Харківський національний університет імені В.Н. Каразіна
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

Кандій Сергій Олегович

УДК 004.056.5

ДИСЕРТАЦІЯ
МЕТОДИ ТА МОДЕЛІ ОЦІНКИ ЗАХИЩЕНОСТІ АСИМЕТРИЧНИХ КРИПТОГРАФІЧНИХ ПЕРЕТВОРЕНЬ НА РЕШІТКАХ ВІД ІСНУЮЧИХ ТА ПОТЕНЦІЙНИХ АТАК

Спеціальність 125 Кібербезпека та захист інформації
(Галузь знань 12 Інформаційні технології)
Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.
_____ С.О. Кандій

Науковий керівник: Горбенко Іван Дмитрович, доктор технічних наук, професор

Харків – 2025

АНОТАЦІЯ

Кандій С.О. Методи та моделі оцінки захищеності асиметричних криптографічних перетворень на решітках від існуючих та потенційних атак – Кваліфікаційна наукова праця на правах рукопису

Дисертація на здобуття ступеня доктора філософії за спеціальністю 125 Кібербезпека та захист інформації (Галузь знань 12 Інформаційні технології). – Харківський національний університет імені В. Н. Каразіна Міністерства освіти і науки України, Харків, 2025.

Дисертаційна робота присвячена розв'язанню актуальної задачі: аналізу та розробці методів та моделей для підвищення захищеності асиметричних криптографічних систем на решітках.

Задача є актуальною, оскільки захист інформації стає одним з ключових викликів у сучасному цифровому світі, де загрози інформаційній безпеці постійно зростають. Криптографічні методи є основою для забезпечення конфіденційності, цілісності та автентичності даних. Сьогоднішня реальність відзначається активним впровадженням криптографічних рішень у різних секторах – від фінансових та урядових структур до промислових і комерційних організацій.

На міжнародній арені спостерігається прагнення до стандартизації криптографічних засобів, що сприяє сумісності систем безпеки різних країн. Такі організації, як Міжнародна організація зі стандартизації (ISO) та Національний інститут стандартів і технологій США (NIST), активно працюють над створенням нових стандартів, які враховують сучасні загрози, зокрема потенційну небезпеку квантових обчислень.

Національний розвиток криптографії також набирає обертів. В Україні активізуються зусилля щодо створення власних стандартів та нормативних документів для регулювання використання криптографічних засобів.

Мета і завдання дослідження. Метою дослідження є аналіз та розробка методів та моделей для підвищення захищеності асиметричних криптографічних систем на решітках від існуючих та потенційних атак.

Для досягнення поставленої мети були розв’язані такі основні задачі:

1. Вибір та обґрунтування моделей редукції решіток, які враховують усі існуючі відомості щодо поведінки базисів під час редукції. Ця задача включала всебічний огляд і порівняння різних моделей редукції решіток, які використовуються для оптимізації базисів решіток, що є ключовим елементом у багатьох криптографічних схемах. Особливо було акцентовано увагу на таких алгоритмах, як BKZ (Block Korkine–Zolotarev), які широко застосовуються для редукції базисів. Проведений аналіз дозволив обґрунтувати вибір найбільш придатних моделей редукції для різних типів задач на решітках та виявити їхні сильні та слабкі сторони. Крім того, були розглянуті аспекти, що впливають на точність і ефективність редукції базисів.

2. Дослідження впливу моделей редукції решіток на складність атак на криптографічні перетворення на решітках та удосконалення методики оцінювання криптографічних перетворень на решітках. Особлива увага приділялась проблемам, які лежать в основі постквантових криптографічних схем: NTRU, LWE, SIS. Проблеми навчання з помилками (LWE) та NTRU є основою багатьох сучасних криптографічних схем. Досліджено, як різні моделі редукції решіток впливають на складність їх розв’язання, а також проаналізовано стійкість криптографічних схем, що базуються на цих задачах. Проблема малого цілочисельного рішення (SIS) є критичною для стійкості багатьох криптографічних протоколів. Досліджено, як зміна параметрів редукції решіток впливає на складність розв’язання SIS, і запропоновано удосконалення для оцінки її криптографічної стійкості. Удосконалена методика оцінювання криптографічних перетворень на решітках враховує особливості цих задач, а також вплив редукції на ключові параметри стійкості криптографічних схем. Це дозволило підвищити точність прогнозування стійкості криптографічних систем

у контексті їх потенційної вразливості до атак квантових і класичних обчислювальних методів.

3. Аналіз криптографічних перетворень на решітках в формальних моделях безпеки, що враховують можливість застосування класичних та квантових атак. У межах цього дослідження був проведений детальний аналіз механізму інкапсуляції ключів, визначеного в стандарті ДСТУ 8961:2019, з урахуванням сучасних викликів у криптографії, пов'язаних із квантовими обчисленнями. Основну увагу було приділено отриманню формального доказу безпеки схеми інкапсуляції ключів у моделі квантового випадкового оракула (QROM).

4. Дослідження впливу обчислень з плаваючою точкою на безпеку криптографічних перетворень на решітках. У рамках цього дослідження особлива увага приділялася аналізу криптографічної стійкості схем електронного підпису, зокрема Falcon — одного з провідних кандидатів для стандартизації постквантової криптографії. Основна мета дослідження полягала в аналізі потенційних атак на основі обчислень з плаваючою точкою, які можуть бути використані для відновлення приватних ключів.

У першому розділі дисертації (*Аналіз сучасних тенденцій у квантово–стійкій криптографії*) на основі проведеного аналізу показано, що розвиток квантових алгоритмів обумовлений двома техніками – квантовим пошуком та квантовою вибіркою Фур'є. Обґрунтовано, що криптографія на решітках є стійкою до застосування цих технік. Розкриті вимоги до квантово–стійкої криптографії. Розкрита сутність доказової безпеки та моделей безпеки IND–CCA (Indistinguishability under Adaptive Chosen Chiphertext Attack) для механізмів інкапсуляції ключів та EUF–CMA (Existentially unforgeable under adaptive chosen message attacks) для електронних підписів. Наведено відомості з теорії решіток та теорії квантових обчислень.

У другому розділі дисертації (*Аналіз та порівняння моделей редукції решіток*) обґрунтовано, що фактор Ерміта відіграє важливу роль при аналізі моделей редукції решіток. Проведено серію експериментів, що направлені на

визначення точності існуючих асимптотичних оцінок фактору Ерміта на криптографічно значущих розмірностях решіток. Показано, що істинне значення фактору Ерміта швидко наближається до асимптотичних оцінок і на криптографічно значущих розмірностях можливо вважати помилку апроксимації фактору Ерміта незначною. Проведено порівняльний аналіз існуючих симуляторів редукції решіток. Для порівняння симуляторів проведено ряд експериментів на решітках малої розмірності. Показано, що симулятор Альбрехта–Лі дає найменшу середньоквадратичну помилку серед усіх симуляторів. Для NTRU решіток розкрито сутність розрідженої підрешітки. Отримано перший науковий результат: Вперше виконано кількісне порівняння точності моделей редукції решіток із застосуванням метрики середньоквадратичної помилки для моделі GSA (Geometric Series Assumption) та симуляторів редукції решіток. Попередні дослідження фокусувалися на якісних або суто теоретичних оцінках якості роботи моделей. Отримані оцінки дозволяють кількісно оцінювати якість роботи симуляторів в залежності від параметрів решіток для оцінки захищеності від класичних та квантових атак.

У третьому розділі дисертації (Методи оцінки складності криптографічних задач з теорії решіток) наведено класифікацію існуючих атак на криптографічні перетворення на решітках. Для атак вкладення та декодування запропонована удосконалена модель оцінки складності атаки, що базується на проведеному у другому розділі аналізі. Для атак декодування запропонований метод визначення оптимальних параметрів атаки. Удосконалено методику оцінювання складності криптографічної задачі SIS (Shortest Integer Solution), що відрізняється від існуючих тим, що у даній методиці враховується алгебраїчна структура решіток під час аналізу процесів редукції для оцінки параметрів та характеристик атак на їх основі.

У четвертому розділі дисертації (Оцінка захищеності механізмів інкапсуляції ключів на алгебраїчних решітках) уточнено оцінки захищеності механізмів інкапсуляції ключів ДСТУ 8961:2019 та Crystals–Kyber. Показано, що врахування структури алгебраїчних решіток, згідно до методу, що був

розроблений в розділі 3, дає більші оцінки безпеки, ніж класична модель GSA. Вперше було отримано узагальнений доказ IND–CCA безпеки перетворень, що використовуються в стандарті ДСТУ 8961:2019, у моделі квантового випадкового оракула. Попередні дослідження не вивчали IND–CCA безпеку перетворень ДСТУ 8961:2019 у моделі квантового випадкового оракула.

У п’ятому розділі дисертації (Оцінка захищеності електронних підписів на алгебраїчних решітках) уточнено оцінки захищеності електронних підписів Falcon та Crystals–Dilithium. Показано, що врахування структури алгебраїчних решіток, згідно до методу, що був розроблений в розділі 3, дає більші оцінки безпеки, ніж класична модель GSA. Для електронного підпису Falcon отримали подальший розвиток обґрунтування оцінки атаки відновлення ключів, що використовує обчислення з плаваючою крапкою, для алгоритмів електронного підпису на основі решіток, що дало змогу підвищити безпеку електронних підписів на решітках.

Практичне значення отриманих результатів полягає у тому, що було уточнено оцінки безпеки криптографічних перетворень на решітках, які підтверджують рівні стійкості квантово–стійких стандартів електронного підпису (ДСТУ 9212:2023) та механізмів інкапсуляції ключів (ДСТУ 8961:2019), що обґрунтовує їх застосування як національних стандартів. Розроблено програмне забезпечення, яке дозволяє оцінювати складність атак вкладення та декодування для проблем LWE та NTRU при процесі практичного оцінювання безпеки криптографічних перетворень на решітках, оцінювати складність проблеми SIS з врахуванням структури решіток та проводити моделювання редукції решіток.

Ключові слова: постквантова криптографія, електронний підпис, механізми інкапсуляції ключів, алгебраїчні решітки, доказова безпека, асиметричні криптосистеми, шифрування, моделі, криптографічні примітиви, конфіденційність, цілісність, аналіз атак, кібербезпека, квантові обчислення.

ABSTRACT

Kandii S.O. Methods and models for evaluating the security of lattice–based asymmetric cryptographic transformations against existing and potential attacks – Qualification scholarly paper: a manuscript.

Thesis for the degree of Doctor of Philosophy in speciality 125 Cybersecurity and information protection (Field of knowledge 12 Information Technology). – V. N. Karazin Kharkiv National University, Ministry of Education and Science of Ukraine, Kharkiv, 2025.

The thesis is devoted to solving a relevant problem: the analysis and development of methods and models to enhance the security of lattice–based asymmetric cryptographic systems.

The problem is relevant because information security has become one of the key challenges in the modern digital world, where threats to information security are constantly increasing. Cryptographic methods form the foundation for ensuring data confidentiality, integrity, and authenticity. Today's reality is marked by the active implementation of cryptographic solutions across various sectors, ranging from financial and governmental institutions to industrial and commercial organizations.

On the international stage, there is a growing effort to standardize cryptographic tools, which facilitates the interoperability of security systems across different countries. Organizations such as the International Organization for Standardization (ISO) and the U.S. National Institute of Standards and Technology (NIST) are actively working on developing new standards that address modern threats, including the potential risks posed by quantum computing.

The national development of cryptography is also gaining momentum. In Ukraine, efforts are being intensified to establish national standards and regulatory documents for governing the use of cryptographic tools.

Research Objective and Tasks. The objective of the research is to analyze and develop methods and models to enhance the security of lattice–based asymmetric cryptographic systems against existing and potential attacks.

To achieve the stated objective, the following key tasks were addressed:

1. Selection and justification of lattice reduction models that account for all known aspects of basis behavior during reduction. This task involved a comprehensive review and comparison of various lattice reduction models used for optimizing lattice bases, which are a crucial element in many cryptographic schemes. Special emphasis was placed on algorithms such as BKZ (Block Korkine–Zolotarev), which are widely applied for basis reduction. The conducted analysis enabled the justification of the most suitable reduction models for different types of lattice problems and the identification of their strengths and weaknesses. Additionally, factors influencing the accuracy and efficiency of basis reduction were examined.

2. Investigation of the impact of lattice reduction models on the complexity of attacks against lattice–based cryptographic transformations and enhancement of evaluation methodologies for such transformations. Special attention was given to fundamental problems underlying post–quantum cryptographic schemes, including NTRU, LWE, and SIS. The Learning With Errors (LWE) problem and the NTRU problem form the foundation of many modern cryptographic schemes. The study examined how different lattice reduction models influence the complexity of solving these problems and analyzed the resilience of cryptographic schemes based on them. The Small Integer Solution (SIS) problem is critical to the security of many cryptographic protocols. The research investigated how changes in lattice reduction parameters affect the complexity of solving SIS and proposed improvements for assessing its cryptographic security. The enhanced evaluation methodology for lattice–based cryptographic transformations takes into account the specific characteristics of these problems, as well as the influence of reduction on key security parameters of cryptographic schemes. This advancement has improved the accuracy of predicting the resilience of cryptographic systems concerning their potential vulnerabilities to both quantum and classical computational attack methods.

3. Analysis of lattice–based cryptographic transformations within formal security models that consider the feasibility of classical and quantum attacks. As part of this study, a detailed analysis was conducted on the key encapsulation mechanism

specified in the DSTU 8961:2019 standard, taking into account contemporary cryptographic challenges related to quantum computing. Special attention was given to obtaining a formal security proof for the key encapsulation scheme within the Quantum Random Oracle Model (QROM).

4. Investigation of the impact of floating–point computations on the security of lattice–based cryptographic transformations. This study focused on analyzing the cryptographic resilience of digital signature schemes, particularly Falcon, one of the leading candidates for post–quantum cryptography standardization. The primary objective was to examine potential floating–point–based attacks that could be exploited to recover private keys.

In the first chapter of the dissertation (Analysis of Modern Trends in Quantum–Resistant Cryptography), the conducted analysis demonstrates that the advancement of quantum algorithms is driven by two key techniques: quantum search and quantum Fourier sampling. It is substantiated that lattice–based cryptography remains resistant to these techniques.

The chapter outlines the requirements for quantum–resistant cryptography and explores the concept of provable security, along with security models such as IND–CCA (Indistinguishability under Adaptive Chosen Ciphertext Attack) for key encapsulation mechanisms and EUF–CMA (Existential Unforgeability under Adaptive Chosen Message Attack) for digital signatures. Additionally, fundamental concepts from lattice theory and quantum computing theory are presented.

In the second chapter of the dissertation (Analysis and Comparison of Lattice Reduction Models), it is substantiated that the Hermite factor plays a crucial role in analyzing lattice reduction models. A series of experiments was conducted to assess the accuracy of existing asymptotic estimates of the Hermite factor for cryptographically significant lattice dimensions. The results demonstrate that the true value of the Hermite factor quickly converges to its asymptotic estimates, and for cryptographically relevant dimensions, the approximation error can be considered negligible.

A comparative analysis of existing lattice reduction simulators was performed. To evaluate these simulators, several experiments were conducted on low–dimensional lattices. The findings indicate that the Albrecht–Lee simulator produces the lowest root–mean–square error among all tested simulators. For NTRU lattices, the concept of a sparse sublattice is explored in detail.

First Scientific Contribution: For the first time, a quantitative comparison of the accuracy of lattice reduction models was performed using the root–mean–square error metric for both the Geometric Series Assumption (GSA) model and lattice reduction simulators. Previous studies primarily focused on qualitative or purely theoretical assessments of model accuracy. The obtained estimates enable a quantitative evaluation of simulator performance based on lattice parameters, contributing to the assessment of security against both classical and quantum attacks.

In the third chapter of the dissertation (Methods for Assessing the Complexity of Cryptographic Problems in Lattice Theory), a classification of existing attacks on lattice–based cryptographic transformations is provided. For embedding and decoding attacks, an enhanced complexity assessment model is proposed, based on the analysis conducted in the second chapter. For decoding attacks, a method for determining optimal attack parameters is introduced.

Additionally, the evaluation methodology for the complexity of the Shortest Integer Solution (SIS) problem has been improved. Unlike existing methods, this approach incorporates the algebraic structure of lattices when analyzing the reduction processes used to estimate the parameters and characteristics of attacks based on SIS.

In the fourth chapter of the dissertation (Security Assessment of Key Encapsulation Mechanisms on Algebraic Lattices), the security estimates of the DSTU 8961:2019 and CRYSTALS–Kyber key encapsulation mechanisms were refined.

The findings demonstrate that considering the structure of algebraic lattices, according to the method developed in Chapter 3, results in higher security estimates compared to the classical Geometric Series Assumption (GSA) model. First Scientific Contribution: For the first time, a generalized IND–CCA security proof for transformations used in DSTU 8961:2019 was obtained within the Quantum Random

Oracle Model (QROM). Previous studies had not explored the IND–CCA security of DSTU 8961:2019 transformations in the QROM framework.

In the fifth chapter of the dissertation (Security Assessment of Digital Signatures on Algebraic Lattices), the security estimates of the Falcon and CRYSTALS–Dilithium digital signature schemes were refined.

The findings demonstrate that considering the structure of algebraic lattices, according to the method developed in Chapter 3, results in higher security estimates compared to the classical Geometric Series Assumption (GSA) model.

For the Falcon digital signature scheme, further development was made in justifying the evaluation of key recovery attacks that utilize floating–point computations for lattice–based signature algorithms. This advancement has contributed to enhancing the security of lattice–based digital signatures.

The practical significance of the obtained results lies in the refinement of security estimates for lattice–based cryptographic transformations, which confirm the security levels of quantum–resistant digital signature standards (DSTU 9212:2023) and key encapsulation mechanisms (DSTU 8961:2019), thereby justifying their adoption as national standards.

Additionally, software was developed to: assess the complexity of embedding and decoding attacks for the LWE and NTRU problems in the context of practical security evaluation of lattice–based cryptographic transformations, evaluate the complexity of the SIS problem, taking into account lattice structure and simulate lattice reduction processes for security analysis.

Keywords: post–quantum cryptography, digital signature, key encapsulation mechanisms, algebraic lattices, provable security, asymmetric cryptosystems, encryption, models, cryptographic primitives, confidentiality, integrity, attack analysis, cybersecurity, quantum computing.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукова публікація у зарубіжних виданнях, що входять до міжнародних наукометричних баз Scopus та Web of Scienc:

1. Gorbenko I.D., Yesina M.V., Kandy S.O., Ostryanska Ye. V. Generation of general system parameters for Falcon cryptosystem for 256, 384, and 512 security bits // Telecommunications and Radio Engineering. 2022. Vol. 81, Is. 2. P. 49–59.

DOI:10.1615/TelecomRadEng.2022037071.

URL:https://www.dl.begellhouse.com/journals/0632a9d54950b268,33bd45d917452b68,54c5c714496ce4ca.html

(Особистий внесок здобувача: розраховано параметри для 256, 384, 512 біт безпеки для електронного підпису Falcon. Особистий внесок Gorbenko I.D.: Обґрунтування вимог та постановка задачі щодо генерації загальносистемних параметрів електронного підпису Falcon. Особистий внесок Yesina M.V.: Пошук та формування сукупностей безумовних, умовних та прагматичних критеріїв для процесу оцінки та порівняння електронних підписів. Особистий внесок Ostryanska Ye. V.: Перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

2. Potii O.V., Kachko O.G., Kandii S.O., Kaptol Y.Y. Determining the effect of a floating point on the Falcon Digital Signature Algorithm Security // Eastern–European Journal of Enterprise Technologies. 2024. Vol. 1, Is. 9. P. 52–59.

DOI:10.15587/1729–4061.2024.295160.

URL: https://journals.uran.ua/eejet/article/view/295160

(Особистий внесок здобувача: запропоновано теоретичне обґрунтування атаки відновлення ключа, виконано моделювання атаки. Особистий внесок Potii O.V.: Обґрунтування вимог та постановка задачі щодо криптоаналізу фіналісту конкурсу NIST PQC, методу ЕП Falcon. Особистий внесок Kachko O.G.: Програмне моделювання процесів та елементів реалізованої атаки. Особистий внесок Kaptol Y.Y.: Верифікація та аналіз результатів здійснення атаки на алгоритм Falcon. Оцінка ймовірності реалізації атаки та впливу

виявленої вразливості методу та потенційних векторів атак на захищеність алгоритму із врахуванням релевантних моделей безпеки.)

3. Kachko O.G., Gorbenko Y.I., Kandii S.O., Kaptol Y.Y. Improving protection of falcon electronic signature software implementations against attacks based on Floating Point Noise // *Eastern–European Journal of Enterprise Technologies*. 2024. Vol. 4, Is. 9, P. 6–17.

DOI:10.15587/1729–4061.2024.310521.

URL: https://journals.uran.ua/eejet/article/view/310521

(Особистий внесок здобувача: знайдено оцінки необхідної кількості підписів для проведення атаки. Особистий внесок Kachko O.G.: Визначення необхідних змін до програмної реалізації для усунення можливості реалізації атаки. Особистий внесок Gorbenko Y.I.: обгрунтування вимог та постановка задачі щодо криптоаналізу та покращення фіналісту конкурсу NIST PQC, методу ЕП Falcon. Особистий внесок Kaptol Y.Y.: оцінено вплив на безпеку електронного підпису застосування фіксованої точки замість плаваючої точки на етапі генерації підпису.)

Наукові публікації у фахових виданнях України

4. Gorbenko Yu.I., Kandii S.O. Comparison of security arguments of promising key encapsulation mechanisms // *Radiotekhnika*. 2022. Vol. 210. P. 22–36.

DOI:10.30837/rt.2022.3.210.02.

URL: http://rt.nure.ua/article/view/268561/264140

(Особистий внесок здобувача: порівняння аргументів безпеки для перспективних механізмів інкапсуляції ключів. Особистий внесок Gorbenko Yu.I.: постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

5. Kandiy S.O. Analysis of DSTU 8961:2019 in random Oracle Model // *Radiotekhnika*. 2022. Vol. 211. P. 22–36.

DOI:10.30837/rt.2022.4.211.02.

URL: http://rt.nure.ua/article/view/278028

(Особистий внесок здобувача: аналіз безпеки стандарту ДСТУ 8961:2019 у моделі випадкового оракула)

6. Kandiy S.O., Ostrianska Ye.V., Gorbenko I.D., Yesina M.V. Classification and analysis of vulnerabilities of modern information systems from classical and quantum attacks. Radiotekhnika. 2022. Vol. 211. P. 7–21.

DOI: 10.30837/rt.2022.4.211.01

URL: http://rt.nure.ua/article/view/278027

(Особистий внесок здобувача: аналіз моделей на основі нерозрізнювальності для алгоритмів шифрування та інкапсуляції ключів. Особистий внесок Ostrianska Ye.V.: Аналіз атак на реалізацію криптографічних перетворень та моделей, що їх враховують. Особистий внесок Gorbenko I.D.: постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи. Особистий внесок Yesina M.V.: классифікація вразливостей сучасних інформаційних систем.)

7. Kandiy S.O., Gorbenko I.D. Security analysis of promising key encapsulation mechanisms in the core–SVP model // *Radiotekhnika*. 2023. Vol. 212. P. 66–84.

DOI:10.30837/rt.2023.1.212.06.

URL: http://rt.nure.ua/article/view/286564

(Особистий внесок здобувача: Отримання оцінок безпеки механізмів інкапсуляції ключів в моделі core–SVP. Особистий внесок Gorbenko I.D.: постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

8. Kandii S.O., Gorbenko I.D. Analysis of DSTU 8961:2019 in the quantum random Oracle Model // *Radiotekhnika*. 2023. Vol. 214. P. 7–16.

DOI:10.30837/rt.2023.3.214.01.

URL: http://rt.nure.ua/article/view/297798/290701

(Особистий внесок здобувача: аналіз ДСТУ 8961:2019 у моделі квантового випадкового оракула. Особистий внесок Gorbenko I.D.: постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

9. Kandii S.O., Gorbenko I.D. Refinement of security estimates of quantum–resistant standards of asymmetric encryption taking into account the structure of q–arry lattices // Radiotekhnika. 2024. Vol 218. P. 76–92

DOI:10.30837/rt.2024.3.218.06

URL: http://rt.nure.ua/article/view/318798/309118

(Особистий внесок здобувача: Уточнення оцінок захищеності квантово–стійких стандартів асиметричного шифрування та електронного підпису з врахуванням q–арної структури решіток. Особистий внесок Gorbenko I.D.: постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

10. Kandii S.O., Gorbenko I.D. Assessing the influence of the algebraic structure of q–ary lattices on the complexity of cryptanalysis of problems on lattices // *Radiotekhnika.* 2024. Vol. 217. P. 79–99.

DOI:10.30837/rt.2024.2.217.07.

URL: http://rt.nure.ua/article/view/310856

(Особистий внесок здобувача: оцінка впливу q–арної структури решіток на складність криптоаналізу складних проблем. Особистий внесок Gorbenko I.D.: постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

Наукові праці, які додатково відображають результати дисертації:

11. Кандій С.О., Горбенко І.Д. Аналіз фактору Ерміта алгоритму BKZ на решітках малої розмірності. Computer Science and Cybersecurity. 2024. Is. 1(25). P. 22–36

DOI: 10.26565/2519–2310–2024–1–02

URL:https://periodicals.karazin.ua/cscs/article/download/2519–2310–2024–1–02/22054/

(Особистий внесок здобувача: експерементальна оцінка фактора Ерміта та аналіз його впливу на криптоаналіз. Особистий внесок Gorbenko I.D.:

постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

12. Кандій С.О. Аналіз CPA–to–CCA перетворення ДСТУ 8961:2019 у моделі випадкового оракула. Мат. та комп'ютер. моделювання. Сер. Фіз.–мат. науки. 2023. Вип. 36. С. 101–105.

DOI: 10.15407/10.15407/fmmit2023.36.101

URL: http://www.fmmit.lviv.ua/index.php/fmmit/article/download/285/245/

(Особистий внесок здобувача: аналіз CPA–to–CCA перетворення ДСТУ 8961:2019 у моделі випадкового оракула.)

Наукові праці, які засвідчують апробацію матеріалів дисертації:

13. Кандій С.О. Острянська Є.В. Генерація загальносистемних параметрів для схеми електронного підпису Rainbow. І міжнародна науково–технічна коференція «Системи і технології зв’язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку». 2021. С. 226–227.

14. Кандій С.О. Порівняння аргументів безпеки перспективних механізмів інкапсуляції ключів. Праці 8–ої Міжнародній конференції «Комп’ютерне моделювання в наукоємних технологіях» (КМНТ–2022). 2022. С. 97–100.

15. Горбенко Ю.І., Острянська Є.В., Кандій С.О. Експериментальна оцінка точності фактора ерміта. IV Міжнародна науково-технічна конференція «Системи і технології зв’язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку». 2024. С. 48-49.

16. Потій О.В., Качко О.Г., Кандій С.О., Каптьол Є.Ю. Дослідження впливу плаваючої точки на безпеку алгоритму електронного підпису Falcon. II Міжнародна науково–практична конференція “Кіберборотьба: розвідка, захист та протидія”. 2024. С. 28–30.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

BKZ	– block korkin–zolotarev reduction, блочна редукція Коркіна–Золотарьова
CPA–to–CCA перетворення	– алгоритм побудови захищеного від атак з адаптивно підібраним шифротекстом механізму інкапсуляції ключів на основі схеми асиметричного шифрування, що є захищеною від атак з адаптивно підібраним повідомленням.
CVP	– closest vector problem, проблема пошуку найближчого вектора
DSA	– стандарт електронного підпису на основі проблеми дискретного логарифмування
ECDSA	– стандарт електронного підпису на основі проблеми дискретного логарифмування в групі точок еліптичної кривої
EUF–CMA	– existential unforgeability under chosen message attack, екзистенціальна непідробність для атак з обраним повідомленням; формальна модель безпеки для електронних підписів
GH(L)	– евристика Ерміта для решітки L; очікувана довжина найменшого вектору на решітці L.
GSA	– geometric series assumption, припущення про геометричний ряд; визначає форму профілю решітки
HKZ	– Hermite–Korkin–Zolotarev reduction, редукція Ерміта–Коркіна–Золотарьова
HSP	– hidden subgroup problem, проблема пошуку прихованої підгрупи
IND–CCA	– indistinguishability under chosen–ciphertext attack, нерозрізнювальність для атак з адаптивно обраним шифротекстом; модель безпеки для механізмів інкапсуляції ключів

IND–CPA – indistinguishability under chosen–plaintext attack, атака з адаптивно обраним повідомленням; модель безпеки для механізмів інкапсуляції ключів

IPSec – набір протоколів для забезпечення захисту даних, що передаються за допомогою протоколу IP

KEM – key encapsulation mechanism, механізм інкапсуляції ключів

LLL – Lenstra–Lenstra–Lovasz reduction, редукція Ленстри–Ленстри–Ловаса

LWE – learning with errors, проблема навчання з помилками

Module–LWE – module learning with errors, проблема навчання з помилками на модульних решітках

NearestPlane – алгоритм найближчої площини Баба для вирішення проблеми CVP

NIST – національний орган зі стандартизації у США

NTRU – Nth Degree Truncated Polynomial Ring

OW–CPA – one–way under chosen plaintext attack, формальна модель безпеки для асиметричного шифрування

PKE – public key encryption, шифрування з відкритим ключем

Ring–LWE – ring learning with errors, навчання з помилками на ідельних решітках

ROM – random oracle model, модель випадкового оракула

RSA – Rivest–Shamir–Adleman, криптографічна система з відкритим ключем, стійкість якої ґрунтується на проблемі факторизації цілих чисел

SIS – short integer solution problem, проблема пошуку малого цілого рішення

SSH – secure shell, протокол захищеної передачі команд через незахищену мережу

SUF–CMA – strong existential unforgeability under chosen message attack, сильна екзистенціальна непідроблюваність для атак з обраним повідомленням

SVP – shortest vector problem, проблема пошуку найменшого вектора

TLS – Transport Layer Security, криптографічний протокол захисту транспортного рівня

QROM – quantum random oracle model, квантова модель випадкового оракула

ZGSA – удосконалена модель GSA для q–арних решіток

ВСТУП

Обґрунтування вибору теми дослідження

Сучасний світ важко уявити без інформаційно–телекомунікаційних мереж. Фактично, уся фінансова та економічна діяльність відбувається з використанням інформаційно–телекомунікаційних мереж. Проте, данні, що поширюються такими мережами, потребують надійного захисту. Зокрема, криптографічного захисту інформації. Сучасні криптографічні протоколи, такі як TLS [1], SSH [2] та IPSec [3], використовують як симетричні криптографічні перетворення, так і асиметричні схеми шифрування.

Більшість сучасних асиметричних криптографічних перетворень, що використовуються для забезпечення безпеки інформації, базуються на таких складних математичних проблемах, як факторизація цілих чисел і дискретне логарифмування. Такі перетворення широко застосовуються в практичних системах захисту даних, включаючи стандарти шифрування, електронних підписів і механізмів обміну ключами. Алгоритми RSA, DSA та системи на основі еліптичних кривих стали невід'ємною частиною сучасних інформаційно–телекомунікаційних технологій. Однак, ці алгоритми мають одну спільну уразливість: вони вразливі до атак, здійснених за допомогою квантових комп'ютерів.

Квантові комп'ютери, що наразі перебувають на стадії активної розробки, пропонують принципово новий підхід до обчислень, що суттєво відрізняється від класичних методів. Одним з найбільш революційних досягнень у цій сфері став квантовий алгоритм, запропонований американським математиком і дослідником Пітером Шором ще в 1994 році. Алгоритм Шора відкрив можливість вирішувати задачу факторизації цілих чисел за поліноміальний час на квантовому комп'ютері. У класичних обчисленнях ця задача є основою безпеки багатьох сучасних криптографічних систем, і її вирішення вимагає експоненційних ресурсів часу та пам'яті. Проте, квантові обчислення суттєво знижують цю складність, що означає потенційну загрозу для всіх систем, заснованих на факторизації.

Після оприлюднення алгоритму Шора інші дослідники також почали адаптувати його підходи для вирішення низки інших проблем, що раніше вважалися складними для класичних комп'ютерів. Зокрема, була розроблена адаптація алгоритму для розв'язання проблеми дискретного логарифмування, що також використовується в багатьох криптографічних системах. Проблема дискретного логарифмування у різних математичних структурах, включаючи мультиплікативні групи цілих чисел та групи точок еліптичних кривих, є основою безпеки таких популярних алгоритмів, як DSA (цифровий підпис) та ECDSA (цифровий підпис на основі еліптичних кривих). Адаптація алгоритму Шора показала, що задача дискретного логарифмування також може бути розв'язана за поліноміальний час на квантовому комп'ютері. Це ставить під загрозу всі сучасні криптографічні системи, що базуються на дискретному логарифмуванні, оскільки їх безпека перестане відповідати необхідному рівню стійкості в умовах появи квантових обчислювальних машин.

Таким чином, виникає нагальна необхідність розробки нових криптографічних перетворень, стійких до квантових атак. Це спонукає дослідників усього світу шукати нові математичні основи для криптографічних систем, які могли б бути стійкими квантових обчислень і забезпечити надійний захист інформації у майбутньому квантовому світі.

Протягом тривалого часу квантові комп'ютери залишалися більше теоретичною концепцією, яка існувала лише у вигляді математичних моделей та абстракцій. Загроза для класичних криптографічних систем, пов'язана з потенційними можливостями квантових комп'ютерів, здавалася далекою і суто гіпотетичною. Проте, за останнє десятиріччя в галузі квантових обчислень було досягнуто вражаючого прогресу, який призвів до змін у сприйнятті цієї загрози. Квантові обчислювальні технології почали активно розвиватися, і сьогодні існують реальні прототипи квантових комп'ютерів, здатні виконувати обчислення на невеликій кількості кубітів.

Зокрема, кілька провідних технологічних компаній зробили значні інвестиції в дослідження та розробку квантових обчислень, серед яких особливо

варто відзначити компанію IBM. Вона виділяється своєю комплексною та системною стратегією розробки квантових чіпів. IBM є однією з компаній, яка не тільки активно займається теоретичними дослідженнями в цій галузі, але й робить значні кроки в напрямку створення практичних рішень для квантових обчислень. Вони поступово покращують свої квантові процесори, і кожен новий прототип відрізняється підвищеною кількістю кубітів, кращою стабільністю та продуктивністю.

Згідно з їх офіційними планами та звітами, зокрема зі звітом, оприлюдненим у 2023 році, компанія IBM має амбітну мету досягти значного прориву в розробці квантових процесорів. Їх стратегія спрямована на створення робочих квантових чіпів, що містять не менше 5000 кубітів, до 2030 року. Якщо ці плани втіляться в життя, квантові комп'ютери такого масштабу зможуть вирішувати задачі, які знаходяться далеко за межами можливостей сучасних класичних комп'ютерів. Це означає, що загроза для сучасних криптографічних систем стане не лише гіпотетичною, але й цілком реальною.

Така перспектива спричиняє велике занепокоєння в сфері інформаційної безпеки. Адже традиційні криптографічні алгоритми, що ґрунтуються на складності таких задач, як факторизація цілих чисел чи дискретне логарифмування, можуть бути легко зламані квантовими комп'ютерами, здатними працювати з великою кількістю кубітів. Саме тому криптографічні дослідження сьогодні фокусуються на пошуку нових методів захисту інформації, які зможуть забезпечити безпеку навіть у епоху квантових обчислень.

Зважаючи на такий стрімкий розвиток, NIST США провели відкритий конкурс для створення нових квантово–стійких стандартів електронного підпису та механізмів інкапсуляції ключів. За результатами цього конкурсу було обрано чотири кандидати на стандартизацію. При чому, три з них ґрунтуються на складних проблемах з теорії решіток.

У той же час, в Україні паралельно відбувалася розробка власних квантово–стійких стандартів інкапсуляції ключів та електронного підпису. За результатами досліджень були створені стандарти ДСТУ 8961:2019 та ДСТУ

9212:2023, які також ґрунтуються на проблемах з теорії решіток. Особливістю українських стандартів, у порівнянні з стандартами NIST, є наявність додаткових рівнів безпеки для 384 та 512 біт безпеки. Використання таких нестандартних для світової практики рівнів безпеки вимагає більш детальних досліджень та обережності при виборі загальносистемних параметрів.

Розвиток криптографії на решітках призвів до кращого розуміння складності проблем в теорії решіток. Проте, багато моделей, що застосовується на практиці, досі використовують спрощені представлення про решітки, що призводить до деякого зміщення у оцінках безпеки. Тож, розробка моделей оцінки безпеки електронних підписів та механізмів інкапсуляції ключів на решітках є важливим та актуальним напрямком досліджень.

Зв'язок роботи з науковими програмами, планами, темами

Дисертаційні дослідження проводились в рамках науково–дослідницьких робіт: № 28–20 «Механізми та засоби асиметричних криптоперетворень у постквантовий період» (Шифр «Квант–2021»), № 34–21 «Методи та алгоритми постквантових криптоперетворень, їх стандартизація та впровадження» (Шифр «Квант–2022»), № 09–22 «Методи та засоби генерування псевдовипадкових та випадкових послідовностей на основі класичних та квантових ефектів» (Шифр «Квант–2023»). № 10–23 «Методи та алгоритми постквантових перетворень типу електронний підпис, їх стандартизація та впровадження» (Шифр «Квант–2024»).

Результати дисертаційних досліджень були використані при підготовці та проведенні лабораторних робіт по дисципліні «Прикладна Криптологія» для спеціальності «Кібербезпека».

Мета і завдання дослідження

Мета дослідження: аналіз та розробка методів та засобів для підвищення захищеності асиметричних криптографічних систем на решітках для постквантового періоду від існуючих та потенційних атак.

Для досягнення поставленої мети були розв'язані такі задачі:

- Вибір та обґрунтування моделей редукції решіток, які враховують існуючі відомості щодо поведінки базисів під час процесу редукції решіток та оцінка точності існуючих моделей редукції решіток.
- Дослідження впливу моделей редукції решіток на складність атак на криптографічні перетворення на решітках та удосконалення методики оцінювання криптографічних перетворень на решітках.
- Аналіз криптографічних перетворень на основі решіток у формальних моделях безпеки, які враховують потенціал класичних та квантових атак.
- Дослідження впливу обчислень з плаваючою точкою на безпеку алгоритмів електронного підпису на решітках.

Об’єкт дослідження: процеси оцінки захищеності існуючих та перспективних асиметричних криптографічних перетворень від класичних та квантових атак.

Предмет дослідження: методи та моделі забезпечення захищеності асиметричних криптографічних систем на решітках від існуючих та потенційних квантових атак.

Методи дослідження: методи комп’ютерного моделювання, теорії ймовірностей та математичної статистики. Експериментальні дослідження та чисельні розрахунки виконувалися на мовах програмування C++ та Python.

Наукова новизна отриманих результатів

1. Вперше виконано кількісне порівняння точності моделей редукції решіток із застосуванням метрики середньоквадратичної помилки для моделі GSA (Geometric Series Assumption) та симуляторів редукції решіток. Попередні дослідження фокусувалися на якісних або суто теоретичних оцінках якості роботи моделей. Отримані оцінки дозволяють кількісно оцінювати якість роботи симуляторів в залежності від параметрів решіток для оцінки захищеності від класичних та квантових атак.

2. Вперше було отримано узагальнений доказ IND–CCA безпеки перетворень, що використовуються в стандарті ДСТУ 8961:2019, у моделі

квантового випадкового оракула. Попередні дослідження не вивчали IND–CCA безпеку перетворень ДСТУ 8961:2019 у моделі квантового випадкового оракула.

3. Удосконалено методику оцінювання складності криптографічної задачі SIS (Shortest Integer Solution), що відрізняється від існуючих тим, що у даній методиці враховується алгебраїчна структура решіток під час аналізу процесів редукції для оцінки параметрів та характеристик на їх основі атак.

4. Отримали подальший розвиток обґрунтування оцінки атаки відновлення ключів, що використовує обчислення з плаваючою крапкою, для алгоритмів електронного підпису на основі решіток, що дало змогу підвищити безпеку електронних підписів на решітках.

Практичне значення отриманих результатів

1. Уточнено оцінки безпеки криптографічних перетворень на решітках, які підтверджують рівні стійкості квантово–стійких стандартів електронного підпису (ДСТУ 9212:2023) та механізмів інкапсуляції ключів (ДСТУ 8961:2019), що обґрунтовує їх застосування як національних стандартів.

2. Розроблено програмне забезпечення, яке дозволяє:

- Оцінювати складність атак вкладення та декодування для проблем LWE та NTRU при процесі практичного оцінювання безпеки криптографічних перетворень на решітках (у тому числі для стандартів ДСТУ 8961:2019, ДСТУ 9212:2023).

- Оцінювати складність проблеми SIS з врахуванням структури решіток (у тому числі для міжнародного проекту стандарту Crystlas–Kyber та федерального стандарту FIPS 203)

- Проводити моделювання редукції решіток (у тому числі при дослідженні безпеки стандартів ДСТУ 8961:2019, ДСТУ 9212:2023)

Результати дисертаційних досліджень впровадженні у Приватному акціонерному товаристві «Інститут інформаційних технологій», м. Харків та були використані при обчисленні вхідних та вихідних тестових векторів стандартів ДСТУ 8961:2019, ДСТУ 9212:2023. Математичні моделі та аналітичні

співвідношення знайшли практичне застосування в ХНУ імені В. Н. Каразіна на кафедрі БІСТ в дисциплінах першого рівня вищої освіти “Прикладна криптологія”, другого рівня освіти “Криптографічні методи в кібербезпеці” та третього рівня освіти «Математичні методи в кібербезпеці» при проведенні лабораторних робіт.

Особистий внесок здобувача

У наукових статтях, опублікованих у співавторстві, автору належать наступні результати:

– Обґрунтування точності асимптотичних оцінок фактора Ерміта при аналізі редукції решіток з криптографічно значущими розмірностями.

– Порівняння якості роботи симуляторів редукції решіток на експериментальних даних.

– Уточнено оцінки атак вкладення та декодування для проблем NTRU та LWE.

– Удосконалено метод оцінки атак на проблему SIS.

– Отримано доказ безпеки ДСТУ 8961:2019 у моделі квантового випадкового оракула.

– Для атаки відновлення ключів на Falcon знайдена оптимальна кількість підписів.

– Удосконалено існуючі аналітичні оцінки безпеки для електронних підписів Falcon та Crystals–Dilithium та механізмів інкапсуляції ключів ДСТУ 8961:2019 та Crystals–Kyber.

Апробація результатів дисертації здійснювалася на І міжнародній науково–технічній коференції «Системи і технології зв’язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку» 25–26 листопада 2021 року, м. Київ; 8–й Міжнародній конференції «Комп’ютерне моделювання в наукоємних технологіях» (КМНТ–2022) *20–22 квітня 2022 року*, м. Харків; IV Міжнародної науково–технічної конференції «Системи і технології зв'язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку» 28

листопада 2024 року, м. Київ, ІІ Міжнародній науково–практичній конференції “Кіберборотьба: розвідка, захист та протидія” 23–24 квітня 2024 року.

Публікації. Основні наукові результати за темою дисертації опубліковані у 10 статтях, із яких 7 статей у фахових наукових журналах, які входять до переліку МОН України, та 3 статті в науковому зарубіжному виданні, включеному до наукометричної бази Scopus.

Структура та обсяг дисертації. Дисертація містить вступ, п’ять розділів, висновки, 3 додатки, список використаних джерел. Загальний обсяг дисертації складає 188 сторінки, у тому числі 24 сторінок додатків, 9 сторінок списку використаних джерел в кількості 89 найменувань.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ ТЕНДЕНЦІЙ В КВАНТОВО–СТІЙКІЙ КРИПТОГРАФІЇ

1.1. Огляд основних напрямків квантово–стійкої криптографії

Технологія квантових обчислень є одним з найбільш потужних інструментів для проведення досліджень у різних областях науки. Використання таких властивостей квантових систем, як суперпозиція та заплутаність станів, дозволяє пришвидшувати обчислення за рахунок природнього паралелізму [4]. На даний момент вже знайдені алгоритми, що дозволяють вирішувати такі задачі як факторизація цілих чисел та дискретне логарифмування [5] за поліноміальний час, що значно швидше за існуючі аналоги для класичних систем.

Оскільки більшість стандартизованих асиметричних криптосистем, таких як RSA [6], DSA [7] та ECDSA [8], базуються на задачах, для яких були знайдені швидкі та ефективні квантові алгоритми, розробка та аналіз нових квантово–стійких криптосистем є однією з найважливіших задач в сучасній криптології.

Квантово–стійка криптографія, за визначенням [9], є розділом криптографії, що вивчає криптографічні перетворення, які мають захист від атак на квантових комп’ютерах. Проте, питання того, які криптографічні системи є стійкими до квантових атак, а які ні, не має чіткої відповіді, оскільки досі не має повного розуміння можливостей квантових комп’ютерів [4]. У науковій спільноті домінує думка, що квантові комп’ютери не здатні вирішувати NP–повні задачі за поліноміальний час [10]. Це є необхідною умовою для існування квантово–стійкої криптографії.

Існуючі квантові алгоритми були отримані за допомогою застосування доволі обмеженої кількості технік, що не можуть бути реалізовані за поліноміальний час на класичних комп’ютерах [4].

На рисунку 1 показані основні ідеї та можливості квантових комп’ютерів відносно криптографічних задач. З рисунка видно, що потужність квантових обчислень випливає з двох «квантових» ідей.

