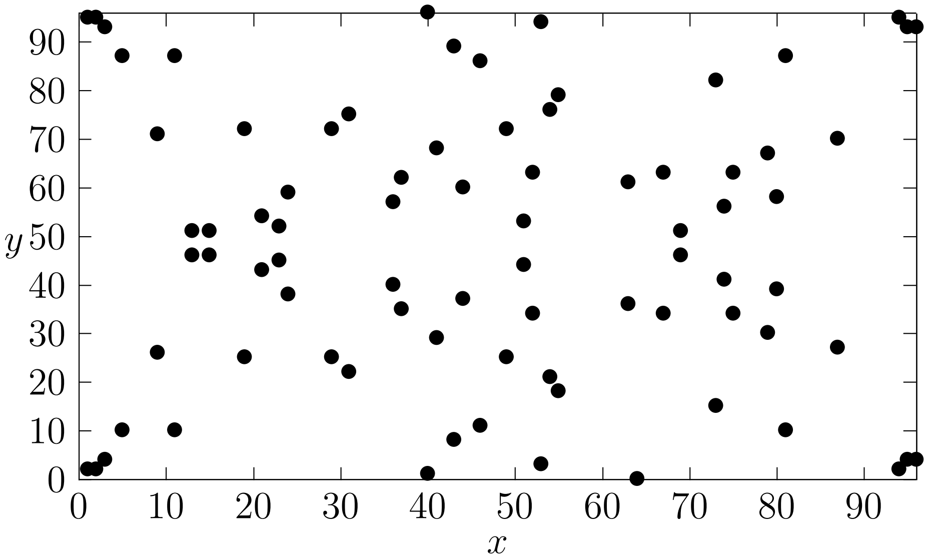

Figure 7.6: Elliptic curve $y^2 = x^3 - 7x + 10$ over the field $\mathbb{F}_{97}$

a point that is a generator of the subgroup of the selected order is sought. To do this, the following fact is used. For any point $g \in E$, the following relation holds

$$Ng = 0,$$

where $N = |E|$ -the order (number of points) of the elliptic curve. Suppose p is some prime divisor of the number N:

$$N = hp$$

then

$$Ng = p\,(hg) = 0.$$

Thus, if $hg \neq 0$, then the point $g' = hg$ will be the generator of a cyclic subgroup of order p.

Remark 7.1.2. *It makes sense to choose the order of the subgroup to be a prime number since if the group order is prime, then by Lagrange's theorem (see theorem 8.6.2) , it only has trivial subgroups - the group itself and the subgroup consisting of the unit element. Thus, due to hg* $\neq 0$*, only the group itself remains, which is cyclic with order p.*

Example 7.1.2 (Choosing the Base Point)**.** *Consider the elliptic curve*

$$E = E\,(\mathbb{F}_{97}) = \{(x, y) \in \mathbb{F}_{97} \times \mathbb{F}_{97} : y^2 \equiv x^3 - 7x + 10\} \cup \{0\},$$

depicted in Figure 7.6. The order of this curve:

$$N = |E| = 82.$$

The number 82 has 2 divisors: 41 and 2. Thus, there is a cyclic subgroup of order 41, i.e., $h = 2$.

Take the point $g = (1, 2) \in E$, its order: $|\langle g \rangle| = 82$, i.e. this point is not suitable. Calculate

$$g' = hg = 2g = (96, 93) \neq 0.$$

At the same time $|\langle g' \rangle| = 41$, so we found the required base point.

If there are two points on the curve $a, b \in E(\mathbb{F}_p)$, then it makes sense to ask about the existence of such an $x \in \mathbb{N}$:

$$x \cdot a = b$$

this problem is called the discrete logarithm on elliptic curves.

7.2 ECDH Algorithm

The ECDH algorithm is a modification of the Diffie-Hellman algorithm (see section 6.2) for elliptic curves. The Diffie-Hellman protocol is a key exchange protocol. In our case, the following elliptic curve parameters are published: (p, a, b, g, n, h), where p, a, b define the curve

$$E(\mathbb{F}_p) = \{(x, y) : y^2 \equiv x^3 + ax + b \mod p\} \cup \{0\},$$

g is the base point of order n: $|\langle g \rangle| = n$, h is the cofactor of the group $\langle g \rangle$, i.e., the order of the curve (see def. 7.1.1) $|E| = nh$.

Alice chooses a private key $d_a \in \{1, \ldots, n-1\}$ and forms a public key $A = d_a g$. Bob also forms private $d_b \in \{1, \ldots, n-1\}$ and public $B = d_b g$ keys. Alice and Bob exchange these keys. Then each of them computes the actual key by the rule $K = d_a B = d_b A$.