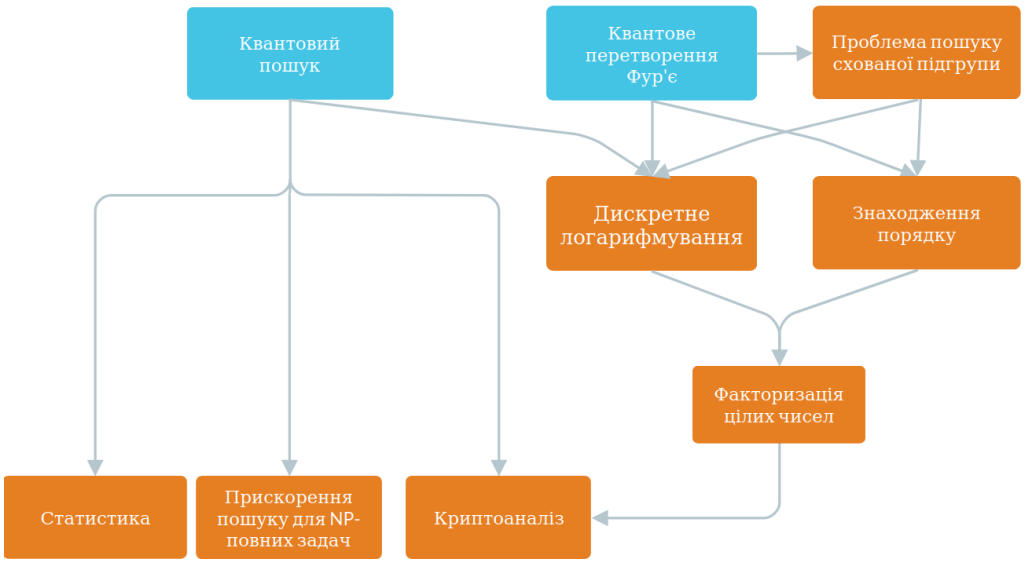

Рис. 1.1. Основні ідеї (сині блоки) та практичні застосування (червоні блоки) квантових алгоритмів

Квантовий пошук забезпечує можливість ітеративного наближатися до точки у просторі станів без знання будь–якої інформацію про цю точку. Вперше ця ідея була реалізована у алгоритмі Гровера [11], що реалізує пошук у несортованій базі даних за $O(\sqrt{N})$. Згодом ця ідея була узагальнена для різноманітних задач [12]. У криптографії квантовий пошук широко використовується для аналізу симетричних криптопримітивів, таких як блокові шифри та геш функції [13, 14]. Проте, для асиметричних криптопримітивів алгоритм Гровера також може бути адаптований. Прикладом є так звані алгоритми просіювання [15] (англ. sieving algorithms), які використовуються для вирішення задачі пошуку найменшого вектора на решітках. Для них застосування алгоритму Гровера дає пришвидшення на деякий експоненціальний фактор в залежності від алгоритму.

Квантова вибірка Фур’є є технікою квантових обчислень, що використовує квантове перетворення Фур'є для аналізу функцій або для вирішення задач, що важко піддаються класичним методам. Одним з перших алгоритмів, що використав цю техніку, був алгоритм Шора для факторизації цілих чисел [5]. Сутність квантового перетворення Фур’є полягає у зміні обчислювального

базису на базис, у якому зручно проводити обчислення з суперпозицією станів. Спершу готується квантовий стан, який представляє всі можливі вхідні дані функції або проблеми, яку потрібно вирішити. До квантового стану застосовується квантове перетворення Фур'є. Це перетворення переводить інформацію про функцію з часового простору до частотного простору, дозволяючи виявити приховані періодичні структури або властивості. Після застосування квантового перетворення Фур'є виконується вимірювання квантового стану. Результати вимірювання надають інформацію про частотні компоненти функції, що допомагає у розв'язанні задачі. Можливість таких обчислень є основною причиною переваги квантових алгоритмів над класичними для ряду важливих задач.

Найбільш потужним застосуванням квантового перетворення Фур'є є вирішення задачі схованої підгрупи (англ. Hidden Subgroup Problem – HSP) для абелевих груп. Більшість теоретико–числових проблем, на які спирається сучасна стандартизована криптографія, можливо звести до цієї проблеми [16]. Формально, проблема схованої підгрупи визначена наступним чином. Нехай задана деяка група *G*, її підгрупа *H* та деяка скінченна множина X. Функція $f: G \to X$ ховає підгрупу *H* якщо для усіх $g_1, g_2 \in G$ виконується $f(g_1) = f(g_2)$ тільки якщо $g_1 H = g_2 H$. Проблема схованої підгрупи полягає у тому, щоб для заданих G, X та функції f, що приховує деяку невідому підгрупу H, визначити генератори підгрупи *H*. Для абелевих груп цю проблему можливо вирішити за поліноміальний час на квантовому комп'ютері за допомогою квантового перетворення Фур'є [17]. Алгоритм Шора є прикладом такого підходу.

Для не абелевих груп ситуація дещо складніша. Досі невідомо як ефективно вирішувати проблему на квантових комп'ютерах. У дослідженні [18] було показано, що HSP для не абелевих груп може бути вирішена за поліноміальну кількість викликів до оракула, що реалізує функцію f, проте запропонований підхід працює за експоненційний час. Тобто, теоретично, не існує заборон щодо існування поліноміального алгоритму для HSP в не абелевних групах. Проте, такий алгоритм має використовувати техніки, що

принципово відрізняються від тих, що відомі на сьогоднішній час. У науковій спільності поява такий нових технік вважається доволі малоймовірною подією [4].

Втім, не зважаючи на те, що для загального випадку HSP не відомо алгоритму, для деяких не абелевих груп вдалося отримати деяке прискорення. У дослідженні [19] було запропоновано квантовий алгоритм для HSP у діедральній групі (група симетрій правильного багатокутника), що має субекспоненційний час. У дослідженні [20] було запропоновано поліноміальний алгоритм для групи HSP у групі Гейзенберга, не зважаючи на те, що вона є не абелевою. Тож, хоча питання складності HSP у не абелевих групах потребує додаткового вивчення, схоже, що криптографія, що ґрунтується на проблемах, що можуть бути зведені до HSP у не абелевих групах, є перспективним кандидатом на роль постквантової криптографії.

У звіті Європейського Інституту Телекомунікаційних стандартів (ETSI) [21] було виділено п’ять перспективних напрямків досліджень у постквантовій криптографії:

- Криптографія на решітках. Цей напрямок вивчає криптографічні схеми, що ґрунтуються на складності вирішення проблем з теорії решіток, таких як пошук найменшого вектора на решітці та пошук найближчого вектора на решітці. Пошук найменшого вектора може бути зведений до HSP у діедральній групі [19], що є додатковим аргументом безпеки.
- Криптографія на багатовимірних перетвореннях. Цей напрямок використовує ідею складності вирішення систем поліноміальних рівнянь над скінченним полем. Використовується переважно для електронних підписів. Відомі конструкції для схем асиметричного шифрування, нажаль, часто виявлялися не безпечними [22].
- Криптографія на завадостійких кодах. Цей напрямок є історично одним з перших і з’явився ще до того, як сила квантових обчислень була усвідомлена. Ґрунтується на складності декодування синдрому випадкових завадостійких кодів. Ця проблема є доволі схожою на проблеми з теорії решіток.

Проте, на відміну від останніх, найкращі відомі атаки носять комбінаторний характер [23], що дає гарне розуміння безпеки.

- Криптографія на геш функціях. Цей напрямок ґрунтується на складності знаходження колізій в геш функціях. Наразі відомі тільки електронні підписи на геш функціях. Побудова асиметричного шифрування на геш функціях є відкритим не вирішеним питанням [24]. Оскільки можливості квантових комп'ютерів є доволі обмеженими для аналізу симетричних криптопримітивів, то це дає привід вважати цей напрям постквантовим.
- Криптографія на ізогеніях еліптичних кривих. Цей напрям є доволі маловивченим. У 2022 році була представлена класична поліноміальна атака відновлення ключа на механізм інкапсуляції ключів SIKE [25]. Поява цієї атаки поставила під питання умови складності вирішення теоретико–числових проблем, що лежать в основі криптографічних схем цього напрямку. Для криптографії на ізогеніях еліптичних кривих необхідні додаткові фундаментальні дослідження, оскільки вона є найменш вивченим напрямком досліджень.

Американський Національний Інститут Стандартизації NIST у 2016 року оголосив про проведення конкурсу на створення постквантових стандартів для асиметричного шифрування, механізмів інкапсуляції ключів та електронних підписів.

За результатами 3 етапу конкурсу NIST PQC було визначено фіналістів. Як серед схем асиметричного шифрування, так і серед електронних підписів, фіналістами стали схеми на решітках, оскільки окрім гарних теоретичних аргументів безпеки, у порівнянні з іншими кандидатами, вони мали кращі технічні характеристики. Це робить питання дослідження схем шифрування на решітках надзвичайно актуальним.

1.2. Аналіз вимог до квантово–стійкої криптографії

Свої вимоги NIST описали у документі [26]. Вимоги поділені на декілька класів: вимоги з безпеки, техніко–економічні вимоги, техніко–експлуатаційні вимоги, інше.

Однією з основних вимог є відкритість алгоритмів, відповідно до принципу Кіргоффа. Алгоритми мають бути публічно відомими та добре дослідженими. Крім того, теоретико–числові проблеми, на яких ґрунтуються криптографічні перетворення, мають мати добре обґрунтовані оцінки складності. NIST не конкретизує які саме обґрунтування вважаються достатніми. Проте, аналіз літератури показав [26, 24, 27, 28], що можливо виділити наступні класи аргументів безпеки теоретико–числових проблем у криптографії:

- Практичні аргументи. Сутність таких аргументів полягає у тому, що для відповідної складної проблеми усі відомі атаки вимагають надто великих обчислювальних ресурсів. Наявність таких аргументів є обов’язковою.
- Теоретичні аргументи. Існує можливість зведення відомих складних проблем до заданої проблеми за поліноміальний час. Для криптографічних задач цікаві два типи таких зведень: зведення від найгіршого до середнього та зведення від середнього до середнього. У першому випадку гарантується, що складність вирішення задачі у середньому є не меншою, ніж для певної задачі у найгіршому випадку. У другому випадку, відповідно, складність вирішення задачі у середньому є такою ж, як і для деякої складної задачі.

Теоретичні аргументи безпеки показують, що якщо відповідна криптографічна теоретико–числова проблема може бути вирішена за поліноміальний час, то буде існувати метод для вирішення безлічі інших проблем не тільки у криптографії, а у математиці взагалі. Практичні аргументи безпеки показують захищеність лише від існуючих атак, у той час як теоретичні аргументи безпеки показують, що поява нових більш потужних атак є доволі малоймовірною подією. Теоретичні аргументи безпеки здебільшого характерні для криптографії на решітках та криптографії на завадостійких кодах.

NIST також вимагає від схем асиметричного шифрування та механізмів інкапсуляції ключів безпеку в формальній моделі безпеки IND–CCA та від електронних підписів безпеку у формальній моделі EUF–CMA. Але, варто зазначити, що на практиці докази у таких моделях важко отримати, якщо застосовуються геш функції у якості компонента відповідного криптографічного перетворення. Для доказів у таких випадках використовується модель (квантового) випадкового оракула, у межах якої геш функції замінюються на ідеалізовані аналоги – випадкові оракули. NIST обмежує кількість шифротекстів/підписів для атаки числом 2^{64}.

Для усіх фіналістів конкурсу NIST існують докази у моделі квантового випадкового оракула. У той же час для українського постквантового стандарту ДСТУ 8961:2019 не було опубліковано відповідних доказів, що стає всупереч вимогам NIST. Одним з внесків даної дисертаційної роботи є аналіз ДСТУ 8961:2019 у моделі квантового випадкового оракула. У четвертому розділі роботи отримано оцінки безпеки для перетворень стандарту ДСТУ 8961:2019. Більш детально сутність формальних моделей безпеки буде розглянута у наступному підрозділі.

Визначення конкретних рівнів безпеки для постквантових перетворень є доволі складною задачею, адже можливості квантових комп'ютерів, як було розглянуто у першому підрозділі на прикладі задачі схованої підгрупи, ще не до кінця вивчені. NIST вводить означення рівнів безпеки на основі безпеки симетричних криптопримітивів. Будь–яка атака у межах відповідної для криптографічного перетворення формальної моделі безпеки (IND–CCA або EUF–CMA) вимагає кількість обчислювальних ресурсів, яку можливо порівняти з кількістю ресурсів для пошуку:

- 128–бітного ключа блочного шифру (наприклад, для AES–128) для 1 рівня безпеки.
- колізій 256–бітної геш функції (наприклад, для SHA–256/SHA3–256) для другого рівня безпеки.

- 192–бітного ключа блочного шифру (наприклад, для AES–192) для третього рівня безпеки.
- колізій 384–бітної геш функції (наприклад, для SHA–384/SHA3–384) для четвертого рівня безпеки.
- 256–бітного ключа блочного шифру (наприклад, для AES–256) для п’ятого рівня безпеки.

Якщо вважати алгоритм Гровера за найкращий алгоритм для прискорення криптоаналізу блочних шифрів та геш функцій, то в бітах кожен рівень можливо оцінити наступним чином:

- 1 рівень – 128 біт для класичних комп’ютерів та 64 біта для квантових комп’ютерів.
- 2 рівень – 128 біт для класичних комп’ютерів та 64 біта для квантових комп’ютерів.
- 3 рівень – 192 біт для класичних комп’ютерів та 96 біт для квантових комп’ютерів.
- 4 рівень – 192 біт для класичних комп’ютерів та 96 біт для квантових комп’ютерів.
- 5 рівень – 256 біт для класичних комп’ютерів та 128 біт для квантових комп’ютерів.

В Україні від криптографічних перетворень на рівні стандартів часто вимагається більший рівень безпеки в бітах, ніж передбачено NIST. Для вирішення цієї проблеми, у дослідженнях [27–30] було запропоновано шостий та сьомий рівень безпеки, які передбачають 384 біта безпеки та 512 біт безпеки для класичних комп’ютерів та відповідно 192 та 256 біт безпеки для квантових комп’ютерів.

При оцінці захищеності від атак на квантових комп’ютерах також варто розуміти, що можливості конкретного квантового комп’ютера залежать не тільки від кількості кубітів, що в ньому реалізовано, але й від максимальної кількості операцій, що може бути обчислено за один запуск. Цей показник залежить від декількох факторів [4]:

- Точність операцій. Ідеалізовані моделі квантових обчислень передбачають ідеальну точність операцій. В реальних існуючих квантових комп'ютерах точність операцій досягає лише 10^{-3}. Це призводить до накопичення помилок з часом, що обмежує максимальну кількість операцій, або вимагає введення додаткового механізму корекції помилок.

- Ізольованість системи. Квантові обчислення передбачають використання суперпозиції станів кубітів. Такий стан системи легко порушити стороннім впливом. Тож, виникає необхідність сильної ізоляції системи і навіть ізоляції окремих кубітів. Максимальний час зберігання стану суперпозиції для кожного квантового комп'ютера є обмеженим, тому це накладає обмеження на кількість операцій.

NIST рекомендує використовувати обмеження у моделі безпеки від 2^{40} до 2^{64} квантових операцій.

До додаткових вимог можливо віднести наступні:

- Властивість «Perfect Forward Secrecy». Ця властивість забезпечує, що навіть у разі компрометації приватного ключа сервера минулі сеансові ключі залишаються захищеними. Вона ґрунтується на використанні асиметричних криптографічних примітивів, які дозволяють створювати протоколи узгодження ключів із такими характеристиками. Ця властивість є компромісом між високими вимогами до безпеки та практичною ефективністю. Наприклад, механізми інкапсуляції ключів та асиметричні схеми шифрування, які потребують значного часу для генерації ключових пар, не підходять для реалізації таких протоколів.

- Стійкість до атак через побічні канали. У випадку витоку інформації через побічні канали під час обчислень, вплив такого витоку на загальну безпеку системи повинен бути мінімальним. При цьому захищена реалізація має забезпечувати високий рівень продуктивності і не значно поступатися еталонній реалізації за швидкістю роботи.

- Захист від багатоключових атак. Схема має залишатися стійкою навіть у випадку, якщо зловмисник отримав велику кількість шифротекстів,

зашифрованих за допомогою різних ключів. Такі атаки, як правило, виходять за межі стандартної моделі безпеки і потребують окремого аналізу.

- Розміри ключів, шифротекстів та підписів мають бути оптимізованими для популярних інтернет–протоколів. Криптографічні дані повинні відповідати розмірам пакетів, що використовуються в таких протоколах, як TCP, UDP, HTTP/2 та інших, щоб забезпечити їхню ефективну передачу. У випадку, якщо шифротекст, підпис або ключ перевищують розмір одного пакету, це може призвести до розбиття даних на декілька частин, що, у свою чергу, розширює можливості для атак, таких як атаки на порядок передачі або повторення даних. Крім того, це створює додаткові незручності, пов’язані з підвищеними вимогами до пропускної здатності мережі, збільшенням затримок та зростанням складності обробки пакетів.

- Ефективність реалізації операцій з відкритим і приватним ключем. Операції, що виконуються з використанням відкритого ключа (шифрування, перевірка підпису), а також операції з приватним ключем (дешифрування, підписання), повинні бути розроблені з урахуванням високої ефективності як у програмному, так і в апаратному середовищах. Це забезпечить швидке виконання криптографічних алгоритмів навіть в умовах обмежених ресурсів або при обробці великої кількості запитів.

- Для програмних реалізацій бажано використовувати конструкції, які дозволяють максимально ефективно задіяти векторизовані інструкції сучасних процесорів, таких як SSE, AVX, AVX2, AVX512 для архітектури x86/x64 та NEON для ARM. Використання таких інструкцій дає змогу значно прискорити обчислення, особливо для алгоритмів з високим рівнем паралелізму, таких як модульна арифметика, матричні операції чи перетворення Фур’є.

- В апаратному середовищі важливо забезпечити підтримку оптимізованих реалізацій через криптографічні акселератори, що використовуються в сучасних процесорах, таких як Intel AES–NI або ARM Cryptography Extensions. Векторизовані операції дозволяють не лише збільшити

швидкість виконання, але й знизити енергоспоживання, що є критичним для мобільних пристроїв та систем з обмеженими ресурсами.

- Оптимізація розміру ключів та надійність роботи криптосистеми. Розмір ключів має бути якомога меншим для забезпечення ефективного використання ресурсів, але при цьому відповідати сучасним вимогам безпеки, встановленим для криптосистеми. У разі, якщо схема не гарантує повної точності дешифрування у загальному випадку, кількість помилок при дешифруванні повинна бути мінімальною, а також має бути забезпечена можливість багаторазового шифрування з гарантованим успіхом.

1.3. Доказова безпека в сучасній криптографії

Концепція доказової безпеки була запропонована у дослідженні [31] для асиметричного шифрування. Доказова безпека є методологічним підходом, сутність якого полягає у наступному. Спочатку створюється формальне визначення мети, якої потрібно досягти. Далі формалізується криптографічна конструкція і надається формальний доказ з деякими теоретико–числовими припущеннями складності. У контексті асиметричної криптографії прикладами таких теоретико–числових проблем можуть слугувати факторизація цілих чисел та пошук найменшого вектора на решітці [26].

Формальний доказ має показувати, що для будь–якого супротивника A, що може побудувати ефективну атаку на схему шифрування, існує також супротивник B, що може використати A для вирішення складної теоретико–числової проблеми [32]. Тож, формальний доказ зводить більш складну задачу криптоаналізу до задачі вивчення теоретико–числової проблеми, на якій ґрунтується безпека схеми.

На сьогоднішній день формальні докази є стандартним засобом аналізу схем шифрування [33]. Можливо сказати, що формальний доказ показує наскільки відрізняється складність атаки схеми шифрування від складності вирішення теоретико–числової проблеми, що лежить в основі [32]. Якщо для супротивник B може звести задачу для іншої задачі для супротивника A таким

чином, щоб час роботи супротивника B не сильно відрізнявся від часу роботи A, то таке зведення називається сильним (англ. tight). У цьому випадку можливо вважати, що складність атаки схеми шифрування точно дорівнює складності вирішення теоретико–числової проблеми і загальносистемні параметри для схеми шифрування можливо обирати з огляду на це.

Згідно визначення [24], протокол інкапсуляції ключів є трійкою алгоритмів $(Gen, Encaps, Decaps)$, де:

- $Gen: 1^\lambda \to (pk, sk)$ – поліноміальний ймовірнісний алгоритм генерації ключової пари. Приймає параметр безпеки 1^λ та повертає ключову пару (pk, sk).
- $Encaps: pk \to (K, C)$ – поліноміальний ймовірнісний алгоритм інкапсуляції ключів. Приймає публічний ключ pk і повертає випадковий ключ K та його інкапсуляцію C.
- $Decaps: (sk, C) \to \{K, \perp\}$ – детермінований поліноміальний алгоритм декапсуляції ключів. Приймає секретний ключ sk та інкапсуляцію ключа C і повертає ключ K у разі вдалої декапсуляції та символ помилки $\perp$ у разі виникнення помилок.

На механізми інкапсуляції ключів накладається вимога коректності:

$$\Pr\left[Decaps(sk, C) = K \middle| \begin{matrix}(pk, sk) \leftarrow Gen\left(1^\lambda\right) \\ (K, C) \leftarrow Encaps(pk)\end{matrix}\right] = 1 - negl(\lambda) \qquad (1.1)$$

Де $negl(\lambda)$ позначає незначну функцію (тобто функцію, що зменшується швидше за будь–який поліном) від параметра λ.

Протоколи інкапсуляції ключів використовуються для побудови складних інтерактивних криптографічних протоколів. Зокрема, на основі протоколу інкапсуляції ключів можливо побудувати протокол узгодження ключів [24,34,35,36]. В залежності від вимог до сторін протоколу, існують різні доказово безпечні конструкції. На рисунку 1.2 наведено протокол узгодження ключів з взаємною автентифікацією сторін на основі CCA безпечного протоколу

інкапсуляції ключів, що задається трійкою алгоритмів $(Gen, Encaps, Decaps)$. Наведений протокол є доказово безпечним у моделі Канетті–Кравчека [37], яка є достатньо популярною моделлю для оцінки безпеки протоколів узгодження ключів. У протоколі узгодження ключів передбачається, що сторони протоколу (Користувач А та Користувач Б) мають довгострокові ключі (pk_A, sk_A) та $(pk_Б, sk_Б)$.

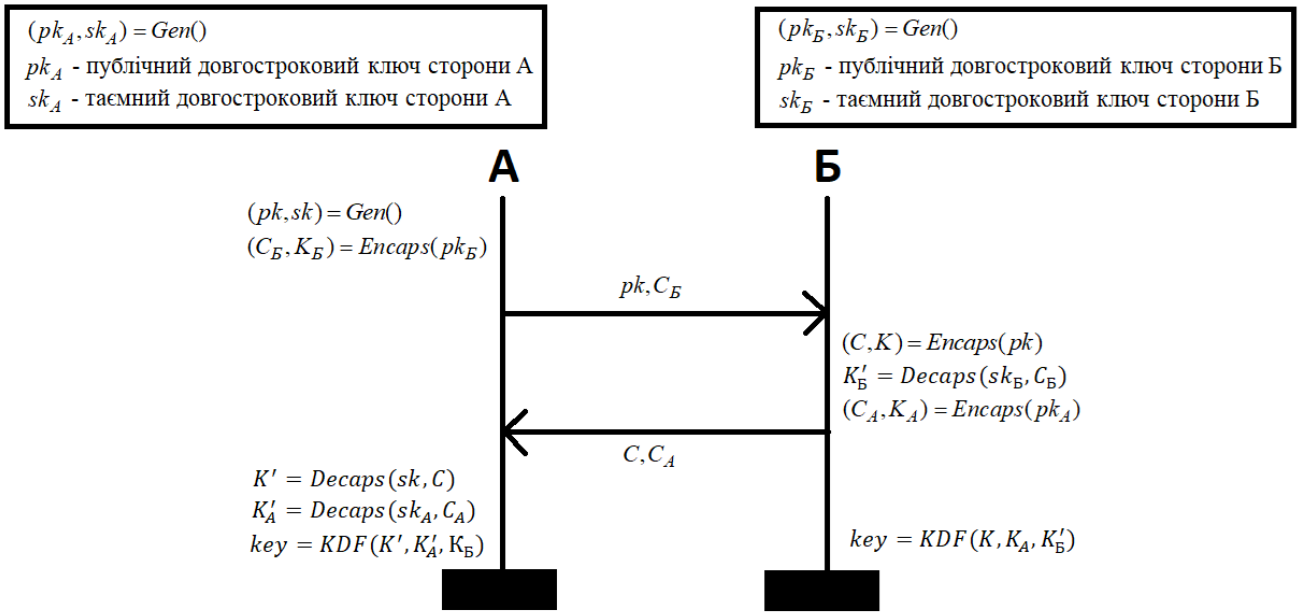

Рис. 1.2. Протокол узгодження ключів на основі механізму інкапсуляції ключів

Ключ pk_A є відомим для Користувачу Б і ключ $pk_Б$ відомий Користувачу А. Ініціатором протоколу є Користувач А. Протокол складається з наступних кроків.

1. Користувач А генерує асиметричну пару короткострокових ключів $(pk, sk) = Gen()$ та обчислює за допомогою публічного ключа $pk_Б$ ключ інкапсуляції $K_Б$ для Користувача Б та інкапсуляцію $C_Б$.

2. Користувач А надсилає пару $(pk, C_Б)$ до Користувача Б.

3. Користувач Б отримує короткостроковий відкритий ключ pk та інкапсуляцію $C_Б$. За допомогою публічного ключа pk Користувач Б обчислює короткостроковий ключ K та його інкапсуляцію C. За допомогою секретного ключа $sk_Б$ Користувач Б відновлює ключ інкапсуляції $K_Б$ та за допомогою

публічного ключа pk_A обчислює ключ інкапсуляції K_A для Користувача А та інкапсуляцію C_A.

4. Користувач Б надсилає пару (C, C_A) до Користувача А.

5. Користувач А відновлює ключ K за допомогою таємного короткострокового ключа sk та ключ K_A за допомогою таємного ключа sk_A.

6. Після кроків 1–5 кожна з сторін має три ключі: $K, K_A, K_Б$. На останньому кроці протоколу узгодження ключів обчислюється спільний таємний ключ key за допомогою деякої криптографічної геш функції KDF.

Для подальшого аналізу наведемо формальне визначення схеми асиметричного шифрування. Схема асиметричного шифрування є трійкою алгоритмів (Gen, Enc, Dec), де:

- $Gen: 1^\lambda \to (pk, sk)$ – поліноміальний ймовірнісний алгоритм генерації ключової пари, що приймає параметр безпеки 1^λ та повертає ключову пару (pk, sk).
- $Enc: (pk, m) \to C$ – поліноміальний ймовірнісний алгоритм шифрування, що приймає публічний ключ pk, повідомлення m та повертає шифротекст C.
- $Dec: (sk, C) \to \{m, \perp\}$ – детермінований поліноміальний алгоритм розшифрування. Приймає секретний ключ sk, шифртекст C та повертає повідомлення m у разі вдалої декапсуляції та символ помилки $\perp$ у разі виникнення помилок.

При аналізі довільного механізму інкапсуляції ключів $KEM = (Gen, Encaps, Decaps)$ зручно виділити деяку схему асиметричного шифрування $PKE = (Gen, Enc, Dec)$ і представити алгоритми KEM як конструкцію, що робить виклики до алгоритмів PKE. При такій декомпозиції можливо окремо аналізувати безпеку PKE і окремо безпеку перетворення (CPA–to–CCA перетворення), що дозволяє отримати алгоритми KEM. Зазвичай, у якості CPA–to–CCA перетворення обирається перетворення, що має доказову безпеку. Прикладами таких перетворень є перетворення Фуджісакі–Окамото [38] та

перетворення Дента [39]. У четвертому розділі використовується така декомпозиція для аналізу стандарту ДСТУ 8961:2019.

На практиці часто буває важко провести формальний аналіз через використання геш функцій, оскільки геш функції можуть складним не передбачуваним чином взаємодіяти з конструкцією схеми. Модель випадкового оракула є евристикою, у межах якої кожна геш функція замінюється на ідеалізований варіант – випадкового оракула.

Модель випадкового оракула була вперше запропонована в роботі [40]. З тих пір ця модель була суттєво дороблена. Наразі вона є стандартним інструментом при роботі з асиметричними перетвореннями. Для квантово–стійкої криптографії часто використовується узагальнення цієї моделі – модель квантового випадкового оракула [41], яка враховує вплив квантових комп'ютерів на безпеку. Формальне викладення моделі квантового випадкового оракула та аналіз асиметричних перетворень у цій моделі наведено в четвертому розділі.

1.4. Основні положення теорії решіток

Введемо необхідні позначення з теорії решіток, згідно до [42]. Решітка Λ з базисом B є множиною цілочисельних комбінацій лінійно незалежних векторів $b_1, \dots, b_n$:

$$\Lambda(b_1, \dots, b_n) = \{\sum_{i=1}^{n} x_i b_i | x_i \in \mathbb{Z}\}. \quad (1.2)$$

Довжиною вектору v є стандартна евклідова норма $\|v\| = \sqrt{v \cdot v}$, де операція $\cdot$ є скалярним добутком і для двох векторів $v = (v_1, \dots, v_n)$ і $w = (w_1, \dots, w_n)$ визначена як $v \cdot w = \sum_{i=1}^{n} v_i w_i$.

Для заданого базису $B = (b_1, \dots, b_n)$ ортогоналізований за Граммом–Шмідтом базис є $B^* = (b_1^*, \dots, b_n^*)$, де $b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j^*$ для $1 \le j < i \le n$, де $\mu_{ij} = (b_i \cdot b_j^*)/\|b_j^*\|^2$ – коефіцієнти Грамма–Шмідта, $\|b_j^*\|$ – довжини векторів Грамма–Шмідта (ГШ–довжини). Сукупність логарифмів ГШ–довжин будемо називати профілем базису.

Для решітки $\Lambda = \Lambda(B) \subseteq \mathbb{Z}^n$ з базисом $B \in \mathbb{Z}^{n\times k}$ фундаментальний паралелепіпед визначений як $P(B) = \{B \cdot x | x \in [0,1)^k\}$. Детермінант базису решітки є інваріантом і може бути обчислений як $\det(\Lambda) = \sqrt{\det(B^T B)} = \prod_{i=1}^{n} \|b_i^*\|$. При цьому, детермінант решітки чисельно дорівнює об'єму фундаментального паралелепіпеда $vol(\Lambda)$.

Ортогональна проекція є відображення $\pi_i: \mathbb{R}^n \longmapsto span(b_i, \dots, b_{i-1})^{\perp}$ для $i \in \{1, \dots, n\}$. Проективна решітка $\Lambda_{[i:j]}$ – решітка, яка задається наступним чином:

$$\Lambda_{[i:j]} = \Lambda_i = \Lambda(\pi_i(b_i), \pi_i(b_{i+1}), \dots, \pi(b_j)) \tag{1.3}$$

Для $j \in \{i, i+1, \dots, n\}$.

У кожній решітці L існує найменший ненульовий вектор. $\lambda_1(L)$ – норма найменшого вектора. Проблема пошуку найменшого вектора (SVP) полягає у пошуку вектора довжини $\lambda_1(L)$. Проблема апроксимації найменшого вектора γ–SVP полягає у пошуку вектора, що має норму, шо менша за $\gamma(n)\lambda_1(L)$, де $\gamma(n)$ – деяка константа, що залежить від розмірності решітки. Константа Ерміта γ_n визначає обмеження на найменший вектор серед усіх решіток розмірності n і визначена як

$$\gamma_n = \sup \{\frac{\lambda_1^2(L)}{vol(L)^{\frac{2}{n}}}\} \tag{1.4}$$

Для константи Ерміта відомі наступні оцінки [35]:

$$\frac{n}{2\pi e} + \frac{\log(\pi n)}{2\pi e} \leq \gamma_n \leq \frac{1.744n}{2\pi e} + o(n) \tag{1.5}$$

Безпека криптографії на решітках переважно ґрунтується на проблемах NTRU та LWE (і її різновидах) та SIS.

Проблема NTRU. Нехай $n, q > 0$ – цілі числа і задано кільце поліномів $R_q \cong \mathbb{Z}_q[x]/\phi(x)$ ступеня n над кільцем лишків за модулем q. Нехай $f, g \in R_q$ – поліноми з деякого розподілу χ і $h = g/f$. Проблема NTRU (обчислювальна версія) полягає у пошуку поліномів f, g для заданого полінома h.

Проблема LWE. Нехай $n, q > 0$ – цілі числа, χ – деякий розподіл ймовірностей над множиною цілих чисел $\mathbb{Z}$ та s – секретний вектор з рівномірного розподілу над $\mathbb{Z}_q^n$. $L_{s,\chi}$ є розподілом ймовірностей над $\mathbb{Z}_q^n \times \mathbb{Z}_q$, який отримується наступним чином. Обирається вектор $a \in \mathbb{Z}_q^n$ з рівномірного розподілу, значення помилки $e \in \mathbb{Z}_q$ з розподілу χ та повертається пара $(a, c) = (a, (a, a \cdot s + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^n$. Проблема LWE (обчислювальна версія) полягає у тому, щоб для поліноміальної кількості пар (a, c) з розподілу $L_{s,\chi}$ знайти вектор s.

Проблема LWE має багато узагальнень. Для подальшого аналізу варто згадати про проблеми Ring–LWE [41] та Module–LWE [42]. Загальне визначення цих проблем потребує введення ряду понять з алгебраїчної теорії чисел, проте для вирішуваних у даній роботі задач достатньо визначення для випадку поля $R_q = \mathbb{Z}_q[X]/(X^n + 1)$, яке використовується у переважній більшості криптографічних схем на решітках. У цьому випадку формальне визначення стає значно простішим.

Проблема Ring–LWE. Нехай $n, q > 0$ – цілі числа і задано поле $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. χ – деякий розподіл ймовірностей над R_q та s – секретний поліном з рівномірного розподілу над R_q. $L_{s,\chi}$ є розподілом ймовірностей над $R_q \times R_q$, який отримується наступним чином. Обирається поліном $a \in R_q$ з рівномірного розподілу, поліном помилки $e \in R_q$ з розподілу χ та повертається пара $(a, c) = (a \cdot s + e) \in R_q \times R_q$. Проблема Ring–$LWE$ (обчислювальна версія) полягає у тому, щоб для поліноміальної кількості пар (a, c) з розподілу $L_{s,\chi}$ знайти поліном s.

Проблема Module–LWE визначена аналогічним чином, тільки a, s є векторами поліномів: $a, s \in (R_q)^d$, де $d > 0$ деяке ціле число.

Проблема SIS. Нехай задано деяку норму $\|\cdot\|_p$, ціле число q, дійсне число β та матрицю $A \in \mathbb{Z}_q^{n \times m}$. Необхідно знайти ненульовий вектор $e \in \mathbb{Z}^m$, для якого виконується $Ae = 0 \; mod \; q$ та $\|e\|_p \leq \beta$.

У якості норми зазвичай використовується евклідова норма $\|\cdot\|_2$ або супремум норма $\|\cdot\|_\infty$. На практиці також є важливим варіант проблеми SIS для неоднорідної системи рівнянь – ISIS. У випадку проблеми ISIS вимагається, щоб виконувалося рівняння $Ae = u \ mod \ q$ для деякого заданого вектору u.

Криптоаналіз криптографічних проблем на решітках зводиться до аналізу q–арних решіток. Для $\zeta \in \mathbb{Z}$, простого $q \geq 2$ та матриці $A \in \mathbb{Z}^{n \times n}$ q–арна решітка визначається базисом

$$B = \begin{pmatrix} qI_m & A \\ 0 & \zeta I_n \end{pmatrix} \in \mathbb{Z}^{(m+n)\times(m+n)} \tag{1.6}$$

Базиси решіток, що відповідають проблемам LWE, NTRU та SIS є q–арними решітками. Криптоаналіз таких решіток зводиться до редукції решіток. Процес редукції решітки полягає у зменшенні норм базисних векторів решітки. В залежності від алгоритму редукції редукований базис решітки матиме різні властивості.

Алгоритм LLL [43] виконує над базисом дві операції: редукція за розміром $b_i \leftarrow b_i - round(\mu_{ij})b_j$ для $j \in [i-1]$ та перестановки b_i і b_{i+1} якщо $\|b_{i+1}^*\|^2 \leq \frac{1}{2}\|b_i^*\|^2$ поки відбуваються зміни.

У алгоритмі BKZ (та його варіаціях) фіксується розмір блоку β і відбувається пошук найменшого вектору на решітках $\Lambda_{[i,i+\beta'-1]}$ для i від 1 до $n-1$, де $\beta' = \min(\beta, n-i+1)$. Пошук вектора відбувається окремою процедурою.

Для індексу i стандартна реалізація алгоритма BKZ викликає алгоритм пошуку найменшого вектора для решітки $\Lambda_{[i,i+\beta-1]}$ і знаходить найкоротший вектор v на цій решітці. Далі BKZ вставляє v у старий базис між b_{i-1} та b_i. Для базису $(b_1, \dots, b_{i-1}, v, b_i, \dots, b_{\min(i+\beta-1,n)})$ застосовується LLL (або BKZ з меншим розміром блока) для отримання нового базису з меншими векторами. Процедура повторюється для усіх можливих значень індексу i. Ці процедури складають один раунд алгоритму. У оригінальній версії BKZ алгоритм зупинявся, коли оновлень базису не відбувалось. Для отримання редукованого базису необхідно

$O(\frac{n^2}{\beta^2})$ раундів. На практиці вже десяток раундів дає результат, що є близьким до фінальної форми базису. Усі інші раунди вносять дрібні зміни до базису. Тому при криптоаналізі не має сенсу проводити редукцію аж до завершення, а достатньо провести тільки перші 8–10 раундів.

Наразі, існує багато узагальнень BKZ. Особливо варто відмітити слайд редукцію [44] та SD–BKZ [45]. Ці алгоритми мають кращу асимптотичну поведінку. Проте, варто зауважити, що усі узагальнення BKZ роблять поліноміальну кількість викликів до певних алгоритмів пошуку найменшого (або малого) вектору, тому для криптоаналізу складність не сильно змінюється і при теоретичній оцінці безпеки можливо спиратися безпосередньо на BKZ.

Базис $(b_1, \dots, b_n)$ є HKZ–редукованим якщо $|\mu_{ij}| \leq 1/2$ для всіх i та j і $\pi_i(b_i)$ є найменшим вектором на проективній підрешітці $L_{[i,n]}$ для всіх i. ГШ–довжини при цьому можливо оцінити як $\|b_i^*\| = GH(L_{[i,n]})$.

Двома найпоширенішими підходами до побудови процедури пошуку найменшого вектору є алгоритми просіювання (англ. sieving algorithms) та алгоритми переліку точок (англ. enumeration algorithms) [42,43].

Алгоритми просіювання ґрунтуються на суто геометричних ідеях. На початку алгоритму генерується субекспоненційна кількість векторів на решітці. Далі обчислюється множина векторів, що є алгебраїчною різницею згенерованих векторів. У наступний раунд потрапляють тільки ті вектори, що мають менші довжини, ніж у згенерованих (відбувається “просіювання”). Процес “просіювання” повторюється ітеративно, поки “просіювання” повертає не пусту множину. Якщо на початку була достатньо велика кількість векторів, то через поліноміальну кількість ітерацій буде знайдено близький до найменшого вектор. Найкращий алгоритм [46] цього класу на сьогоднішній день потребує $\sqrt{3/2}^{\frac{\beta}{2}+o(\beta)}$ операцій. До таблиці 1.1 зведені оцінки роботи найкращих алгоритмів просіювання, згідно до [47]. Зауважимо, що фактор $o(\beta)$ під час аналізу ігнорується, проте він може бути доволі великим. Реальна вартість атаки може коштувати значно більше, тому у таблицю 1.1 зведено як теоретичну

оцінку, так і оцінку, що отримана на основі практичних досліджень. Квантові комп'ютери дають відносно не велике прискорення алгоритмів просіювання. Це пов'язано з тим, що операцію «просіювання» важко прискорити за допомогою квантових ефектів.

Таблиця 1.1.

Оцінки часу роботи алгоритмів просіювання

Оцінка	Формула
Алгоритм просіювання на класичному комп'ютері	$2^{0.29248125036\beta+o(\beta)}$
Алгоритм просіювання на класичному комп'ютері	$2^{0.3924062518\beta-5}$ $2^{0.265\beta+o(\beta)}$

Алгоритми переліку точок є комбінаторним методом вирішення задачі SVP на певній решітці. Для заданого базису $(b_1, \dots, b_n)$ та ГШ–базису $(b_1^*, \dots, b_n^*)$ алгоритм переліку полягає у побудові дерева пошуку, у якому вузлами є вектори. Корінь дерева є нульовим вектором. Для кожного вузла $v \in L$ на глибині $k \in 1, \dots, n$ потомки міснять вектори $v + a_{n-k}b_{n-k}(a_{n-k} \in \mathbb{Z})$ з довжиною $\|\pi_{n-k}(\sum_{i=n-k}^{n} a_i b_i)\|$ меншою за задану константу $R_{k+1} \in (0, \|b_1\|]$. Після проходу по всім можливим вузлам, алгоритм знаходить вектор на решітці, що менший за R_n на певній глибині.

У таблицю 1.2 зведені оцінки сучасних алгоритмів переліку. Квантові комп'ютери здатні значно краще прискорити алгоритми переліку, порівняно з алгоритмами просіювання.

Таблиця 1.2.

Оцінки часу роботи алгоритмів переліку

Оцінка	Формула
Класичний комп'ютер	$2^{0.125\beta\cdot\log_2(\beta)-0.547\beta+10.4}$ $2^{0.1839\beta\cdot\log_2(\beta)-0.995\beta+16.25}$
Квантовий комп'ютер	$2^{(0.125\beta\cdot\log_2(\beta)-0.547\beta+10.4)/2}$

	$2^{(0.1839\beta\cdot\log_2(\beta)-0.995\beta+16.25)/2}$

Прискорення отримується за допомогою застосування варіантів алгоритму Гровера. Алгоритми переліку потребують значно менше пам'яті. Можливість їх реалізації на квантових комп'ютерах ймовірно з'явиться значно раніше, ніж для алгоритмів просіювання. Проте, з таблиці 1.2 видно, що асимптотично вони працюють гірше.

1.5. Основні положення теорії квантових обчислень

У межах цієї роботи використовується стандартна модель квантових обчислень. Гільбертів простір є векторним n–вимірним простором над полем комплексних чисел. Для двох векторів у Гільбертовому просторі – $|\Psi\rangle = (\psi_1, \psi_2, \dots, \psi_n)$ та $|\Phi\rangle = (\phi_1, \phi_2, \dots, \phi_n)$ – скалярний добуток визначений як $\langle\Psi, \Phi\rangle = \sum_{i=1}^{n} \psi_i^* \phi_i$, де ψ_i^* – комплексне спряження до ψ_i. Норма для вектора $|\Phi\rangle$ відповідно визначена як $\||\Phi\rangle\| = \sqrt{\langle\Phi, \Phi\rangle}$. Стан квантової системи описується вектором з нормою 1 у Гільбертовому просторі. Унітарний оператор U над Гільбертовим простором є лінійним перетворенням, для якого виконується $UU^\dagger = U^\dagger U = I$, де $U^\dagger$ є ермітово спряженою матрицею для U та I – тотожній оператор. Норма для оператора U визначена наступним чином: $\|U\| = \max_{|\psi\rangle} \|U|\psi\rangle\|$. Комутатор операторів U, V визначений як $[U, V] = UV - VU$. Обчислювальний базис для n–вимірного Гільбертового простору складається з $\log(n)$ векторів $|b_i\rangle$ довжини $\log(n)$ з 1 на позиції i та 0 у всіх інших позиціях.

Ортогональна проекція P є лінійним перетворенням, для якого виконується $P^2 = P = P^\dagger$. Операція вимірювання визначена сімейством ортогональних проекторів, що попарно ортогональні. Прикладом є вимірювання у обчислювальному базисі, у якому проектори визначені базисними векторами. Результатом вимірювання в обчислювальному базисі стану $|\Psi\rangle$ є число i з ймовірністю $\|\langle b_i, \Psi\rangle\|^2$. Для загального випадку, коли задано довільне сімейство

проекторів $\{P_1, P_2, \dots, P_m\}$, результатом виміру буде стан $\frac{P_i|\Psi\rangle}{\|P_i|\Psi\rangle\|}$ з ймовірністю $\|P_i|\Psi\rangle\|^2$.

Будь–яка класична функція $f: X \to Y$ може бути реалізована як унітарний оператор U_f:

$$U_f: |x, y\rangle \to |x, y \oplus f(x)\rangle \tag{1.6}$$

Квантовий алгоритм має доступ до певної функції f^{St} якщо він може зробити запит до оператора $U_{f^{St}}$. Надалі позначення f^{St} та $U_{f^{St}}$ ототожнюються для квантових обчислень для зручності викладення.

Більш детальну операцію про оператори та квантові обчислення можливо знайти у [4].

1.6. Постановка задач досліджень роботи

Метою дослідження є аналіз та розробка методів та моделей для підвищення захищеності асиметричних криптографічних систем на решітках від існуючих та потенційних атак та обгрунтування на їх основі національних стандартів асиметричних криптоперетворень електронного підпису та інкапсуляції ключів на модулярних решітках.

Об’єктом дослідження є процеси оцінки захищеності існуючих та перспективних асиметричних криптографічних перетворень від класичних та квантових атак.

Наукова задача включає в себе комплекс взаємопов’язаних досліджень, що спрямовані на підвищення ефективності та безпеки сучасних криптографічних перетворень на решітках, які відіграють ключову роль у квантово–стійкій криптографії.

1. Вибір і обґрунтування моделей редукції решіток.

Перша задача спрямована на вибір моделей редукції решіток, які найбільш адекватно описують поведінку базисів під час редукції. Це є важливим, оскільки моделі редукції є основою для розуміння алгоритмів, що забезпечують захищеність криптографічних перетворень. Врахування сучасних відомостей

про поведінку базисів, таких як динаміка зміни довжин векторів і кути між ними, дозволяє створювати більш точні оцінки ефективності редукції, які є критично важливими для оцінки безпеки алгоритмів на решітках.