Example 7.2.1 (ECDH Algorithm)**.** *Take the curve and base point from ex. 7.1.2. Thus,*

$$(p, a, b, g, n, h) = (97, -7, 10, (96, 93), 41, 2)$$

Alice chooses $d_a = 5$, i.e., $A = (37, 35)$. Bob chooses $d_b = 15$, thus $B = (15, 51)$. The key for Alice $K = d_a B = (46, 11)$ and the key for Bob $K = d_b A = (46, 11)$ match.

7.3 Shor's Algorithm and the Discrete Logarithm on Elliptic Curves

Consider the elliptic curve

$$E\left(\mathbb{F}\right) = \left\{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p, y^2 = x^3 + ax + b, \right\}$$

with a given base point $g \in E\left(\mathbb{F}_p\right)$ such that:

$$ng = 0.$$

The task is to solve the following: for the given $q \in E\left(\mathbb{F}_p\right)$ find an x such that

$$xg = q \mod n \tag{7.5}$$

Consider the following auxiliary function

$$f\left(x_1, x_2\right) = x_1 q + x_2 g = \left(x x_1 + x_2\right) g, \tag{7.6}$$

where $q, g \in E\left(\mathbb{F}_p\right)$ and are taken from the conditions of our problem (7.5). This function is analogous to (**??**) used in solving the discrete logarithm problem. Then we measure this function. The result of this measurement is some point $c \in E\left(\mathbb{F}_p\right)$. Moreover, from (7.6) it follows that $c \in \langle g \rangle$, i.e., $\exists x_0$ such that $c = x_0 g$.

Thus, by analogy with (**??**), we compose the following function

$$f'\left(x_1, x_2\right) = \begin{cases} 1, x x_1 + x_2 \equiv x_0 \mod n \\ 0, x x_1 + x_2 \not\equiv x_0 \mod n \end{cases} \tag{7.7}$$

The coordinates (j_1, j_2) of the maximum of the Fourier image $\tilde{f}'$ provide according to the formula (**??**) some value of the desired number x. In our case, almost always $n \neq M$ so we can only use an approximate estimate

$$x \approx \frac{j_1}{j_2}.$$

Example 7.3.1 (Discrete Logarithm on an Elliptic Curve)**.** *Consider the problem from example 7.2.1 The curve and base point (see example 7.1.2) are as follows*

$$(p, a, b, g, n, h) = (97, -7, 10, (96, 93), 41, 2)$$

Suppose we know Alice's public key

$$A = (37, 35)$$

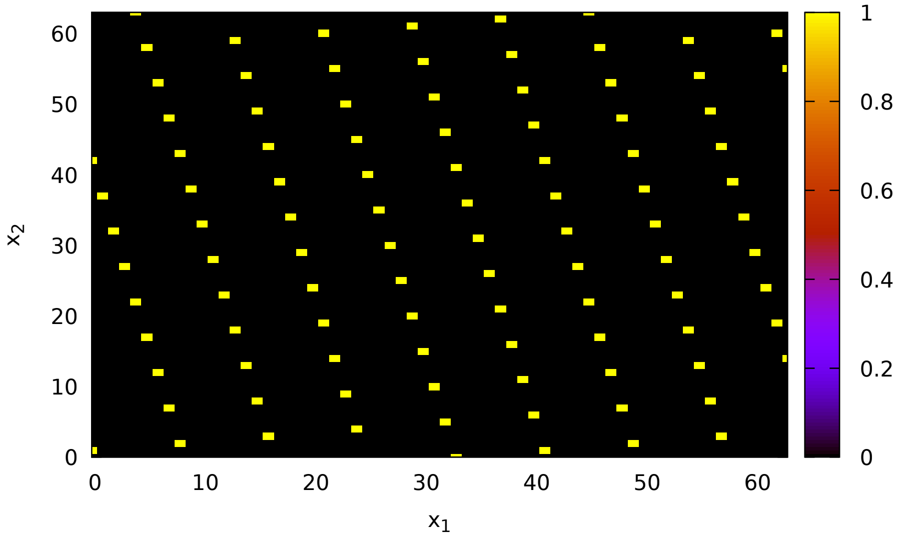

Figure 7.7: Graph of the function $f'(x_1, x_2)$ at $x_0 = 1$. Thus, the points x_1, x_2 are shown corresponding to the relationship $x_1 A + x_2 g = g$: $x_1(37, 35) + x_2(96, 93) = (96, 93)$, for example $7(37, 35) + 7(96, 93) = (96, 93)$, $8(37, 35) + 2(96, 93) = (96, 93)$ or $16(37, 35) + 3(96, 93) = (96, 93)$. It is worth noting that the selected pairs of points correspond to the condition (7.7), indeed we have $x \cdot 7 + 7 \equiv x \cdot 8 + 2 \mod 41$. That is, if you subtract one from the other, we get an equation of the form $x\,(8 - 7) = -(2 - 7) = 5 \mod 41$. Or $x \equiv 5 \mod 41$