2. Дослідження впливу моделей редукції решіток на складність атак.

Друга задача зосереджується на вивченні впливу різних моделей редукції на складність проведення криптографічних атак, таких як атаки декодування та вкладення. Результати досліджень дозволяють удосконалити методики оцінювання криптографічних перетворень на решітках, забезпечуючи більш точну відповідність між математичними оцінками та реальними показниками безпеки. Це надзвичайно важливо для формування стандартів безпеки, зокрема в умовах активного розвитку квантових обчислень.

3. Аналіз криптографічних перетворень у формальних моделях безпеки.

Третя задача полягає в аналізі криптографічних алгоритмів на решітках у контексті формальних моделей безпеки, таких як IND–CCA (захист від вибору шифротекстів) і EUF–CMA (захист від підроблення підписів). Особливу увагу приділено врахуванню можливості квантових атак. Ця задача є фундаментальною для розробки криптосистем, здатних протистояти загрозам, що виникають із розвитком квантових обчислень.

4. Дослідження впливу обчислень з плаваючою точкою на безпеку.

Четверта задача присвячена вивченню впливу обчислень з плаваючою точкою на безпеку криптографічних перетворень. Оскільки такі обчислення часто використовуються в алгоритмах на решітках для забезпечення точності, вони можуть створювати потенційні вразливості, пов’язані з некоректними обчисленнями або атаками через побічні канали. Дослідження цієї проблеми дозволяє розробити методики зменшення впливу помилок обчислень і підвищити загальну стійкість криптографічних алгоритмів.

Значення розв’язання цих задач: Результати дослідження сприятимуть удосконаленню існуючих криптографічних перетворень на решітках, забезпечать більш точні оцінки їх безпеки та адаптацію до сучасних викликів,

таких як квантові атаки та вразливості, пов’язані з обчисленнями. Це дозволить створювати нові та більш ефективні стандарти квантово–стійкої криптографії, які є критичними для захисту даних у глобальній цифровій інфраструктурі.

1.7. Висновки до розділу

1. Обчислювана сила квантових комп’ютерів ґрунтується на двох техніках: квантовому пошуку та квантовому перетворенні Фур’є. Для криптографічних цілей найбільш важливим застосуванням квантових алгоритмів є задача пошуку схованої підгрупи. Для випадку абелевих груп існують поліноміальні алгоритми, що призводить до ефективних атак на такі криптосистеми як RSA, DSA, ECDSA. Проте задачі, що зводяться до випадку не абелевих груп, є перспективними для криптографічного застосування, адже поява ефективних квантових алгоритмів для цього випадку вважається малоймовірною.

2. Обчислювальна сила квантових комп’ютерів, окрім кількості кубітів, обмежена також кількістю операцій, що можуть бути виконані над ними за одне обчислення. На практиці саме цей фактор значно обмежує можливості, адже точність виконання операцій та неможливість повної ізоляції системи не дозволяють реалізовувати квантові алгоритми, навіть якщо кубітів вистачає. Ці особливості мають враховуватися при побудові моделі безпеки.

3. Криптографія на решітках особливо виділяється серед постквантових напрямків через гарні теоретичні та практичні аргументи безпеки. Іншою її перевагою є різноманітність перетворень, що робить її перспективним напрямком досліджень.

4. Проведений аналіз показав, що формальна модель безпеки IND–CCA є стандартною моделлю безпеки для асиметричних схем шифрування та механізмів інкапсуляції ключів. Формальні докази в IND–CCA гарантують відсутність атак з адаптивно підібраними шифротекстами при умові складності теоретико–числових проблем, на яких ґрунтується безпека схеми шифрування. При цьому, на практиці для оцінок безпеки в IND–CCA використовується модель

(квантового) випадкового оракула, яка передбачає заміну усіх геш функцій на ідеалізовані варіанти – випадкові оракули.

5. Було виявлено, що теоретико–числові проблеми NTRU та LWE є двома основними проблемами в криптографії на решітках. При цьому, на практиці часто використовуються варіанти проблеми LWE – R–LWE та M–LWE, які дозволяють створювати швидкі та ефективні схеми шифрування. Але, їх аналіз потребує більш детальної уваги.

6. Найбільш поширеним підходом до редукції решіток є алгоритм BKZ, проте існують і інші підходи, такі як слайд редукція, які є асимптотично швидшими. Проте, у всіх цих алгоритмів є одна спільна риса. Вони роблять поліноміальну кількість викликів до деякого іншого алгоритму, який шукає малий вектор на решітці меншої розмірності. Далі знайдений вектор додається до базису та оброблюється в залежності від конкретного алгоритму.

7. У цьому розділі буда вирішена перша задача наукового дослідження. Було проведено аналіз міжнародних вимог до квантово стійких перетворень. Проведений аналіз показав, що одним з найбільш перспективних напрямків для досліджень є криптографія на решітках. Одним з напрямків підвищення безпеки криптографічних перетворень на решітках є створення комплексних моделей безпеки, що враховували би поведінку q–арних решіток при криптоаналізі.

РОЗДІЛ 2. АНАЛІЗ ТА ПОРІВНЯННЯ МОДЕЛЕЙ РЕДУКЦІЇ РЕШІТОК

Криптографія на решітках ґрунтується переважно на складності вирішення теоретико–числових проблем на q–арних решітках (NTRU–решітки, LWE–решітки, тощо). Як правило, криптоаналіз зводиться до редукції базису решіток, а комбінаторні та алгебраїчні методи мають обмежену ефективність [42] і виконують другорядну роль. Для оцінки складності редукції виникає необхідність у виборі моделі, яка б враховувала структурні особливості q–арних решіток та мала б найменшу кількість модельних припущень. Цей розділ присвячений аналізу та порівнянню існуючих моделей редукції решіток та евристичних припущень, що лежать в їх основі.

Надалі вважаємо, що задана деяка (не обов'язково q–арна) d–вимірна решітка Λ з базисом $B = (b_1, \dots, b_d)$ та ГШ–профілем $B^* = (b_1^*, \dots, b_d^*)$. Усі експерименти з решітками у цьому розділі виконувалися за допомогою бібліотеки fplll.

2.1. Евристика Гауса

В основі аналізу сучасних моделей редукції решіток лежить евристика Гауса [43], сутність якої полягає у тому, що кількість $|\Lambda \bigcap \Omega|$ точок решітки Λ у довільному вимірюваному тілі $\Omega \subset \mathbb{R}^d$ складає $vol(\Omega)/vol(\Lambda)$. Використовуючи d–вимірний шар у якості вимірюваного тіла, для випадкової решітки $\Lambda \subset \mathbb{R}^d$, очікуваний найменший вектор, згідно до евристики Гауса, можливо оцінити як

$$GH(\Lambda) = \left(\frac{vol(\Lambda)}{vol(\Omega)}\right)^{1/d} = \frac{\Gamma\left(1+\frac{d}{2}\right)}{\sqrt{\pi}} \cdot vol(\Lambda)^{\frac{1}{d}} \approx \sqrt{\frac{d}{2\pi e}} \cdot vol(\Lambda)^{1/d}. \qquad (2.1)$$

Практичні експерименти з алгоритмами LLL та BKZ у дослідженнях [43–45], показують, що $\|b_i^*\|/\|b_{i+1}^*\| \approx const$ для $\beta \ll d$. Для перевірки цього твердження додатково була проведена BKZ редукція решіток малих розмірностей з різними значеннями блока β. На рисунку 2.1 наведено ісходний профіль для 230–мірної випадкової q–арної решітки та профілі для блоку

редукції $\beta = 2,10,20,30,40,50$. З рисунка 2.1 видно, що, дійсно, виконується припущення $\|b_i^*\|/\|b_{i+1}^*\| \approx const$. Для інших розмірностей спостерігається схожа ситуація.

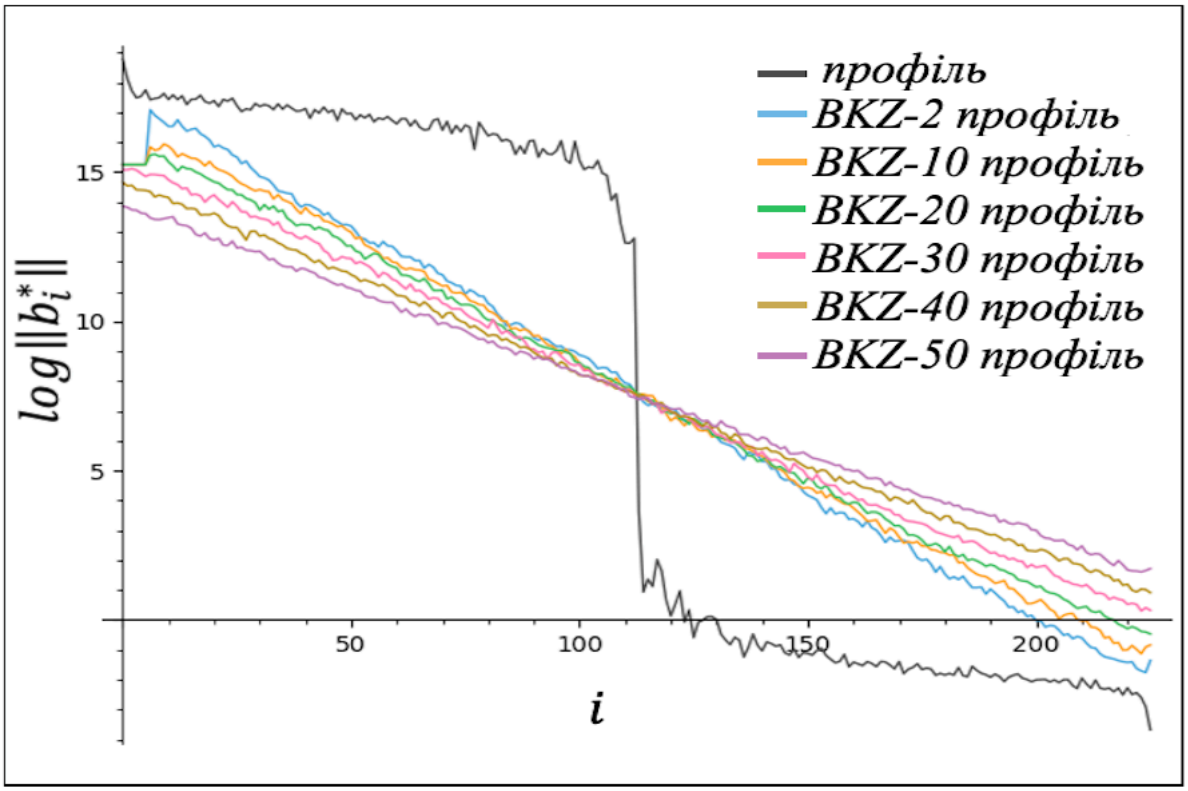

Рис. 2.1. Профілі 230–мірної q–арної решітки для $\beta = 2,10,20,30,40,50$

Застосовуючи евристику Гауса (2.1) до першого вектору в BKZ–β редукованого базисі $B = (b_1, \dots, b_d)$, та враховуючи припущення $\|b_i^*\|/\|b_{i+1}^*\| \approx const$, маємо:

$$log\|b_i^*\| = \frac{d-1-2i}{2} \cdot \log(\alpha_\beta) + \frac{1}{d}\log\left(vol(\Lambda)\right) \qquad (2.2)$$

Для деякої константи α_β, що залежить від властивостей BKZ–β.

Рівняння (2.2) є моделлю редукції решіток GSA (англ. Geometric Series Assumption) [48]. На рисунку 2.2 наведено приклад застосування моделі GSA. Припущення $\|b_i^*\|/\|b_{i+1}^*\| \approx const$ також є наслідком з евристики Гауса та з визначення β–BKZ редукованого базису, тож можливо сказати, що адекватність моделі GSA цілком залежить від того, чи виконується евристика Гауса.

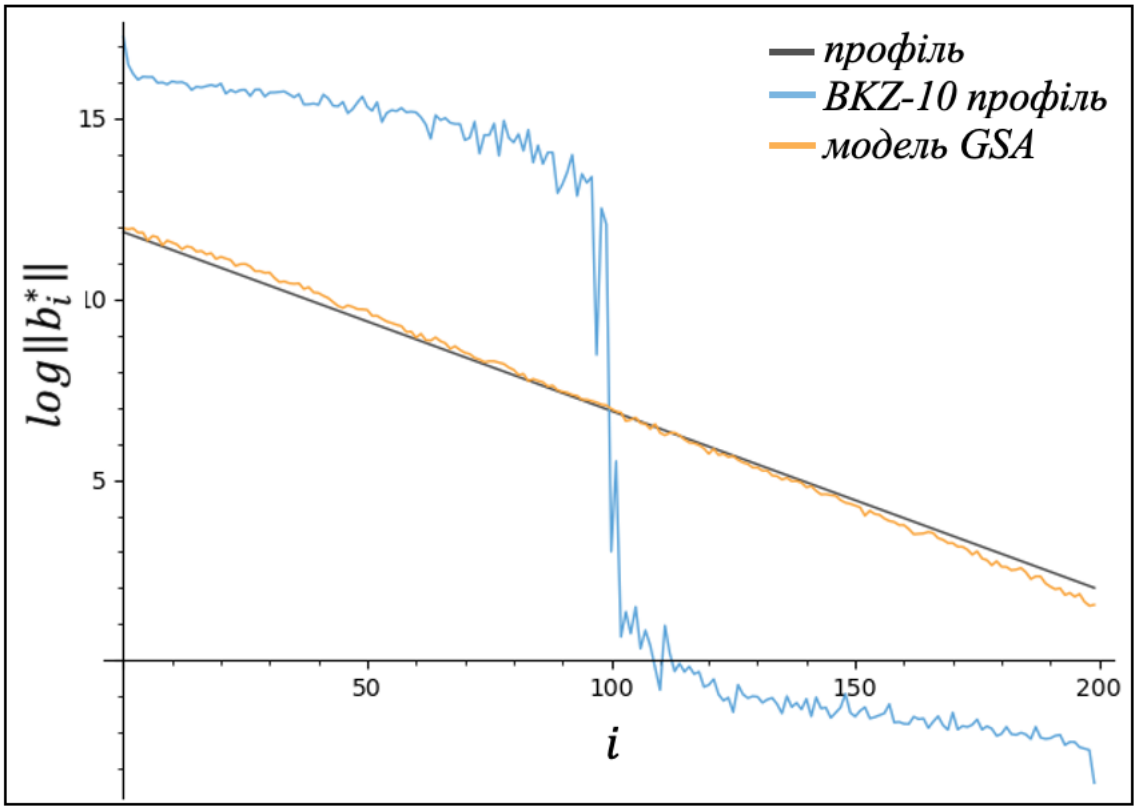

Рис. 2.2. Приклад застосування моделі GSA для 200–мірної q–арної решітки

Варто зауважити, що у більшості робіт замість константи α_β використовується інша метрика. Доволі зручною метрикою є так званий кореневий фактор Ерміта, який для базису $B = (b_0, \dots, b_d)$ визначається як

$$\delta_\beta = (\|b_0\|/vol(\Lambda)^{1/d})^{1/d} \tag{2.3}$$

З формули (2.2) випливає $\delta_\beta = \sqrt{\alpha_\beta}^{1-1/d}$. У дослідженні [49] була отримана асимптотична оцінка

$$\lim_{\beta\to\infty} \delta_\beta = \left(\frac{\beta}{2\pi e}\cdot(\pi\beta)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}} \tag{2.4}$$

У дослідженні [50] було показано, що ця оцінка є лише першим наближенням і може бути уточнена наступним чином:

$$\lim_{\beta\to\infty} \delta_\beta = \left(\frac{\beta}{2\pi e}\cdot(\pi\beta)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}+\frac{\beta}{2d^2}} \tag{2.5}$$

З теоретичної точки зору фактор Ерміта обмежується значенням константи Ерміта. На рисунку 2.3. зображено асимптотичний графік фактора Ерміта та графік для константи Ерміта. З рисунку видно, що оцінка (2.4) наближається до нижньої теоретичної межі, у той час як оцінка (2.5) є більш помірною. Оскільки фактор Ерміта сильно впливає на модель GSA, то важливо визначити яка оцінка є точнішою і наскільки точними є асимптотичні оцінки для криптографічно значущих розмірностей [51].

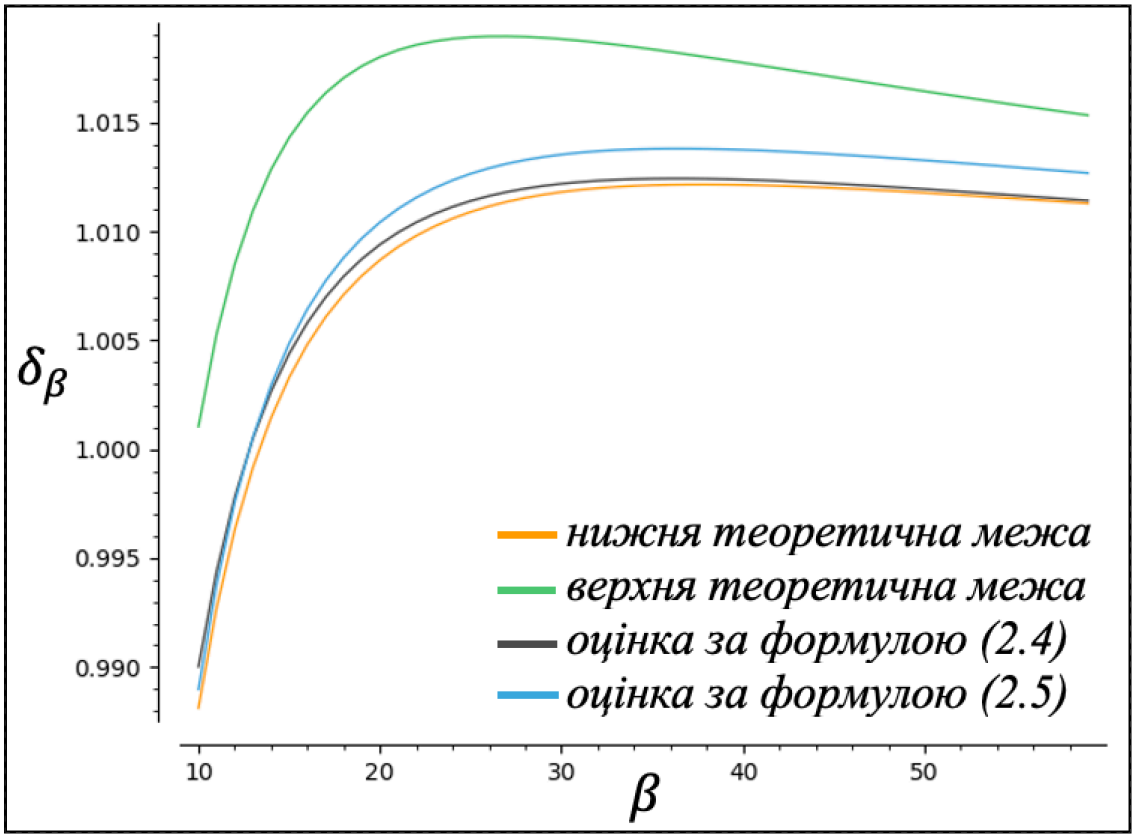

Рис. 2.3. Оцінка фактора Ерміта

Для криптографічно значущих розмірностей, звісно, отримати значення похибки неможливо прямим шляхом, проте можливо протестувати на малих розмірностях і екстраполювати результати на більші розмірності.

2.2. Експериментальна оцінка фактора Ерміта

У загальному випадку на значення фактора Ерміта можуть впливати розмір блоку редукції, розмірність решітки та розмір в бітах кожного коефіцієнта

векторів в базисі [51,52]. Для виявлення впливу кожного з цих факторів були проведені експериментальні дослідження поведінки кореневого фактора Ерміта на випадкових q–арних решітках. У таблиці 2.1 наведено результати виміру за формулою (2.4) фактора Ерміта для вимадкових q–арних решіток. Для досліджень використовувалися випадкові решітки розмірностей 120, 145, 170. Для кожної з цих розмірностей розглядалися значення $\log_2 q = 10{,}20{,}40$ для розмірів блоку від 3 до 60.

Таблиця 2.1.

Експериментальні оцінки фактора Ерміта для $\log_2 q = 10$

β	Розмірність 120	Розмірність 145	Розмірність 170
3	1.017378	1.017690	1.016867
5	1.017378	1.017690	1.016867
10	1.015669	1.016230	1.015883
15	1.014016	1.014413	1.014484
20	1.013434	1.013680	1.013671
25	1.012825	1.013024	1.013012
30	1.012559	1.012833	1.012833
35	1.012352	1.012610	1.012653
40	1.012187	1.012415	1.012458
45	1.011986	1.012163	1.012241
50	1.011449	1.011726	1.011718
55	1.011132	1.011373	1.011397
60	1.010854	1.011097	1.011049

З таблиці 2.1 вже видно, що розмірність решітки має вплив на значення фактора Ерміта. Так, з ростом розмірності при фіксованих параметрах редукції спостерігається зменшення фактору Ерміта, що цілком відповідає очікуванням. У таблиці 2.2. наведено аналогічні оцінки для $\log_2 q = 20$.

Таблиця 2.2.

Експериментальні оцінки фактора Ерміта для $\log_2 q = 20$

β	Розмірність 120	Розмірність 145	Розмірність 170
3	1.017318	1.017987	1.018275
5	1.017318	1.017987	1.018275
10	1.015702	1.016254	1.016846
15	1.013923	1.014496	1.015003
20	1.013243	1.013654	1.013946
25	1.012820	1.012977	1.013223
30	1.012630	1.012746	1.013053
35	1.012296	1.012596	1.012792
40	1.012116	1.012393	1.012651
45	1.011863	1.012116	1.012360
50	1.011399	1.011695	1.011787
55	1.011200	1.011372	1.011453
60	1.010895	1.011065	1.011194

З таблиці 2.2 видно, що значення q впливає лише на малих значеннях β, у той час, як при збільшенні розмірності цей фактор нівелюється, що пояснює відсутність прямої залежності від q в асимптотичних оцінках, проте, варто зауважити, що на практиці залежність все ж присутня, хоча і є настільки малою, що на неї можливо не звертати увагу.

У таблиці 2.3 наведені аналогічні оцінки для $\log_2 q = 40$. Отримані оцінки підтверджуються розглянуті вище спостереження. Фактор Ерміта переважно залежиться від розмірності решіток, над якими проводяться процеси редукції, у той час як параметр q вносить лише незначні зміни.

Наведені в таблицях 2.1 – 2.3 оцінки показують реальні значення фактора Ерміта, проте інтерес представляє оцінка точності асимптотичних формул, оскільки саме вони використовуються при оцінці безпеки криптографічних перетворень.

Таблиця 2.3.

Експериментальні оцінки фактора Ерміта для $\log_2 q = 40$

β	Розмірність 120	Розмірність 145	Розмірність 170
3	1.017322	1.017685	1.018242
5	1.017322	1.017685	1.018242
10	1.015547	1.016249	1.016789
15	1.013991	1.014420	1.014897
20	1.013351	1.013708	1.013915
25	1.012738	1.013018	1.013307
30	1.012563	1.012835	1.013085
35	1.012377	1.012607	1.012849
40	1.012162	1.012407	1.012661
45	1.012019	1.012167	1.012385
50	1.011495	1.011734	1.011870
55	1.011121	1.011356	1.011508
60	1.010811	1.011107	1.011172

Для отримання конкретних оцінок введемо наступний функціонал середньоквадратичної похибки для еталонних значень δ_{etalon} та експериментально отриманих значень $\delta_{experiment}$:

$$MSE\left(\delta_{etalon}, \delta_{experiment}\right) = \frac{1}{d}\sum_{i=0}^{d-1}\left(\delta_{etalon}[i] - \delta_{experiment}[i]\right)^2 \quad (2.6)$$

На рисунку 2.4 наведено графіки δ_{etalon} та $\delta_{experiment}$, усереднені за параметром q для розмірності 170.

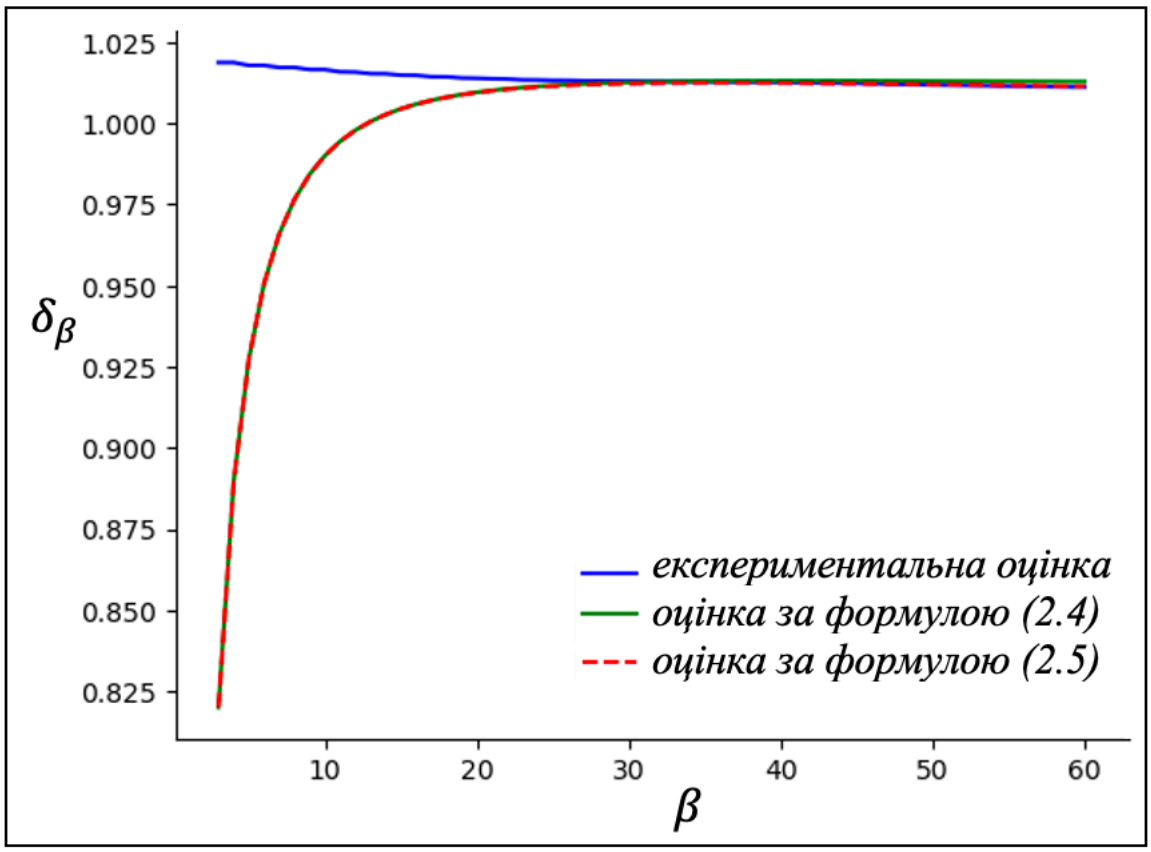

Рис. 2.4. Експериментальна оцінка фактора Ерміта для розмірності 170

З рисунка 2.4 видно, що на малих значеннях $\beta < 30$ оцінки (2.4) та (2.5) не працюють через свій асимптотичний характер. Далі експериментальна оцінка наближається до (2.4).

У таблиці 2.4 наведено значення функціоналу помилки (2.6) для оцінок фактора Ерміта починаючи від $\beta > 30$ для (2.4) та (2.5) відповідно.

З таблиці 2.4 випливає, що середньоквадратична помилка для оцінки (2.4) є меншою.

Таблиця 2.4.

Значення MSE для оцінок фактора ерміта

Асимптотична оцінка (2.4)			
	Розмірність 120	Розмірність 145	Розмірність 170
$\log_2 q = 10$	0.0003728998	0.0002477775	0.0002575795
$\log_2 q = 20$	0.0003651147	0.0002409057	0.0003197217
$\log_2 q = 40$	0.0003566181	0.0002485526	0.0003261832

Продовження таблиці 2.4.

Асимптотична оцінка (2.5)			
	Розмірність 120	Розмірність 145	Розмірність 170
$\log_2 q = 10$	0.0021920808	0.0013799646	0.0010351942
$\log_2 q = 20$	0.0021856184	0.0014050912	0.0009452115
$\log_2 q = 40$	0.0021794041	0.0013804805	0.0009179317

На рисунку 2.5 зображено фрагмент рисунка 2.4 для значень $20 \leq \beta \leq 60$. З рисунка 2.5 видно, що реальне експериментально обчислене значення фактора Ерміта на малих розмірностях навіть менше, ніж дає оцінка (2.4). При цьому з ростом розмірності реальні значення фактору Ерміта збільшуються і наближається до оцінки (2.5).

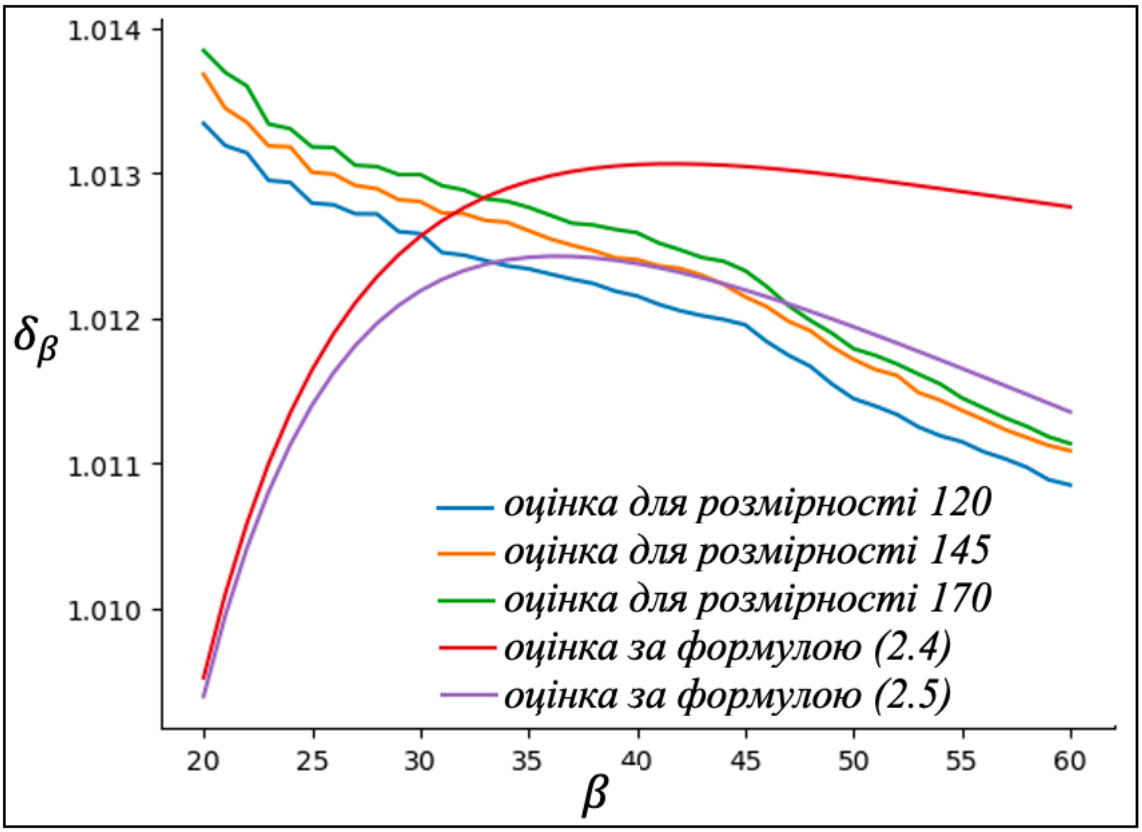

Рис. 2.5. Експериментальна оцінка фактора Ерміта для розмірностей 120, 145, 170

Для того, щоб знайти вплив похибки вимірювання ϵ_{mse} фактора Ерміта для великих розмірностей скористаємося розкладенням в ряд Тейлора по малому параметру ϵ_{mse} першого вектора $\|b_0\|$ у β–BKZ редукованому базисі:

$$\|b_0\| = \delta^d \cdot vol(\Lambda)^{\frac{1}{d}} \approx (\delta_{real} + \epsilon_{mse})^d \cdot vol(\Lambda)^{\frac{1}{d}}$$

$$\approx \delta_{real}^d \cdot vol(\Lambda)^{\frac{1}{d}} + d\delta_{real}^{d-1}\epsilon_{mse} vol(\Lambda)^{\frac{1}{d}} = \|b_0\|_{etalon}(1 + \frac{d\epsilon_{mse}}{\delta_{real}}) \quad (2.7)$$

Оскільки гарантовано, що $\delta_{real} < 1.02$ і ϵ_{mse} з зростанням d зменшується до 0, то враховуючи, що на малих розмірностях $\epsilon_{mse} \approx 0.0001$, маємо

$$\|b_0\| < \|b_0\|_{etalon}(1 + d \cdot 0.00009803921) = \|b_0\|_{etalon}(1 + \epsilon_\delta) \quad (2.8)$$

Типовими значеннями розмірності d для криптографічних застосувань є 1024 та 2048 [52]. Припускаючи, що ϵ_{mse} з ростом d зменшиться хоча б на 10^{-2} для для цих значень d, то відповідно маємо $\epsilon_\delta < 0.001$ для $d = 1024$ і $\epsilon_\delta < 0.002$ для $d = 2024$. Тобто похибка фактора Ерміта є настільки малою, що її можливо не враховувати при реальних обчисленнях і оцінки фактора Ерміта (2.4) та (2.5) можливо вважати точними для криптографічних наборів параметрів.

2.3. Порівняння моделей базису решіток

Модель GSA є корисним, проте доволі грубим наближенням форми базису редукованої решітки. Перш за все, GSA не враховує того факту, що останній блок буде HKZ редукованим, а не BKZ редукованим. Втім, існує можливість форму HKZ базису можливо аналогічно, використовуючи наступну евристику [50]:

$$h_i = \log GH(d-i) - \frac{1}{d-i}\sum_{j<i} h_j \quad (2.9)$$

Узагальнюючи для довільного базису:

$$l_i = \frac{d-1-2i}{2} \cdot log\alpha_\beta + s, \text{ якщо } 0 \le i \le d-\beta$$

$$l_i = h_{i-(d-\beta)} + l_{d-\beta} - h_0, \text{ якщо } d-\beta \le i \le d \quad (2.10)$$

Де s є нормуючим фактором, таким, що виконується $\sum l_i = logV$.

Окремо варто розглянути ефекти, що виникають під час редукції q–арних решіток. Такі решітки містять вектор (q, 0, …, 0) та усі його перестановки. Типовою формою профіля базису таких решіток є так звана «Z–форма»:

Приклад Z–форми наведено на рисунку 2.6:

- Перші вектори мають розмір q
- Останні вектори мають розмір 1
- Для усіх інших векторів виконується $\|b_i^*\|/\|b_{i+1}^*\| \approx const$.

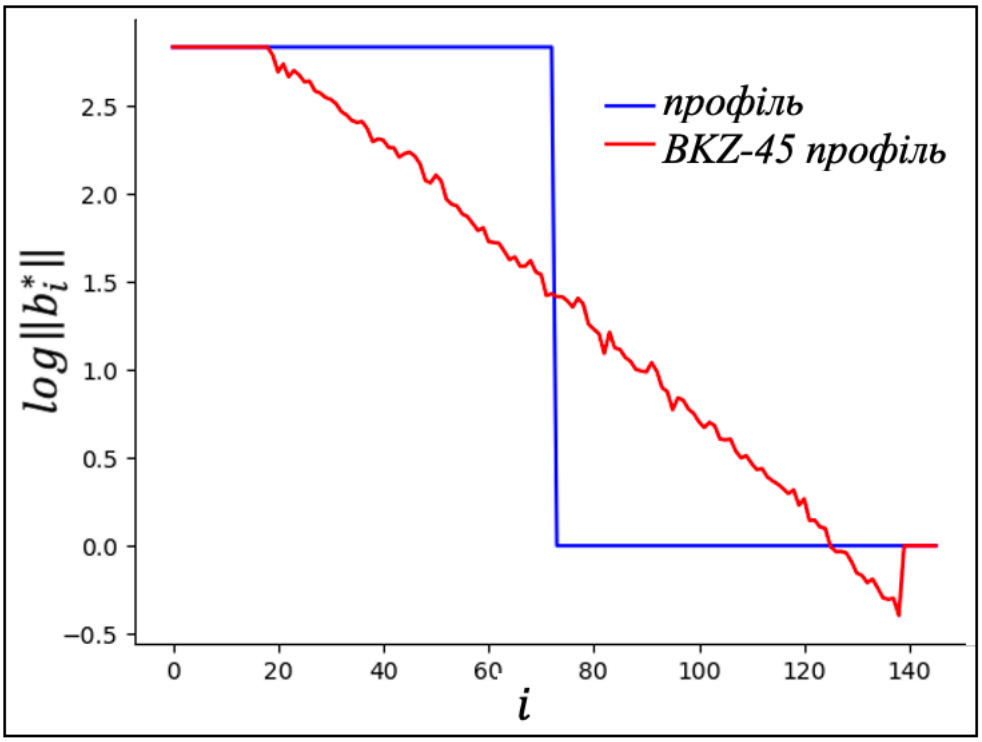

Рис. 2.6. Z–форма базису решітки

Одним з підходів для моделювання такого базису є модифікація евристики GSA, що була запропонована в роботі [50]. Такий підхід, зокрема, був популяризований авторами ЕП Crystals–Dilithium. Модифікована евристика GSA (модель ZGSA) визначена наступним чином:

$$\|b_i^*\| = \begin{cases} q, \text{якщо } i \le d-m \\ \sqrt{q}\cdot \alpha_\beta^{\frac{(2d-1-2i)}{2}}, \text{якщо } d-m < i < d+m-1 \\ 1, \text{ якщо } i \ge d+m-1 \end{cases} \quad (2.11)$$

Де $\alpha_\beta = GH(\beta)^{2/(\beta-1)}$ і $m = \frac{1}{2} + \frac{lnq}{2ln\alpha_\beta}$.

Інший підхід базується на основі симуляції. Ідея симуляції ГШ–профілю була запропонована у роботі [49] (Симулятор Чена–Нгуєна). Симулятор замість запуску SVP–оракула визначає очікувану довжину нового вектора за допомогою евристики Гауса. Особливістю симулятора Чена–Нгуєна було використання предобчислених даних для моделювання останнього блоку. У роботі [53] була запропонована нова рандомізована версія симулятора Чена–Нгуєна, у якій враховувалася ймовірнісна природа евристики Гауса. Замість точного значення евристики Гауса використовувалися випадкові зміні. Це дало більшу точність симуляції для перших векторів в ГШ–профілі. В роботі [54] був запропонований варіант симулятора (симулятор Альбрехта–Лі), що враховує «Z–форму» q–арних решіток.

Для того, щоб порівняти якість роботи симуляторів було проведено ряд експериментів. Для решіток розмірності 120, 146, 170 і $q \in \{17{,}257\}$ було проведено симуляцію для розмірів блоків 45,50, 55, 60. Для кожного випадку за адаптованою для базисів формулою (2.6) було обчислено похибку. У таблицях 2.6–2.7 наведено експериментально обчислені значення похибки симуляції. У таблицях використовуються наступні позначення:

- ZGSA – модифікована модель GSA
- NChS – симулятор Чена–Нгуєна
- RNchS – рандомізований симулятор Чена–Нгуєна
- ALS – симулятор Альбрехта–Лі
- RALS – рандомізований симулятор Альбрехта–Лі

З таблиці 2.6 видно, що при екстримально малих значеннях паарметра q на модель ZGSA може показувати малі значення середньоквадратичної помилки. На розмірності 170 модель ZGSA навіть показує результати, що є кращими за більшість симуляторів. Це можна пояснити тим, що на цій розмірності останній блок редукції для розглянутих значення блоку редукції є достатньо малим, щоб модель ZGSA була адекватною, а явищ, які враховують симулятори не відбувалося.

Таблиця 2.6.

Похибка симуляції для $q = 17$

Розмір блоку редукції	Модель	Розмірність 120	Розмірність 146	Розмірність 170
45	ZGSA	0.0918627409	0.0894449220	0.0875595488
	NChS	0.0767525560	0.1031007579	0.1760477260
	RNchS	0.0655758504	0.1020406858	0.1811824129
	ALS	0.0554163968	0.0492938887	0.0471614769
	RALS	0.0572503430	0.1039426219	0.1892227936
50	ZGSA	0.0676759627	0.0895910545	0.0936503555
	NChS	0.0498812581	0.0714566622	0.1548769166
	RNchS	0.0496923086	0.0730467183	0.1570434753
	ALS	0.0491067359	0.0729945262	0.0510986377
	RALS	0.0519962792	0.0744609171	0.1578382409
55	ZGSA	0.0765852785	0.0905551754	0.0982828898
	NChS	0.0552639857	0.0489293062	0.1374005196
	RNchS	0.0546754607	0.0505093230	0.1391918609
	ALS	0.0539157170	0.0492426122	0.0452180625
	RALS	0.0566630060	0.0513962549	0.1405860991
60	ZGSA	0.0663607301	0.0911430468	0.1171579021
	NChS	0.0375912047	0.0363198544	0.1119574250
	RNchS	0.0377482240	0.0378406751	0.1134538197
	ALS	0.0354854945	0.0345229111	0.0554450885
	RALS	0.0388490447	0.0374229545	0.1149085830

У таблиці 2.7 наведено аналогічні результати для $q = 257$. З таблиці 2.7 видно, що результати відрізняються від даних в таблиці 2.6. При збільшенні параметра q модель ZGSA вже перестає гарно себе показувати.

Таблиця 2.7.

Похибка симуляції для $q = 257$

		Розмірність 120	Розмірність 146	Розмірність 170
45	ZGSA	0.0856547953	0.1143844549	0.1625077569
	NChS	0.0703067513	0.0818412572	0.0949361512
	RNchS	0.0577576377	0.0636084311	0.0690679321
	ALS	0.0459138584	0.0474779560	0.0536077376
	RALS	0.0496283293	0.0527500953	0.0557647428
50	ZGSA	0.0750900281	0.0728182579	0.0943241435
	NChS	0.0604062554	0.0474888101	0.0477398951
	RNchS	0.0604734344	0.0475457279	0.0458887251
	ALS	0.0600554860	0.0466250962	0.0468464543
	RALS	0.0620524373	0.0471859090	0.0469704474
55	ZGSA	0.0577970192	0.0609955853	0.0712081067
	NChS	0.0368758025	0.0380612775	0.0397216612
	RNchS	0.0372464477	0.0386950257	0.0390605797
	ALS	0.0358205944	0.0364759089	0.0385183694
	RALS	0.0397465001	0.0385757386	0.0396534254
60	ZGSA	0.0750109988	0.0603822820	0.0604462450
	NChS	0.0499051195	0.0338782000	0.0347489650
	RNchS	0.0508975278	0.0348965254	0.0353331483
	ALS	0.0478939300	0.0317893058	0.0330983680
	RALS	0.0518109223	0.0347100224	0.0351538199

Для більшої наглядності усереднені значення помилок наведено на рисунках 2.7–2.8. Доволі цікавим на рисунку 2.7 є те, що значення середньоквадратичних помилок майже не збільшуються зі збільшенням розмірності. При цьому, з рисунка 2.8 випливає, що на розмірності 170 середньоквардатична помилка майже всіх симуляторів різко збільшується.

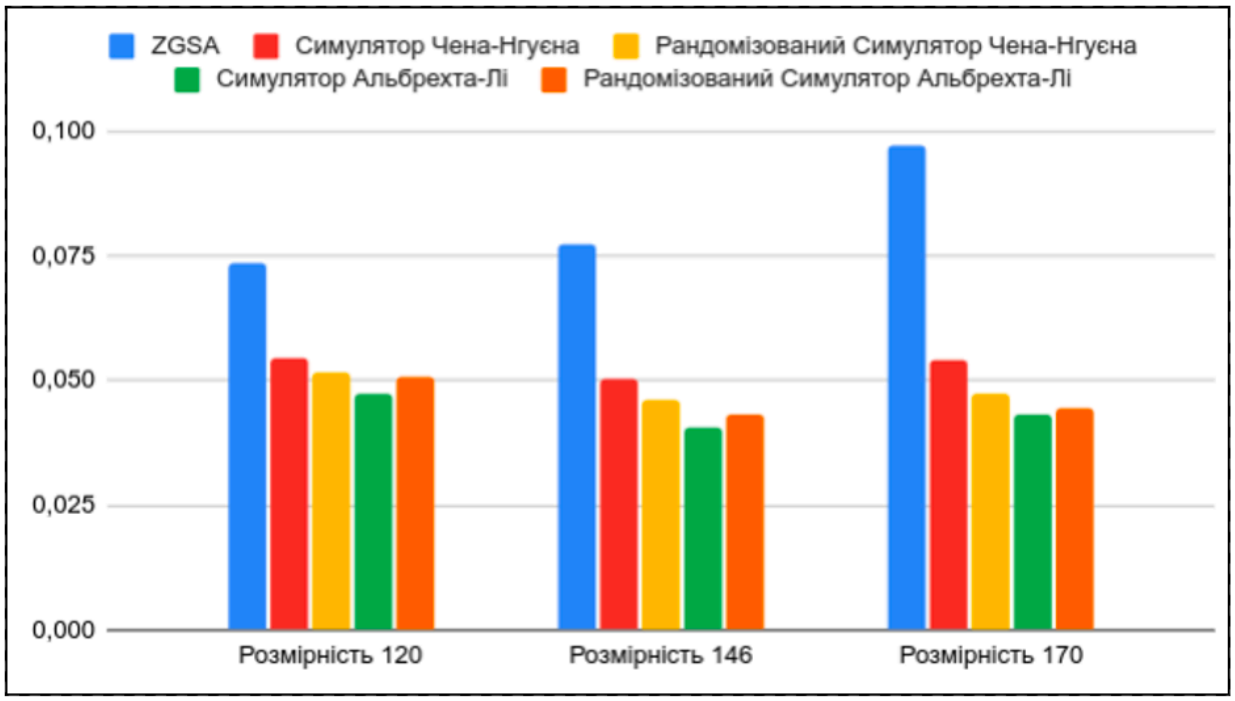

Рис. 2.7. Середньоквадратичні помилки симуляторів для $q = 256$

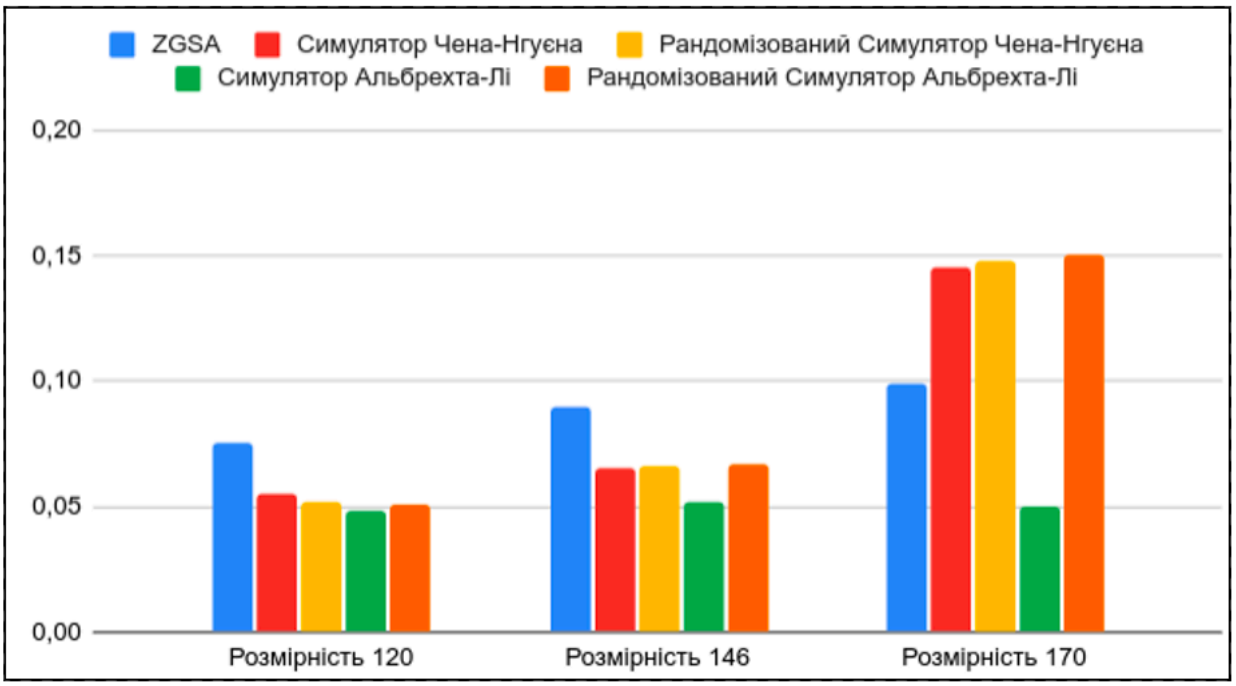

Рис. 2.8. Середньоквадратичні помилки симуляторів для $q = 17$

Оскільки з рисунків 2.7–2.8 та таблиць 2.6–2.7 видно, що симулятор Альбрехта–Лі в обох випадках показує найменшу середньоквадратичну помилку, то має сенс використовувати його для моделювання редукції решіток [55]. Цікаво, що рандомізований симулятор Альбрехта–Лі через рандомізацію

погіршив свою якість передбачення профілю редукованого базису. На рисунку 3.7 наведено приклад такої ситуації.

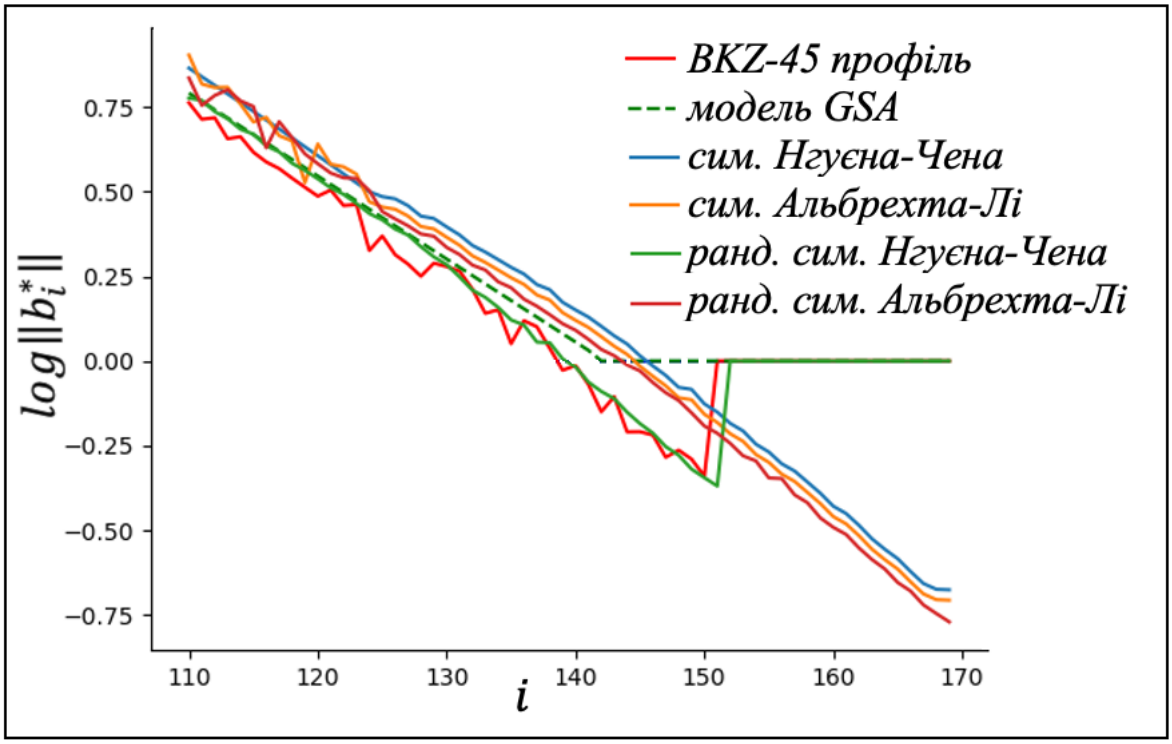

Рис. 2.9. Приклад не коректної роботи рандомізованого симулятора Чена–Нгуєна.

2.4. Вплив розріджених підрешіток на швидкість редукції

Особливістю NTRU решіток є велика кількість малих векторів. Зафіксуємо деяке поле $\mathbb{Z}_q[X]/(\phi(X))$ для деякого незвідного полінома $\phi, \deg(\phi) = n$. Якщо $h = g \cdot f^{-1}(mod q)(mod \phi(X))$ для деяких поліномів f, g, то решітка розмірності $2n$ з базисом

$$B_{NTRU} = \begin{pmatrix} qI_n & rot(h) \\ 0 & I_n \end{pmatrix} \tag{2.12}$$

Міститиме вектори $(g, f), (x \cdot g, x \cdot f), \dots, (x^{n-1} \cdot g, x^{n-1} \cdot f)$. Ці вектори формують підрешітку розмірності n з базисом

$$B_{NTRU}^{dense} = \begin{pmatrix} rot(g) \\ rot(f) \end{pmatrix} \tag{2.13}$$

У криптографічному випадку поліноми f, g є малими (мають малі значення коефіцієнтів), тому підрешітка з базисом (2.12) міститиме велику кількість векторів, що значно менші за евристику Гауса. Знаходження векторів на підрешітці (2.13) під час редукції решітки дуже швидко призводить до знаходження інших векторів підрешітки, що розбиває редукцію базису (2.12) на дві незалежні частини і для багатьох параметрів призводить до швидкого знаходження f, g. Приклад такої ситуації наведено на рисунку 2.8.

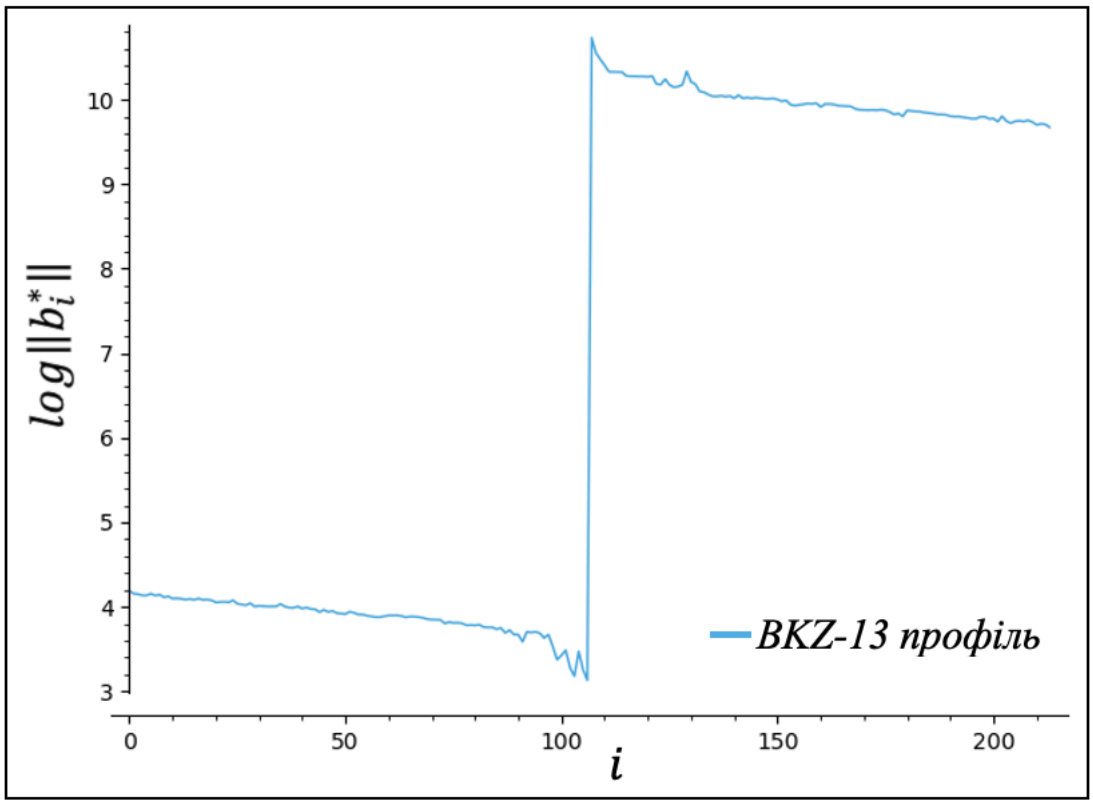

Рис. 2.8. Перехід на розріджену решітку. Кожна з підрешіток під час редукції не залежить від іншої, що зменшує складність редукції

Вперше ця особливість NTRU решіток була помічена у контексті алгебраїчних атак. Зокрема, так звані атаки на підполе [56]. Проте, у роботі [57] було показано, що розріджені підрешітки призводять до пришвидшення редукції решіток без алгебраїчних технік, що використовувалися в попередніх атаках.

Фактично, при криптоаналізі NTRU решіток є цікавими 2 події:

- Відновлення таємного ключа – вектор (g, f) буде міститися у базисі решітки.

- Потрапляння на розріджену підрешітку – деякий вектор (значно довший за (g, f)) з розрідженої підрешітки буде міститися у базисі

У роботі [58] був отриманий наступний результат. Нехай Λ_{NTRU} – NTRU–решітка розмірності $2n$ з розрідженою решіткою $\Lambda_{NTRU}^{dense} \subset \Lambda_{NTRU}$. Якщо Λ_{NTRU} має базис, що має Z–форму, то під час BKZ–β редукції буде знайдено вектор з розрідженої решітки (що значно більший за таємний ключ), якщо

$$vol\left(\Lambda_{NTRU}^{dense}\right) < q^{(m-1)/2} \cdot \alpha_{\beta}^{-\frac{1}{2}(m-1)^2} \tag{2.14}$$

Де $\alpha_{\beta} = gh(\beta)^{2/(\beta-1)}$ і $m = \frac{1}{2} + \frac{lnq}{2ln\alpha_{\beta}}$.

У роботі [58] був запропонований інший підхід до виявлення події потрапляння на розріджену підрешітку. Згідно [58], потрапляння на розріджену решітку $\Lambda_{NTRU}^{dense} \subset \Lambda_{NTRU}$ відбувається, якщо має місце нерівність

$$\pi_{n+k-\beta}(v) < \left\|b_{n+k-\beta}^{*}\right\| \tag{2.15}$$

Де v – найменший вектор на решітці $\Lambda_{proj}^{dense} = \Lambda(\pi_0(b_0), \dots, \pi_0(b_{n+k-1})) \cap \Lambda_{NTRU}^{dense}$. Значення v можливо оцінити на основі евристики Гауса, проте для цього необхідна оцінка детермінанту решітки Λ_{proj}^{dense}. Автори [58], виходячи з тих самих міркувань, що і оцінка (2.14), запропонували наступну оцінку об'єма Λ_{proj}^{dense}:

$$vol\left(\Lambda_{proj}^{dense}\right) \leq vol\left(\Lambda_{NTRU}^{dense}\right) \cdot \left(\prod_{j=n+k}^{d-1}\left\|b_j^{*}\right\|\right)^{-1} \tag{2.16}$$

Оцінку (3.14) зручно використовувати, коли використовується модель ZGSA, проте оцінка (3.15) є більш гнучкою у тому сенсі, що дозволяє використовувати симулятори. Вона теж неявно вимагає Z–форми базису.

2.5. Висновки до розділу

1. Модель GSA є основною моделлю редукції решіток у криптографії. Модель GSA цілком залежить від одного параметра – кореневого фактора Ерміта. Існуючі оцінки кореневого фактора Ерміта є асимптотичними і невідомо наскільки швидко відбувається сходимість. Для оцінки можливої похибки були проведені дослідження на решітках малої розмірності. Була отримана формула,

що враховує значення похибки оцінки фактора Ерміта і для малих розмірностей експериментально обчислено значення похибки. Вже на малих розмірностях похибка є доволі малою. Враховуючи, що зі збільшенням розмірності похибка буде зменшуватися, то можливо вважати, що для криптографічно значущих розмірностей значення похибки апроксимації фактора Ерміта є достатньо малими, щоб ними можливо було нехтувати.

2. Модель GSA є доволі грубою і не враховує багатьох факторів. ZGSA є модифікацією GSA, що враховує особливості q–арних решіток. Іншим підходом є використання симуляторів решіток. Був проведений порівняльний аналіз точності симуляторів редукції решіток відносно базової моделі ZGSA для оцінки форми базису. Серед розглянутих симуляторів найбільшу точність показав симулятор Альбрехта–Лі для усіх розміностей. При цьому рандомізована версія цього симулятора показала гірші значення середньоквадратичної помилки. Рандомізація дозволяє врахувати відхилення перших векторів профіля решітки від моделі GSA, проте, отриманий результат вказує на те, що така рандомізація заважає врахуванню q–арної структури решіток під час моделювання.

3. NTRU решітки мають розріджену підрешітку. Це необхідно враховувати при оцінці складності проблеми NTRU. Оцінки ймовірності переходу на розріджену решітку залежать від форми базису під час редукції. Існуючі оцінки ймовірностей переходу на розріджену решітку вже враховують форму q–арних решіток.

4. У цьому розділі була вирішена друга задача дисертаційної роботи – був проведений аналіз існуючих моделей редукції та було виявлено найкращі моделі. Для оцінки профілю решіток найкращим вибором є використання симулятора Альбрехта–Лі. Для NTRU решіток додатково необхідно враховувати можливість переходу на розріджену решіток.

РОЗДІЛ 3. МЕТОДИ ОЦІНКИ СКЛАДНОСТІ КРИПТОГРАФІЧНИХ ЗАДАЧ З ТЕОРІЇ РЕШІТОК

У криптографії на решітках основними складними проблемами є LWE, SIS, їх структуровані різновиди та проблема NTRU. Складність цих проблем доказово зводиться до складності класичних проблем з теорії решіток (проблема пошуку найменшого вектору, проблема пошуку найближчого вектору, тощо). Проте, як видно з дослідів у розділі 2, форма базису може відрізнятися в залежності від моделі редукції решітки, що визначає поведінку базису під час процесу редукції, а отже і складність криптоаналізу. Цей розділ присвячений уточненню оцінок складності вирішення проблем LWE, SIS та NTRU, шляхом врахування відомих на даний момент особливостей редукції решіток, на основі отриманих у попередньому розділі результатів.

3.1. Класифікація та аналіз відомих атак

Відомі атаки на проблеми LWE, SIS та NTRU можливо поділити на наступні класи:

- Комбінаторні атаки [59, 60];
- Алгебраїчні атаки [56, 61, 62];
- Атаки декодування [63];
- Атаки розпізнавання [64];
- Гібридні атаки [65];
- Атаки вкладення [66].

До комбінаторних атак належать атака повного перебору та атака MITM (англ. Meet In The Middle). Зазвичай MITM використовується не самостійно, а як складова більш комплексних гібридних атак. Також до комбінаторних атак можливо віднести атаку BKW.

До алгебраїчних атак на LWE відносять атаку Arora–Ge [61], сутність якої полягає у зведенні проблеми LWE до вирішення системи поліноміальних рівнянь над скінченним полем. Атака є суто теоретично, оскільки складність вирішення

такої системи зростає набагато швидше, ніж складність редукції решіток. Також, якщо розглядати проблему LWE на ідеальних решітках, то існує ряд квантових атак [62], що використовують структуру ідеалів для значного пришвидшення. Проте, такі атаки як правило працюють лише для не криптографічних випадків. Наразі не має суттєвих доказів того, що такі атаки можуть бути розширені для криптографічних параметрів. Для NTRU алгебраїчною атакою є атака на підполе, яка розглядалася у другому розділі.

Атаки вкладення є одними з найбільш ефективних атак на LWE та NTRU. Сутність таких атак полягає у побудові решіток спеціального вигляду, найменший вектор яких містить шуканий секрет. Такі атаки ще називають атаками первинними (англ. Primal) атаками. Також їх можливо використовувати для вирішення проблеми SIS.

Сутність атак декодування полягає у зведенні проблеми LWE або NTRU до проблеми CVP. Атаки такого роду вимагають побудови та редукції базису решітки таким чином, щоб було можливо вирішити проблему CVP для шуканого таємного вектору. Проведення атак декодування є технічно складнішим за атаки вкладення через двоетапну структуру атаки. Атаки декодування, як і атаки вкладення, також іноді називають первинними атаками (англ. BDD Primal attacks).

Атаки розпізнавання часто називають дуальними атаками через те, що вони зводяться до редукції дуальних (відносно атак вкладення) решіток. У таких атаках використовуються статистичні методи аналізу. Перед криптоаналітиком стоїть задача відрізнення двох розподілів ймовірностей. У поєднанні з комбінаторними методами дуальні атаки можуть давати гарні результати для проблеми LWE. Проте, дуальні атаки, як правило, дають гірші результати у порівнянні з атаками вкладення та декодування.

Гібридні атаки поєднують комбінаторні методи криптоаналізу з атаками вкладення або атаками розпізнавання (гібридні дуальні атаки). Такі атаки при використанні розріджених секретів часто є найкращими для багатьох криптографічних систем. Це є особливо актуальним для ДСТУ 8961:2019.

Розглянемо більш детально атаки для оцінки впливу моделей редукції решіток на безпеку криптографічних перетворень.

3.2. Атаки вкладення

Розглянемо атаку вкладення на прикладі проблеми LWE. Зафіксуємо пару $(A, b = A \cdot s + e)$ відповідно до визначення проблеми LWE. Надалі вважаємо, без втрати загальності, що вектори e, s мають центрований нормальний розподіл з дисперсіями σ_e та σ_s відповідно.

Атака ґрунтується на тому факті, що решітка

$$\Lambda_\omega = \left(x \in \mathbb{Z}^{m+n+1} : \left(I_m \left| \frac{1}{\omega} A \right| - \frac{1}{\omega} b\right) x = 0 mod q\right) \tag{3.1}$$

містить найменший вектор $v = \lambda_1(\Lambda_\omega) = (e, \omega \cdot s, \omega)$, де $\omega \in \mathbb{R}$ – параметр масштабування. Значення цього вектору задовільнять LWE рівнянню для матриці A. Параметр масштабування може бути корисним у випадку, якщо розподіли e та s мають різну дисперсію. Типовим значенням цього параметра є $\omega = \sigma_e / \sigma_s$. При такому виборі складність атаки є найменшою серед усіх можливих значень ω, проте оскільки ω не сильно впливає на складність атаки, то часто використовується для простоти $\omega = 1$ для простоти реалізації.

Для оцінки складності пошуку вектору $\lambda_1(\Lambda_\omega)$ не можна застосовувати стандартні припущення на основі Евристики Гауса, оскільки вектор $\lambda_1(\Lambda_\omega)$ набагато менший за $GH(\dim(\Lambda_\omega))$ для типових криптографічних параметрів. На практиці себе добре зарекомендував [42] критерій

$$\left\|\pi_{d-\beta+1}(v)\right\| \leq \left\|b^*_{d-\beta+1}\right\|. \tag{3.2}$$

Ідея, що лежить за критерієм (3.2), полягає у наступному. Малі вектори у алгоритмі BKZ знаходяться за допомогою SVP–оракула і далі за допомогою алгоритму LLL (більш конкретно – за допомогою кроку редукції за розміром) ці малі вектори вставляються в новий базис. Якщо задано найменше β, для якого виконується (3.2), то, скоріш за все, потрібний вектор буде знайдено під час останнього виклику SVP–оракула. Знайдений вектор буде вставлений у базис

тільки у тому випадку, якщо (3.2) виконується, за визначенням алгоритму BKZ [36].

У роботі [68] було доведено, що ймовірність відновлення таємного вектору, якщо виконується критерій (3.2), складає

$$p = \sum_{i=1}^{d-\beta} \Pr\left[\|\pi_i(v)\| < \min\{\|\pi_i(v) + b_i^*\|, \|\pi_i(v) - b_i^*\|\}\right] \quad (3.3)$$

При $\beta > 50$ ймовірність (3.3) швидко наближається до 1. На малих розмірностях іноді таємний вектор може відновлюватися за менших значень β. Це явище пояснюється геометрією решіток [68] і на великих розмірностях не спостерігається, тому при подальшому аналізі ігнорується.

Якщо припустити, що таємний вектор є однорідним (його часто можливо зробити таким, використовуючи параметр масштабування), то $\left\|\pi_{d-\beta+1}(v)\right\| \approx \sqrt{\beta/d}\|v\|$. У свою чергу, оскільки v має нормальний розподіл, то маємо оцінку $\|v\| \approx \sigma\sqrt{d}$. Тож, $\left\|\pi_{d-\beta+1}(v)\right\| \approx \sqrt{\beta}\sigma$, де σ – середньоквадратичне відхилення для компонентів вектору v. Цей факт може бути використаним для перевірки формули (3.2) на реальних параметрах.

Обчислення правої частини нерівності залежить від моделі редукції решіток. Якщо використовується модель GSA, що досліджувалася у попередньому розділі, то, відповідно до визначення GSA, маємо:

$$\left\|b_{d-\beta+1}^*\right\| \approx \delta_\beta^{2\beta-d} \cdot vol(\Lambda_\omega)^{\frac{1}{d}} = \delta_\beta^{2\beta-d} \cdot q^{\frac{m}{d}}\omega^{(n+1)/d} \quad (3.4)$$

У випадку використання симуляторів $\left\|b_{d-\beta+1}^*\right\|$ має бути обчислено експериментально.

Варто зауважити, що складність атаки залежить не монотонно від значення m . У роботі [69] було доведено, що оптимальним значенням є

$$m_{opt} = \left\lfloor \sqrt{\frac{(n+1)(logq - log\omega)}{log\delta_\beta}} - (n+1) \right\rfloor \quad (3.5)$$

Таким чином, оцінка атаки вкладення зводяться до знаходження найменшого β, для якого виконується (3.2). При цьому мають бути задані:

- Параметри решітки (n, q, σ)
- Модель редукції решіток, що визначає $\left\|b_{d-\beta+1}^*\right\|$

- Модель часу роботи редукції

Оскільки алгоритм BKZ та усі інші методи редукції решіток роблять поліноміальну кількість запитів до SVP–оракула, то можливо вважати час роботи BKZ часом роботи SPV–оракула, що помножений на деякий поліноміальний фактор. Враховуючи, що для реальної атаки можливо виконувати лише перші 8–10 раундів алгоритму редукції, то поліноміальним фактором можливо знехтувати і у якості моделі часу роботи редукції обрати час роботи SPV–оракула. Оцінки часу роботи SVP–оракулів були наведені в першому розділі.

Оскільки модель часу роботи редукції решіток слугує для перетворення оптимальних параметрів редукції у конкретну оцінку безпеки, тому при дослідженні впливу моделей редукції решіток на безпеку, її можливо не задавати, а досліджувати безпосередньо оптимальні параметри атаки.

На рисунку 3.1 наведено результати моделювання атаки вкладення з використанням моделі GSA та симулятора для $n = 256$ при $\sigma = \{1.1, 1.3, 1.5, 1.8, 2.1\}$. З рисунка 3.1 видно, що при збільшенні параметра q зменшується оптимальне для атаки значення β, що є логічним, враховуючи формулу (3.2): при збільшенні q права частина рівняння (3.2) збільшуватиметься, у той час як ліва частина залишатиметься такою ж. Втім, графік для симуляторів має дві ключові відмінності. Відрізняється сила впливу параметра σ на оптимальне значення β. При використанні моделі GSA спостерігається трохи більший розмах між випадками $\sigma = 1.1$ та $\sigma = 2.1$ на більшій частині досліджуваного простору параметрів. При малих значеннях q це може суттєво впливати на значення β.

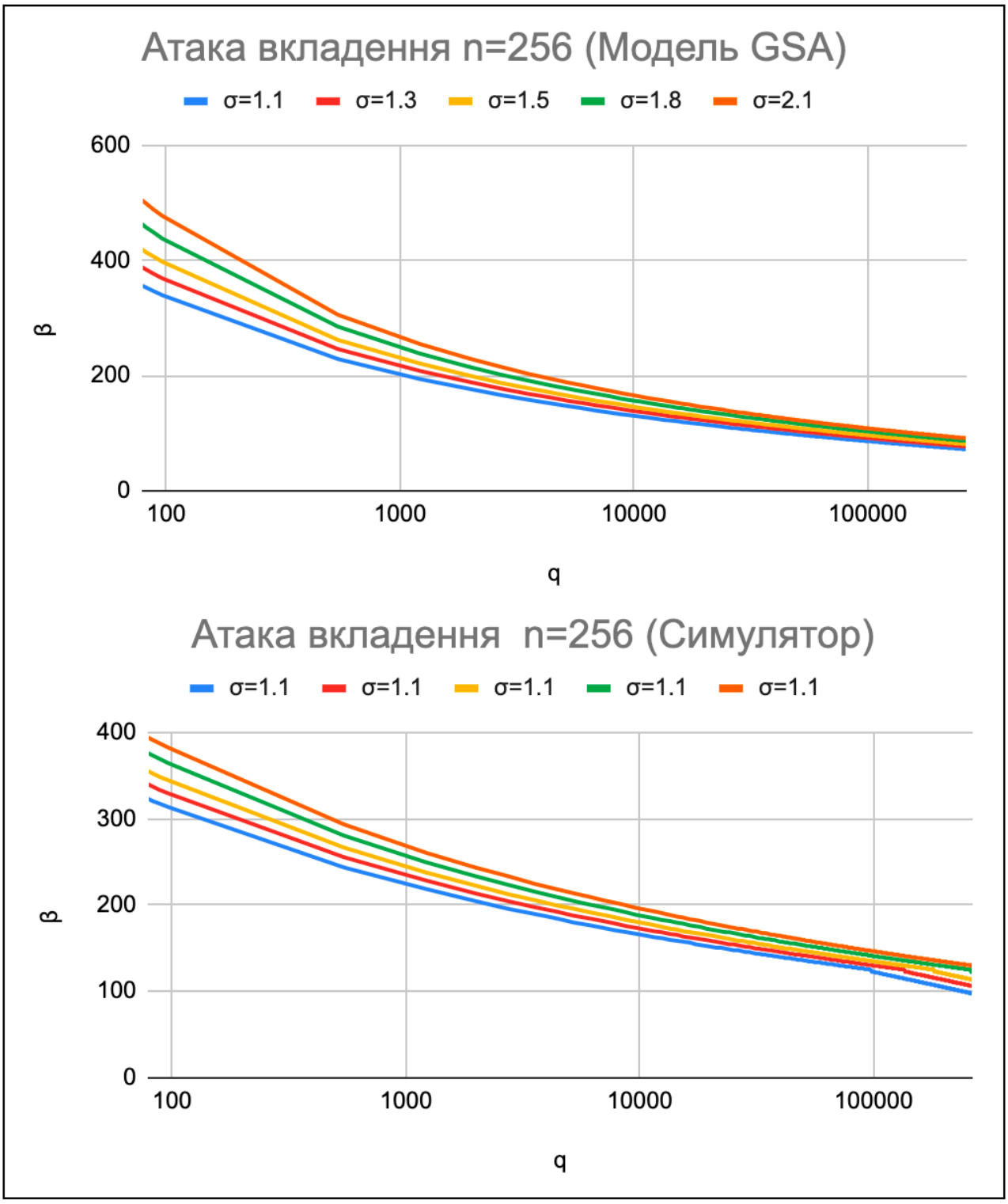

Рис. 3.1. Результати моделювання атаки вкладення

Для більш наглядної демонстрації різниці між симулятором та моделлю GSA на рисунку 3.2 наведено накладені один на одного графіки з рисунку 3.1. З рисунка 3.2 видно, що на малих значеннях q симулятор дає менші значення параметра β, у порівнянні з моделлю GSA. У той же час, для великих значень q

симулятор дає більші значення β, що свідчить про те, що модель GSA дещо занижує рівень безпеки. Отриманні свідчення можливо пояснити тим, що останній блок у профілі решітки буде HKZ редукованим, а отже $\left\|b^*_{d-\beta+1}\right\|$ буде мати трохи більше значення, ніж передбачене GSA значення. Оскільки симулятор враховує це явище, то данні на рисунку 3.2 виглядають цілком природніми [55].

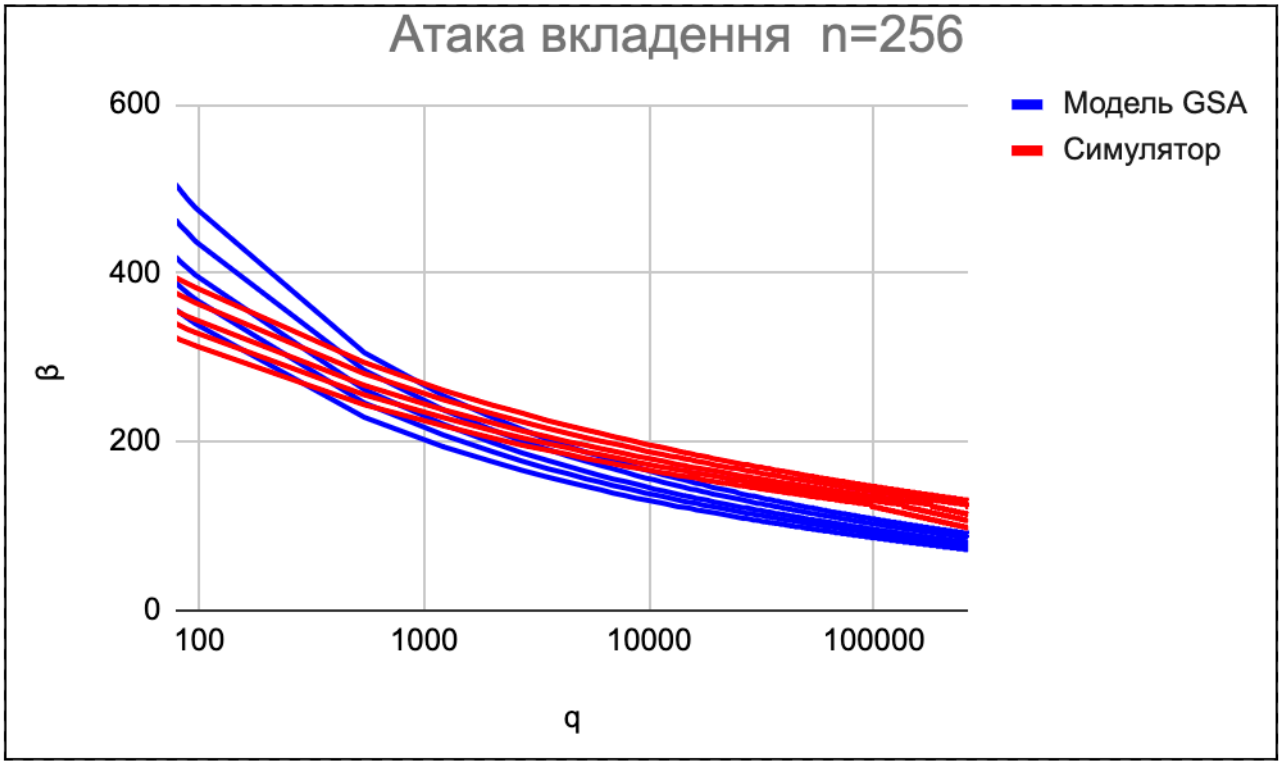

Рис. 3.2. Порівняння моделі GSA та симулятора Чена–Нгуєна

При збільшенні параметра n результати моделювання виглядають схожим чином і для них зберігаються усі описані явища. На рисунку 3.3 наведено результати моделювання для $n = 256,512,1024,2048$ при фіксованому $\sigma = 1.1$.

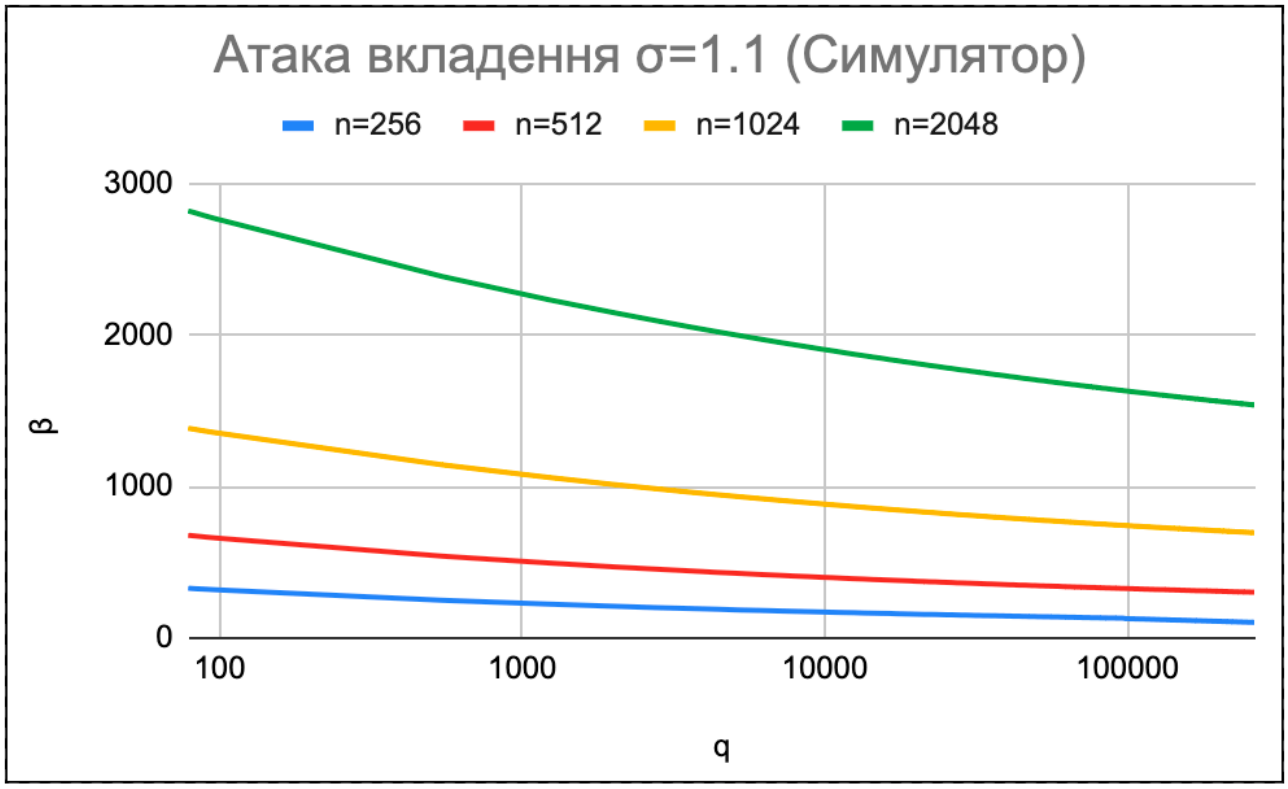

Рис. 3.3. Результати моделювання атаки вкладення для $n = 256,512,1024,2048$ при фіксованому $\sigma = 1.1$

З рисунка видно, що збільшення n дає лінійний приріст значення β, з чого випливає стратегія пошуку оптимальних загальносистемних параметрів: зафіксувати значення n, що дає близьке до необхідного рівня безпеки значення і підлаштовувати рівень безпеки змінюючи параметри q, σ.

Для проблеми NTRU базис решітки веде себе так само, окрім ситуацій, коли виникає перехід на розріджену решітку. Згідно до [58], точкою переходу є $q \approx 0.0038 \cdot n^{2.484}$. До цієї точки поведінка не буде відрізнятися, проте починаючи з цієї точки параметри не будуть криптостійкими [55]. Це потрібно враховувати при аналізі.

3.3. Атаки декодування

Атака декодування зводить проблему LWE до задачі BDD (англ. Bounded Distance Decoding), сутність якої полягає у знаходженні достатньо близької

точки решітки до заданої точки. Точка $t = A \cdot s + e$ може розглядатися як деяка зашумлена точка $v = A \cdot s$ решітки.

Для проведення атаки будується решітка

$$\Lambda = \{y = A \cdot x \; mod \; q | x \in \mathbb{Z}^m\} \tag{3.6}$$

Базисом такої решітки буде стандартний базис q–арної решітки.

Далі для вектору $t = A \cdot s + e$ вирішується задача BDD, результатом вирішення якої є вектор $v = A \cdot s$, знаючи який можливо легко відновити таємний ключ s, вирішуючи систему $A \cdot s = 0$ стандартними методами лінійної алгебри за поліноміальний час.

Щоб знайти найближчий вектор v необхідно спочатку провести редукцію базису (3.6). У загальному випадку складність атаки можливо знайти за формулою:

$$(T_{red} + T_{bdd})/p_{succ} \tag{3.7}$$

Де T_{red} – час редукції базису решітки (3.6), T_{bdd} – час алгоритму пошуку найближчого вектора, p_{succ} – ймовірність вдалого завершення атаки.

У якості алгоритму пошуку найближчого вектору, як правило, використовується алгоритм Бабаї та його узагальнення. Алгоритм Бабаї гарантує, що $v - t \in P_{\frac{1}{2}}(B^*)$, тож для цього випадку необхідною умовою вдалого завершення атаки є $e \in P_{\frac{1}{2}}(B^*)$.

У випадку нормального розподілу помилки, у роботі [52] запропонована наступна оцінка:

$$p_{succ} = \Pr\left[e \in P_{\frac{1}{2}}(B^*)\right] = \prod_{i=1}^{m} erf\left(\frac{\|b_i^*\|\sqrt{\pi}}{2\sigma_e}\right) \tag{3.8}$$

Для класичного алгоритму Бабаї T_{bdd} є поліноміальним і може не враховуватися. Проте, використання більш складних алгоритмів пошуку найближчого вектора, що працюють за субекспоненційний час, може зменшити (3.7). Такі алгоритми, як правило, тісно повязані з SVP оракулами. Ідеї, що використовуються для побудови SVP оракулів, також можуть бути адаптованими для побудови алгоритмів вирішення задачі BDD, як як це було показано у роботі [63]. Сутність ідеї полягає у тому, щоб у алгоритмі Бабаї

обчислювати не тільки найоптимальніші координати $c_0, \dots, c_{d-1}$, а для кожної координати c_i перебираються d_i найоптимальніших значень. Тоді, складність алгоритму пошуку найближчого вектору при використанні найпростішої стратегії перебору складатиме $O(\prod_i d_i)$, яка вже буде не поліноміальною. Тоді, оцінка (3.7) буде мінімізуватися, коли $T_{red} \approx T_{bdd}$. Ймовірність p_{succ} відповідно складатиме

$$p_{succ} = \prod_{i=1}^{m} p_{succ}^{i} = \prod_{i=1}^{m} erf\left(\frac{d_i \|b_i^*\| \sqrt{\pi}}{2s}\right) \tag{3.9}$$

Питання вибору значень d_i в літературі є малодослідженим. Якщо використовувати модель GSA, то значення $\|b_i^*\|$ будуть розподілені за експоненціальним розподілом. Значення d_i, що мають відмінне від одиниці значення, будуть згруповані у хвості профіля решітки. Проте, оскільки у q–арних решіток останні значення $\|b_i^*\|$ сильно відхиляються від GSA і $\|b_i^*\| \approx 1$, що повинно зменшувати відповідні значення d_i для цих векторів, а отже і зменшувати відповідне значення T_{bdd}.

Для пошуку відповідних значень d_i зафіксуємо мінімальну ймовірність успішного виконання атаки p_0. Розглянемо стратегію пошуку значень d_i. Стратегія випливає з міркувань, що існує така ймовірність p_{avg}, що

$$p_0 \leq \prod_{i=1}^{m} p_{succ}^{i} < \left(p_{avg}\right)^m \tag{3.10}$$

Зрозуміло, що мінімальне таке значення є $p_{avg} = (p_0)^{1/m}$. Тоді, якщо кожне p_{succ}^{i} буде більшим за p_{avg}, то буде виконуватися $p_0 \leq p_{succ}$. Звідси, маємо:

$$p_{succ}^{i} = erf\left(\frac{d_i \|b_i^*\| \sqrt{\pi}}{2s}\right) \leq p_{avg} \Rightarrow$$

$$d_i \leq \frac{erfinv(p_{avg}) \cdot 2\sigma_e}{\|b_i^*\| \sqrt{\pi}} \tag{3.11}$$

На рисунку 3.4 зображена залежність p_{avg} від p_0 для різних значень параметра m.

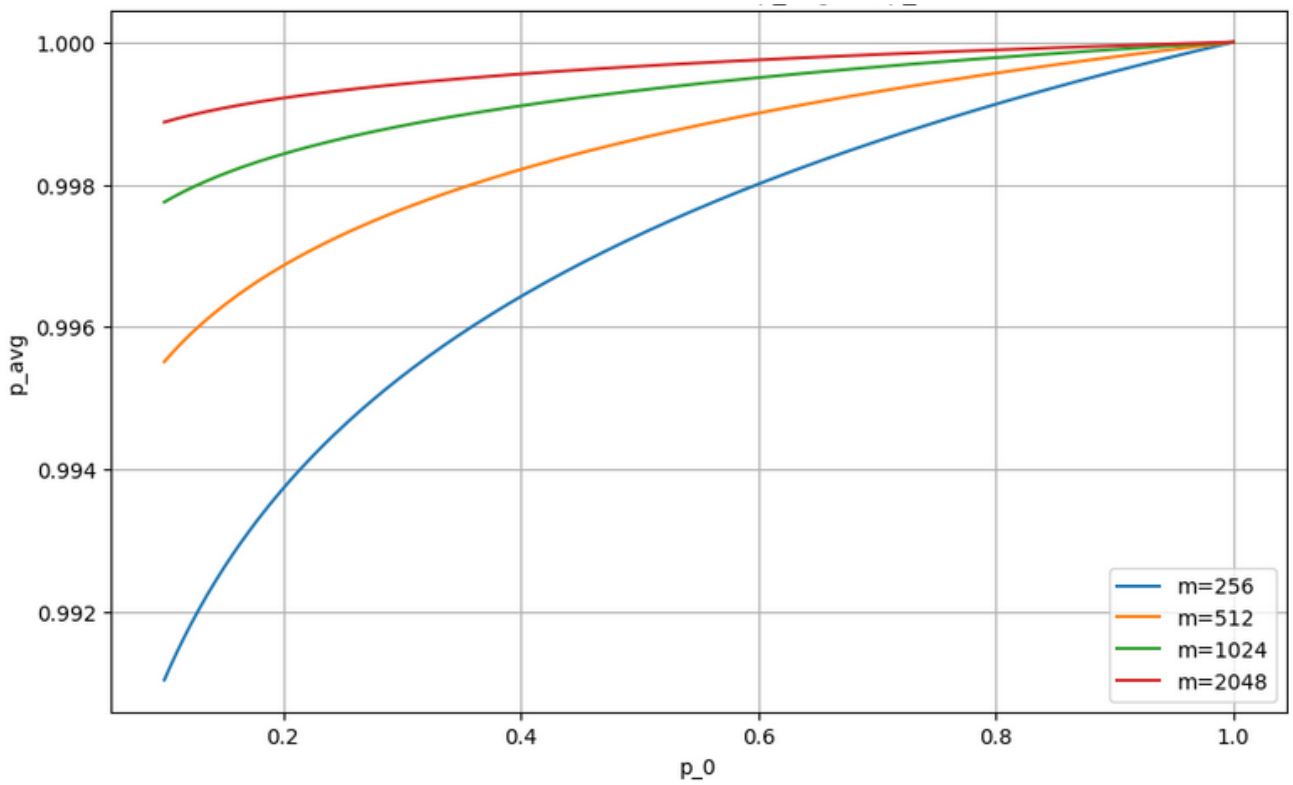

Рис. 3.4. Залежність p_{avg} від p_0 для різних значень параметра m.