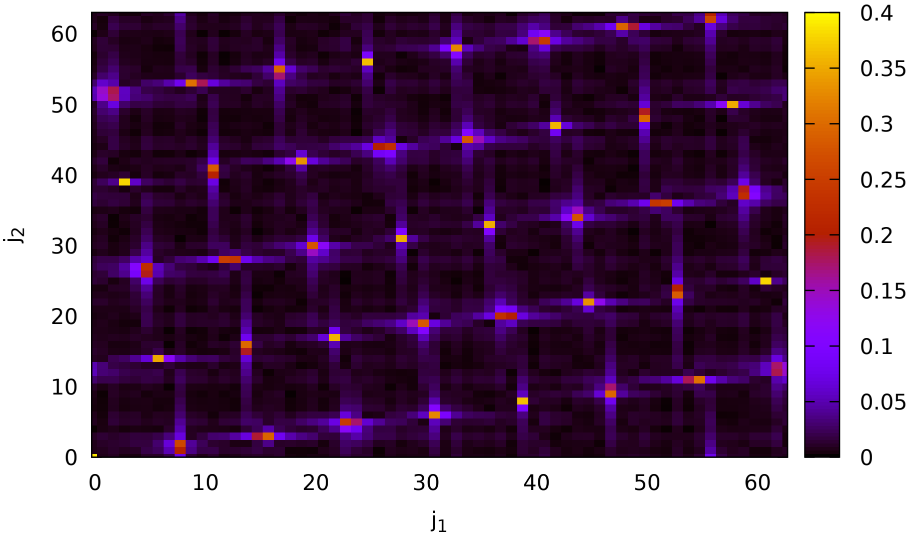

Figure 7.8: Fourier transform of the samples of the function $f'(x_1, x_2)$ Number of samples $M = 64$. The three lower left maxima have coordinates $\approx (8, 2), (15, 3), (24, 5)$, giving the following estimates for x: $x \approx 4, 5, 4.8$, which is close to the actual value $x = 5$

and we want to find an $x \in \{0, 1, \ldots 40\}$ *such that* $xg = A$*, as follows from example 7.2.1 the answer is* $x = d_a = 5$*. Our function under investigation is*

$$f(x_1, x_2) = x_1 A + x_2 g = x_1(37, 35) + x_2(96, 93)$$

As the measurement result, we choose $c = g$*, i.e.,* $x_0 = 1$*. The graph of the function* $f'(x_1, x_2)$*, corresponding to this measurement is shown in Figure 7.7. Note that the function* $f'(x_1, x_2)$ *is periodic and if we take any two closely located points such as* $(8, 2)$ *and* $(16, 3)$ *we can see that the period in coordinate* x_1 *is* $T_1 = 8$*, and in coordinate* x_2 *it is* $T_2 = 1$*. Solving the equation* $x = T_2 T_1^{-1} \mod n$ *we get* $x = 8^{-1} \equiv 5 \mod 41$*, which corresponds to the desired solution.*

The Fourier image of the function $f'(x_1, x_2)$ *is shown in Figure 7.8. From it, we can find the desired* $x = 5$*.*

Chapter 8

Appendices

8.1 Greatest Common Divisor. Euclidean Algorithm

Definition 8.1.1. The greatest common divisor of numbers a and b (GCD (a, b)) is the largest of their common divisors.

Theorem 8.1.1 (Theorem on Common Divisors)**.** *Suppose the following inequalities hold $a > b > 0$, and the number r is the remainder of dividing a by b. Thus, it can be written as*

$$a = x \cdot b + r, \tag{8.1}$$

where $x \geq 1$, $b > r \geq 0$. If $r = 0$, then b is the largest number that divides both a and b without a remainder. In case $r > 0$, then

$$GCD\,(a, b) = GCD\,(b, r)\,. \tag{8.2}$$

Proof. To prove (8.2) we show that any divisor of the pair (a, b) is a divisor of the pair (b, r). Let d be a common divisor of the numbers a and b, i.e., $a = d \cdot x_1$, $b = d \cdot x_2$. Thus, from (8.1) it follows

$$r = a - x \cdot b = d \cdot (x_1 - x \cdot x_2)\,,$$

i.e., d is a divisor of the number r.

Now we prove that any common divisor of the numbers b and r will be a common divisor of the numbers a and b. Indeed, let d be a common divisor of numbers b and r, i.e., $b = y_1 \cdot d$ and $r = y_2 \cdot d$, thus (8.1) can be rewritten as

$$a = x \cdot y_1 \cdot d + y_2 \cdot d = d \cdot (x \cdot y_1 + y_2)\,,$$

i.e., d is a divisor of the number a.

Thus, the pairs of numbers (a, b) and (b, r) have common divisors, including the greatest divisor for one pair will be the same for the second. □

The relation (8.2) leads to the following algorithm for computing the greatest common divisor

Algorithm 3 Euclidean Algorithm

```
$a > b$
if $b = 0$ then
    return $a$
end if
$a \Leftarrow 0$
$r \Leftarrow b$
$b \Leftarrow a$
repeat
    $a \Leftarrow b$
    $b \Leftarrow r$
    $r \Leftarrow$ remainder of dividing $a$ by $b$
until $(r \neq 0)$
return $b$
```

To estimate the complexity of algorithm 3 we write it in the following form

$$\begin{aligned} f_k = x_k \cdot f_{k+1} + f_{k+2}, \\ f_0 = a,\ f_1 = b, \\ x_k \geq 1,\ f_k > f_{k+1} > f_{k+2}, \end{aligned}$$

i.e., $f_k > 2 \cdot f_{k+2}$, or $f_0 > 2 \cdot f_2 > \cdots > 2^n f_{2n}$ meaning the algorithm will stop at $n = log_2(f_0) = log_2(a)$ [1]. The number of algorithm steps in this case is obviously $2n$ or $2 \cdot log_2(a)$. Thus, the algorithmic complexity of the Euclidean Algorithm can be written as $O(\log(a))$.

Example 8.1.1. $(\mathrm{GCD}(2345, 1456))$

$$\begin{aligned} 2345 &= 1456 + 889, \\ 1456 &= 889 + 567, \\ 889 &= 567 + 322, \\ 567 &= 322 + 245, \\ 322 &= 245 + 77, \\ 245 &= 3 \cdot 77 + 14, \\ 77 &= 5 \cdot 14 + 7, \\ 14 &= 2 \cdot 7. \end{aligned}$$

[1] Further derivations assume that $log_2(a)$ is an integer

Therefore, $GCD\,(2345, 1456) = 7$. *The number of algorithm steps is* $8 < 2 \cdot log_2 2345 \approx 2 \cdot 11.2 = 22.4$.

8.1.1 Bézout's Identity

Theorem 8.1.2 (Bézout)**.** *If the numbers a and b are coprime, then the equation*

$$ax + by = 1$$

has integer solutions.

Proof. We use the Euclidean Algorithm 3 to find GCD (a, b). Assume $a > b$, then

$$\begin{aligned} r_1 &= a - bq_0, \\ r_2 = b - r_1q_1 = b - (a - bq_0)q_1 &= b(1 + q_1q_0) - aq_1, \\ r_3 = r_1 - r_2q_2 = a - bq_0 - (b(1 + q_1q_0) - aq_1)\,q_2 &= \\ = a(1 + q_1q_2) - b(q_0 + q_2 + q_0q_1q_2)&, \\ &\ldots \\ \text{GCD}\,(a, b) = r_n = r_{n-2} - r_{n-1}q_{n-1} = \cdots &= ax + by, \end{aligned} \tag{8.3}$$

which proves our statement. □

Remark 8.1.1 (On the Complexity of Computing Bézout's Identity)**.** *The calculations in* (8.3) *are equivalent to the steps in algorithm 3. The number of these steps is* $O\,(log_2(a))$*, thus the algorithm described by equations* (8.3) *is quite efficient and has complexity* $O\,(log_2(a))$*.*

Example 8.1.2 (Bézout's Identity)**.** *Suppose* $a = 25, b = 14$*. The Euclidean algorithm formulas are*

$$\begin{aligned} 11 &= 25 - 14 \cdot 1, \\ 3 &= 14 - 11 \cdot 1, \\ 2 &= 11 - 3 \cdot 3, \\ 1 &= 3 - 2 \cdot 1. \end{aligned}$$

From these formulas we get

$$\begin{aligned} 11 &= 25 - 14 \cdot 1, \\ 3 &= 14 - 25 + 14 = 2 \cdot 14 - 25, \\ 2 = 11 - 3 \cdot 3 = 25 - 14 - 3 \cdot (2 \cdot 14 - 25) &= 4 \cdot 25 - 7 \cdot 14, \\ 1 &= 3 - 2 \cdot 1 = 9 \cdot 14 - 5 \cdot 25. \end{aligned}$$

Thus

$$25 \cdot (-5) + 14 \cdot 9 = 1.$$

8.2 Comparison by Modulus

Definition 8.2.1. The notation

$$a \equiv b \mod c \tag{8.4}$$

means that a and b have the same remainders when divided by c or a and b are comparable by the modulus of the natural number c. Here, the number c is called the modulus of comparison.