З рисунка 3.4 видно, що для типових значень параметра m значення параметра p_{avg}, лежить близько до 1, тобто вплив ймовірності p_0 на значення d_i є не значним.

Тож, для атаки декодування мають бути задані

- Модель редукції решіток, що визначає профіль решітки
- Модель часу роботи редукції решітки
- Модель часу роботи алгоритму Бабаї або його узагальнення

У межах дослідження було проведено моделювання атаки декодування для моделі GSA та симулятора для розмірностей $N = 256,512,1024,2048$ та значень параметра $\sigma = 1.1,1.3,1.5$. Результати моделювання для розмірності 512 наведені на рисунку 3.5.

З рисунка 3.5 видно, що вартість атаки з використанням симулятора є вищою, проте такої великої різниці, як в атаках вкладення не має. Це пояснюється тим, що вартість атак вкладення залежить лише від значення $\left\|b^*_{d-\beta+1}\right\|$, у той час, як вартість атак декодування залежить цілком від всього

профіля редукованої решітки. Також видно, що складність етапу редукції решітки та етапу пошуку найближчого вектору хоч і наближаться один до одного, проте не дорівнюють один одному, тобто мінімум формули (3.7) не досягається через дискретність параметрів.

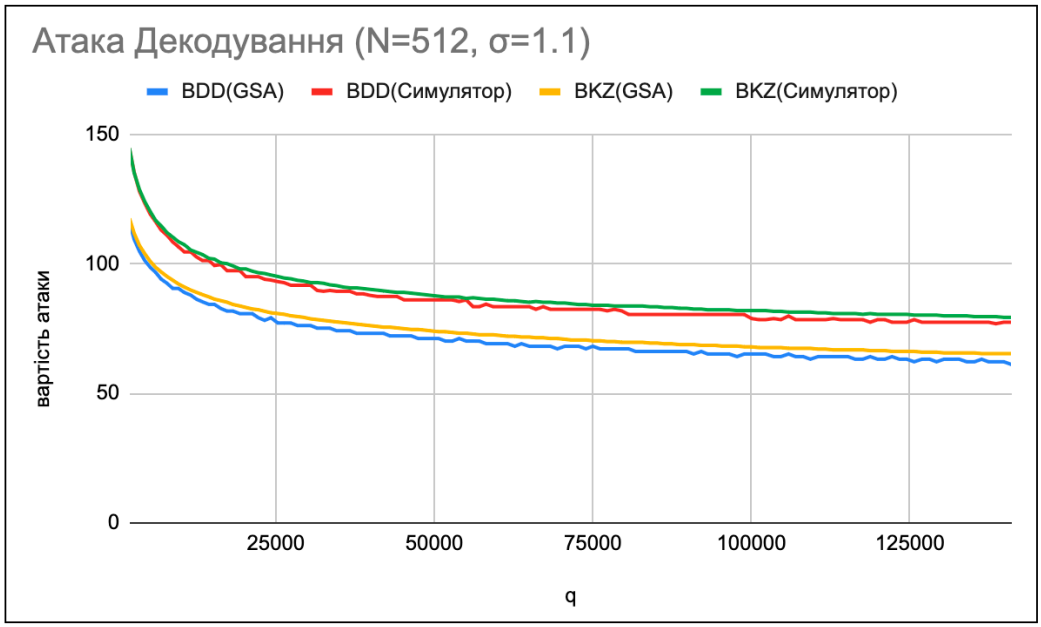

Рис. 3.5. Результати моделювання атаки декодування для $N = 512, \sigma = 1.1$

На рисунку 3.6 показано вплив збільшення параметра N на складність атак декодування.

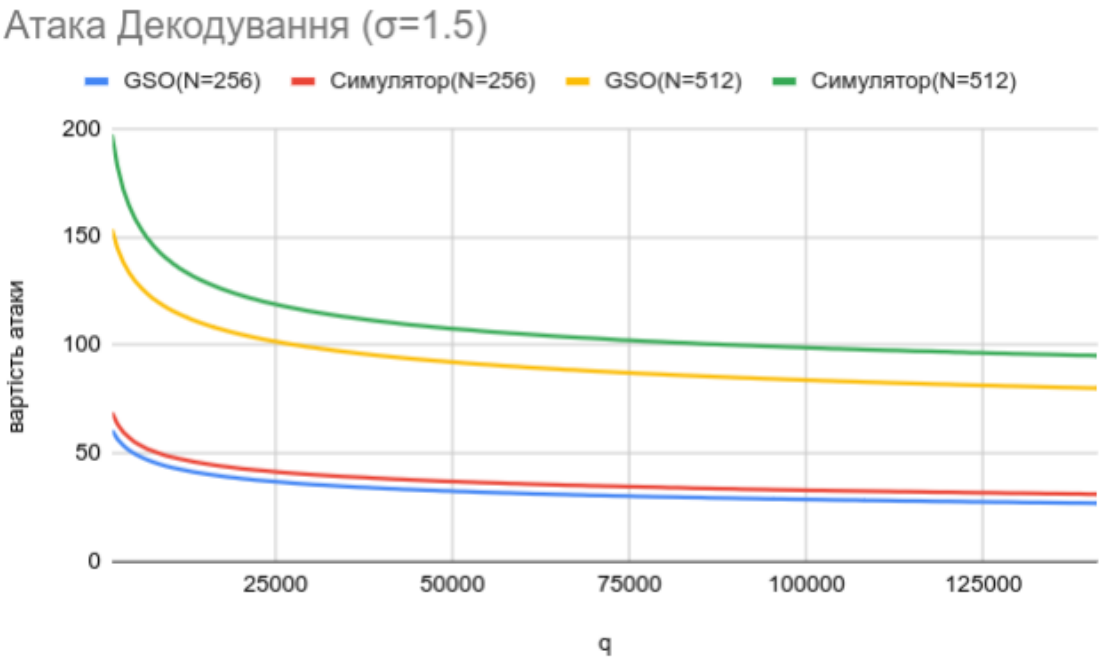

Рис. 3.6. Вплив параметра N на складність атак декодування

Як видно з рисунка 3.6, збільшення параметра N вдвічі збільшує складність атаки приблизно вдвічі. Цікаво, що при цьому зростає вплив симулятора на складність атаки, чого так явно не спостерігалося для атак вкладення, що , знов ж таки, пояснюється тим, що форма профіля базису в атаках декодування впливає сильніше на складність атаки.

На рисунку 3.7 показано вплив параметра σ на вартість атак декодування

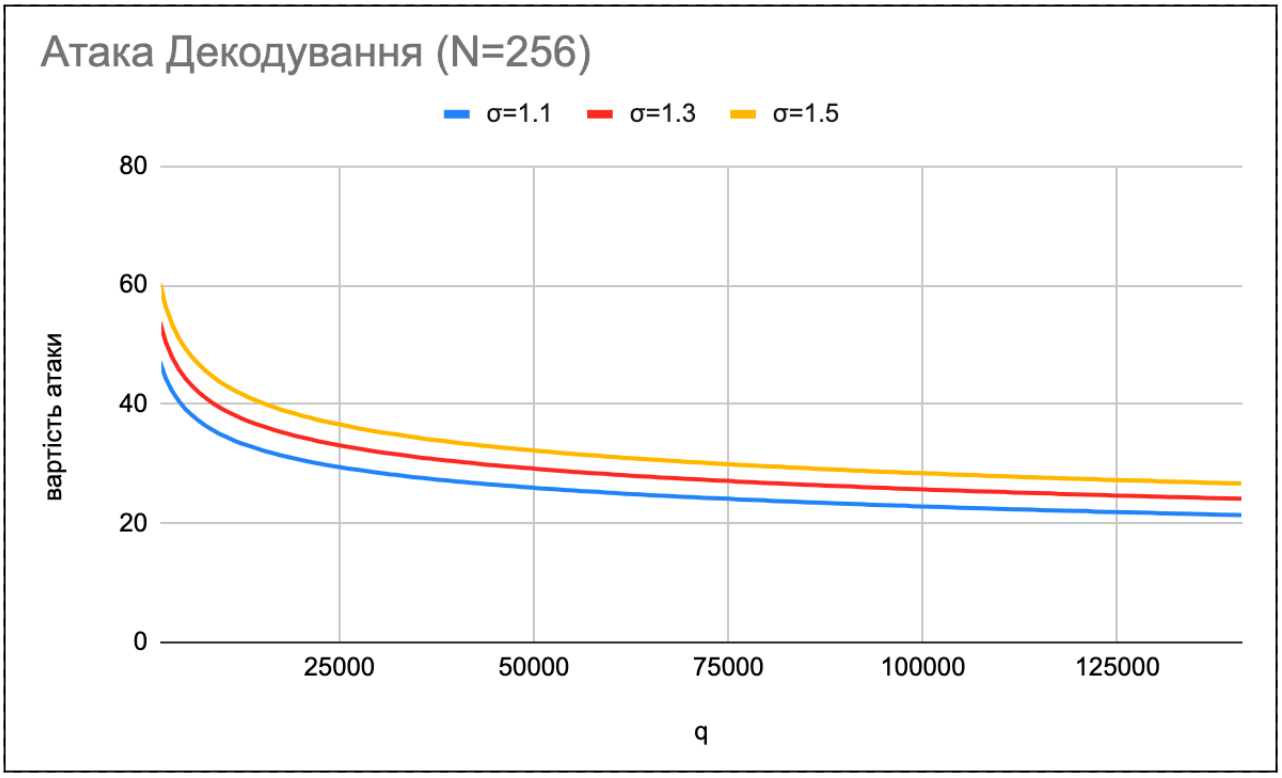

Рис. 3.7. Вплив параметра σ на вартість атак декодування

Великої різниці у впливі параметра σ при використанні моделі GSA та симулятору не має. Як і в атаках вкладення, параметр σ можливо використовувати для уточнення параметрів безпеки.

На рисунку 3.8 наведено порівняння складності атак вкладення та декодування.

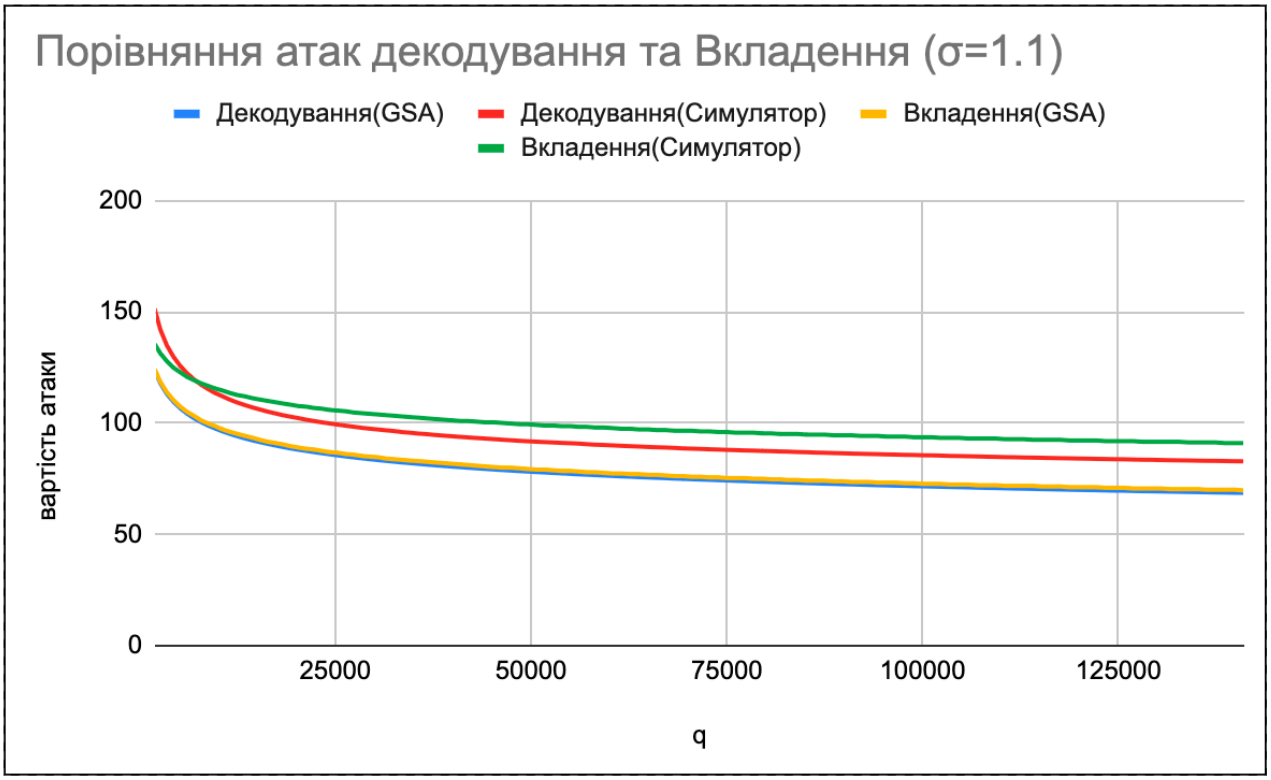

Рис. 3.8. Порівняння атак вкладення та декодування

З рисунку 3.8 можливо зробити декілька висновків. При використанні моделі GSA складність атак декодування та вкладення є майже однаковою. Фактично, різниця настільки не суттєва, що їх вартість можливо вважати однаковою. Проте, при використанні симуляторів різниця між атаками вкладення та декодування стає помітною [55]. Для переважної частини параметрів атака декодування перевершує атаку вкладення, проте на малих значеннях параметра q атака вкладення все ж стає кращою. Різниця є достатньо малою, проте при виборі параметрів, все ж, її варто враховувати. Тож, при оцінці безпеки не можна нехтувати атаками декодування.

Для проблеми NTRU атаки декодування можливо застосовувати так само, як і для проблеми LWE. Базис NTRU решітки можливо інтерпретувати як базис q–арної решітки і для довільної точки виконати атаку. Проте, як і для атак вкладення, необхідно враховувати точку переходу на розріджену підрешітку, за якої параметри NTRU стає небезпечно використовувати.

3.4. Атаки розпізнавання

Атаки розпізнавання є статистичними атаками, у яких супротивник намагається відрізнити пари $(A, t = A^T s + e)$ від пари (A, b) з рівномірного розподілу. Це можливо зробити, якщо відомо деякий малий вектор v, для якого виконується $Av = 0 mod q$ (тобто він лежить на дуальній решітці $\Lambda_q^{\perp}(A^T)$). Для вектора $t = A^T s + e$ скалярний добуток $\langle v, t \rangle mod q$ буде мати нормальний розподіл, оскільки $\langle v, t \rangle = vA^T s + \langle v, e \rangle mod q = \langle v, e \rangle$. Для вектору b відповідний розподіл буде рівномірним.

Атаки розпізнавання враховуються у багатьох моделях безпеки, наприклад у Crystals–Dilithium, чи New–Hope, аналізуються чисто як атаки, що дозволяють відрізнити розподіл LWE від рівномірного розподілу. Звичайно, факт відрізнення LWE розподілу від рівномірного руйнує усі докази безпеки у таких моделях безпеки, як IND–CCA для схем асиметричного шифрування, чи IND–CMA для електронних підписів, проте в реальному світі зловмисників цікавить саме відновлення таємного ключа, а не лише відрізнення розподілів. Перетворення атаки розрізнення на атаку відновлення ключів потребує деяких додаткових обчислювальних ресурсів, тому такі оцінки є дещо заниженими. Враховуючи, що навіть такі занижені оцінки є гіршими за атаки вкладення та декодування, то уточнені оцінки будуть ще гіршими. Тож, у моделі безпеки їх можливо не враховувати.

3.5. Гібридні атаки

Основна ідея гібридної атаки ґрунтується на тому, що якщо q– просте число, то для будь–якої n–вимірної q–арної решітки Λ базис можливо представити у вигляді

$$B = \begin{pmatrix} B_1 & B_2 \\ 0 & I_r \end{pmatrix} \in \mathbb{Z}^{n \times n} \tag{3.12}$$

Де $0 < r < n$, $B_1 \in \mathbb{Z}^{(n-r)\times(n-r)}$, $B_2 \in \mathbb{Z}^{(n-r)\times r}$.

Використовуючи структурованість базису, довільний вектор $v \in \Lambda$ можливо представити як конкатенацію векторів меншої розмірності:

$$v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = B \begin{pmatrix} x \\ v_2 \end{pmatrix} = \begin{pmatrix} B_1 x + B_2 v \\ v_2 \end{pmatrix} \tag{3.13}$$

$$v_1 \in \mathbb{Z}^{(n-r)}, v_2 \in \mathbb{Z}^r$$

для деякого $x \in \mathbb{Z}^{(n-r)}$.

З формули (3.12) маємо рівняння $B_2 v_2 = -B_1 x + v_1$. Оскільки v_1 є малим вектором, то вектор $B_2 v_2$ знаходиться близько до решітки $\Lambda(B_1)$ і за умови, що B_1 є достатньо редукованим базисом, може бути відновлений за допомогою алгоритма найближчої площини Бабаї [70]. Позначимо відновлений вектор як v_1.

Гібридна атака складається з трьох етапів:

- Редукувати базис B_1.
- Знайти вектор v_2 за допомогою комбінаторних технік.
- Знайти вектор v_1 за допомогою алгоритма Найближчної площини Бабаї.

Гібридна атака схожа на атаку декодування, проте оскільки розмірності, у яких виконується пошук, є меншими, то загальний час виконання буде значно меншим за умови, що комбінаторна частина атаки має не велику вартість. Оскільки реалізація комбінаторної частини атаки сильно залежить від конкретної криптографічної схеми, то конкретні оцінки комбінаторної частини атаки будуть сильно відрізнятися для різних схем. Фактично, гібридна атака є модифікацією атаки декодування, тому графік для простору параметрів буде схожим на рисунок 3.8, тільки зміщеним на деякий фактор, який визначається комбінаторною частиною.

Є цікавим той момент, що формула (3.9), яка, фактично, визначає ймовірність знаходження вектору v_1, отримана з припущення, що таємний вектор матиме нормальний розподіл, у той час, як гібридна атака застосовується переважно для векторів, що мають розподіл відмінний від нормального. Ця проблема у літературі зазвичай обходиться стороною.

Для вирішення проблеми при оцінці гібридної атаки для розподілів, що відмінні від нормального, можливо апроксимувати цей розподіл ймовірностей нормальним розподілом, що мінімізує відстань Колмогорова–Смірнова, яка визначена як максимальна абсолютна різниця між двома емпіричними функціями розподілу.

Для рівномірного розподілу у межах $[-\epsilon, +\epsilon]$ для $\epsilon = 1, \dots, 10$ було обчислено оптимальні значення дисперсії σ у таблиці 3.1.

Таблиця 3.1.

Оптимальні значення σ для мінімізації відстані Колмогорова–Смірнова

ϵ	Оптимальне σ	Відстань Колмогорова–Смірнова	ϵ	Оптимальне σ	Відстань Колмогорова–Смірнова
1	1.03	0.16666	6	4.14	0.07339
2	1.71	0.12074	7	4.73	0.06963
3	2.30	0.09644	8	5.34	0.06701
4	2.93	0.08589	9	5.94	0.064911
5	3.53	0.07807	10	6.54	0.063163

На рисунку 3.9 у якості ілюстрації наведено порівняння кумулятивних функцій розподілу для $\epsilon = 3$. З рисунку добре видно, що отримана апроксимація доволі близько знаходиться до відповідного дискретного розподілу, що робить можливим застосування формули (3.9) для обчислення ймовірностей гібридних атак та атак декодування.

Звичайно, такий підхід є доволі грубим, проте для довільних розподілів не відомо аналогів (3.9), тому для атак декодування та гібридних атак значно важче отримати точні оцінки.

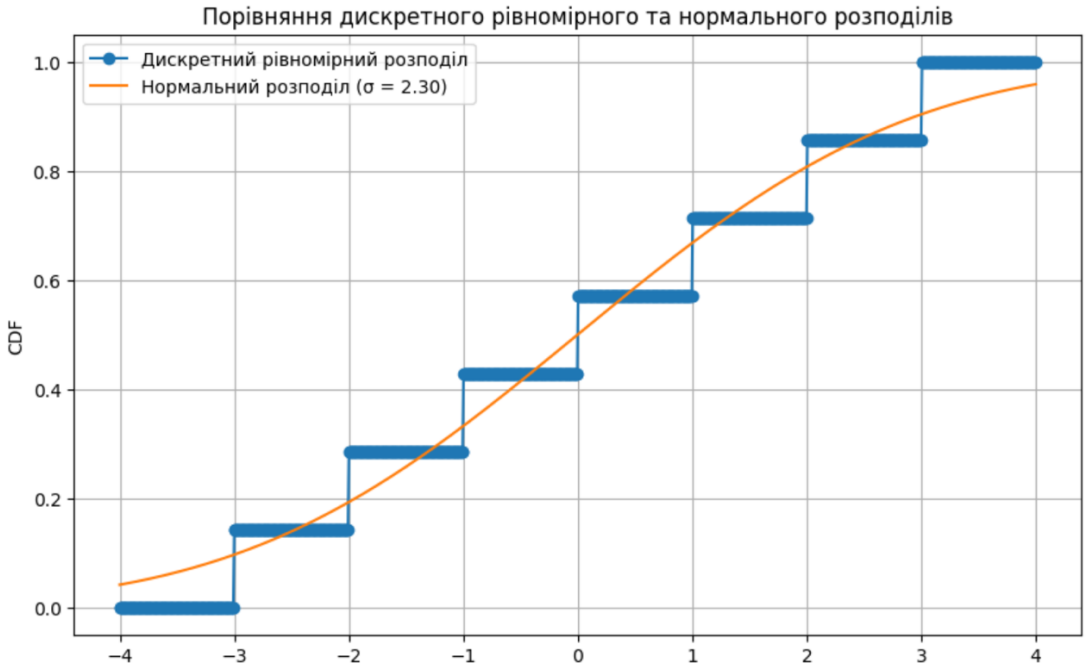

Рис. 3.9. Порівняння дискретного рівномірного розподілу та його апроксимації нормальним розподілом

Отримані значення, на нашу думку, дозволяють з достатньою точністю апроксимувати заданий розподіл нормальним розподілом і застосувати описаний вище підхід до атаки декодування та гібридної атаки.

3.6. Розробка методу для оцінки безпеки проблеми SIS

Якщо проблем SIS визначена для норми $\|\cdot\|_2$, то для її вирішення можливо використовувати звичайні атаки вкладення. Проте, аналіз проблеми SIS дещо відрізнятиметься від проблем LWE та NTRU, якщо необхідно знайти вектор, що є малим у $\|\cdot\|_\infty$ нормі. Відповідний вектор лежить на наступній решітці

$$\Lambda(A) = \{z \in \mathbb{Z}^d | Az = 0 \; mod \; q\} \tag{3.14}$$

Існуючі в літературі підходи до оцінки складності проблеми SIS через редукцію решіток вважають, що нам відомий розмір шуканого вектору у $\|\cdot\|_2$ нормі. Проте, з визначення проблеми для $\|\cdot\|_\infty$ норми не випливає, що нам необхідний вектор саме з конкретною $\|\cdot\|_2$ нормою. Тому було розроблено

власний підхід, який враховує те, що для шуканого вектора з $\|\cdot\|_\infty$ нормою можуть бути знайдені рішення з різними $\|\cdot\|_2$ нормами.

Для проблеми SIS існуючі методи оцінки зазвичай не дозволяютья застосувати симулятори редукції, оскільки ці методи напряму ґрунтуються на ZGSA. Тому в процесі дисертаційного дослідження для оцінки ймовірності події знаходження вектора v, що має l_∞ норму зі значенням B було запропоновано використати властивість концентрації мери на гіперсфері [71], яка полягає у тому, що для d–вимірної гіперсфери ймовірність того, що довільна компонента вектора v_i буде відрізнятися від середнього значення експоненціально зменшується з збільшенням відстані. Більш конкретно, для кожного v_i маємо

$$\Pr[|v_i| \geq B] \leq 2exp\left(-\frac{\left(\frac{B\sqrt{d}}{\|v\|_2}\right)^2}{2}\right) \tag{3.15}$$

Відповідно, для усього вектора маємо:

$$\Pr\left[\|v\|_\infty \leq B\right] \approx \left(1 - 2exp\left(-\frac{\left(\frac{B\sqrt{d}}{\|v\|_2}\right)^2}{2}\right)\right)^d \tag{3.16}$$

Запропонований підхід до вирішення SIS полягає у наступному:

- Провести редукцію базису SIS–решітки з параметром β_1
- За допомогою алгоритму просіювання у розмірності β_2 отримати N малих векторів з значенням норми α
- Для заданого значення α знайти ймовірність p_{succ} того, що l_∞ норма не перевищує значення B за формулою (3.16).
- Оцінити складність атаки як $(T_{red} + T_{sieve})/(\min(1, N \cdot p_{succ}))$

Алгоритм просіювання може повернути н $2^{0.2075\beta}$ векторів з нормою $\alpha = \rho \cdot \|b_0\|_2$, де ρ оцінюється як

$$\sqrt{4/3} \cdot \delta_{\beta_1}^{\beta_1 - 1} \delta_{\beta_2}^{1-\beta_2} \tag{3.17}$$

Моделювання атаки показало, що застосування симуляторів майже не впливає на оцінки безпеки для атаки на SIS. На рисунку 16 показані результати моделювання.

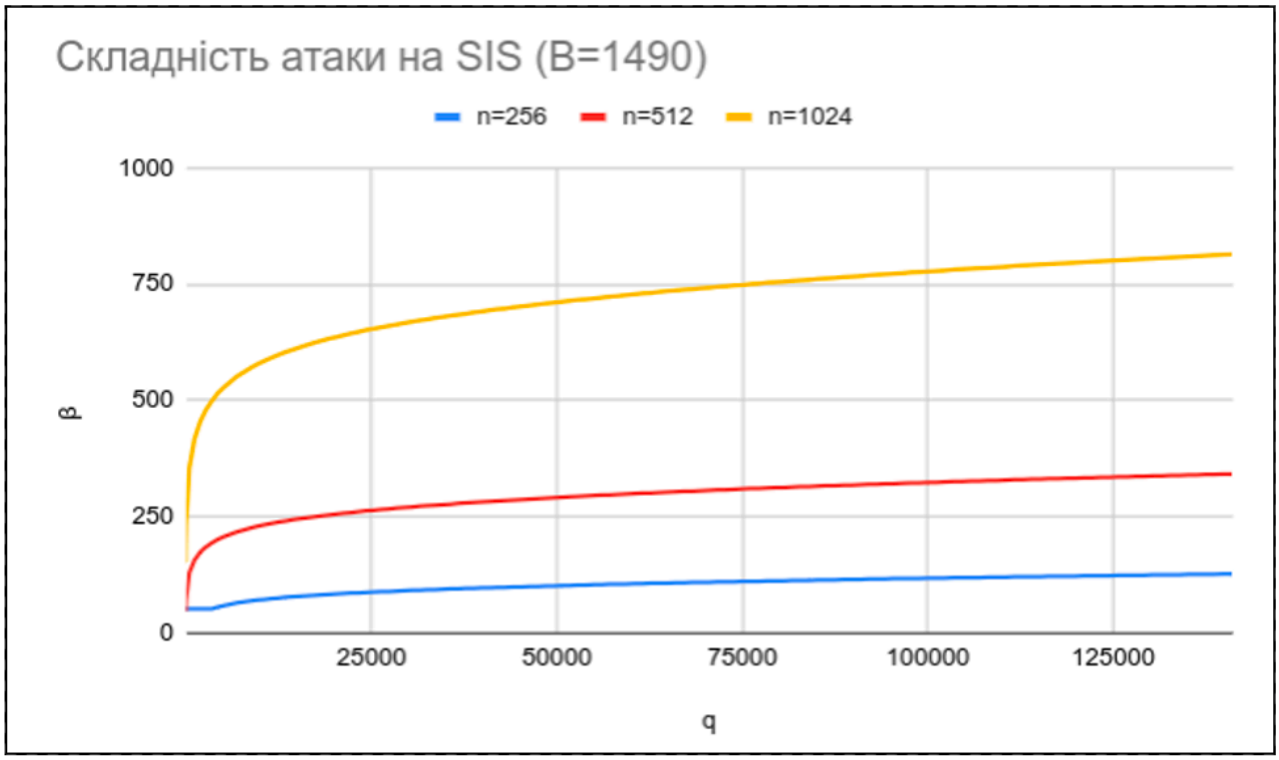

Рис. 3.10. Результати оцінки атаки на SIS

3.7. Висновки до розділу

1. При врахуванні алгебраїчної структури q–арних решіток в атаках вкладення було виявлено, що модель GSA занижує значення безпеки. Цей ефект пояснюється тим, що GSA не враховує того, що останній блок в базисі є HKZ–редукованим і має іншу форму. На малих значеннях параметра q уточнені оцінки показують менші показники безпеки, проте зі збільшенням праметра ситуація повністю змінюється. Оцінки безпеки стають більшими, ніж для GSA. Більшість існуючих криптографічних параметрів потрапляють у другу зону. Це вказує на те, що існуючи схеми переважно є безпечнішими, ніж вважалося раніше. Для NTRU решіток також необхідно враховувати можливість переходу на розріджену решітку.

2. Для атак декодування було запропоновано стратегію вибору параметрів атаки d_i. З використанням цих параметрів було показано, що атаки

декодування можуть бути кращими за атаки вкладення. Вплив алгебраїчної структури q–арних решіток на атаки відновлення є не таким сильним, як при атаках вкладення. Це пояснюється тим, що на атаки декодування впливає вся форма профіля, а не лише конкретне значення в останньому блоці, як у атаках вкладення.

3. Гібридні атаки є, фактично, узагальненням атак декодування, хоча вони історично з'явилися раніше. Оцінки гібридних атак та атак декодування ґрунтуються на тому, що розподіл таємного вектора є нормальним. Проте, це не так для більшості параметрів, для яких гібридні атаки можливо застосувати. Щоб подолати цю ситуацію було запропоновано апроксимувати відповідні розподіли нормальним розподілом, мінімізуючи відстань Колмогорова–Смірнова та обчисленні конкретні оптимальні параметри апроксимуючих нормальних розподілів.

4. Існуючі в літературі підходи до оцінки складності проблеми SIS через редукцію решіток вважають, що нам відомий розмір шуканого вектору у l_2 нормі. Проте, з визначення проблеми не випливає, що нам необхідний вектор саме з конкретною l_2 нормою. У межах дослідження було розроблено метод, що враховує факт того, що при фіксованих вимогах до l_∞ норми, l_2 норма може мати різні значення.

РОЗДІЛ 4. ОЦІНКА ЗАХИЩЕНОСТІ МЕХАНІЗМІВ ІНКАПСУЛЯЦІЇ КЛЮЧІВ НА АЛГЕБРАЇЧНИХ РЕШІТКАХ

У сучасній криптографії однією з обов'язкових вимог до будь–якої криптографічної системи є наявність доказової безпеки. Тобто, має існувати математичний доказ безпеки, який гарантує відсутність атак у межах обраної формальної моделі, за умови виконання невеликої кількості модельних припущень. Типовими припущеннями є складність таких теоретико–числових проблем, таких як LWE та NTRU. Проте, при оцінці реальних систем її треба застосовувати разом з деякою формальною моделлю безпеки. Модель квантового випадкового оракула часто застосовується на практиці. Цей розділ присвячений аналізу механізмів інкапсуляції ключів у моделі квантового випадкового оракула з використанням розробленої у третьому розділі методики оцінки складності проблем LWE та NTRU.

У цьому розділі використовується наступна нотація. Для позначення предикатів використовується позначення $[\![\cdot]\!]$. Якщо b є деяким твердженням, то предикат $[\![b]\!]$ приймає значення 1, якщо b є істинним, та 0 інакше. Якщо змінна x приймає значення детермінованим чином, то використовується знак «=». Якщо змінна x приймає значення з деякого випадкового процесу, то використовується символ «←». Для визначеної множини X позначення $x \leftarrow X$ означає, що змінна x приймає випадкове значення з рівномірного розподілу над X. Символом «==» позначатимемо перевірку на рівність аргументів. Ймовірність деякої події W надалі позначатимемо символом $\Pr[W]$, математичне очікування для деякого розподілу S надалі позначатимемо як $\mathbb{E}[S]$. Для заданої множини X вираз |X| означає потужність множини. Для числа x вираз |x| означає абсолютне значення.

Тут і надалі запис $Adv(\lambda) = negl(\lambda)$ означає, що значення функції Adv при збільшенні параметра λ зменшується швидше за будь–який поліном. Більш формально: для будь якого полінома $p(\lambda)$ виконується $\lim_{\lambda\to\infty} Adv(\lambda)p(\lambda) = 0$.

4.1. Модель безпеки IND–CCA

Модель безпеки IND–CCA ґрунтується на ідеї нерозрізнювальності: якщо супротивник не може розрізнити шифротекст повідомлення m_0 від шифротекста повідомлення m_1, то він не може отримати жодної інформації про зашифровані повідомлення.

Для побудови доказу безпеки шифру E для супротивника A вводяться дві гри (експерименти): $Exp_{A,E}^{IND-CCA-0}(\lambda)$ та $Exp_{A,E}^{IND-CCA-1}(\lambda)$ для параметра безпеки λ [34]. У кожній грі іспитувач генерує випадкову ключову пару $(pk, sk) \leftarrow Gen(1^{\lambda})$ та передає відкритий ключ супротивнику A. Супротивник A обирає два повідомлення m_0, m_1 однакової довжини та надсилає їх іспитувачу. Іспитувач генерує випадковий біт $b \in \{0,1\}$, чим обирає гру. Якщо $b = 0$, то іспитувач зашифровує повідомлення m_0 та надсилає шифротекст $c^* = Enc(sk, m_0)$ у якості завдання (гра $Exp_{A,E}^{IND-CCA-0}(\lambda)$). Якщо біт $b = 1$, то, іспитувач зашифровує повідомлення m_1 та надсилає шифротекст $c^* = Enc(sk, m_1)$ у якості завдання (гра $Exp_{A,E}^{IND-CCA-1}(\lambda)$). Супротивник має визначити у яку гру він грає (яке повідомлення було зашифровано) та повернути біт b_A. Результатом ігор $Exp_{A,E}^{IND-CCA-0}(\lambda)$ та $Exp_{A,E}^{IND-CCA-1}(\lambda)$ є значення предиката $[\![b == b_A]\!]$. Супротивник A може робити запити до оракула дешифрування O_{Dec}, який може розшифрувати будь–який шифротекст окрім шифротексту завдання. Розрізняють IND–CCA1 безпеку, де супротивник може робити запити тільки до моменту отримання шифротекста та IND–CCA2 безпеку, де запити можливо робити і після отримання завдання. У межах цього дослідження IND–CCA2 буде вважатися синонімом IND–CCA. Псевдокод $Exp_{A,E}^{IND-CCA}(\lambda)$ наведений нижче.

Гра $Exp_{A,E}^{IND-CCA}(\lambda)$:

1. $(pk, sk) \leftarrow Gen(1^{\lambda})$
2. $pk \rightarrow A^{O_{Dec}}$
3. $m_0, m_1 \leftarrow A^{O_{Dec}}$
4. $b \in \{0,1\}$
5. $c^* = Enc(sk, m_b) \rightarrow A^{O_{Dec}}$
6. $b_A \leftarrow A^{O_{Dec}}$
7. $return\ [\![b == b_A]\!]$

Перевага супротивника у розрізненні ігор визначає безпеку в моделі IND–CCA. Якщо перевага є незначною у теоретико–числовому сенсі, то схема асиметричного шифрування вважається безпечною в моделі IND–CCA:

$$Adv_{A,E}^{IND-CCA}(\lambda) = \Pr\left[Exp_{A,E}^{IND-CCA-0}(\lambda) - Exp_{A,E}^{IND-CCA-1}(\lambda)\right] = negl(\lambda) \quad (4.1)$$

У квантовому випадку для забезпечення унітарності системи оракул дешифрування вводиться наступним чином:

$$U_{Dec}|c,y\rangle \rightarrow \begin{cases} |c, y \oplus \perp\rangle, c == c^* \\ |c, y \oplus Dec(sk,c)\rangle, c \neq c^* \end{cases} \quad (4.2)$$

Тобто, використовується допоміжний регістр $|y\rangle$, до якого записується результат.

Від схем асиметричного шифрування при побудові механізмів інкапсуляції ключів вимагається безпека у моделі IND–CPA (Indistinguishability under Chosen–Plaintext Attacks), або у моделі OW–CPA (One–Wayness under Chosen–Plaintext Attacks). Відповідні експерименти зображені нижче

Гра $Exp_{A,E}^{OW-CPA}(\lambda)$:	Гра $Exp_{A,E}^{IND-CPA}(\lambda)$:
1. $(pk, sk) \leftarrow KeyGen(1^\lambda)$ 2. $m^* \leftarrow \{0,1\}^\lambda$ 3. $c^* \leftarrow Enc(pk, m^*)$ 4. $m' \leftarrow A(pk, c^*)$ 5. $return\ [\![m' == m^*]\!]$	1. $(pk, sk) \leftarrow KeyGen(1^\lambda)$ 2. $b \leftarrow \{0,1\}$ 3. $(m_0^*, m_1^*) \leftarrow A_1(pk)$ 4. $c^* \leftarrow Enc(pk, m_b^*)$ 5. $b' \leftarrow A_2(pk, c^*)$ 6. $return\ [\![b' == b]\!]$

Перевагу супротивника A у іграх IND–CPA та OW–CPA для схеми асиметричного шифрування PKE позначимо як $Adv_{PKE}^{OW-CPA}(A)$ та $Adv_{PKE}^{IND-CPA}(A)$ відповідно. Стандартним визначенням для переваги супротивника є:

$$Adv_{PKE}^{OW-CPA}(A) = \Pr[OW - CPA(A) == 1]$$

$$Adv_{PKE}^{IND-CPA}(A) = |\Pr[IND - CPA(A) == 1] - 1/2| \quad (4.3)$$

Схема асиметричного шифрування у загальному випадку може мати помилки дешифрування, тобто для деяких правильно обчислених шифротекстів розшифрування може давати не правильний результат. Існують різні підходи до врахування помилок дешифрування. У межах цього дослідження ми будемо слідувати роботі [72]. Для оцінки ймовірності виникнення помилок дешифрування введемо наступну величину:

$$\delta_{wc} = \mathbb{E}_{(pk,sk)}\left[\max_{m\in M} \Pr[Dec(sk,c) \neq m] \,|(pk,sk) \leftarrow Keygen()\right] \quad (4.4)$$

Величина δ_{wc} характеризує ймовірність появи помилок дешифрування у найгіршому випадку.

Для того, щоб схема асиметричного шифрування була безпечною необхідно, щоб рівень помилок був незначним. Для оцінки складності отримання помилки дешифрування введемо гру COR–RO [73]. У цій грі супротивник має доступ до деякого випадкового оракула G. Задача супротивника полягає у тому, щоб повернути список повідомлень. Якщо хоча б одне повідомлення викликатиме помилку дешифрування, то супротивник перемагає. Формальне визначення у вигляді псевдокоду наведено нижче.

Гра $Exp_{A,E}^{COR-RO}(\lambda)$:

1. $(pk, sk) \leftarrow PKE.KeyGen(1^{\lambda})$
2. $L_M \leftarrow A^G(sk, pk)$
3. For $m \in L_m$
4. $\quad c \leftarrow Enc(pk, m)$
5. $\quad if\ Dec(sk, c) \neq m$
6. $\quad\quad return\ 1$
7. Return 0

Перевага супротивника A відповідно визначається як

$$Adv_{PKE}^{COR-RO}(A) = \Pr\,[COR - RO(A) = 1] \quad (4.5)$$

У роботі [73] був отриманий важливий результат щодо оцінки ймовірності появи помилок дешифрування.

Лема 1 ([73]). Якщо PKE є δ_{wc} –коректною схемою асиметричного шифрування, тоді для будь–якого супротивника A, що робить q_G квантових запитів до оракула G та повертає одне повідомлення, має місце нерівність

$$Adv_{PKE}^{COR-RO}(A) \leq 8 \cdot (q_G + 1)^2 \cdot \delta_{wc} \quad (4.6)$$

Важливою лемою при доказі тверджень у моделях на основі нерозрізнювальності є так звана лема Union Bound.

Лема 2 (Union Bound, [39]). Нехай A,B та E – події у деякому просторі ймовірностей. Якщо $\Pr[A|\neg E] = \Pr[B|\neg E]$, то має місце нерівність $|\Pr[A] - \Pr[B]| \leq \Pr[E]$.

У класичній моделі випадкового оракула типовою стратегією доказу є показати, що супротивник A не може відрізнити значення випадкового оракула від випадкового, якщо A не робив раніше відповідного запиту до оракула. Проте, у квантовому випадку цю стратегію важко реалізувати, оскільки A може робити запити в суперпозиції і з деякою незначною ймовірністю отримати відповідне значення. Щоб оцінити ймовірність успіху A необхідно оцінити наскільки важко витягти цю інформацію з запиту. Одним з перших рішень для цієї проблеми була OW2H Лема. Нижче наведений варіант цієї леми, який є зручним для доказу безпеки ДСТУ 8961:2019.

Лема 3 (OW2H Лема [74]). Нехай $H: \{0,1\}^n \to \{0,1\}^m$ є випадковим оракулом і задано деякий алгоритм A, що робить не більше q запитів до H. Нехай B є оракулом, що приймає на вхід деяку змінну x та робить наступне:

- Обирає випадкове значення $i \leftarrow \{1, \dots, q\}$
- Обирає випадкове значення $y \leftarrow \{0,1\}^m$
- Запускає $A^H(x, y)$ допоки не буде здійснено i–й запит до H
- Вимірює значення аргументу у i–му запиті до H
- Повертає виміряне значення аргументу (Якщо A робить менше i запитів, то B повертає $\perp \notin \{0,1\}^n$).

Тоді виконується нерівність

$$|P_A^1 - P_A^2| \leq 2q\sqrt{P_B}$$

де

$$P_A^1 = \Pr\left[b' == 1 : x \leftarrow \{0,1\}^n, b' \leftarrow A^H\big(x, H(x)\big)\right]$$

$$P_A^2 = \Pr\left[b' == 1 : x \leftarrow \{0,1\}^n, y \leftarrow \{0,1\}^m, b' \leftarrow A^H(x, y)\right]$$

$$P_B = \Pr\left[x == x' : x \leftarrow \{0,1\}^n, x' \leftarrow B^H(x)\right]$$

Випадковий оракул є функцією $H: X \to Y$ (де $X = \{0,1\}^m$ та $Y = \{0,1\}^n$ для деяких m, n), яка обрана з рівномірного розподілу над множиною усіх можливих функцій Ω_H. Квантовий випадковий оракул задається наступним оператором:

$$H^{St}: |x, y\rangle \to |x, y \oplus H(x)\rangle \tag{4.9}$$

4.2. Оцінка безпеки ДСТУ 8961:2019 у моделі квантового оракула

ДСТУ 8961:2019 [57] використовує перетворення у полі $R_q = \mathbb{Z}_q[X]/(X^n - X - 1)$ і грунтується на проблемі NTRU. Позначимо як R_3 множину усіх поліномів поля R_q, усі коефіцієнти яких належать до множини $\{-1,0,1\}$, як $R_3^{a,b}$ множину усіх поліномів у R_3, що мають кількість ненульових елементів у діапазоні $[a, b]$. Якщо $a == b$, то використовується скорочене позначення $R_3^{t,t} = R_3^t$.

Текст стандарту ДСТУ 8961:2019 не містить зручного математичного опису криптографічних перетворень. Визначення в стандарті містять багато технічних подробиць. Тому для проведення аналізу введемо наступні геш функції:

$$BPGM: \{0,1\}^L \times R_q \to R_q$$

$$MGF: R_q \to R_3$$

$$H: R_q \to \{0,1\}^\lambda$$

$$KDF: R_q \to \{0,1\}^{K_{len}} \tag{4.11}$$

Де λ – параметр безпеки, t –загальносистемний параметр, від якого залежить кількість ненульових елементів в поліномах, L – повна довжина повідомлення, K_{len} – довжина ключа інкапсуляції. Додатково використовується бієктивне відображення

$$Pad: \{0,1\}^L \times \{0,1\}^{db} \times R_q \to \{R_3, \perp\}$$
$$Pad^{-1}: R_3 \to \{0,1\}^L \quad (4.12)$$

Де L є максимальна довжина повідомлення в бітах, db – довжина випадкового рядка бітів. Символ $\perp$ повертається якщо не вдалося декодувати повідомлення. Зауважимо, що в тексті стандарту не має функції Pad, як і введених вище геш функцій. Ці функції були введені для простоти опису перетворень у межах данної роботи.

Протокол інкапсуляції ключів ДСТУ 8961:2019 використовує конструкцію механізму інкапсуляції ключів власної розробки [75]. Нижче наведено псевдокод асиметричної схеми шифрування, що лежить в основі стандарту. Для простоти представлення технічні деталі щодо перетворення поліномів у бітові строки, механізми стискання підпису та оптимізації (що не впливають на оцінки безпеки до загальних атак) не показані.