Definition 8.4 can also be interpreted as the difference $a - b$ being divisible by c.

Example 8.2.1. Comparison by Modulus $30 \equiv 8 \mod 11$, *because* $30 = 2{\cdot}11{+}8$.

Definition 8.2.2 (Negative Element)**.** If $a < n$, then $n - a$ will be called the element negative relative to a and denoted by $-a \mod n$.

Example 8.2.2. Negative Element

$$-5 \equiv 6 \mod 11,$$

since $5 < 11, 6 = 11 - 5$.

8.2.1 Arithmetic Operations

Lemma 8.2.1 (Addition by Modulus)**.** *If* $a_1 \equiv a_2 \mod n, b_1 \equiv b_2 \mod n$*, then*

$$a_1 + b_1 \equiv a_2 + b_2 \mod n$$

Proof. We can write $a_1 = k_1 n + r_a, a_2 = k_2 n + r_a, b_1 = l_1 n + r_b, b_2 = l_2 n + r_b$ from which

$$a_1 + b_1 = (k_1 + l_1)n + r_a + r_b \equiv r_a + r_b \mod n$$

and

$$a_2 + b_2 = (k_2 + l_2)n + r_a + r_b \equiv r_a + r_b \mod n$$

from which

$$a_1 + b_1 \equiv a_2 + b_2 \equiv r_a + r_b \mod n$$

□

Lemma 8.2.2 (Multiplication by Modulus)**.** *If* $a_1 \equiv a_2 \mod n, b_1 \equiv b_2 \mod n$*, then*

$$a_1 \cdot b_1 \equiv a_2 \cdot b_2 \mod n$$

Proof. If $a_1 \equiv a_2 \mod n, b_1 \equiv b_2 \mod n$, then

$$a_1 + b_1 \equiv a_2 + b_2 \mod n$$

We can write $a_1 = k_1 n + r_a, a_2 = k_2 n + r_a, b_1 = l_1 n + r_b, b_2 = l_2 n + r_b$ from which

$$a_1 \cdot b_1 = k_1 l_1 n + l_1 n r_a + k_1 n r_b + r_a r_b \equiv r_a r_b \mod n$$

and

$$a_2 \cdot b_2 = k_2 l_2 n + l_2 n r_a + k_2 n r_b + r_a r_b \equiv r_a r_b \mod n$$

from which

$$a_1 \cdot b_1 \equiv a_2 \cdot b_2 \equiv r_a r_b \mod n$$

□

8.2.2 Solving Equations

Very often in cryptography, one deals with equations of the form

$$ax \equiv b \mod n, \tag{8.5}$$

where a, b, n are known integers, and x is an unknown parameter to be determined.

It is obvious that if we find an integer a^{-1}, such that

$$aa^{-1} \equiv 1 \mod n,$$

then

$$x \equiv ba^{-1} \mod n.$$

If $\gcd(a, n) = 1$, then in accordance with Bézout's identity (see Bézout's theorem (Theorem 8.1.2)) $\exists x, y : ax + ny = 1$, i.e.,

$$x \equiv a^{-1} \mod n.$$

Moreover, in accordance with remark 8.1.1, a^{-1}, and the solution to equation (8.5), can be found quite efficiently.

8.2.3 Field $\mathbb{F}_p$

As we have seen in modular arithmetic, one can add, subtract, multiply, and even divide if compared by the modulus of a prime number. In this case, addition and multiplication operations are commutative and satisfy the distributive condition. Thus, the remainders form a field (see definition 8.7.1) which is called a Galois field and is denoted by $\mathbb{F}_p$.

8.3 Euler's Function

8.3.1 Definition

Definition 8.3.1 (Euler's Function)**.** Euler's function $\phi(n)$ indicates how many numbers $k \in \{1, ...n - 1\}$ are coprime with n, i.e., $\gcd(k, n) = 1$.