<table>
<tr>
<td>$SkelyaPKE$.Gen(1^λ):
1. $G \leftarrow R_3^{\left\lfloor \frac{2n}{3}+1 \right\rfloor}$
2. $F \leftarrow R_3^{2t}$
3. $f = (1 + p \cdot F) mod\ q$
4. Якщо $\nexists f^{-1}$, goto 2
5. $h = G \cdot f^{-1} \in R_q$
6. Повернути $pk = h, sk = (f, h)$</td>
<td>$SkelyaPKE$.Enc($msg, coins, pk = h$):
1. $m = Pad(msg, coins)$
2. $r = BPGM(msg, coins, h)$
3. $R = r \cdot h \in R_q$
4. $m' = m + MGF(R)$
5. Якщо $m' \notin R_3^{2t,n-2t}$, повернути $\perp$
6. $c = R + m'$
7. Повернути c</td>
</tr>
<tr>
<td colspan="2">$SkelyaPKE$.Dec$\big(c, sk = (f, h)\big)$:
1. $a = f \cdot e \in R_q$
2. $m' = a\ mod\ p$
3. Якщо $m' \notin R_3^{2t,n-2t}$, повернути $\perp$
4. $R = c - m'$</td>
</tr>
</table>

5. $m = m' - MGF(R)$
6. $(msg, coins) = Pad^{-1}(m)$
7. Якщо msg $=\perp$, повернути $\perp$
8. $r' = BPGM(msg, coins, h)$
9. $R' = r' \cdot h$
10. Якщо $R' = R$, повернути msg
11. Повернути $\perp$

Нижче наведено протокол інкапсуляції ключів ДСТУ 8961:2019.

Encaps($pk = h$):

1. // msg є константою
2. $seed \leftarrow \{0,1\}^{seed\_len}$
3. $r = BPGM(msg, seed, h)$
4. $C_1 = Enc(msg, seed, pk)$
5. $C_2 = H(r)$
6. $K = KDF(r)$
7. $C = (C_1, C_2)$
8. Повернути (C, K)

Decaps($C = (C_1, C_2), sk = (f, h)$):

1. $msg, seed = DEc(C_1, sk)$
2. Якщо msg $=\perp$, повернути $\perp$
3. $r = BPGM(msg, seed, h)$
4. $C_2' = H(r)$
5. $K = \begin{cases} KDF(r), C_2' = C_2 \\ \perp, \text{інакше} \end{cases}$
6. Повернути K

KeyGen(1^{λ}):

1. $(pk, sk) \leftarrow PKE.KeyGen()$
2. Повернути (sk, pk)

Зауважимо, що наведений псевдокод має деякі особливості, про які варто згадати. По–перше, алгоритм шифрування $SkelyaPKE.Enc$ може повертати помилку шифруваяння $\perp$, що дещо відрізняється від звичайної нотації асиметричного шифрування. По–друге, зазвичай при декомпозиції механізму інкапсуляції ключів схема АСШ є CPA безпечною, проте у випадку ДСТУ

8961:2019 декомпозиція дає CCA безпечну схему, що вказує на те, що перетворення можуть бути спрощені без втрати безпеки.

У таблиці 4.1 перелічені загальносистемні параметри ДСТУ 8961:2019. Параметри N, q, p визначають поле (і ідеал у цьому полі) у якому будуть виконуватися перетворення, параметри t, d_g, d_f задають кількість ненульових коефіцієнтів у поліномах.

Таблиця 4.1.

Основні загальносистемні параметри ДСТУ 8961:2019

Параметр	Значення
N	Параметр поля. Визначає степінь поліномів.
q	Параметр поля. Визначає максимальні значення коефіцієнтів поліномів.
p	«Малий модуль». Визначає структуру таємного ключа. Для всіх наборів параметрів має фіксоване значення – 3.
t	Визначає кількість коефіцієнтів в таємному поліномі
d_g	$d_g = \left\lfloor \frac{2N}{3} + 1 \right\rfloor$
d_f	$d_f = 2t$

ДСТУ 8961:2019 підтримує три набори загальносистемних параметрів для 256, 384, 512 біт безпеки. Набори загальносистемних параметрів зведені в таблиці 4.2. Навідміну від інших стандартів, ДСТУ 8961:2019 орієнтований на підвищені рівні безпеки.

Таблиця 4.2.

Загальносистемні параметри ДСТУ 8961:2019

Набір параметрів	N	q	p	t	d_g	d_f
Skelya256	881	7673	3	159	588	318
Skelya384	1201	9227	3	192	801	384
Skelya512	1471	12269	3	255	981	510

4.2.1. Аналіз в моделі квантового випадкового оракула

Стандарт ДСТУ 8961:2019 [76] визначає асиметричне перетворення та механізм інкапсуляції ключів на основі NTRU [77]. Асиметричне перетворення є реалізацією схеми NAEP, для якого вже існують докази безпеки. Для отримання механізму інкапсуляції ключів використовується перетворення власної розробки, формального аналізу якого в літературі існує доволі мало. У межах цієї роботи ми будемо називати це перетворення як SkelyaTransform.

Метою даного розділу є аналіз перетворення SkelyaTransform у моделі квантового випадкового оракула. Аналіз ґрунтується на роботах [72,73], у яких проводився аналіз доволі схожого на SkelyaTransform перетворення.

Алгоритм Enc є ймовірнісним, тобто він має деяку внутрішню випадковість r. У межах аналізу зручно виносити цю випадковість у аргумент функції і вважати Enc детермінованим алгоритмом, що має сигнатуру $Enc\colon (pk, m, r) \to C$. Цей прийом має назву дерандомізація і широко використовується у формальних доказах [32, 78].

Схема асиметричного шифрування має властивість відновлення випадковості, якщо існує алгоритм RandomRecovery, що приймає у якості аргументів відкритий ключ pk, повідомлення m, відповідний шифротекст c=Enc(pk,m) та повертає значення r, що використовувалося під час шифрування.

Схема асиметричного шифрування має властивість відновлення повідомлення, якщо існує алгоритм MessageRecovery, що приймає у якості

аргументів відкритий ключ pk, випадкове значення r та шифротекст c, що використовує r під час шифрування. Алгоритм MessageRecovery повертає повідомлення m, що зашифроване у шифротексті c, або символ помилки розшифрування, якщо шифротекст є не коректним.

Якщо схема асиметричного шифрування має властивість відновлення випадковості та властивість відновлення повідомлення, то надалі казатимемо, що схема асиметричного шифрування має властивість однозначного відновлення. Зі структури процедури шифрування видно, що ДСТУ 8961:2019 має таку властивість і це використовується під час доказу безпеки [79].

У межах цієї роботи досліджується перетворення SkelyaTransform, яке визначено (у неявному вигляді) стандартом ДСТУ 8961. Формалізуємо це перетворення для довільної схеми асиметричного шифрування.

Нехай λ – параметр безпеки, PKE=(Gen,Enc,Dec) – деяка схема асиметричного шифрування, що використовує простір повідомлень MSpace, простір шифротекстів CSpace, простір випадковості RSpace і задано геш функції:

$$H: RSpace \to \{0,1\}^{\lambda}$$
$$BPGM: MSpace \to RSpace$$
$$KDF: RSpace \to \{0,1\}^{\lambda} \quad (4.13)$$

Перетворення SkelyaTransform задано наступним чином:

Encaps(pk): 1. $m \leftarrow MSpace$ 2. $r = BPGM(m)$ 3. $C_1 = PKE.Enc(m, r, pk)$ 4. $C_2 = H(r)$ 5. $K = KDF(r)$ 6. $C = (C_1, C_2)$ 7. $return\ (C, K)$	Decaps$(C = (C_1, C_2), sk)$: 1. $m' = PKE.Dec(C_1, sk)$ 2. If $m' == \perp\ return\ \perp$ 3. $r' = BPGM(m)$ 4. $C_2' = H(r')$ 5. $C_1' = PKE.Enc(m, r, pk)$ 6. If $C_1' == C_1\ and\ C_2' == C_2$ return $K = KDF(r)$ 7. retuturn $\perp$
KeyGen(1^{λ}):	

1. $(sk, pk) \leftarrow PKE.Gen(1^{\lambda})$
2. $return\ (sk, pk)$

В процесі дисертаційного дослідження було отримано доказ безпеки перетворення SkelyaTransform[PKE] та сформульовано в теоремі 1.

Теорема 1 [78, 88]. Нехай PKE є OW–CPA безпечною та δ_{wc} –коректною схемою асиметричного шифрування з властивістю однозначного відновлення, тоді SkelyaTransfom[PKE] є IND–CCA безпечним механізмом інкапсуляції ключів. Більш формально – для кожного квантового алгоритму A у грі IND–CCA проти KEM=SkelyaTransform[PKE], що робить $q_H, q_{BPGM}, q_{KDF}, q_D$ запитів до оракулів H,BPGM,KDF та оракула дешифрування, існує квантовий алгоритм B у грі OW–CPA проти схеми асиметричного шифрування PKE, для якого виконується нерівність

$$Adv_{KEM}^{IND-CCA}(A) \leq (2 \cdot q_H + 2 \cdot q_D + q_{KDF}) \cdot$$
$$\cdot \sqrt{Adv_{PKE}^{OW-CPA}(B)} + 8 \cdot (q_{BPGM} + q_D + 1)^2 \cdot \delta_{wc} \qquad (4.14)$$

Доказ.

Перед тим, як перейти безпосередньо до доказу, розглянемо загальну структуру доказу. Для доказу використовується стандартна техніка “game hopping”. Для того, щоб довести нерівність (4.14) розглядається серія ігор GAME0 – GAME6. Гра GAME0 відтворює гру IND–CCA. Кожна наступна гра спрощується у тому сенсі, що значення змінних замінюються на дійсно випадкові або змінна взагалі виводиться з використання. При цьому фіксується зміна переваги супротивника. Цей процес відбувається до тих пір, допоки не буде простого способу оцінити ймовірність перемоги супротивника у поточній грі.

Гра GAME0 зображена у псевдокоді нижче

Гра $GAME0_{A,PKE}(\lambda)$:	Оракул $O_{Dec}((c_0, c_1) \neq (c_0^*, c_1^*))$:
1. $(pk, sk) \leftarrow PKE.KeyGen(1^{\lambda})$	1. $m' = PKE.Dec(c_0, sk)$
2. $b \leftarrow \{0,1\}$	2. $if\ m' == \perp$

3. $m^* \leftarrow \{0,1\}^n$ 4. $r^* = BPGM(m^*)$ 5. $c_0^* \leftarrow PKE.Enc(pk, m^*, r^*)$ 6. $K_0^* = KDF(r^*)$ 7. $K_1^* \leftarrow \{0,1\}^n$ 8. $K^* = K_b^*$ 9. $c_1^* = H(r^*)$ 10. $c^* = (c_0^*, c_1^*)$ 11. $b' = A^{O_{Dec}}(pk, c^*, K^*)$ 12. $return \llbracket b' == b \rrbracket$	3. $return \perp$ 4. $r' = BPGM(m')$ 5. $c_1' = H(r')$ 6. $if(\llbracket c_0 == PKE.Enc(m', r', pk) \rrbracket)$ 7. $and \llbracket c_1 == c_1' \rrbracket)$ 8. $return\, K = KDF(r')$ 9. $return \perp$

Ця гра в точності повторює IND–CCA гру для SkelyaTransfom[PKE], тому перевагю супротивника є:

$$Adv_{KEM}^{IND-CCA}(A) = |\Pr[GAME0(A) == 1] - 1/2|$$

У грі GAME1 замість використання оракула BPGM генеруватимемо випадкове значення r^*. В оракулі декапсуляції відповідно замість BPGM використовуватимемо функцію RandomRecovery:

Гра $GAME1_{A,PKE}(\lambda)$:	Оракул $O_{Dec}((c_0, c_1) \neq (c_0^*, c_1^*))$:
1. $(pk, sk) \leftarrow PKE.KeyGen(1^\lambda)$ 2. $b \leftarrow \{0,1\}$ 3. $m^* \leftarrow \{0,1)^n$ 4. $r^* = \{0,1\}^n$ 5. $c_0^* \leftarrow PKE.Enc(pk, m^*, r^*)$ 6. $K_0^* = KDF(r^*)$ 7. $K_1^* \leftarrow \{0,1\}^n$ 8. $K^* = K_b^*$ 9. $c_1^* = H(r^*)$ 10. $c^* = (c_0^*, c_1^*)$	1. $m' = PKE.Dec(c_0, sk)$ 2. $if\ m' == \perp$ 3. $return \perp$ 4. $r' = RandomRecovery(m', c_0)$ 5. $c_1' = H(r')$ 6. $if\ (\llbracket c_0 == PKE.Enc(m', r, pk) \rrbracket$ 7. $and \llbracket c_1 == c_1' \rrbracket)$ 8. $return\, K = KDF(r')$ 9. $return \perp$

11. $b' = A^{O_{Dec}}(pk, c^*, K^*)$ 12. $return\ [\![b' == b]\!]$	

Розглянемо наскільки зміниться перевага супротивника при переході від гри GAME0 до GAME1. Позначимо як DIFF подію, яка полягає у тому, що супротивник зможе відрізнити ігри GAME0 та GAME1. З Леми 2 маємо:

$$|\Pr[GAME0(A) == 1] - \Pr[GAME1(A) == 1]| \leq \Pr[DIFF]$$

$$\left|\Pr[GAME0(A) == 1] - \frac{1}{2} - \Pr[GAME1(A) == 1] + \frac{1}{2}\right| \leq \Pr[DIFF]$$

$$\left|\Pr[GAME0(A) == 1] - \frac{1}{2}| - |\Pr[GAME1(A) == 1] - \frac{1}{2}\right| \leq \Pr[DIFF]$$

$$Adv_{KEM}^{IND-CCA}(A) \leq Adv_{KEM}^{GAME1}(A) + \Pr[DIFF]$$

Різниця між іграми буде помітна якщо супротивник зможе знайти повідомлення, яке викликає помилку дешифрування. Тобто, якщо супротивник A сформує запит $(c_0 = PKE.Enc(pk, m, BPGM(m)), c_1)$, для якого $PKE.Dec(sk, c_0) \neq m$. Тоді можливо побудувати супротивника D у грі COR–RO, що ідеально симулює середовище для супротивника A. Супротивник D симулює гру IND–CCA та усі оракули для A, використовуючи алгоритм гри GAME1, та записує усі запити A до оракулів BPGM та оракула дешифрування. Для супротивника A симуляція буде ідеальною допоки не станеться подія DIFF. Застосовуючи Лему 1 отримуємо, що ймовірність події обмежена $8 \cdot (q_{BPGM} + q_D + 1)^2 \cdot \delta_{wc}$. Отже, маємо:

$$Adv_{KEM}^{IND-CCA}(A) \leq Adv_{KEM}^{GAME1}(A) + 8 \cdot (q_{BPGM} + q_D + 1)^2 \cdot \delta_{wc} \quad (2.44)$$

У грі GAME2 замінимо K^* та c_1^* на дійсно випадкові значення:

Гра $GAME2_{A,PKE}(\lambda)$: 1. $(pk, sk) \leftarrow PKE.KeyGen(1^\lambda)$	Оракул $O_{Dec}((c_0, c_1) \neq (c_0^*, c_1^*))$: 1. $m' = PKE.Dec(c_0, sk)$

2. $b \leftarrow \{0,1\}$ 3. $m^* \leftarrow \{0,1)^n$ 4. $r^* = \{0,1\}^n$ 5. $c_0^* \leftarrow PKE.Enc(pk, m^*, r^*)$ 6. $K_0^* = \{0,1\}^n$ 7. $K_1^* \leftarrow \{0,1\}^n$ 8. $K^* = K_b^*$ 9. $c_1^* = \{0,1\}^n$ 10. $c^* = (c_0^*, c_1^*)$ 11. $b' = A^{O_{Dec}}(pk, c^*, K^*)$ 12. $return \, [\![b' == b]\!]$	2. $if \; m' == \perp$ 3. $return \perp$ 4. $r' = RandomRecovery(m', c_0)$ 5. $if \; [\![c_0 == PKE.Enc(m', r, pk)]\!]$ 6. $return \, K = KDF(r')$ 7. $return \perp$

Застосовуючи визначення переваги супротивника, отримуємо вираз :

$$Adv_{KEM}^{IND-CCA}(A) \leq \frac{1}{2} \cdot \left| \begin{matrix} \Pr[GAME1(A) == 1 | b == 0] \\ - \Pr[GAME1(A) == 1 | b == 1] \end{matrix} \right| +$$

$$+8 \cdot (q_{BPGM} + q_D + 1)^2 \cdot \delta_{wc}$$

$$Adv_{KEM}^{IND-CCA}(A) \leq 8 \cdot (q_{BPGM} + q_D + 1)^2 \cdot \delta_{wc} +$$

$$+\frac{1}{2} \cdot | \Pr[GAME1(A) == 1 | b == 0] - \Pr[GAME2(A) == 1]$$

$$+ \Pr[GAME2(A) == 1] - \Pr[GAME1(A) == 1 | b == 0]|$$

Звідки витікає

$$Adv_{KEM}^{IND-CCA}(A) \leq +8 \cdot (q_{BPGM} + q_D + 1)^2 \cdot \delta_{wc} +$$

$$+\frac{1}{2} \cdot | \Pr[GAME1(A) == 1 | b == 0] - \Pr[GAME2(A) == 1]|$$

$$+ | \Pr[GAME2(A) == 1] - \Pr[GAME1(A) == 1 | b == 0]|$$

Для оцінки значень $| \Pr[GAME1(A) == 1 | b == 0] - \Pr[GAME2(A) == 1] |$ та $| \Pr[GAME2(A) == 1] - \Pr[GAME1(A) == 1 | b == 0] |$ можливо застосувати лему OW2H. Якщо покласти $O(\cdot) = H(\cdot)$, то гра OW2H буде

ідентичною до гри GAME1, за умови, що b=1 і до GAME2 якщо у є випадковим. Аналогічно, якщо покласти $O(\cdot) = H(\cdot) \times KDF(\cdot)$, то гра OW2H буде ідентичною до гри GAME1, за умови, що b=0 і до GAME2 якщо у є випадковим. З Леми 3 маємо нерівності:

$$|\Pr[GAME1(A) == 1 | b == 0] - \Pr[GAME2(A) == 1]| \leq 2 \cdot (q_{KDF} + q_H) \cdot \sqrt{\Pr[GAME3(A) == 1}$$

$$|\Pr[GAME1(A) == 1 | b == 1] - \Pr[GAME2(A) == 1]| \leq 2 \cdot (q_{KDF} + q_H + q_{KDF}) \cdot \sqrt{\Pr[GAME4(A) == 1]}$$

Де GAME3, GAME4 зображені у псевдокоді нижче, де позначення E^A позначає запуск алгоритму А до тих пір, доки не буде обрано випадково чергу з запитів до відповідних геш функцій, над якою потім робиться вимір для того, щоб отримати повідомлення m', відповідно до формулювання теореми 3.

Гра $GAME3_{A,PKE}(\lambda)$:	Гра $GAME4_{A,PKE}(\lambda)$:
1. $(pk, sk) \leftarrow PKE.KeyGen(1^\lambda)$	1. $(pk, sk) \leftarrow PKE.KeyGen(1^\lambda)$
2. $b \leftarrow \{0,1\}$	2. $b \leftarrow \{0,1\}$
3. $m^* \leftarrow \{0,1\}^n$	3. $m^* \leftarrow \{0,1)^n$
4. $r^* = \{0,1\}^n$	4. $r^* = \{0,1\}^n$
5. $c_0^* \leftarrow PKE.Enc(pk, m^*, r^*)$	5. $c_0^* \leftarrow PKE.Enc(pk, m^*, r^*)$
6. $K^* = \{0,1\}^n$	6. $K^* \leftarrow \{0,1\}^n$
7. $c_1^* = \{0,1\}^n$	7. $c_1^* \leftarrow \{0,1\}^n$
8. $c^* = (c_0^*, c_1^*)$	8. $c^* = (c_0^*, c_1^*)$
9. $m' = E^{A,H}(pk, c^*, K^*)$	9. $m' = E^{A,H,KDF}(pk, c^*, K^*)$
10. $return [\![m' == m^*]\!]$	10. $return [\![m' == m^*]\!]$

Для того, щоб оцінити ймовірність успіху супротивника у іграх GAME3, GAME4 змінимо оракул декапсуляції таким чином, щоб він не використовував секретний ключ, а відповідні ігри, що використовують змінений оракул декапсуляції позначимо як GAME5,GAME6. При побудові нового оракула декапсуляції NewDecaps використаємо той факт, що квантовий випадковий

оракул, до якого робиться q запитів, є невідрізнимим від випадкового полінома степені 2q над відповідним полем Галуа [83]. Відповідно, множина усіх значень r, для яких H(r)=d, може бути розглянута як множена коренів полінома H(X)–d. Новий оракул декапсуляції NewDecaps представлений у псевдокоді нижче. Замість таємного ключа для розшифрування повідомлення використовується множина значень r, що були вже запитані у оракула H.

Оракул $O_{Dec}((c_0, c_1) \neq (c_0^*, c_1^*))$:
1. $if\ \exists r \in Roots(H(x) - c_1): PKE.Dec(sk, c_0) = m$
2. $return\ K = KDF(r)$
3. $return\ \perp$

Розглянемо як зміниться перевага супротивника від GAME3 до GAME5 та від GAME4 до GAME6. Нехай супротивник A робить запит до оракула декапсуляції з деяким шифротекстом $(c_0, c_1) \neq (c_0^*, c_1^*)$. Оракул декапсуляції Decaps для цього шифротексту може повернути ключ декапсуляції або символ помилки декапсуляції ⊥.

Припустимо, що Decaps повертає ⊥ для шифротексту (c_0, c_1), тоді, якщо NewDecaps не повертає ⊥ у іграх GAME5–GAME6, то існує значення r для якого виконується $H(r) = c_1$. Різниця між іграми буде, якщо для c_0, r існує повідомлення m, для якого $m = MessageRecovery(c_0, r) \neq \perp$. Проте, якщо таке m існує, то Decaps не буде повертати ⊥, маємо протиріччя. Отже, таких m не існує і ігри в цьому випадку є невідрізнимими.

Припустимо, що Decaps не повертає ⊥. Тоді існує деяке r, що є коренем H і NewDecaps повертає $K = KDF(r)$. Ігри і в цьому випадку є не відрізнимими і має місце рівність:

$$\Pr[GAME3(A) == 1] = \Pr[GAME5(A) == 1]$$

$$\Pr[GAME4(A) == 1] = \Pr[GAME6(A) == 1]$$

Тож, задача звелася до оцінки складності ігор GAME5, GAME6. Для кожної з ігор можливо побудувати супротивників B_1, B_2 у грі OW–CPA проти

PKE, які є обгорткою над A. Супротивник B_i симулює середовище для A наступним чином:

- Генерує випадкові значення K^* та c_1^*
- Викликає $E^{A,Oracles}(pk,(c^*,c_1^*),K^*)$, де $Oracles = H$ для i=1 і Oracles=H,KDF для i=0.
- Повертає будь–що, що поверне $A^{E,Oracles}$.

Маємо:

$$Adv_{PKE}^{OW-CPA}(B_1) = Adv_{KEM}^{GAME5}(A), Adv_{PKE}^{OW-CPA}(B_2) = Adv_{KEM}^{GAME6}(A)$$

Нехай супротивник B у грі OW–CPA проти PKE паралельно викликає B_1, B_2 для OW–CPA проти PKE. Зрозуміло, що $Adv_{PKE}^{OW-CPA}(B) = \min\left(Adv_{PKE}^{OW-CPA}(B_1), Adv_{PKE}^{OW-CPA}(B_2)\right)$. Поєднуючи формули, маємо результат:

$$Adv_{KEM}^{IND-CCA}(A) \leq (2 \cdot q_H + 2 \cdot q_D + q_{KDF}) \cdot$$
$$\cdot \sqrt{Adv_{PKE}^{OW-CPA}(B)} + 8 \cdot (q_{BPGM} + q_D + 1)^2 \cdot \delta_{wc}$$

Що і треба було довести.

Оскільки в стандарті ДСТУ 8961:2019 використовується асиметрична схема, що ґрунтується на проблемі NTRU та загальносистемні параметри забезпечують відсутність помилок дешифрування, то маємо:

$$Adv_{\text{ДСТУ8961:2019}}^{IND-CCA}(A) \approx \sqrt{Adv_{NTRU}^{OW-CPA}(B)} \quad (4.15)$$

Де $Adv_{NTRU}^{OW-CPA}(B)$ – складність аналізу проблеми NTRU, яка вважається складною.

4.2.2. Оцінки безпеки

З теореми 1 випливає, що оцінка безпеки ДСТУ 8961:2019 може бути доказово зведена до проблеми NTRU. Поліноми f та g мають коефіцієнти у множині $\{-1,0,1\}$, проте кількість ненульових елементів сильно відрізняються. Для полінома f маємо $\|f\|_\infty = 2t$, де t – загальносистемний параметр, який для усіх наборів параметрів дає кількість ненульових елементів $d_f \approx n/3$. У той же

час $\|g\| = \frac{2n}{3} + 1$, що дає близький до рівномірного розподіл на множині $\{-1,0,1\}$ для полінома g. Тож, можливо вважати, що поліном g має рівномірний розподіл і використовувати апроксимовані параметри розподілів, що отримані в розділі 3. Для полінома f експерименти показали, що центрований нормальний розподіл з параметром $\sigma_f = 0.6$ достатньо гарно апроксимує розподіл ймовірностей, що зображено на рисунку 4.1.

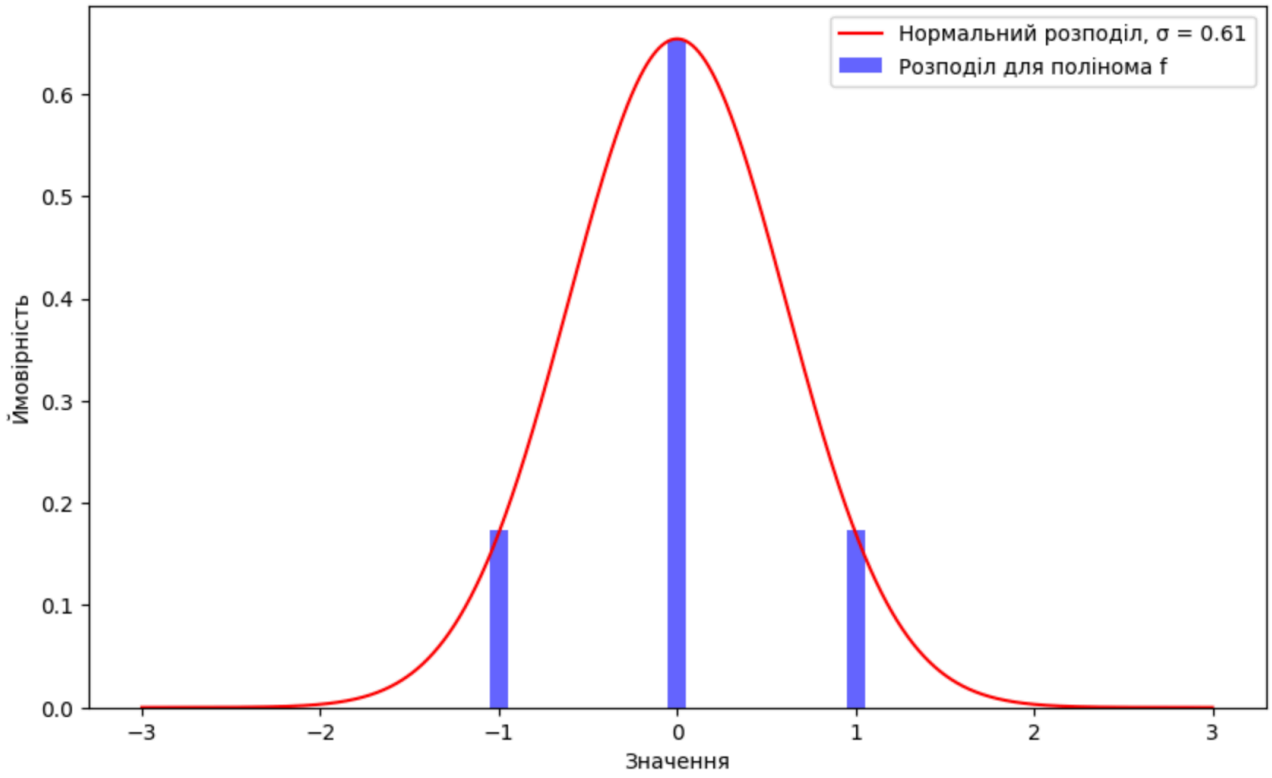

Рис. 4.1. Апроксимація розподілу ймовірностей для полінома f

У таблиці 4.2 наведено оцінки безпеки ДСТУ 8961:2019 у моделі GSA та з використанням симуляторів. Вартість атаки наведена для класичних комп'ютерів та квантових через «/».

З таблиці 4.2 видно, що врахування q–арної структури решіток дозволяє отримати більш точні оцінки безпеки криптографічних алгоритмів. Різниця між симулятором і моделлю GSA становить від 4 до 6 біт безпеки для всіх наборів параметрів. Це свідчить про те, що симулятор редукції краще враховує специфіку структури решіток у порівнянні з узагальненою моделлю GSA, яка дає більш грубі оцінки. Така різниця є особливо помітною для великих розмірностей

решіток, які використовуються в сучасних криптографічних схемах, що підтверджується експериментальними результатами для ДСТУ 8961:2019. Це означає, що для схем із більшими параметрами модель GSA стає менш точною.

Таблиця 4.2.

Оцінки безпеки для ДСТУ 8961:2019

Набір параметрів	Вартість атаки (біт, GSA)	Розмір блоку редукції	Вартість атаки (біт, симулятор)	Розмір блоку редукції	Найкраща атака
Скеля 256	178/161	611	182/165	624	Вкладення
Скеля 384	253/229	865	258/233	882	Вкладення
Скеля 512	312/283	1071	318/288	1090	Вкладення

Збільшення різниці між симулятором і моделлю GSA також пояснюється зростанням розмірності решіток, оскільки в таких умовах геометричні властивості базисів починають все більше впливати на складність алгоритмів редукції. Таким чином, для великих розмірностей решіток модель GSA стає менш придатною для оцінювання складності, що підтверджується більшою різницею в оцінках між нею та симулятором. Крім того, врахування квантових алгоритмів знижує оцінки безпеки на 20–30 біт.

Найкращою атакою для ДСТУ 8961:2019 виявилася атака вкладення, яка демонструє найменші оцінки безпеки порівняно з іншими підходами. Це може бути пояснено тим, що атака вкладення менше залежить від норм ортогоналізованого базису, у порівнянні з атаками декодування.

На рисунку 4.2 візуалізовано оцінки безпеки для ДСТУ 8961:2019, що дозволяє наочно продемонструвати різницю між використанням моделі GSA, симулятора редукції та впливом квантових атак. Візуалізація підтверджує важливість більш точного врахування структури решіток для оцінки безпеки сучасних криптографічних стандартів.

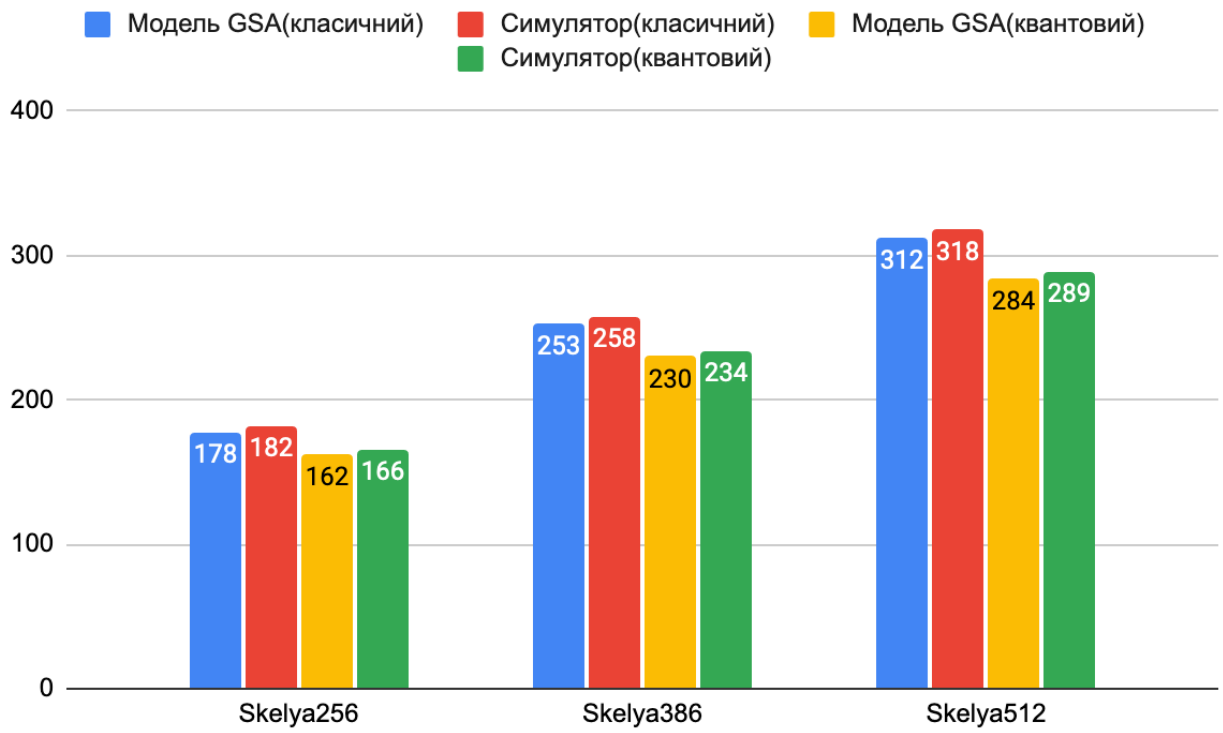

Рис. 4.2. Оцінки безпеки ДСТУ 8961:2019 у моделі GSA та з використанням симулятора

4.3. Оцінка безпеки Crystals–Kyber

Протокол інкапсуляції ключів CRYSTALS–Kyber [79] використовує перетворення у полі $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ і грунтується на проблемі Module–LWE [71]. Елементи поля представляються у вигляді поліномів.

У механізмі інкапсуляції ключів CRYSTALS–Kyber використовуються наступні криптографічні геш функції:

$$PRF: \{0,1\}^{32} \times \{0,1\}^8 \to \{0,1\}^*$$

$$XOF: \{0,1\}^* \times \{0,1\}^8 \times \{0,1\}^8 \to \{0,1\}^*$$

$$H: \{0,1\}^* \to \{0,1\}^{32}$$

$$G: \{0,1\}^* \to \{0,1\}^{32} \times \{0,1\}^{32}$$

$$KDF: \{0,1\}^* \to \{0,1\}^*$$

Додатково використовується геш функція $Parse$, яка перетворює бітову строку на елемент поля з рівномірного розподілу (при умові, якщо вхідні данні з рівномірного розподілу).

Для генерації вектору шуму використовується біноміальний розподіл B_η з параметром η. Відповідно, для генерації векторів поліномів з біноміального розподілу використовується функція CBD_η.

Схема асиметричного шифрування у дерандомізованому вигляді, що використовується у CRYSTALS–Kyber, зображена на рисунку 4.6. Для простоти представлення технічні деталі щодо перетворення поліномів у бітові строки, механізми стискання підпису та оптимізації (що не впливають на оцінки безпеки до загальних атак) не показані.

<table>
<tr>
<td>KyberPKE.Gen(1^λ):
1. $d \leftarrow \{0,1\}^{32}$
2. $(\rho, \sigma) = G(d)$
3. $A = (a_{ij} = XOF(\rho, j, i))$
4. $s = (s_0, \dots, s_{k-1}), s_i = CBD_{\eta_1}(\sigma, i)$
5. $s = (e_0, \dots, e_{k-1}), e_i = CBD_{\eta_1}(\sigma, k+i)$
6. $t = A \cdot s + e$
7. Return $pk = (t, \rho), sk = s$</td>
<td>KyberPKE.Enc($pk = (t, \rho), m, coins$):
1. $A = \left(a_{ij} = XOF(\rho, j, i)\right)$
2. $r = (r, \dots, r_{k-1}), r_i = CBD_{\eta_2}(PRF(coins, i))$
3. $e_1 = \left(e_{1(1)}, \dots, e_{1(k-1)}\right), e_{1(i)} = CBD_{\eta_2}(PRF(coins, k+i))$
4. $e_2 = CBD_{\eta_2}(PRF(coins, 2k))$
5. $u = Ar + e_1$
6. $v = t \cdot r + e_2 + m$
7. $c = (u, v)$
8. Return c</td>
</tr>
<tr>
<td colspan="2">KyberPKE.Dec($sk = s, c = (u, v)$):
1. $m = v - s \cdot u$
2. Return m</td>
</tr>
</table>

Для отримання протокола інкапсуляції ключів використовується варіант перетворення Фуджісакі–Окамото з неявним відхиленням [38,73]. Відповідний псевдокод наведено нижче.

KyberKEM.Encaps(pk):

1. $seed \leftarrow \{0,1\}^{32}$
2. $m = H(seed)$
3. $(key, r) = G(m||H(pk))$
4. $c = KyberPKE.Enc(pk, m, r)$
5. $K = KDF\big(key, H(c)\big)$
6. Return (c, K)

KyberKEM.Decaps$\big(c, sk = (sk_1, pk, H(pk), z)\big)$:

1. $m' = KyberPKE.Dec(sk_1, c)$
2. $(key', r') = G(m'||H(pk))$
3. $c' = KyberPKR.Enc(pk, m', r')$
4. $K = \begin{cases} KDF(key'||H(c)), c == c'' \\ KDF(z||H(c)), c \neq c'' \end{cases}$
5. Return K

KyberKEM.Gen$\big(1^{\lambda}\big)$:

1. $z \leftarrow \{0,1\}^{32}$
2. $(pk, sk_1) = KyberPKE.Gen\big(1^{\lambda}\big)$
3. $sk = (sk_1, pk, H(pk), z)$
4. Return (pk, sk)

Згідно до специфікації, CRYSTALS–Kyber підтримує три набори загальносистемних параметрів. Параметри зведені у таблицю 4.3.

Таблиця 4.3.

Загальносистемні параметри ПІК CRYSTALS–Kyber

Набір параметрів	n	k	q	η_1	η_2	(d_u, d_v)	δ
Kyber512	256	2	3329	3	2	(10,4)	2^{-139}
Kyber768	256	3	3329	2	2	(10,4)	2^{-164}
Kyber1024	256	4	3329	2	2	(11,5)	2^{-174}

Параметри n та q визначають поле, параметр k задає розмірність векторів, параметри η_1 та η_2 є параметрами біноміального розподілу для векторів s та e відповідно. Параметри d_u, d_v використовуються під час кодування поліномів у бітову строку, параметр δ є ймовірністю помилки декапсуляції.

З таблиці 4.3. добре видна суто практична перевага проблеми Module–LWE: для всіх рівнів безпеки використовується одне поле. Такий підхід дає змогу значно спростити реалізацію та масштабувати систему для довільного рівня безпеки. Окрім того, використання відносно малого значення для параметра q дозволяє ефективно використовувати векторизацію обчислень.

4.3.1. Аналіз в моделі квантового випадкового оракула

В роботі [79] авторами Crystals–Kyber був проведений детальний аналіз безпеки у моделі квантового випадкового оракула. Цими результатами можливо скористатися для подальшого аналізу.

Теорема 2 [79]. Нехай XOF, H та G є випадковими оракулами. Тоді для будь–якого класичного супротивника A, що робить не більше q_{RO} запитів до випадкових оракулів XOF, H та G, існують класичні супротивники B та C, для яких

$$Adv_{Kyber}^{IND-CCA}(A) \leq 2Adv_{k+1,k,\eta}^{MLWE}(B) + Adv_{PRF}^{prf}(C) + 4q_{RO}\delta \quad (4.16)$$

Теорема 3 [59]. Нехай XOF, H та G є квантовими випадковими оракулами. Тоді для будь–якого квантового супротивника A, що робить не більше q_{RO} запитів до випадкових оракулів XOF, H та G, існують квантові супротивники B та C, для яких

$$Adv_{Kyber}^{IND-CCA}(A) \leq 4q_{RO} \cdot \sqrt{Adv_{k+1,k,\eta}^{MLWE}(B)} + Adv_{PRF}^{prf}(C) + 8q_{RO}\delta \quad (4.17)$$

З теорем 2,3 видно, що формальні докази безпеки для Crystals–Kyber мають таку ж структуру, що і отримані докази для ДСТУ 8961:2019.

4.3.2. Оцінки безпеки

З теорем 2,3 випливає, що якщо вважати симетричні криптопримітиви безпечними, то безпека Crystals–Kyber цілком зводиться до проблеми MLWE. Оскільки для криптографічних наборів параметрів не відомо як використовувати алгебраїчну структуру MLWE, то можливо вважати, що безпека Crystals–Kyber зводиться до LWE. Оскільки Crystals–Kyber використовує біноміальний розподіл, який є дискретним аналогом нормального розподілу, то аналіз полегшується. У таблиці 4.3 наведені оцінки атак вкладення на проблему MLWE, що асоційована з кожним набором загальносистемних параметрів.

Таблиця 4.4.

Оцінки безпеки CRYSTALS–Kyber

Набір параметрів	Вартість атаки (біт, GSA)	Розмір блоку редукції	Вартість атаки (біт, симулятор)	Розмір блоку редукції	Найкраща атака
Kyber512	118/108	406	121/110	449	Вкладення
Kyber786	183/166	625	187/170	687	Вкладення
Kyber1024	256/233	878	263/239	950	Вкладення

З таблиці 4.4 можна побачити, що врахування q–арної структури решіток має вплив на рівень безпеки криптографічних алгоритмів, створюючи різницю в межах 3–10 біт безпеки для різних наборів параметрів. Різниця є більшою у порівнянні з ДСТУ 8961:2019. Це пояснюється різними розмірностями решіток.

Квантові атаки можуть знизити рівень безпеки на 20–30 біт для Crystals–Kyber. Це вказує на необхідність врахування квантового аспекту при розробці сучасних криптографічних алгоритмів, особливо в контексті постквантової криптографії [71].

На рисунку 4.3 наведено графічне представлення оцінок безпеки для криптографічного алгоритму Crystals–Kyber, що дозволяє візуально проаналізувати різницю між різними сценаріями атак та рівнями безпеки.

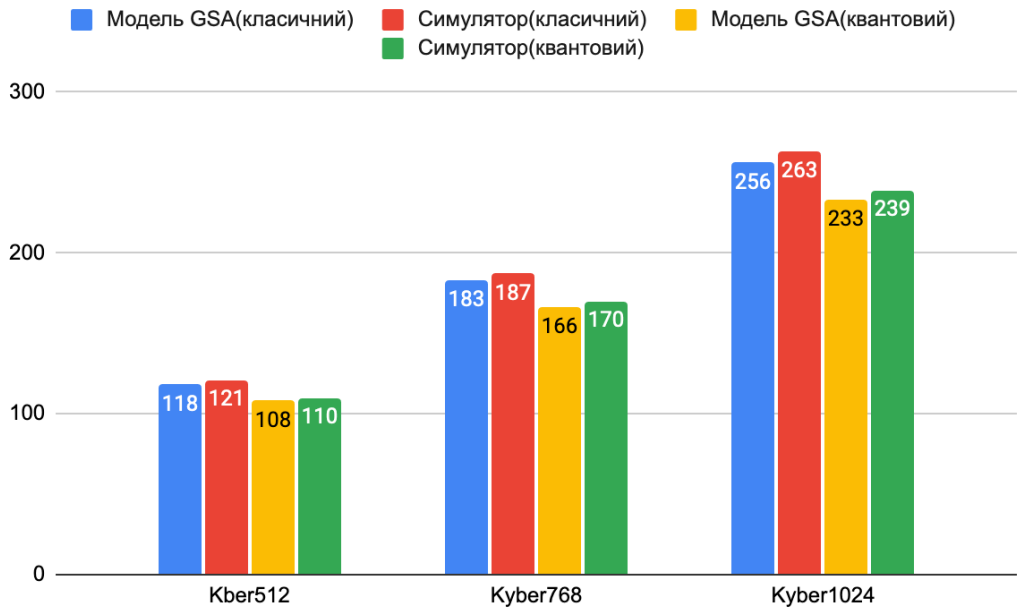

Рис. 4.3. Оцінки безпеки Crystals–Kyber у моделі GSA та з використанням симулятора

4.4. Висновки до розділу

1. Для ДСТУ 8961:2019 було вперше отримано доказ безпеки в моделі квантового випадкового оракула. Цей доказ гарантує, що механізм інкапсуляції ключів стійкий до атак з адаптивно підібраними шифротекстами, як класичних, так і квантових, за умови, що проблема NTRU залишається нерозв'язаною. Отримання такого доказу не лише підвищує довіру до безпеки ДСТУ 8961:2019, але й закриває важливу прогалину у відповідності до міжнародних стандартів, де аналогічні докази вже є базовою вимогою. Це також підтверджує конкурентоспроможність стандарту на міжнародному рівні, особливо в умовах швидкого розвитку квантових обчислень.

2. Було уточнено рівні безпеки механізмів інкапсуляції ключів відповідно до стандартів ДСТУ 8961:2019 та Crystals–Kyber. Аналіз показав, що різниця між оцінками у моделі GSA (Geometric Series Assumption) та моделі, яка враховує алгебраїчну структуру q –арних решіток, становить 4–10 біт безпеки, залежно від обраних параметрів. Уточнені оцінки свідчать, що сучасні атаки, особливо в класичних умовах, менш ефективні, ніж припускалося раніше. Крім того,

врахування квантових атак показало зниження рівня безпеки на 20–30 біт. Це підкреслює важливість комплексного підходу до оцінки криптографічних систем із врахуванням майбутніх квантових загроз.

3. Уточненні оцінки безпеки та отримані формальні докази в моделі квантового випадкового оракула свідчать про високий рівень зрілості ДСТУ 8961:2019 як криптографічного стандарту. Ці результати демонструють, що ДСТУ 8961:2019 може конкурувати з провідними міжнародними стандартами, такими як Crystals–Kyber, особливо з урахуванням зниження ефективності існуючих атак і підтвердження стійкості до квантових загроз. Це створює передумови для ширшого застосування ДСТУ 8961:2019 у національних і міжнародних криптографічних системах, забезпечуючи високий рівень захисту інформації в умовах стрімкого технологічного розвитку.