Example 8.3.1 (Euler's Function)**.** *If we take the number* $n = 15$*, there are 8 numbers coprime with* 15*:* $1, 2, 4, 7, 8, 11, 13, 14$*. The remaining 7 numbers are not coprime with n as they have a greatest common divisor different from 1, for example,* $\gcd(6, 15) = 3$*. Thus,* $\phi(15) = 8$*.*

8.3.2 Properties

Property 8.3.1 (Euler's Function of a Prime Number)**.** *If p is a prime number, then* $\phi(p) = p - 1$

Proof. This follows from definition 8.3.1. □

Property 8.3.2 (Euler's Function of a Product (Generalized Multiplicativity))**.** *If* $\gcd(n, m) = 1$*, then* $\phi(n \cdot m) = \phi(n)\,\phi(m)$

Proof. TBD □

Remark 8.3.1 (On the Complexity of Calculating Euler's Function)**.** *The calculation of Euler's function for large numbers is a very complex task. Most often, property 8.3.1 is used in combination with 8.3.2. Applying these properties to an arbitrary number requires its factorization, so the complexity of calculating Euler's function is comparable to the complexity of the factorization task.*

8.4 Fermat's Little Theorem

Theorem 8.4.1 (Fermat's Little Theorem)**.** *If p is a prime number, and a is not divisible by p, then*

$$a^{p-1} \equiv 1 \mod p \tag{8.6}$$

Proof. Consider the following relation

$$a \cdot k_i \mod p,$$

where $k_i \subset \{1, \ldots, p - 1\}$.

It is obvious that

$$a \cdot k_i \equiv k_j \mod p. \tag{8.7}$$

Indeed

$$a \cdot k_i \mod p \subset \{1, \ldots, p-1\},$$

as any remainder when divided by p takes values $0, 1, \ldots, p-1$. A zero remainder is impossible because a and p are coprime.

Additionally, each of the remainders $a \cdot k_i \mod p$ occurs only once. Assume that $a \cdot k_i \mod p = a \cdot k_j \mod p$ or $a \cdot (k_i - k_j) \equiv 0 \mod p$, meaning a is divisible by p, contradicting the coprimeness condition.

Multiplying all expressions from Equation 8.7, we get

$$a \cdot 2a \cdot 3a \cdot \cdots \cdot a\,(p-1) \equiv 1 \cdot 2 \cdot 3 \cdot \cdots \cdot (p-1) \mod p.$$

or

$$a^{p-1}\,(p-1)! \equiv (p-1)! \mod p,$$

from which we obtain the required equality by the coprimeness of p and $(p-1)!$:

$$a^{p-1} \equiv 1 \mod p$$

□

8.4.1 Pseudoprimes

The generalization of Fermat's Little Theorem is not true, that is, if a and p are coprime numbers that satisfy the relation Equation 8.6 then p may not be prime. For example

$$2^{341-1} \equiv 1 \mod 341,$$

even though $341 = 11 \cdot 31$.

Numbers p that satisfy the relation Equation 8.6, but are not prime, are called pseudoprimes to the base a. For example, 341 is the first pseudoprime to the base 2.

8.5 Chinese Remainder Theorem

Theorem 8.5.1. *If there are pairwise coprime integers $n_1, n_2, \ldots, n_k$, then for any set of integers $a_1, a_2, \ldots a_k$ $\exists x$ such that*

$$\begin{aligned} x &\equiv a_1 \mod n_1, \\ x &\equiv a_2 \mod n_2, \\ &\vdots \\ x &\equiv a_k \mod n_k, \end{aligned} \tag{8.8}$$

Moreover, for any x_1, x_2 satisfying this relation, the equality

$$x_1 \equiv x_2 \mod N,$$

holds, where $N = n_1 \cdot n_2 \cdot \dots \cdot n_k$.

8.6 Introduction to Group Theory

Definition 8.6.1. A group $(\mathcal{G}, \circ)$ is a set of elements $g \in \mathcal{G}$ for which a certain binary operation $\circ$ is defined (often called multiplication or addition):

$$\begin{aligned} &\forall g_1, g_2 \in \mathcal{G}, \\ &g_1 \circ g_2 \in \mathcal{G}. \end{aligned} \tag{8.9}$$

The operation defined by (8.9) has the property of associativity:

$$g_1 \circ (g_2 \circ g_3) = (g_1 \circ g_2) \circ g_3.$$

The set under consideration must contain an element $e_{\mathcal{G}}$ with the following property, valid for any element in the set g:

$$g \circ e_{\mathcal{G}} = e_{\mathcal{G}} \circ g = g.$$

For each element of the group g, there must exist an inverse element $g^{-1} \in \mathcal{G}$ with the following property

$$g \circ g^{-1} = g^{-1} \circ g = e_{\mathcal{G}}$$

Definition 8.6.2 (Monoid)**.** If for a certain set of elements $\mathcal{G}$ the last property of a group (existence of an inverse element) is not satisfied, then this set is called a monoid or a semigroup.

Definition 8.6.3 (Abelian Group)**.** A group $(\mathcal{A}, \circ)$ is called abelian or commutative if $\forall a_1, a_2 \in \mathcal{A}$: $a_1 \circ a_2 = a_2 \circ a_1$.

Example 8.6.1. Group $(\mathbb{Z}, +)$ *The set of integers $\mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}$ forms a group under the operation of addition.*

Definition 8.6.4 (Cyclic Group)**.** A cyclic group G is a group generated by a single element $g : G =< g >$, i.e., all its elements are powers of g. The element g is called the generator of the group G.

Definition 8.6.5 (Multiplicative Group of the Ring of Residues)**.** Consider the set of integers coprime to n and less than n, denoted by $(\mathbb{Z}/n\mathbb{Z})^\times$. As the operation of multiplying two elements $a, b \in (\mathbb{Z}/n\mathbb{Z})^\times$, we take

$$a \circ b = a \cdot b \mod n.$$

The unit element $e_{(\mathbb{Z}/n\mathbb{Z})^\times}$ is 1. Furthermore, it can be shown that for each $a \in (\mathbb{Z}/n\mathbb{Z})^\times$ there exists $a^{-1} \in (\mathbb{Z}/n\mathbb{Z})^\times$ such that $a \circ a^{-1} = 1$. Thus, $(\mathbb{Z}/n\mathbb{Z})^\times$ is a group.

Theorem 8.6.1 (On the Order of $(\mathbb{Z}/n\mathbb{Z})^\times$)**.** *The order of the group $(\mathbb{Z}/n\mathbb{Z})^\times$ is determined by the following relation*

$$\left|(\mathbb{Z}/n\mathbb{Z})^\times\right| = \phi(n),$$

where $\phi(n)$ is Euler's function (8.3.1) . Moreover, if $n = p$ is a prime number, then

$$\left|(\mathbb{Z}/p\mathbb{Z})^\times\right| = \phi(p),$$

and if $a \in (\mathbb{Z}/p\mathbb{Z})^\times, a \neq 1$, then a is a generator of the group in question, i.e.,

$$a^{p-1} = 1.$$

Proof. TBD □

Theorem 8.6.2 (Lagrange)**.** *For any finite group $\mathcal{G}$, the order (number of elements) of any subgroup $\mathcal{H}$ divides the order of $\mathcal{G}$:*

$$|\mathcal{G}| = h\,|\mathcal{H}|\,,$$

where the integer h is called the index of the subgroup.

8.7 Fields

Definition 8.7.1 (Field (algebra))**.** Let there be an Abelian group (see definition 8.6.3) $(\mathcal{F}, +)$. The identity element of this group $e_\mathcal{F}$ is 0. Let also $(\mathcal{F} \setminus \{0\}, \cdot)$ be some other group (also Abelian) with the identity element 1. Additionally, the operations $+, \cdot$ satisfy the distributive property, i.e., $\forall a, b, c \in \mathcal{F}$:

$$c \cdot (a + b) = c \cdot a + c \cdot b,$$
$$(a + b) \cdot c = a \cdot c + b \cdot c.$$

In this case, $(\mathcal{F}, +, \cdot)$ is called a field.

Example 8.7.1 (Field $\mathbb{Q}$). *Note that $\mathbb{Z}$ is not a field because not every integer has an inverse with respect to multiplication. However, the following set will be a field: $\mathbb{Q} = \{a/b \mid a \in \mathbb{Z}, b \in \mathbb{Z} \setminus \{0\}\}$. The inverse with respect to $a/b \in (\mathbb{Q} \setminus \{0\}, \cdot)$ will be b/a.*

Example 8.7.2 (Field $\mathbb{R}$). *The real numbers form a field.*

Example 8.7.3 (Field $\mathbb{C}$). *The complex numbers form a field.*

8.8 Main Theorem on Recurrence Relations

Theorem 8.8.1 (Main Theorem on Recurrence Relations). *If there is the following recurrence relation for the complexity of some algorithm*

$$T(n) = aT\left(\frac{n}{b}\right) + f(n),$$

it is possible to determine the asymptotic behavior of the function $T(n)$ in the following cases

1. *If $f(n) = O\left(n^{log_b a - \epsilon}\right)$, for some $\epsilon > 0$, then $T(n) = \Theta\left(n^{log_b a}\right)$*
2. *If $f(n) = \Theta\left(n^{log_b a} log^k n\right)$, then $T(n) = \Theta\left(n^{log_b a} log^{k+1} n\right)$*
3. *If $f(n) = \Omega\left(n^{log_b a + \epsilon}\right)$, for some $\epsilon > 0$ and $af\left(\frac{n}{b}\right) \leq cf(n)$ for some constant $c < 1$ and large n, then $T(n) = \Theta(f(n))$*

8.9 Divide and Conquer

"Divide and conquer" is an important paradigm in solving algorithmic problems, which involves dividing the original problem into simpler ones.