РОЗДІЛ 5. ОЦІНКА ЗАХИЩЕНОСТІ ЕЛЕКТРОННИХ ПІДПИСІВ НА АЛГЕБРАЇЧНИХ РЕШІТКАХ

Розділ присвячено аналізу електронних підписів. Як і для механізмів інкапсуляції ключів, які досліджувалися в четвертому розділі, для електронних підписів спочатку досліджується формальний доказ безпеки, потім складні проблеми, до криптоаналізу яких зводиться формальний аналіз. Оцінка відбувається в моделі EUF–CMA.

Для позначення предикатів використовується позначення $[\![\cdot]\!]$. Якщо b є деяким твердженням, то предикат $[\![b]\!]$ приймає значення 1, якщо b є істинним, та 0 інакше. Якщо змінна x приймає значення детермінованим чином, то використовується знак «=». Якщо змінна x приймає значення з деякого випадкового процесу, то використовується символ «←». Для визначеної множини X позначення $x \leftarrow X$ означає, що змінна x приймає випадкове значення з рівномірного розподілу над X. Символом «==» позначатимемо перевірку на рівність аргументів. Ймовірність деякої події W надалі позначатимемо символом Pr[W], математичне очікування для деякого розподілу S надалі позначатимемо як E[S]. Для заданої множини X вираз |X| означає потужність множини. Для числа x вираз |x| означає абсолютне значення.

Тут і надалі запис $Adv(\lambda) = negl(\lambda)$ означає, що значення функції Adv при збільшенні параметра λ зменшується швидше за будь–який поліном. Більш формально: для будь якого полінома $p(\lambda)$ виконується $\lim_{\lambda\to\infty} Adv(\lambda)p(\lambda) = 0$.

У межах цього дослідження використовуються наступні статистичні методи:

– критерій Хі–Квадрат: це статистичний тест, що використовується для перевірки гіпотез про незалежність чи відповідність розподілу спостережуваних частот теоретичним частотам. Використовується для категорійних даних і базується на порівнянні очікуваних і фактичних частот;

– критерій Колмогорова–Смірнова: це непараметричний тест, який використовується для порівняння двох вибірок або для перевірки відповідності

вибірки теоретичному розподілу. Тест заснований на найбільшій відстані між кумулятивними функціями розподілу вибірки і теоретичного розподілу;

– квантиль–квантиль графік (Q–Q графік): це графічний інструмент для порівняння двох розподілів шляхом нанесення їх квантілів один проти одного. Якщо точки на графіку утворюють приблизно пряму лінію, це означає, що розподіли подібні;

– критерій Краскела–Уолліса: це непараметричний метод для порівняння медіан декількох незалежних груп. Він є узагальненням критерію Манна–Уітні і використовується, коли припущення нормальності для даних не виконується;

– t–критерій Стьюдента: це параметричний тест, який використовується для порівняння середніх значень двох вибірок або для перевірки, чи середнє значення однієї вибірки відповідає заданому значенню. Застосовується, коли дані мають нормальний розподіл і відомі або невідомі дисперсії;

– поліноміальна регресія: це метод регресійного аналізу, в якому залежність між незалежною змінною і залежною змінною моделюється як поліном. Поліноміальна регресія дозволяє враховувати нелінійні залежності між змінними.

5.1. Модель EUF–CMA

У моделі EUF–CMA супротивник може звертатися до оракула підпису $Sign(sk,\cdot)$ для отримання підписів довільно обраних повідомлень. Схема підпису вважається безпечною, якщо ймовірність того, що супротивник зможе підробити підпис для будь–якого повідомлення є не значною. Так само, як і IND–CCA, доказова безпека у моделі EUF–CMA формуються через ігри (експерименти). Позначимо відповідний експеримент $Exp_{A,S}^{EUF-CMA}(1^{\lambda})$ для схеми підпису S та супротивника A. У цьому експерименті випробовувач генерує ключову пару (sk, pk) та надає супротивнику відкритий ключ pk. Супротивник може роботи запити $m_1, \dots, m_q$ до оракула $Sign$. Усі запити до оракула зберігаються у списку Q. Після чого супротивник має повернути пару (m^*, σ^*).

Якщо $S.Verify(pk, m^*, \sigma^*) = 1$ і $m^* \notin Q$, то супротивник перемагає. Більш формально експеримент $Exp_{A,S}^{EUF-CMA}(1^\lambda)$ описаний у псевдокоді нижче.

$Exp_{A,S}^{EUF-CMA}(1^\lambda)$:

1. $(pk, sk) \leftarrow S.KeyGen(1^\lambda)$
2. $(m^*, \sigma^*) \leftarrow A^{Sign(sk,\cdot)}(pk) \,//\, Q = \{m_1, m_2, \dots, m_q\}$
3. $return\ [\![S.Verify(pk, m^*, \sigma^*)]\!]\ and\ [\![m^* \notin Q]\!]$

Перевага супротивника визначається як

$$Adv_{A,S}^{EUF-CMA}(1^\lambda) = \Pr\left[[\![Exp_{A,S}^{EUF-CMA}(1^\lambda)]\!]\right] \quad (5.1)$$

Якщо $Adv_{A,S}^{EUF-CMA}(1^\lambda) = negl(\lambda)$, то схема підпису вважається безпечною у моделі EUF–CMA.

Посиленим варіантом моделі безпеки EUF–CMA є модель SUF–CMA. Якщо у моделі EUF–CMA вимагається створити підпис для повідомлення, що раніше не було підписано, то у моделі SUF–CMA вимагається створити підпис для будь–якого повідомлення, навіть якщо воно було вже підписано. Відповідний формальний експеримент $Exp_{A,S}^{SUF-CMA}(1^\lambda)$ відрізняється від $Exp_{A,S}^{EUF-CMA}(1^\lambda)$ лише тим, що список Q містить не тільки запити до оракула підпису, а й відповіді. І у кінці перевіряється, що $(m^*, \sigma^*) \notin Q$, як наведено у псевдокоді нижче.

$Exp_{A,S}^{SUF-CMA}(1^\lambda)$:

1. $(pk, sk) \leftarrow S.KeyGen(1^\lambda)$
2. $(m^*, \sigma^*) \leftarrow A^{Sign(sk,\cdot)}(pk) \,//\, Q = \{(m_1, \sigma_1), \dots, (m_q, \sigma_q)\}$
3. $return\ [\![S.Verify(pk, m^*, \sigma^*)]\!]\ and\ [\![(m^*, \sigma^*) \notin Q]\!]$

Перевага супротивника аналогічно до EUF–CMA.

$$Adv_{A,S}^{SUF-CMA}(1^\lambda) = \Pr\left[[\![Exp_{A,S}^{SUF-CMA}(1^\lambda)]\!]\right]$$

Якщо $Adv_{A,S}^{SUF-CMA}(1^\lambda) = negl(\lambda)$, то схема підпису вважається безпечною у моделі SUF–CMA.

5.2. Оцінка безпеки електронного підпису Falcon

Схема електронного підпису Falcon має в своїй основі фреймворк GPV, що був вперше запропонований в роботі [80] для побудови квантово–стійких електронних підписів на решітках. Сутність фреймворку GPV полягає у наступному:

- Відкритий ключ задається матрицею $A \in \mathbb{Z}_q^{n \times m}$ (де $m > n$). Ця матриця задає базис q–арної решітки Λ.
- Таємний ключ задається матрицею $B \in \mathbb{Z}_q^{m \times m}$. Ця матриця задає базис дуальної решітки $\Lambda_q^{\perp}$, яка, згідно до визначення, є ортогональною до Λ за модулем q. Тобто, для будь–яких векторів $x \in \Lambda$ та $y \in \Lambda_q^{\perp}$ виконується $x \cdot y = 0 mod q$, де $\cdot$ – операція скалярного добутку.

Для заданого повідомлення m підписом є малий (у сенсі евклідової норми) вектор $s \in \mathbb{Z}_q^m$, для якого виконується $sA^T = H(m)$, де $H: \{0,1\}^* \to \mathbb{Z}_q^n$ – стійка до колізій геш функція. Для перевірки підпису достатньо перевірити, що виконується рівняння $sA^T = H(m)$.

Для обчислення підпису спочатку обчислюється довільний випадковий вектор $c_0 \in \mathbb{Z}_q^m$, для якого виконується $c_0 A^T = H(m)$. Оскільки до вектора c_0 не накладається вимог щодо значень його евклідової норми, то його знайти можливо стандартними засобами лінійної алгебри за поліноміальний час. Далі використовується таємний базис B для обчислення вектора $z \in \Lambda_q^{\perp}$, який є близьким до вектора c_0. Різниця векторів $s = c_0 - z$ є коректним підписом, оскільки $sA^T = c_0 A^T - zA^T = c - 0 = H(m)$. Якщо c_0 та v є достатньо близькими, то s буде малим.

Електронний підпис Falcon використовує у якості решітки Λ NTRU решітку [77]. Застосовуючи NTRU решітки до фреймворку GPV, Falcon вносить наступні зміни до GPV:

- Відкритим ключем є поліном h, який використовується для обчислення публічного базису NTRU решітки Λ.

- Таємним ключем є поліноми $f, g, F, G \in \mathbb{Z}[x]/(\phi)$, які використовуються для обчислення базису дуальної решітки $\Lambda_q^{\perp}$

- Підпис для повідомлення m складається з пари поліномів (s_1, s_2), для яких виконується $s_1 + s_2 h = H(r||m)$, де r – сіль (salt). При обчисленні підпису використовується таємний ключ для обчислення вектора $z = (z_0, z_1) \in \Lambda_q^{\perp}$, який є близьким до вектора $t = (H(m||r), 0)$. Різниця векторів t та z є коректним підписом.

Для обчислення вектору $z = (z_0, z_1)$ використовується алгоритм семплування (вибірки), що повертає вектор з нормального розподілу. Особливістю алгоритму семплування Falcon [81] є використання для пришвидшення операцій алгебраїчної структури циклотомічного поля та перетворення Фур'є. Falcon також використовує деревовидні структури даних – LDL дерева. Деталі можливо знайти в специфікації [82].

Falcon використовує наступні загальносистемні параметри:

- Параметри поля (n, q)
- Параметр розподілу таємних ключів $\sigma_{\{f,g\}} = 1.17\sqrt{q/2n}$
- Параметр розподілу підписів σ
- Обмеження на максимальний розмір підписів B

Загальносистемні параметр Falcon зведені в таблиці 5.1

Таблиця 5.1.

Загальносистемні параметри Falcon

	Falcon512	Falcon1024
(n, q)	(512,12289)	(1024,12289)
$\sigma_{\{f,g\}}$	4.0531638033	2.86601961058
σ	165.736 617 183	168.388 571 447
B	5833.92886484	8382.43651929

5.2.1. Оцінка в моделі EUF–CMA

Не зважаючи на те, що Falcon є фіналістом конкурсу NIST, безпосередньо його аналізу у моделі EUF–CMA присвячено не так багато робіт. Оскільки схема підпису ґрунтується на фрейворку GPV, то можливо адаптувати докази з оригінальної роботи.

Проте, можливо довести безпеку іншим шляхом. Фреймворк GPV є частковим випадком парадигми Hash–and–Sign. В останні роки для парадигми Hash–and–Sign з'явилося багато робіт щодо безпеки EUF–CMA у моделі квантового випадкового оракула. Кожен результат ґрунтується на певних модельних припущеннях. Результат у роботі [83] зручно використовувати, оскільки він ґрунтується на тих самих припущеннях, що і докази безпеки фреймворку GPV.

У загальному випадку підпис Hash–and–Sign параметризується стійкою до колізій геш функцією H та односторонньою функцією з лазівкою T, що є стійкою до знаходження прообразу (англ. Preimage–resistant trapdoor function). У випадку Falcon геш–функція H реалізується через shake256, тож будемо вважати, що вона є криптографічною. Одностороння функція T в Falcon є функцією з фреймворку GPV, до якої додана структура NTRU решітки.

Адаптуємо основний результат роботи [83] до схеми підпису Falcon наступним чином:

Теорема 5.1 ([83], теорема 1). Для будь–якого квантового супротивника A у грі EUF–CMA для схеми підпису Falcon, що робить не більше q_{sign} класичних запитів до оракулу підпису та q_{qro} квантових запитів до квантового оракулу H, існує супротивник B, що може інвертувати односторонню функцію T та супротивник D, що може знайти прообраз для T, використовуючи q_{sign} запитів до оракулу підпису. При цьому перевага супротивника A становить

$$Adv_{A,Falcon}^{EUF-CMA}\left(1^{\lambda}\right) \leq (2q_{ro}+1)^2 Adv_T^{INV}(B) + Adv_T^{PS}(D)$$
$$+\frac{3}{2}q'_{sign}\sqrt{\frac{q'_{sign}+q_{qro}+1}{|R|}} + 2(q_{ro}+2)\sqrt{\frac{q'_{sign}-q_{sign}}{|R|}} \qquad (5.2)$$

Де $|R|$ – розмір простору бітових строк, що використовуються у якості випадкових значень, q'_{sign} – максимальна загальна кількість запитів до оракула H в усіх запитах на підпис, $Adv_T^{INV}(B)$ – перевага супротивника B в інвертуванні T, $Adv_T^{PS}(D)$ – перевага супротивника D в знаходженні прообразу T.

Якщо A робить тільки класичні запити до оракула H, то

$$Adv_{A,Falcon}^{EUF-CMA}\left(1^{\lambda}\right) \leq (2q_{ro}+1)^2 Adv_T^{INV}(B) + Adv_T^{PS}(D)$$
$$+q'_{sign}\frac{q'_{sign}+q_{qro}+1}{|R|} + (q_{ro}+1)\frac{q'_{sign}-q_{sign}}{|R|} \quad (5.3)$$

Для Falcon $|R| = |\{0,1\}^{384}|$. Тож, доказ безпеки у моделі EUF–CMA зводить безпеку Falcon до безпеки односторонньої функції з лазівкою T: до складності інвертування та складності пошуку прообразу.

Інвертування T означало б вирішення проблеми NTRU, тож

$$Adv_T^{INV}(B) \leq Adv_{n,q,\sigma}^{NTRU} \quad (5.4)$$

Знаходження прообразу T є рішенням $s = (s_1, s_2)$ рівняння $s_1 + s_2 h = H(r||m)$. Знаходження рішення рівняння є в точності проблемою ISIS з параметром B, тому

$$Adv_T^{PS}(D) \leq Adv_{n,q,B}^{ISIS} \leq Adv_{n,q,B}^{SIS} \quad (5.5)$$

Тож, безпеки Falcon можливо звести до проблем NTRU та SIS на NTRU решітках.

5.2.2. Оцінки безпеки

Оскільки задача інвертування односторонньої функції в електронному підписі Falcon зводиться до проблем NTRU та ISIS, то конкретні оцінки складності зводяться до оцінки складності атак вкладення та декодування.

У таблиці 5.2 наведені оцінки безпеки екземплярів проблеми NTRU, на яку спирається Falcon. У роботі [84] були запропоновані загальносистемні параметри для рівня безпеки 512 біт для схеми Falcon. Окрім стандартних наборів параметрів таблиця 5.2 містить також оцінки для набору параметрів з роботи [84] (Falcon2048).

Таблиця 5.2.

Оцінка безпеки Falcon (Проблема NTRU).

	Вартість атаки (біт, GSA)	Розмір блоку редукції	Вартість атаки (біт, симулятор)	Розмір блоку редукції
Falcon512	140/128	483	144/131	494
Falcon1024	268/244	918	276/251	944
Falcon2048	608/552	2083	628/570	2149

Аналіз криптостійкості параметрів Falcon у таблиції 5.2 демонструє чітку залежність між розмірністю параметрів та рівнем безпеки. Зі збільшенням параметрів від Falcon512 до Falcon2048 суттєво зростає як класична, так і квантова стійкість системи. Це підтверджує, що більші розмірності решіток є ефективним підходом для підвищення захисту криптографічних механізмів, особливо в умовах перспективних квантових загроз. Результати оцінки криптостійкості стосуються задачі NTRU, на складності якої базується схема електронного підпису Falcon.

Порівняння моделей GSA та симулятора редукції на рисунку 5.1 показує, що остання дає більш оптимістичні результати безпеки. Симулятор враховує більше факторів, що дозволяє точніше оцінювати криптостійкість проблеми NTRU і демонструє, що реальні атаки можуть бути менш ефективними, ніж оцінювалося раніше за допомогою GSA. Водночас, обчислювальні вимоги до редукції дещо вищі у симуляторі, що є компромісом між точністю оцінки та складністю реалізації.

Окремо варто відзначити зниження рівня безпеки для квантових атак у порівнянні з класичними. Проблема NTRU, хоча і стійка до багатьох атак, усе ж демонструє вразливість до квантових обчислень, що проявляється у зниженні криптостійкості для всіх параметрів. Однак навіть у таких умовах, параметри Falcon2048 забезпечують досить високий рівень захисту, що робить їх придатними для використання у висококритичних системах. Менші параметри,

як–от Falcon512, можуть бути застосовані в середовищах з менш жорсткими вимогами до безпеки.

Загалом, результати підтверджують надійність механізмів Falcon, заснованих на задачі NTRU. Використання сучасних моделей оцінки криптостійкості дозволяє точніше оцінити її стійкість, враховуючи як класичні, так і квантові загрози. Falcon демонструє високу гнучкість, що дозволяє адаптувати параметри під конкретні загрози та ресурси, забезпечуючи баланс між продуктивністю та безпекою.

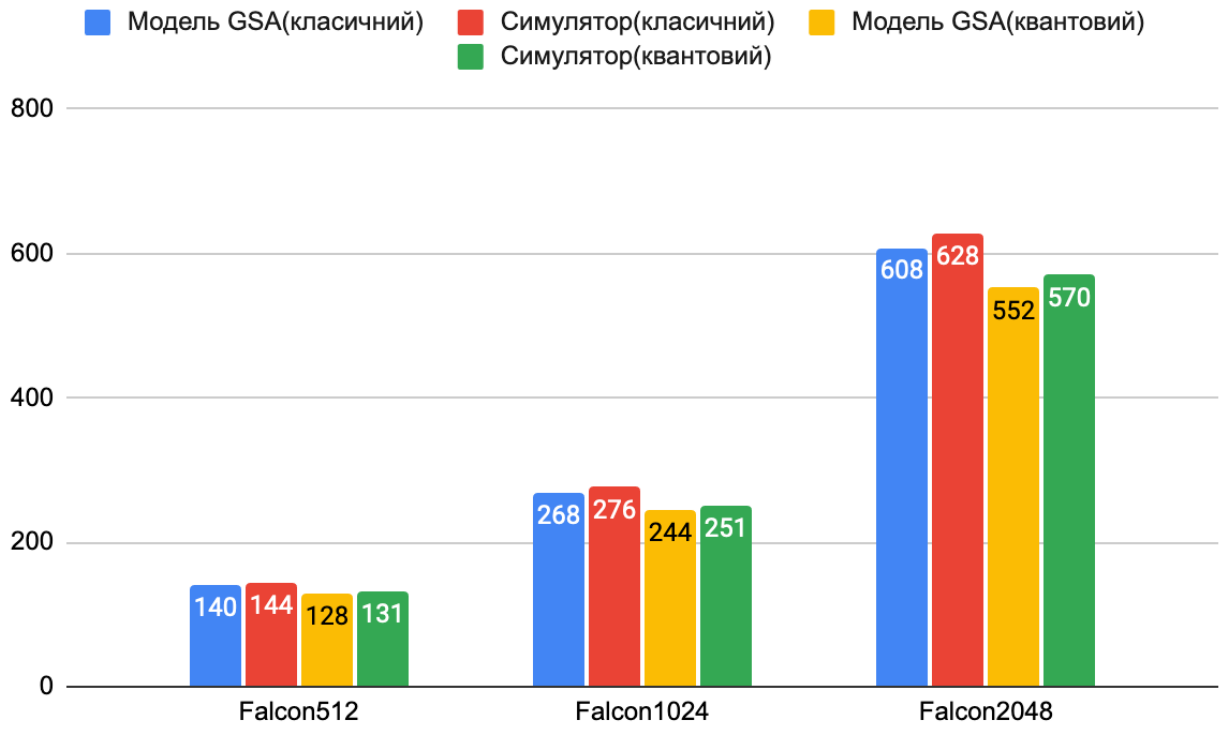

Рис. 5.1. Оцінка безпеки електронного підпису Falcon для моделі GSA та симулятора (Проблема NTRU).

Аналіз криптостійкості механізмів Falcon також охоплює оцінку безпеки проти підробки підпису, що базується на складності задачі SIS (Short Integer Solution). Дані, представлені в таблиці 5.3, узагальнюють результати оцінки для наборів параметрів Falcon, наведених у таблиці 5.1, а також додаткових параметрів із роботи [84]. Відповідно до цих даних, рівень безпеки залежить як від розмірності параметрів, так і від використаної моделі оцінювання.

Таблиця 5.3.

Результати оцінки безпеки Falcon (Проблема SIS)

	Вартість атаки (біт, GSA)	Розмір блоку редукції	Вартість атаки (біт, симулятор)	Розмір блоку редукції
Falcon512	110/100	373	112/103	446
Falcon1024	253/230	878	260/237	1169
Falcon2048	559/508	1915	575/523	1972

Зі збільшенням параметрів (від Falcon512 до Falcon2048) зростає як класична, так і квантова стійкість до підробки підпису. У моделі GSA, класична стійкість для Falcon512 складає відносно низький рівень, який поступово зростає для Falcon1024 і Falcon2048, досягаючи значного рівня захисту для останнього. Симулятор редукції, як і в попередньому аналізі задачі NTRU, демонструє дещо вищий рівень захисту для всіх параметрів у порівнянні з GSA, що свідчить про його більшу точність.

На рисунку 5.2 візуалізовано рівні безпеки для підробки підпису в моделі GSA і симуляторі редукції. Дані показують, що хоча модель GSA з квантовою оцінкою демонструє деяке зниження стійкості у порівнянні з класичною, симулятор редукції забезпечує більший рівень стійкості навіть у квантовому сценарії.

Для Falcon512 квантова стійкість у моделі GSA демонструє зниження до мінімальних значень, що вказує на ризик у використанні таких параметрів для задач із високим рівнем безпеки. Проте Falcon2048 забезпечує значно вищі показники криптостійкості навіть у квантовому середовищі, роблячи його придатним для критичних систем із довготривалими вимогами безпеки.

Дані з таблиці 5.3 також підкреслюють, що різниця між класичною та квантовою стійкістю є значною для задачі SIS, але обидва підходи оцінки (GSA та симулятор редукції) підтверджують, що задача SIS залишається важливим механізмом забезпечення стійкості електронних підписів. Це підкреслює

необхідність вибору відповідних параметрів для забезпечення стійкості як до класичних, так і до квантових атак.

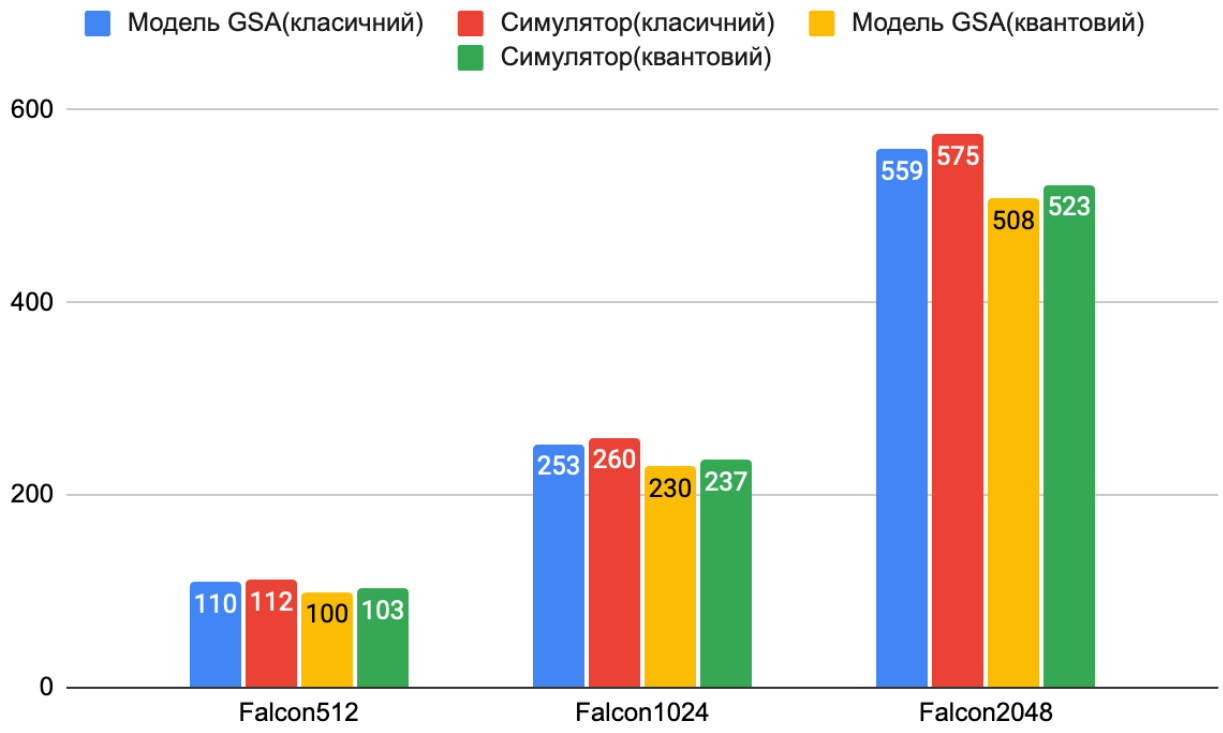

Рис. 5.2. Вартість атаки на SIS для різних моделей безпеки.

5.2.3. Атака на реалізацію електронного підпису Falcon

У роботі [85] запропоновано атаку на реалізацію електронного підпису Falcon. Атака ґрунтується на використанні двох різних реалізацій, що за рахунок шуму округлення дають для однакових повідомлень різні підписи [85, 86]. Ключовим рівнянням атаки є

$$h = \frac{g\delta_0 + G\delta_1}{f\delta_0 + F\delta_1} \tag{5.6}$$

де $\delta_0 = z_0^1 - z_0^0$ та $\delta_1 = z_1^1 - z_1^0$ (верхній індекс позначає порядковий номер реалізації)

Ідея атаки полягає у тому, щоб підібрати повідомлення m таким чином, щоб $\delta_1 = 0$, а δ_0 було деяким достатньо малим для використання алгоритму пошуку найбільшого спільного дільника поліномом (в ідеалі – $\delta_0 \in \mathbb{Z}$). Тоді, можливо буде обчислити таємний ключ $\tilde{f}, \tilde{g}$ як:

$$\tilde{f} = \frac{s_1^1 - s_1^0}{\gcd(s_1^1 - s_1^0, s_0^0 - s_0^1)}$$

$$\tilde{g} = \frac{s_0^0 - s_0^1}{\gcd(s_1^1 - s_1^0, s_0^0 - s_0^1)} \quad (5.7)$$

Де gcd – найбільший спільний дільник поліномів.

У роботі [85] атака реалізована на прикладі реалізації [82]. Цю реалізацію в подальшому будемо називати еталонною реалізацією. В розглянутому варіанті реалізації наведено два варіанти обчислення ЕП.

Варіант 1. Обчислення на основі секретного ключа – поліномів f, g, F, G. В цьому випадку паралельно з обчисленням ЕП виконується обчислення LDL дерева та компонентів ЕП (функція sign_dyn).

Варіант 2. Усі обчислення для LDL дерева залежать тільки від секретного ключа і не залежать від повідомлення для підпису. Вони виконуються заздалегідь (функція expand_privkey). Це значно прискорює формування ЕП в разі, якщо необхідно підписати декілька документів за допомогою одного секретного ключа, але суттєво збільшує розмір потрібної пам'яті (треба зберігати LDL дерево). В разі, якщо носій секретного ключа дозволяє по захищеній пам'яті, ці обчислення можуть бути виконані один раз після генерації ключів, і тоді ефект від застосування другого способу буде навіть в разі формування тільки одного підпису (Функція sign_tree).

Атака [85] дає змогу підписувати документи і успішно перевіряти підпис легальним відкритим ключом в разі отримання різних підписів в умовах застосування однакових повідомлень для підпису, та випадкових даних seed2 та nonce. Тому доцільно більш детально проаналізувати умови для обчислення різних підписів і засоби запобігання цього.

Щоб отримати необхідні оцінки була зібрана наступна статистика: для 10 випадкових ключів для N=2,3,4,5,6,7,8,9,10 обчислювалися підписи на випадкових 33–байтних повідомленнях до тих пір, доки не було отримано 100 подій

- Різниць в підписах при однакових повідомленнях та seed;
- Успішної атаки відновлення ключа.

Таким чином, у процесі дисертаційного дослідження було зібрано дві статистики [86]. Для кожного ключа для кожного значення N статистика підпорядковується експоненціальному розподілу. Цю гіпотезу було перевірено на рівні 0.01 тестами Хі–квадрат та Колмогорова–Смірнова [81]. Для усіх ключів нульова гіпотеза про експоненціальний розподіл була прийнята. На рисунку 1 наведено типовий розподіл та Q–Q графік.

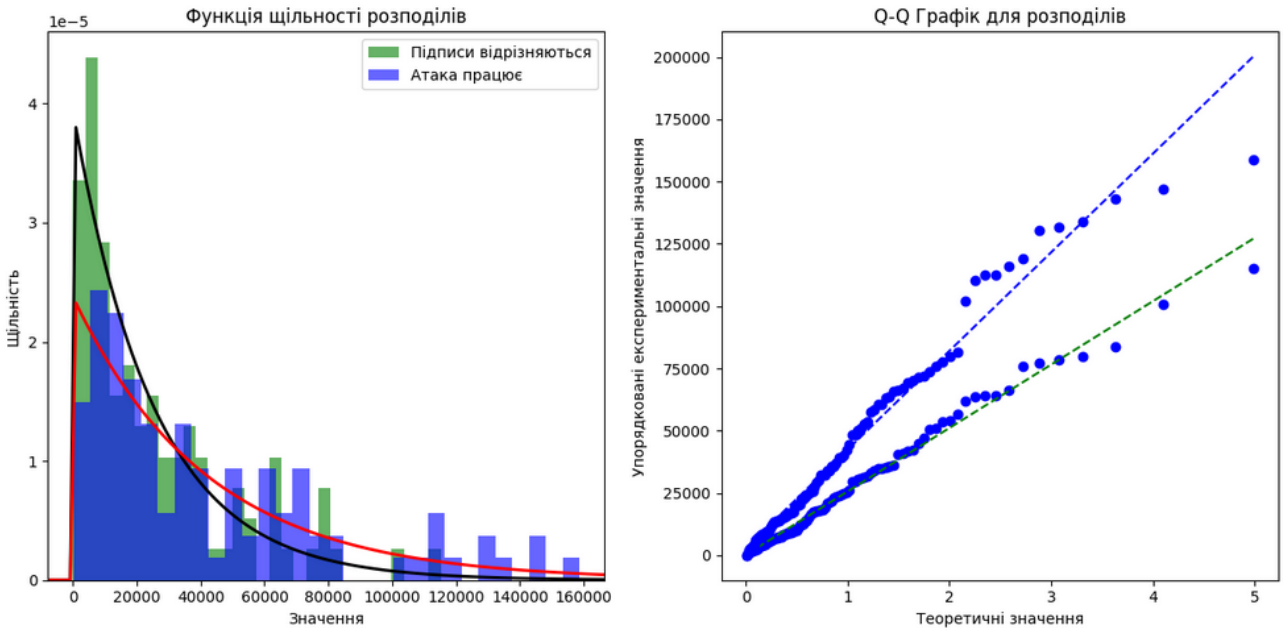

Рис. 5.3. Функція щільності та Q–Q графік для N=9

З рисунку 5.3 видно, що гіпотеза про експоненціальний розподіл ймовірностей є правдоподібною.

Хоча для усіх ключів події мають експоненціальний розподіл, проте параметри розподілу можуть відрізнятися для кожного ключа [81].

Щоб перевірити гіпотезу про еквівалентність ключів був застосований критерій Краскела–Уолліса (H–критерій) до усіх ключів для всіх значень N. Відповідні данні занесено до таблиці 5.5. Якщо p–значення є більшим за рівень значущості, то вважається, що статистично значимої різниці між розподілами не має.

Таблиця 5.5.

Оцінка рівності медіан розподілів за критерієм Краскела–Уолліса

N	Виникає різниця в підписах		Успішна атака	
	Статистика	p–значення	Статистика	p–значення
2	62.911	3.67e–10	131.961	4.68e–24
3	76.360	8.507e–13	45.352	7.93e–07
4	31.218	0.00027	53.805	2.05e–08
5	22.257	0.00809	26.346	0.00179
6	30.343	0.00038	28.641	0.00074
7	48.355	2.19e–07	33.766	9.81e–05
8	18.5688	0.029118	170.475	4.91e–32
9	27.565	0.001126	21.602	0.010226
10	17.2289	0.045249	21.298	0.01139

Для повноти картини також було застосовано критерій Краскела–Уолліса попарно до кожного з розподілів. Результати порівняння наведені на рисунках 5.4–5.5.

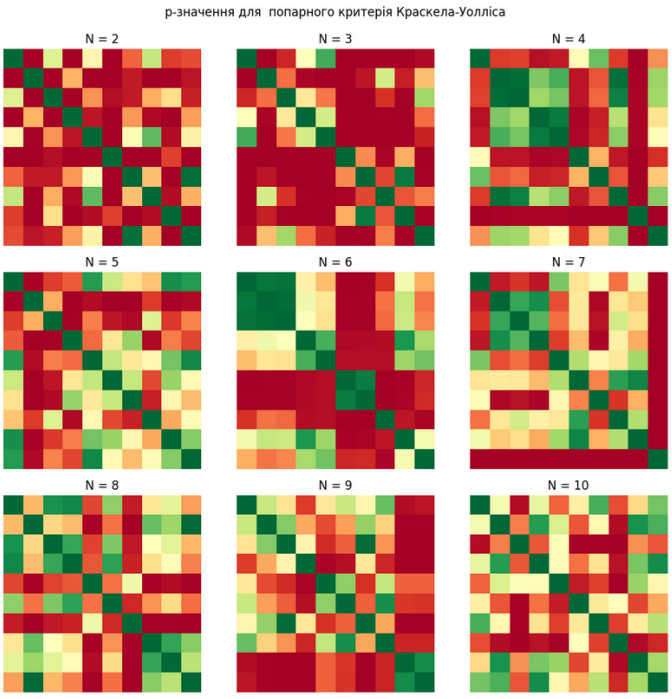

Рис. 5.4. p–значення для попарного критерія Краскела–Уолліса для статистики різниць в підписах

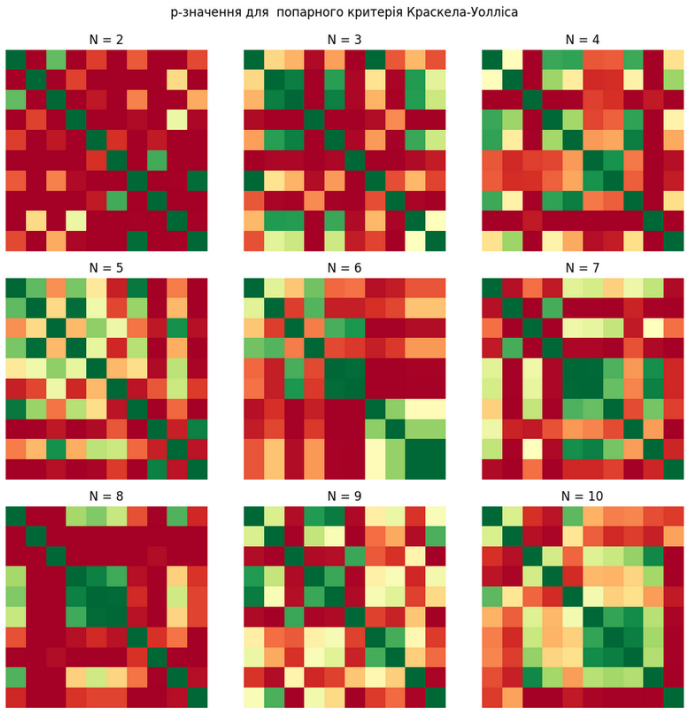

Рис. 5.5. p–значення для попарного критерія Краскела–Уолліса для статистики успішної атаки

З таблиці 5.5 та рисунків 5.4–5.4 можливо зробити висновок, що для N=10 розподіли задовольняють критерію Краскела–Уолліса для рівня значущості 0.01, проте у загальному випадку розподіли для різних ключів є різними і очікувана кількість підписів залежить від ключа.

Щоб обчислити очікувану кількість підписів для випадково обраного ключа можливо обчислити для усіх ключів очікуване значення і усереднити. Такі середні значення будуть розподілені нормально, отже можливо для отриманих значень застосувати t–критерій Стьюдента, щоб оцінити найбільш ймовірне значення кількості підписів для кожного з ключів.

За допомогою t–критерію Стьюдента були обчислені довірчі інтервали для кожного з ключів [81]. Очікувані середні значення були занесені до таблиці 5.6.

Статистика різниць в підписах та статистика успішної атаки очікувано відрізняються. Так, для криптографічно значущих параметрів для отримання різниці в підписах необхідно близько 30000 підписів, у той час як для проведення атаки необхідно аж 50000 підписів.

Таблиця 5.6.

Очікувана середня кількість підписів

Ключ	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
Статистика різниць в підписах									
1	40386	52424	25984	28728	27704	34612	34276	33828	33222
2	23202	31346	33877	46674	29318	26374	39661	28854	29877
3	47154	44637	37725	35368	29782	28839	34582	28055	22245
4	26125	47474	33517	25336	30066	26100	33067	25390	30593
5	39963	49803	34086	32891	31642	36444	24367	37016	28000
6	65627	22292	26366	29067	45190	34893	32164	35695	40353
7	33151	24164	27038	27029	45219	38335	25691	29825	30165
8	36703	32555	29579	34887	26207	30686	33477	36803	42870
9	48764	28125	46406	28709	29673	39975	38210	44872	31468
10	33968	42215	32540	32820	35353	65048	37617	44665	34608
Статистика успішної атаки									
1	76945	109577	49630	48050	59888	55900	50185	47478	50286
2	183296	89302	55542	48871	52282	81730	175649	41240	54094
3	78895	86359	89954	51729	69551	43473	102241	57725	39488
4	135106	142008	56461	41662	58987	80722	40079	42062	35699
5	58165	84637	58854	54557	71641	51648	44177	43061	45783
6	45930	62435	74446	54927	68080	53771	42342	67335	46750
7	92645	113691	70199	43453	47131	48301	57978	54718	40465
8	44826	116400	52626	76233	44067	59914	75243	50474	47772
9	164514	100128	108978	55696	51824	53938	45177	55511	45871
10	86542	80268	54442	78320	51824	38122	34715	36177	62573

На рисунках 5.6–5.7 зображені 95% довірчий інтервали для очікуваної кількості підписів відповідних статистик.

Доволі цікаво, що залежність кількості підписів від N є не лінійною. Це може бути поясненим впливом інших загальносистемних параметрів на ймовірність різниць в підписах.

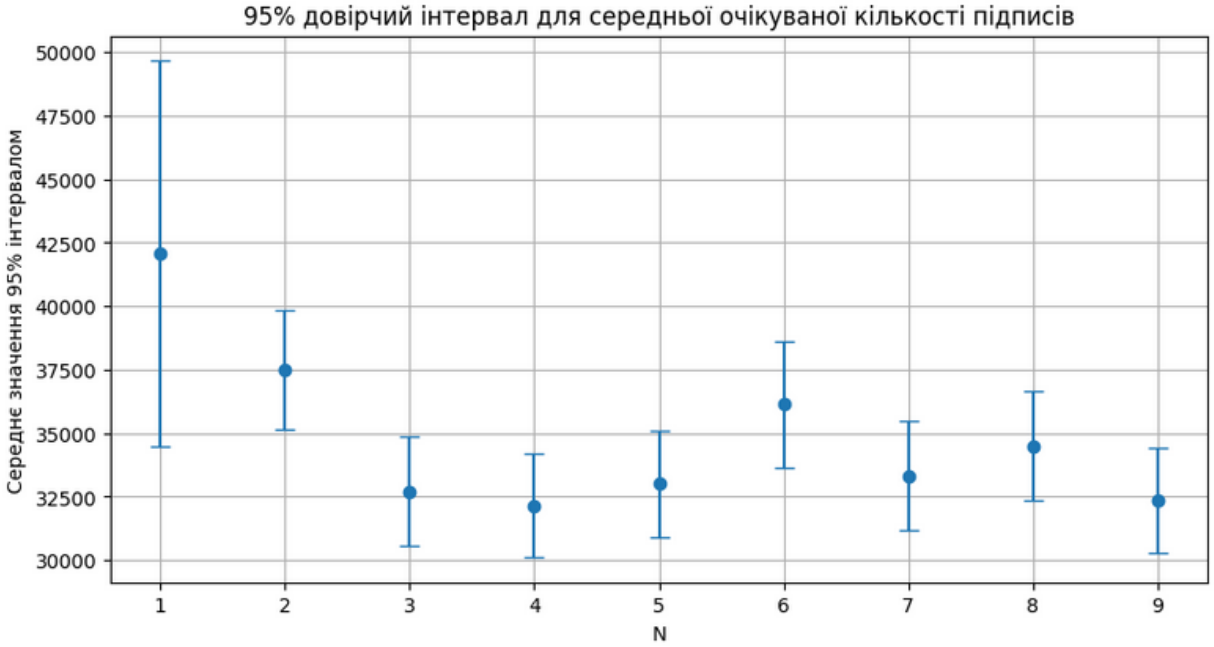

Рис. 5.6. 95% довірчий інтервали для очікуваної кількості підписів для статистики різниць в підписах

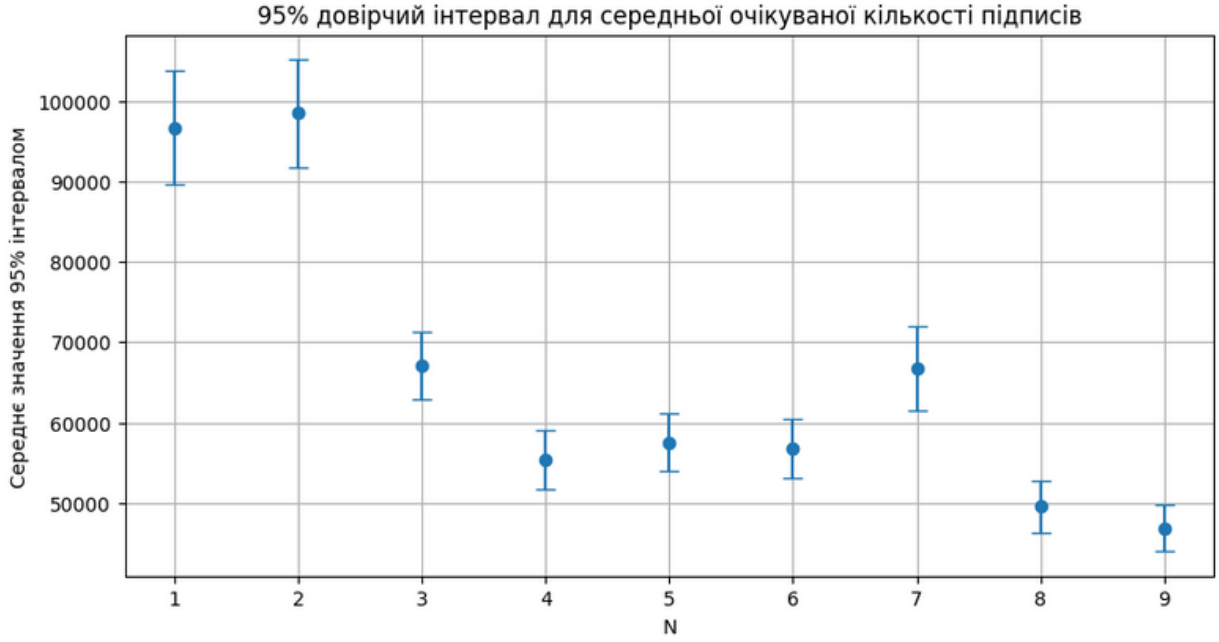

Рис. 5.7. 95% довірчий інтервали для очікуваної кількості підписів для статистики успішної атаки

До отриманих оцінок були застосовані лінійна, квадратична та кубічна регресивні моделі, для оцінки якості моделей було обчислено коефіцієнт детермінації R^2. На рисунках 5.8–5.9 наведені відповідні моделі.