Bibliography

[1] Charles H. Bennett et al. “Strengths and Weaknesses of Quantum Computing”. In: *SIAM J. Comput.* 26.5 (Oct. 1997), pp. 1510–1523. ISSN: 0097-5397. DOI: 10.1137/S0097539796300933. URL: http://dx.doi.org/10.1137/S0097539796300933.

[2] Daniel M. Gordon. “Discrete logarithms in gf(p) using the number field sieve”. In: *SIAM J. Discrete Math* 6 (1993), pp. 124–138.

[3] Lov K. Grover. “A Fast Quantum Mechanical Algorithm for Database Search”. In: *ANNUAL ACM SYMPOSIUM ON THEORY OF COMPUTING*. ACM, 1996, pp. 212–219.

[4] Ivan Murashko and Constantine Korikov. “Analyze of Quantum Fourier Transform Circuit Implementation”. In: *Internet of Things, Smart Spaces, and Next Generation Networks and Systems - 15th International Conference, NEW2AN 2015, and 8th Conference, ruSMART 2015, St. Petersburg, Russia, August 26-28, 2015, Proceedings*. Ed. by Sergey Balandin, Sergey D. Andreev, and Yevgeni Koucheryavy. Vol. 9247. Lecture Notes in Computer Science. Springer, 2015. ISBN: 978-3-319-23125-9. DOI: 10.1007/978-3-319-23126-6. URL: http://dx.doi.org/10.1007/978-3-319-23126-6.

[5] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge Series on Information and the Natural Sciences. Cambridge University Press, 2000. ISBN: 9780521635035. URL: http://books.google.ru/books?id=65FqEKQOfP8C.

[6] J. M. Pollard. “Monte Carlo method for index computation (mod p)”. In: *Mathematics of Computation* 32.143 (July 1978), pp. 918–924. URL: http://levicivita.uniroma2.it/~eal/montecarlo.pdf.

[7] John Proos and Christof Zalka. “Shor’s Discrete Logarithm Quantum Algorithm for Elliptic Curves”. In: *Quantum Info. Comput.* 3.4 (July 2003), pp. 317–344. ISSN: 1533-7146. URL: https://arxiv.org/abs/quant-ph/0301141.

[8] René J. Schoof. "Elliptic curves over finite fields and the computation of square roots mod p". In: *Mathematics of Computation* 44 (1985). URL: `http://cr.yp.to/bib/entries.html#1985/schoof`, pp. 483–494. ISSN: 0025–5718.

[9] Claude E. Shannon. "Communication Theory of Secrecy Systems". In: *Bell Systems Technical Journal* 28 (1949), pp. 656–715.

[10] Peter W. Shor. "Algorithms for Quantum Computation: Discrete Logarithms and Factoring". In: *FOCS*. IEEE Computer Society, 1994, pp. 124–134.

[11] Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography, Second Edition*. 2nd ed. Chapman & Hall/CRC, 2008. ISBN: 9781420071467.

[12] Wikipedia. *Дискретное логарифмирование — Wikipedia, The Free Encyclopedia*. http://ru.wikipedia.org/wiki/Дискретное_логарифмирование. [Online; accessed 29-October-2013]. 2013.

[13] W. H. Zurek. "Decoherence and the Transition from Quantum to Classical—Revisited". In: *Los Alamos Science* 27 (2002), pp. 2–25. URL: `http://vvkuz.ru/books/zurek.pdf`.

[14] А. Цайлингера, ed. *Физика квантовой информации*. Russian. Москва: Постмаркет, 2002, p. 376.

[15] В. А. Ильин and Э. Г. Поздняк. *Линейная алгебра*. Russian. 6th ed. Москва: Физматлит, 2005, p. 115.

[16] Михаил Борисович Менский. *Квантовые измерения и декогеренция. Модели и феноменология*. Russian. Москва: Физматлит, 2001, p. 228.

[17] Поль Дирак. *Принципы квантовой механики*. Russian. Москва: Наука, 1979.