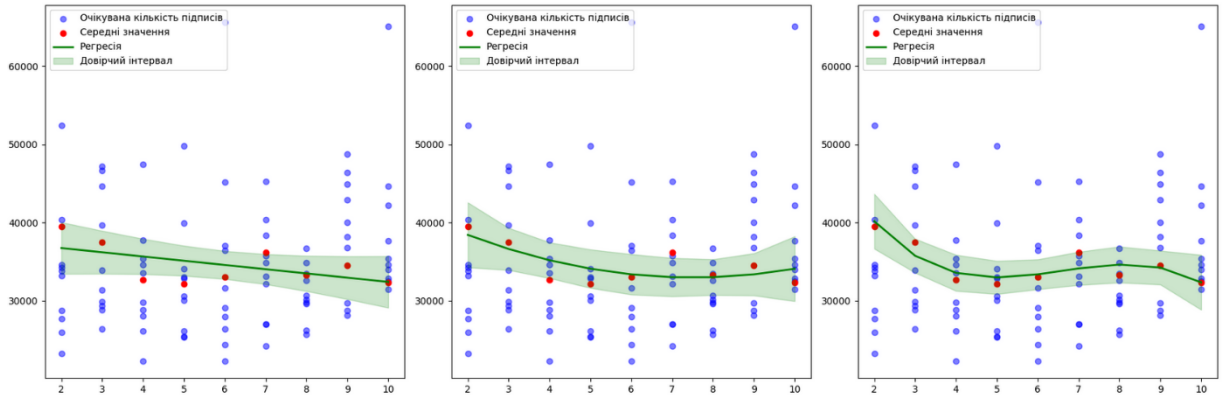

Рис. 5.8. Застосування регресивних моделей до статистики різниць в підписах

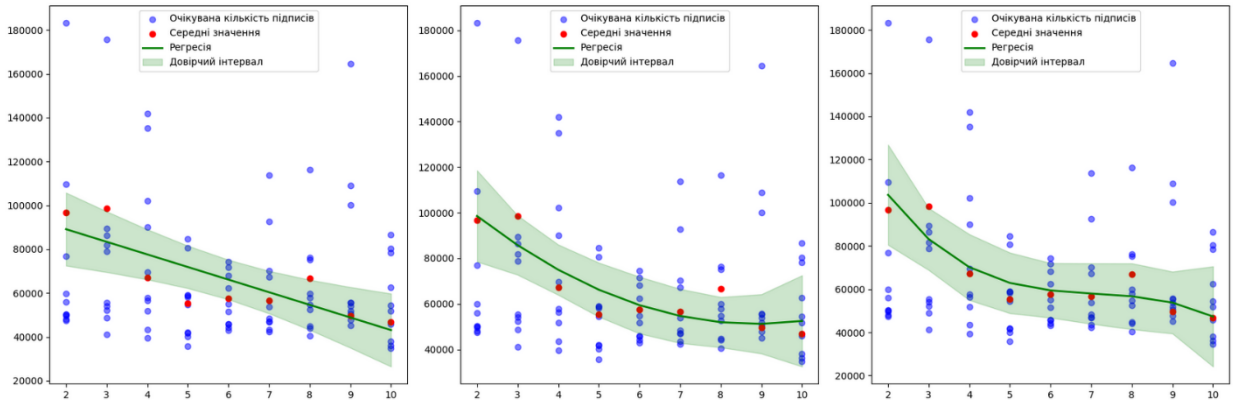

Рис. 5.9. Застосування регресивних моделей до статистики успішної атаки

Статистику різниць в підписах добре описує тільки кубічна модель. Лінійна модель має коефіцієнт детермінації 0.3277, що показує погану відповідність даних, квадратична модель має коефіцієнт детермінації 0.5138, що, хоча і вважається задовільним, проте на самій межі. Кубічна модель добре описує данні.

Варто зауважити, що для криптографічно значущих параметрів (N=9,N=10) оцінки усіх моделей збігаються, незважаючи на різні коефіцієнти детермінації.

Для статистики успішної атаки усі моделі показують високі коефіцієнти детермінації, проте, зважаючи, що пік для N=8 виникає для багатьох ключів незалежно, то доцільно застосовувати все ж кубічну модель.

В таблиці 5.7 занесено отриману середню кількість підписів для кожної з моделей.

Таблиця 5.7.

Очікувана кількість підписів згідно до регресивних моделей

	Статистика різниць в підписах			Статистика успішної атаки		
Модель	Лінійна	Квадратична	Кубічна	Лінійна	Квадратична	Кубічна
R^2	0.3277	0.5138	0.7980	0.6838	0.7909	0.8368
N=2	36740	38421	40158	89141	98523	103660
N=3	36199	36619	35750	83388	85733	83164
N=4	35657	35177	33563	77634	74953	70183
N=5	35116	34095	32978	71881	66184	62882
N=6	34574	33374	33374	66127	59426	59426
N=7	34033	33012	34129	60374	54677	57980
N=8	33491	33011	34625	54620	51940	56710
N=9	32950	33370	34239	48867	51212	53781
N=10	32408	34089	32351	43113	52495	47358

Аналіз коду показав, що розбіжності в порядку виконання операцій та заміна операцій є при реалізації функцій ffSampling_fft_dyntree та ffSampling_fft при $logn = 2$, а саме при реалізації операцій split (функція poly_split_fft) та merge (функція poly_merge_fft) в функції ffSampling_fft_dyntree та їх розгортки в функції ffSampling_fft.

При реалізації операції split в функції ffSampling_fft_dyntree для $logn = 2$ виконується операція множення комплексних чисел, яка потребує 4 операції множення і дві операції додавання, а потім операція ділення навпіл результату множення, тобто при множенні комплексних чисел $x + iy$ та $z = iu$ отримаємо число $v + iw$, де $v = x \cdot z - y \cdot u$, $w = x \cdot u + y \cdot z$, а після ділення пополам $v = v/2$, $w = w/w$.

В функції ffSampling_fft враховується, що друге комплексне число $z + iu$ для $logn = 2$ дорівнює $1/\sqrt{2} - i/\sqrt{2}$, тоді в результаті множення отримаємо $v = x/\sqrt{2} + y/\sqrt{2}$ та $w = y/\sqrt{2} - x/\sqrt{2}$, а з урахуванням ділення пополам отримаємо: $v = (x + y)/\sqrt{8}$, $w = (x - y)/\sqrt{8}$, тобто замість 4 операцій множення застосовують дві операції множення на константу.

Аналогічно для операції merge замість множення двох комплексних чисел враховують, що друге комплексне число $1/\sqrt{2} + i/\sqrt{2}$ і тоді результат множення $v = (x - y)/\sqrt{2}$, $w = (x + y)/\sqrt{2}$, , замість 4 операцій множення застосовують дві операції множення на константу. Таким чином в варіанті функції ffSampling_fft застосовують не тільки більш ефективні з боку часу виконання, а і більш точні обчислення за рахунок зменшення кількості операцій множення. Після застосування в функціях poly_split_fft, poly_merge_fft більш ефективного методу для $logn < 3$ ефект отримання різних ЕП зник.

5.3. Оцінка безпеки електронного підпису CRYSTALS–Dilithium

Для опису ЕП Вершина введемо наступні позначення. Нехай R_q– кільце $\mathbb{Z}_q[X]/(X^n + 1)$, де q– просте число. $R_{\{a_1,\dots,a_m\}}$– множина усіх поліномів в R_q, що мають коефіцієнти з множини $\{a_1, \dots, a_m\}$. $R_{\{a_1,\dots,a_m\},count}$ – множина усіх поліномів $R_{\{a_1,\dots,a_m\}}$, що мають рівно $count$ ненульових елементів. Для парного (непарного) цілого числа a позначимо як $r' = mod^{\pm} a$ – унікальний елемент r' у діапазоні $-\frac{a}{2} < r' \leq a/2$. Аналогічно, позначимо $r' = r\ mod^{\pm} a$ для унікального елемента r' у діапазоні $0 \leq r' < a$. Для елемента $w \in \mathbb{Z}_q$ введемо норму як $\|w\|_\infty = |w\ mod^{\pm} q|$. l_∞ та l_2 норми для елементів $w = w_0 + w_1 X + \dots + w_{n-1} X^{n-1} \in R_q$. введемо наступним чином:

$$\|w\|_\infty = \max_i \|w_i\|, \|w\|_2 = \sqrt{\|w_0\|_\infty^2 + \dots + \|w_{n-1}\|_\infty^2} \qquad (5.8)$$

Для вектора $w \in \left(R_q\right)^k$ норму введемо аналогічно.

Множина S_η визначається як множина усіх елементів $w \in R_q$, для яких виконується $\|w\|_\infty \leq \eta$. ЕП Вершина використовує псевдовипадкову функцію H:

$$H: \{0,1\}^* \to R_{\{-1,0,1\},w_c} \tag{5.9}$$

Схема електронного підпису CRYSTALS–Dilithium має в основі перетворення Фіата–Шаміра з перериваннями (англ. Fiat–Shamir with Aborts). Наведемо спрощену версію схеми для подальшого аналізу.

<table>
<tr>
<td>Генерація Ключів
1. $A \leftarrow R_q^{k \times l}$
2. $(s_1, s_2) \leftarrow S_\eta^l \times S_\eta^k$
3. $t = As_1 + s_2$
4. $pk = (A, t)$
5. $sk = (A, t, s_1, s_2)$
6. $return\ (pk, sk)$</td>
<td>Вироблення підпису (sk, M)
1. $z = \perp$
2. $while\ z == \perp$
3. $y \leftarrow S_{\gamma_1 - 1}^l$
4. $w_1 = Ay|^{2\gamma_2}$
5. $c = H(M||w_1)$
6. $z = y + cs_1$
7. $if\ \|z\|_\infty \geq \gamma_1 - \beta\ or$
8. $\left\|(Ay - cs_2)|_{2\gamma_2}\right\| \geq \gamma_2 - \beta$
9. $then\ z = \perp$
10. $return\ \sigma = (z, c)$</td>
</tr>
<tr>
<td colspan="2">Перевірка підпису $\left(pk, M, \sigma = (z, c)\right)$
1. $w_1' = (Az - ct)^{2\gamma_2}$
2. $return\ [\![\|z\|_\infty < \gamma_1 - \beta]\!]\ and\ [\![c = H(M||w_1')]\!]$</td>
</tr>
</table>

У цій схемі параметрами безпеки є

- Параметри поля (n, q), які визначають поле $R_q = \mathbb{Z}_q[X]/(X^n + 1)$
- Параметр d, що визначає кількість біт, які будуть відкинути з коефіцієнтів вектору t.
- Параметр τ, що визначає кількість ± 1 у поліномі c
- Параметр γ_1, що визначає діапазон коефіцієнтів у векторі y
- Параметр γ_2, що визначає параметри округлення

- Параметри (k, l), що визначають розмірність матриці A
- Параметр η, що визначає діапазон коефіцієнтів в таємному ключі
- Параметр β, що визначає
- Параметр ω

Параметри Crystals–Dilithium зведені у таблиці 5.8.

Таблиця 5.8. Параметри Crystals–Dilithium

Рівень безпеки	2	3	5
(q, n)	(8380417, 512)	(8380417, 512)	(8380417, 512)
d	13	13	13
τ	39	49	60
γ_1	2^{17}	2^{19}	2^{19}
γ_2	$(q-1)/88$	$(q-1)/32$	$(q-1)/32$
(k, l)	(4,4)	(6,5)	(8,7)
η	2	4	2
β	78	196	120
ω	80	55	75

5.3.1. Оцінка безпеки в моделі EUF–CMA

Оскільки CRYSTALS–Dilithium був у центрі уваги конкурсу NIST, то різними авторами для нього був проведений детальний аналіз безпеки у моделі квантового випадкового оракула [89].

Найбільш детальний аналіз був проведений у роботі [87]. Безпека CRYSTALS–Dilithium окрім стандартних проблем Module–SIS та Module–LWE також ґрунтується на не стандартній проблемі SelfTargetMSIS.

Сутність проблеми SelfTragetMSIS полягає у тому, щоб знайти вектор y, для якого виконується

- $\|y\|_\infty \leq \gamma$ для заданного γ

- $H([I|A] \cdot y||M) = c$ для заданих $A \in R_q^{m \times k}, M, c$

Якщо деякий супротивник A має перевагу $Adv_{m,k,2\gamma}^{MSIS}$ у вирішенні проблеми MSIS, то супротивник B, якого перевага увирішенні проблеми SelfTargetMSIS буде

$$Adv_{H,m,k,\gamma}^{SelfTargetMSIS}(B) \approx \sqrt{Adv_{m,k,2\gamma}^{MSIS}(A)/Q_H} \quad (5.10)$$

Де Q_H – кількість запитів до квантового випадкового оракула H.

Для CRYSTALS–Dilithium перевага супротивника A у грі SUF–CMA складає

$$Adv_{Dilithium}^{SUF-CMA} \leq Adv_{k,l,D}^{MLWE} + Adv_{H,k,l+1,\zeta_1}^{SelfTargetMSIS} + Adv_{k,l,\zeta_2}^{MSIS} + 2^{-\alpha+1} \quad (5.11)$$

Де α є мінімальною ентропією схеми,

$\zeta_1 = \max\{\gamma_1 - \beta, 2\gamma_2 + 1 + 2^{d-1} \cdot \rho\}$,

$\zeta_2 = \max\{2(\gamma_2 - \beta), 4\gamma_1 + 2\}$.

Мінімальну ентропію α для CRYSTALS–Dilithium можливо розрахувати як

$$\alpha > nl \cdot log\left(min\left\{\frac{q}{(4\gamma_1+1)(4\gamma_2+1)}\right\}, 2\gamma_2 - 1\right) \quad (5.12)$$

Якщо $2\gamma_1, 2\gamma_2 < \sqrt{q/2}$ та $l \leq k$.

5.3.2. Оцінки безпеки

З формул (5.10) та (5.11) випливає, що для оцінки безпеки схеми необхідно оцінити складність вирішення проблеми MLWE з параметрами k,l,D, проблем MSIS з параметрами k, l, ζ_1 та $k, l+1, \zeta_2$. І мінімальна ентропія (3) схеми повинна перевищувати цільовий рівень безпеки.

Оскільки алгебраїчну структуру проблем MLWE та MSIS для криптографічних наборів параметрів невідомо як використовувати, то можливо розглядати відповідні проблеми LWE та SIS.

Аналіз криптостійкості CRYSTALS–Dilithium від атак, що базуються на проблемі LWE (Learning With Errors), представлений у таблиці 5.9. Дані охоплюють оцінку складності атак для різних наборів параметрів у двох

моделях: GSA (Geometric Series Assumption) та симулятор редукції. Результати демонструють чіткий взаємозв’язок між розміром параметрів Dilithium та рівнем безпеки.

Таблиця 5.9.

Оцінка складності атаки вкладення

Набір параметрів	Складність атаки (біт, GSA)	Розмір блоку редукції	Складність атаки (біт, Симулятор)	Розмір блоку редукції
Dilithium2	124/113	424	134/122	458
Dilithium3	183/166	625	192/175	659
Dilithium5	252/229	864	264/240	904

Набір параметрів Dilithium2 забезпечує найнижчу складність атак у порівнянні з іншими конфігураціями. У моделі GSA рівень складності оцінюється на 124 біти для класичних атак і знижується до 113 біт у квантовому сценарії. Симулятор редукції, враховуючи додаткові аспекти редукції, показує дещо вищу складність атак: 134 біти (класична безпека) та 122 біти (квантова). Розмір блоку редукції для цієї конфігурації також менший у порівнянні з Dilithium3 і Dilithium5, що вказує на нижчі вимоги до обчислювальних ресурсів.

Dilithium3 демонструє суттєве підвищення рівня безпеки. Для класичних атак складність у моделі GSA оцінюється на 183 біти, а для квантових – на 166 біт. Аналогічно, симулятор редукції покращує ці показники до 192 біт (класична безпека) та 175 біт (квантова). Розмір блоку редукції збільшується до 625 (GSA) та 659 (симулятор), що свідчить про зростання обчислювальних витрат, але також про значно вищий рівень стійкості.

Dilithium5 забезпечує найвищий рівень криптостійкості серед представлених конфігурацій. У моделі GSA складність атак становить 252 біти для класичних атак і 229 біт для квантових. У симуляторі редукції ці значення підвищуються до 264 біт (класична безпека) та 240 біт (квантова). Розмір блоку редукції досягає найвищих значень – 864 у моделі GSA та 904 у симуляторі, що

вказує на необхідність значних обчислювальних ресурсів для проведення атаки, особливо при великих розмірностях.

Результати свідчать про поступове підвищення рівня безпеки зі збільшенням розмірності параметрів, причому симулятор редукції завжди демонструє більшу складність атак у порівнянні з GSA. Це підтверджує, що симулятор редукції є точнішим інструментом оцінки безпеки, хоча і потребує більших обчислювальних ресурсів. На рисунку 5.10 візуалізовано отримані результати.

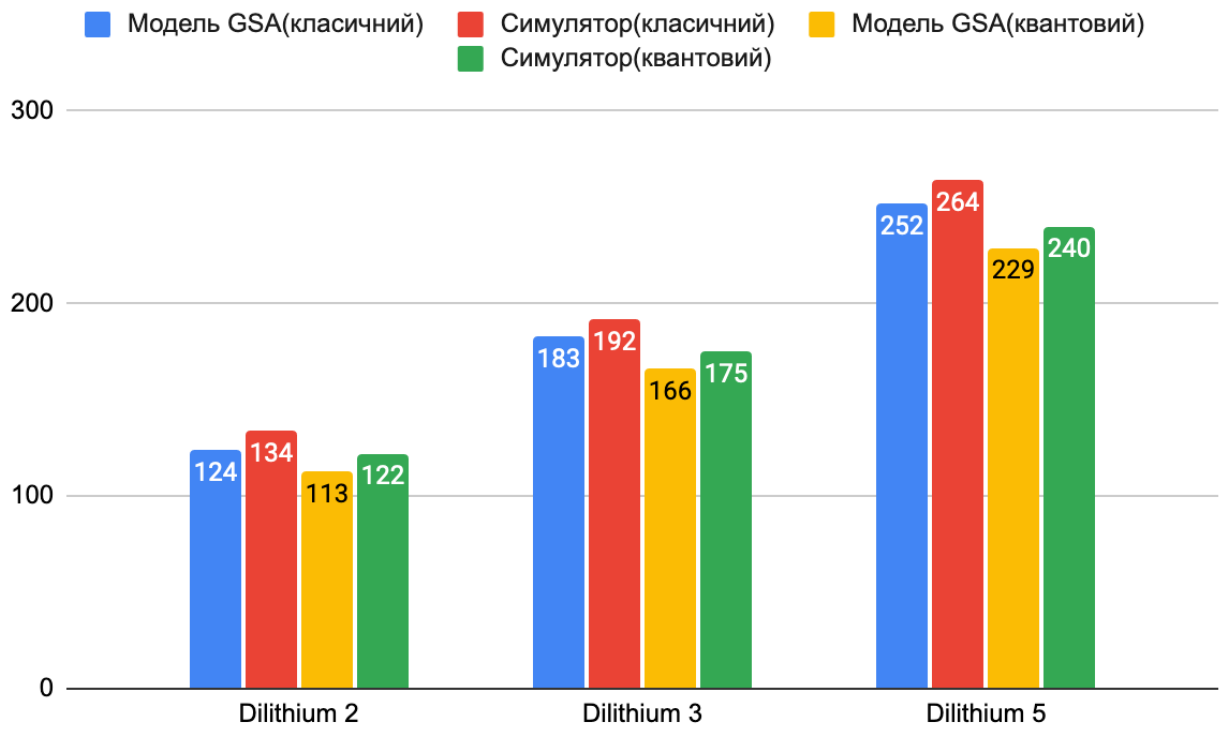

Рис. 5.10. Оцінка безпеки CRYSTALS–Dilithium від атак на LWE

Оцінка криптостійкості CRYSTALS–Dilithium при зведенні до задачі MSIS (Module Short Integer Solution) є важливим аспектом аналізу безпеки цієї криптографічної схеми. У таблиці 5.10 представлені результати для параметрів Dilithium2, Dilithium3 і Dilithium5, а на рисунку 5.11 дані візуалізовані у вигляді порівняння рівнів безпеки в моделі GSA (Geometric Series Assumption) і симуляторі редукції.

Таблиця 5.10.

Оцінки складності проблеми MSIS

	MSIS	Блок редукції	MSIS	Блок редукції
Dilithium2	123/113	423	131/120	450
Dilithium3	186/170	638	195/178	670
Dilithium5	266/242	911	277/253	951

Для параметрів Dilithium2 спостерігається найнижчий рівень безпеки серед представлених конфігурацій. У моделі GSA складність атаки оцінюється на 121 біт для класичної стійкості та 111 біт для квантової. Симулятор редукції демонструє дещо вищі показники: 128 біт (класична безпека) і 117 біт (квантова). Розмір блоку редукції при цьому незначно зростає: 417 для GSA та 441 для симулятора. Ці показники вказують на те, що Dilithium2 підходить для систем із помірними вимогами до безпеки, де важлива ефективність і помірні обчислювальні витрати.

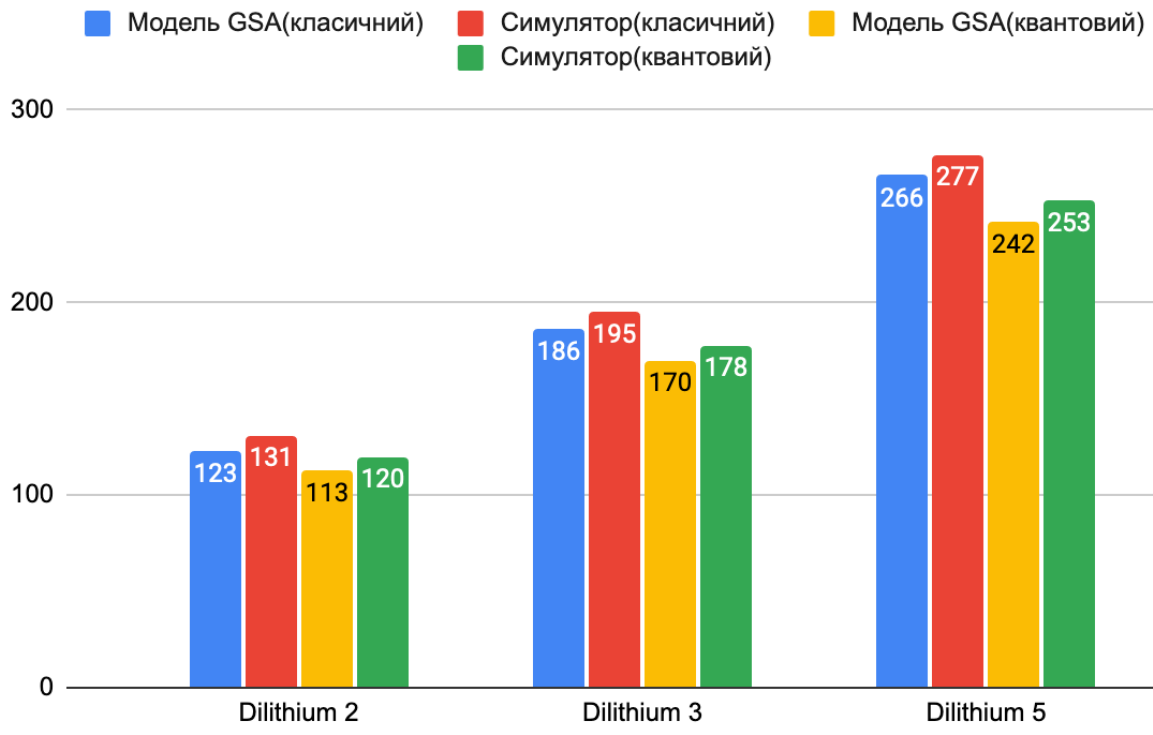

Рис. 5.11. Порівняння атак на SIS

Dilithium3 демонструє суттєве покращення рівня стійкості. У моделі GSA складність атаки становить 175 біт для класичної стійкості та 160 біт для квантової. Для симулятора редукції ці показники підвищуються до 186 і 170 біт відповідно. Розмір блоку редукції також збільшується до 602 (GSA) та 640 (симулятор), що вказує на потребу в більших обчислювальних ресурсах. Dilithium3 забезпечує збалансований рівень захисту, що робить його придатним для застосування у системах із підвищеними вимогами до безпеки.

Для найвищого набору параметрів, Dilithium5, забезпечується максимальна стійкість до атак. У моделі GSA складність оцінюється на рівні 253 біт для класичних атак і 231 біт для квантових. Симулятор редукції, як завжди, підвищує ці показники до 266 біт (класична безпека) і 242 біт (квантова). Розмір блоку редукції досягає найбільших значень – 869 для GSA і 913 для симулятора. Це вказує на високу обчислювальну складність, але одночасно й на максимальний рівень захисту, що робить Dilithium5 придатним для використання в критичних системах з довготривалими вимогами безпеки.

У таблиці 5.11 та на рисунку 5.12 наведені аналогічні результати для проблема SelfTargetMSIS. При оцінці SelfTargetMSIS проблема була зведена до MSIS і було застосовано метод, що був запропонований в розділі 3.

З таблиці 5.11 видно, що спостерігаються такі ж тенденції як і у таблиці 5.10.

Таблиця 5.11.

Оцінка складності атаки для SelfTargetMSIS

	SelfTargetMSIS	Блок редукції	SelfTargetMSIS	Блок редукції
Dilithium2	121/111	417	128/117	441
Dilithium3	175/160	602	186/170	640
Dilithium5	253/231	869	266/242	913

Порівняння моделей GSA та симулятора редукції показує, що симулятор стабільно демонструє вищі рівні безпеки для всіх параметрів. Це свідчить про його більшу точність у моделюванні реальних умов атак, зокрема врахування додаткових факторів редукції. Проте це також супроводжується збільшенням обчислювальних витрат, що варто враховувати при виборі параметрів для практичного застосування.

Для моделі GSA оцінки збігаються з оцінками авторів. Проте, при використанні симулятору редукції видно вплив алгебраїчної структури. Це показує, що запропонована у третьому розділі модель є адекватною та узагальнює раніше відомі результати.

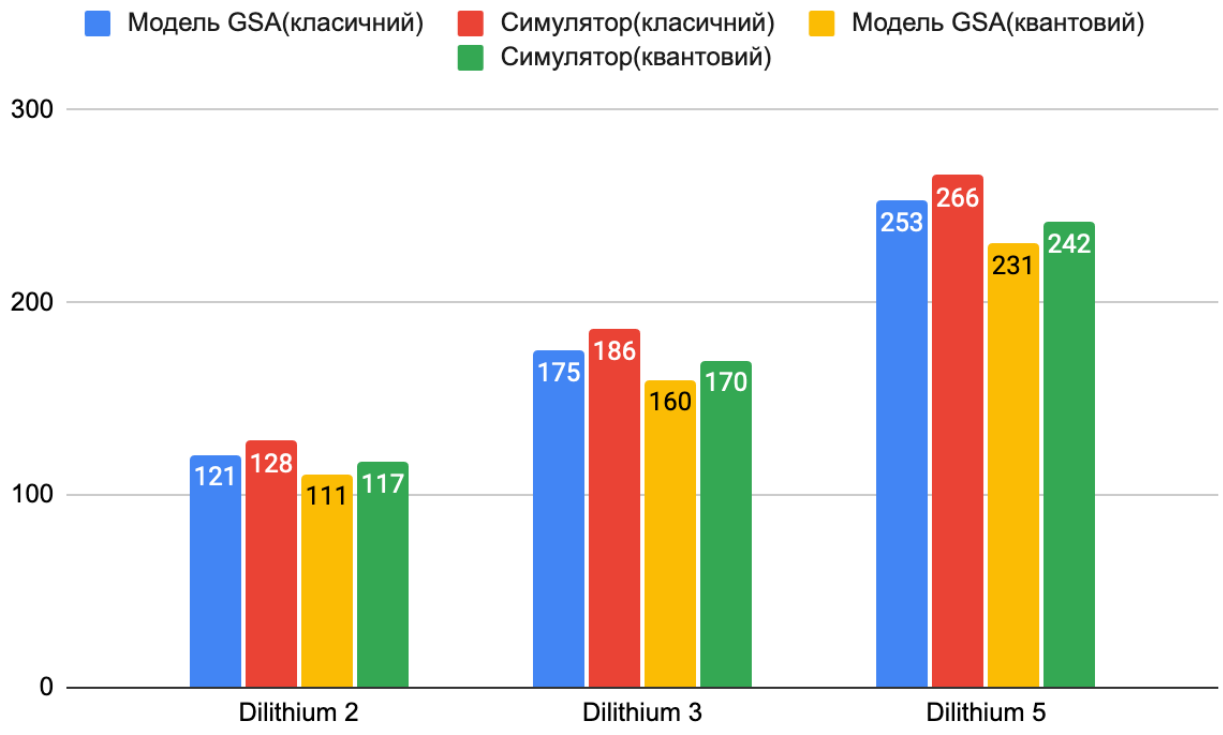

Рис. 5.12. Оцінки складності атак SelfTargetMSIS

5.4. Висновки до розділу

1. Були уточнені оцінок безпеки електронних підписів Falcon та Crystals–Dilithium. В залежності від набору параметрів різниця між оцінками у моделі GSA та моделі, що враховує алгебраїчну структуру q–арних решіток, складає 20–

30 біт безпеки. При чому, уточнені оцінки показують, що існуючі атаки є менш ефективними, ніж припускалося при використанні моделі GSA.

2. Ймовірність проведення атаки відновлення ключів на Falcon є обернено пропорційною до очікуваної кількості підписів. Очікувана кількість підписів для успішної атаки складає 51212–53781 для Falcon512 та 47358–52495 для Falcon1024.

3. Зі збільшенням параметра log$n$ ймовірність проведення атаки відновення ключа збільшується. Це пояснюється тим, що збільшується кількість обчислень з плаваючою крапкою. Отримана залежність необхідної кількості підписів від log$n$ є не монотонно зростаючою. Проте, оскільки різниця між оцінками квадратичної та кубічної регресії потрапляє в 95 % довірчий інтервал для середніх значень, то можна нехтувати цим ефектом і вважати, що ймовірність зростає монотонно.

4. Отриману оцінку кількості підписів можливо використовувати як обмеження кількості підписів на одній парі ключів в інфраструктурі відкритих ключів, щоб захиститися від атаки. Це підвищить безпеку електронного підпису Falcon у якісному сенсі – атака відновлення ключів стає не можливою при обмеженнях на кількість вироблених підписів.

ВИСНОВКИ

В дисертаційній роботі розв’язана актуальна задача аналізу та розробки методів та моделей для підвищення захищеності асиметричних криптографічних систем на решітках від існуючих та потенційних атак та обгрунтування на їх основі національних стандартів асиметричних криптоперетворень електронного підпису та інкапсуляції ключів на решітках.

Для вирішення поставленої наукової задачі використано методи комп’ютерного моделювання, теорії ймовірностей та математичної статистики. Чисельні розрахунки на обчислюваній системі виконувалися з використанням середовища розробки Visual Studio Code з процесором 1.6 GHz Intel Core i5 та обсягом оперативної пам’яті 16 ГБ на базі Ubuntu 22.04. Для редукції решіток використовувалася бібліотека fpylll на мові програмування Python. Для оцінки безпеки загальносистемних використовувалася мова програмування c++. Для роботи з довгою арифметикою використовувалася бібліотека mpfr на мові програмування C++.

Основні наукові та практичні результати, отримані в дисертації.

1. Вперше виконано кількісне порівняння точності моделей редукції решіток із застосуванням метрики середньоквадратичної помилки для моделі GSA (Geometric Series Assumption) та симуляторів редукції решіток. Попередні дослідження фокусувалися на якісних або суто теоретичних оцінках якості роботи моделей. Отримані оцінки дозволяють кількісно оцінювати якість роботи симуляторів в залежності від параметрів решіток для оцінки захищеності від класичних та квантових атак.

2. Вперше було отримано узагальнений доказ IND–CCA безпеки перетворень, що використовуються в стандарті ДСТУ 8961:2019, у моделі квантового випадкового оракула. Попередні дослідження не вивчали IND–CCA безпеку перетворень ДСТУ 8961:2019 у моделі квантового випадкового оракула.

3. Удосконалено методику оцінювання складності криптографічної задачі SIS (Shortest Integer Solution), що відрізняється від існуючих тим, що у

даній методиці враховується алгебраїчна структура решіток під час аналізу процесів редукції для оцінки параметрів та характеристик на їх основі атак.

4. Отримали подальший розвиток обґрунтування оцінок атаки на відновлення ключів для алгоритмів електронного підпису на основі решіток, що використовують обчислення з плаваючою крапкою, спираючись на аналіз статистичних даних.

Достовірність результатів дисертаційної роботи забезпечується адекватністю припущень, які лежать в основі проведених наукових досліджень, а також коректним застосуванням відомих математичних методів. Результати проведених числельних розрахунків узгоджуються з отриманими теоретичними висновками.

Значення наукових результатів дисертації для теорії полягає в тому, що отримані результати мають універсальний характер, що дозволяє використовувати їх в подальшому при дослідженні безпеки широкого класу криптографічних перетворень на решітках.

Практичне значення роботи. Розроблено програмні реалізації, які дозволяють знаходити оцінки безпеки проблем з теорії решіток. Особливістю розроблених програмних реалізацій є їх модульність. Вони дозволяють використовувати як симулятори, так і класичні моделі оцінки та швидко розширювати кодову базу для нових методів. Розроблені програмні засоби можуть бути корисними як при оцінці безпеки криптографічних перетворень, так і при розробці нових методів оцінки безпеки, оскільки кодова база розроблялася з врахуванням можливості подальшого узагальнення існуючих методів.

Уточнено оцінки безпеки криптографічних перетворень на решітках, які підтверджують рівні стійкості квантово–стійких стандартів електронного підпису (ДСТУ 9212:2023) та механізмів інкапсуляції ключів (ДСТУ 8961:2019), що обґрунтовує їх застосування як національних стандартів.

Результати дисертаційних досліджень впровадженні у Приватному акціонерному товаристві «Інститут інформаційних технологій», м. Харків та

були використані при обчисленні вхідних та вихідних тестових векторів стандартів ДСТУ 8961:2019, ДСТУ 9212:2023.

Математичні моделі та аналітичні співвідношення знайшли практичне застосування в ХНУ імені В. Н. Каразіна на кафедрі БІСТ в дисциплінах першого рівня вищої освіти “Прикладна криптологія”, другого рівня освіти “Криптографічні методи в кібербезпеці” та третього рівня освіти «Математичні методи в кібербезпеці» при проведенні лабораторних робіт.

Висновки та рекомендації по науковому та практичному використанню наукових результатів.

1. Отримані експериментальні обґрунтування точності існуючих асимптотичних оцінок фактора Ерміта дозволяють стверджувати, що евритика Гауса, з якої були отримані асимптотичні оцінки фактора Ерміта, добре працює вже на малих розмірностях решіток (порядку 30 – 100), а отже експерименти на цих розміростях можуть давати інформацію про поведінку великих криптографічно значущих розмірностей.

2. Симулятори враховують багато особливостей редукції решіток, що дозволяє отримувати точніші оцінки для атак, що використовують редукцію решіток. Проведене порівняння показує, що симулятор Альбрехта–Лі дає найточніші оцінки.

3. Врахування алгебраїчних особливостей решіток дає збільшення оцінок безпеки, тобто показує, що атаки працюватимуть гірше. Для атаки вкладення зміна є не сильною, проте для атак декодування форма базису решітки сильніше впливає на оцінки і ефективність атак декодування сильно падає з ростом розмірності решіток.

4. Отриманий доказ безпеки ДСТУ 8961:2019 у моделі квантового випадкового оракула дозволяє гарантувати, що не існує квантових чи класичних супротивників, що змогли б провести атаку з адаптивно підібраними шифротекстами на механізм інкапсуляції ключів без вирішення проблеми NTRU. Оскільки для інших криптографічних систем на міжнародному рівні такі докази існують, то отриманий доказ закриває прогалину в безпеці ДСТУ 8961:2019.

5. Запропонована атака відновлення ключів на Falcon показує необхідність стандартизації не лише алгоритмів, але й конкретного порядку обчислень. Наразі проблема використання обчислень з плаваючою точкою залишається не вирішеною повністю.

6. Розроблені програмні реалізації дозволяють робити оцінку безпеки криптографічних перетворень на решітках і буде корисною навіть при розробці нових методів оцінки безпеки, оскільки вона є модульною і її компоненти можуть бути замінені. Реалізація також використовує багатопоточність, що значно пришвидшує час обчислень.

7. Основні наукові та практичні результати дисертаційної роботи реалізовані НДР «Квант», що виконувалися в ПАТ «Інститут Інформаційних Технологій» та університеті імені В.Н. Каразіна. Отримані результати можуть використовуватися при проведенні експертних досліджень для отримання науково обґрунтованих висновків про можливість застосування в Україні перспективних квантово–стійних алгоритмів на алгебраїчних решітках.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Rescorla, E. and Dierks, T. (2008) RFC 5246: The Transport Layer Security (TLS) protocol version 1.2, IETF Datatracker. Available at: https://datatracker.ietf.org/doc/html/rfc5246 (Accessed: 17 October 2024).

2. Lonvick, C.M. and Ylonen, T. (2006) RFC 4253: The Secure Shell (SSH) transport layer protocol, IETF Datatracker. Available at: https://datatracker.ietf.org/doc/html/rfc4253 (Accessed: 17 October 2024).

3. Frankel, S. and Krishnan, S. (2011) RFC 6071: IP Security (IPsec) and Internet Key Exchange (IKE) document roadmap, IETF Datatracker. Available at: https://datatracker.ietf.org/doc/html/rfc6071 (Accessed: 17 October 2024).

4. Wong, H.Y. (2022) Introduction to quantum computing [Preprint]. doi:10.1007/978–3–030–98339–0.

5. Shor, P.W. (1994) ‘Algorithms for quantum computation: Discrete logarithms and factoring’, Proceedings 35th Annual Symposium on Foundations of Computer Science [Preprint]. doi:10.1109/sfcs.1994.365700.

6. Moriarty, K. et al. (2016) RFC 8017: PKCS #1: RSA cryptography specifications version 2.2, IETF Datatracker. Available at: https://datatracker.ietf.org/doc/html/rfc8017 (Accessed: 17 October 2024).

7. National Institute of Standards and Technology (NIST) et al. (2024) Digital Signature Standard (DSS), NIST. Available at: https://www.nist.gov/publications/digital–signature–standard–dss–3 (Accessed: 17 October 2024).

8. Pornin, T. (2013) RFC 6979: Deterministic usage of the digital signature algorithm (DSA) and elliptic curve digital signature algorithm (ECDSA), IETF Datatracker. Available at: https://datatracker.ietf.org/doc/html/rfc6979 (Accessed: 17 October 2024).

9. Bernstein, D.J. (2009) ‘Introduction to post–quantum cryptography’, Post–Quantum Cryptography, pp. 1–14. doi:10.1007/978–3–540–88702–7_1.

10. Aaronson, S. (2005) ‘Limitations of quantum advice and one–way communication’, Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004., 9, pp. 320–332. doi:10.1109/ccc.2004.1313854.

11. Grover, L.K. (1996) ‘A fast quantum mechanical algorithm for database search’, Proceedings of the twenty–eighth annual ACM symposium on Theory of computing – STOC ’96, pp. 212–219. doi:10.1145/237814.237866.

12. Ambainis, A. (2004) ‘Quantum search algorithms’, ACM SIGACT News, 35(2), pp. 22–35. doi:10.1145/992287.992296.

13. Chailloux, A., Naya–Plasencia, M. and Schrottenloher, A. (2017) ‘An efficient quantum collision search algorithm and implications on symmetric cryptography’, Lecture Notes in Computer Science, pp. 211–240. doi:10.1007/978–3–319–70697–9_8.

14. Denisenko, D. (2021) ‘Quantum differential cryptanalysis’, Journal of Computer Virology and Hacking Techniques, 18(1), pp. 3–10. doi:10.1007/s11416–021–00395–x.

15. Chailloux, A. and Loyer, J. (2021) ‘Lattice sieving via quantum random walks’, Lecture Notes in Computer Science, pp. 63–91. doi:10.1007/978–3–030–92068–5_3.

16. Jozsa, R. (2001) ‘Quantum factoring, discrete logarithms, and the hidden subgroup problem’, Computing in Science & Engineering, 3(2), pp. 34–43. doi:10.1109/5992.909000.

17. Lomont, C. (2004) The hidden subgroup problem – review and open problems, arXiv.org. Available at: http://arxiv.org/abs/quant–ph/0411037 (Accessed: 17 October 2024).

18. Ettinger, M. and Høyer, P. (2000) ‘On quantum algorithms for noncommutative hidden subgroups’, Advances in Applied Mathematics, 25(3), pp. 239–251. doi:10.1006/aama.2000.0699.

19. Kuperberg, G. (2005) ‘A subexponential–time quantum algorithm for the dihedral hidden subgroup problem’, SIAM Journal on Computing, 35(1), pp. 170–188. doi:10.1137/s0097539703436345.

20. Krovi, H. and Rötteler, M. (2008) ‘An efficient quantum algorithm for the hidden subgroup problem over Weyl–Heisenberg groups’, Lecture Notes in Computer Science, pp. 70–88. doi:10.1007/978–3–540–89994–5_7.

21. ETSI GR QSC 006 V1.1.1 (2017–02). Available at: https://www.etsi.org/deliver/etsi_gr/QSC/001_099/006/01.01.01_60/gr_qsc006v010101p.pdf (Accessed: 17 October 2024).

22. Øygarden, M. et al. (2020) ‘Cryptanalysis of the multivariate encryption scheme EFLASH’, Lecture Notes in Computer Science, pp. 85–105. doi:10.1007/978–3–030–40186–3_5.

23. Esser, A. and Bellini, E. (2022) ‘Syndrome decoding estimator’, Lecture Notes in Computer Science, pp. 112–141. doi:10.1007/978–3–030–97121–2_5.

24. Katz, J. and Lindell, Y. (2014) Introduction to modern cryptography [Preprint]. doi:10.1201/b17668.

25. Castryck, W. and Decru, T. (2023) ‘An efficient key recovery attack on sidh’, Lecture Notes in Computer Science, pp. 423–447. doi:10.1007/978–3–031–30589–4_15.

26. Alagic, G. et al. (2022) Status report on the third round of the NIST Post–Quantum Cryptography Standardization Process [Preprint]. doi:10.6028/nist.ir.8413–upd1.

27. Gorbenko, Yu.I. and Kandii, S.O. (2022) ‘Comparison of security arguments of promising key encapsulation mechanisms’, Radiotekhnika, (210), pp. 22–36. doi:10.30837/rt.2022.3.210.02.

28. S. O. Kandii, “Comparison of security arguments for promising key encapsulation mechanisms,” in Proc. 8th Int. Conf. “Computer Modeling in High–Tech Technologies” (KMNT–2022), 2022.

29. Gorbenko, I.D. et al. (2019) ‘Algorithms of asymmetric encryption and encapsulation of keys of post–quantum period of 5 –7 stability stability levels and their applications’, Radiotekhnika, 3(198), pp. 5–18. doi:10.30837/rt.2019.3.198.01.

30. Gorbenko, I. D. et al. (2019) ‘Methods of building general parameters and keys for NTRU prime Ukraine of 5th – 7th levels of stability. product form’,

Telecommunications and Radio Engineering, 78(7), pp. 579–594. doi:10.1615/telecomradeng.v78.i7.30.

31. Goldwasser, S. and Micali, S. (1982) ‘Probabilistic Encryption & How To Play Mental Poker keeping secret all partial information’, Proceedings of the fourteenth annual ACM symposium on Theory of computing – STOC ’82, pp. 365–377. doi:10.1145/800070.802212.

32. Kandiy, S.O. (2022) ‘Analysis of DSTU 8961:2019 in random Oracle Model’, Radiotekhnika, (211), pp. 22–36. doi:10.30837/rt.2022.4.211.02.

33. Gorbenko, I.D. et al. (2021) ‘Substantiation and proposals for the selection, improvement and standardization of the post–quantum electronic signature mechanism at the national and International Levels’, Radiotekhnika, (207), pp. 5–26. doi:10.30837/rt.2021.4.207.01.

34. Boyd, C. et al. (2008) ‘Efficient one–round key exchange in the standard model’, Lecture Notes in Computer Science, pp. 69–83. doi:10.1007/978–3–540–70500–0_6.

35. Fujioka, A. et al. (2012) ‘Strongly secure authenticated key exchange from factoring, codes, and lattices’, Lecture Notes in Computer Science, pp. 467–484. doi:10.1007/978–3–642–30057–8_28.

36. Kandiy, S.O. (2023) ‘Security analysis of promising key encapsulation mechanisms in the core–SVP model’, Radiotekhnika, (212), pp. 66–84. doi:10.30837/rt.2023.1.212.06.

37. Canetti, R. and Krawczyk, H. (2001) ‘Analysis of key–exchange protocols and their use for building secure channels’, Lecture Notes in Computer Science, pp. 453–474. doi:10.1007/3–540–44987–6_28.

38. Fujisaki, E. and Okamoto, T. (2011) ‘Secure integration of asymmetric and symmetric encryption schemes’, Journal of Cryptology, 26(1), pp. 80–101. doi:10.1007/s00145–011–9114–1.

39. Dent, A.W. (2003) ‘A designer’s guide to kems’, Lecture Notes in Computer Science, pp. 133–151. doi:10.1007/978–3–540–40974–8_12.

40. Bellare, M. and Rogaway, P. (1993) ‘Random oracles are practical’, Proceedings of the 1st ACM conference on Computer and communications security – CCS ’93, pp. 62–73. doi:10.1145/168588.168596.

41. Boneh, D. et al. (2011) ‘Random oracles in a Quantum World’, Lecture Notes in Computer Science, pp. 41–69. doi:10.1007/978–3–642–25385–0_3.

42. Peikert, C. (2016) ‘A decade of lattice cryptography’, Foundations and Trends® in Theoretical Computer Science, 10(4), pp. 283–424. doi:10.1561/0400000074.

43. Nguyen, P.Q. and Vallee, B. (2011) ‘The LLL algorithm’, Lattice Basis Reduction, pp. 73–104. doi:10.1201/b11066–8.

44. LI, J. and WEI, W. (2013) ‘Slide reduction, successive minima and several applications’, Bulletin of the Australian Mathematical Society, 88(3), pp. 390–406. doi:10.1017/s0004972713000257.

45. Micciancio, D. and Walter, M. (2016) ‘Practical, predictable lattice basis reduction’, Lecture Notes in Computer Science, pp. 820–849. doi:10.1007/978–3–662–49890–3_31.

46. Aono, Y., Nguyen, P.Q. and Shen, Y. (2018) ‘Quantum lattice enumeration and tweaking discrete pruning’, Lecture Notes in Computer Science, pp. 405–434. doi:10.1007/978–3–030–03326–2_14.

47. Bernstein, D.J. et al. (2017) ‘NTRU prime: Reducing attack surface at low cost’, Lecture Notes in Computer Science, pp. 235–260. doi:10.1007/978–3–319–72565–9_12.

48. Schnorr, C.P. (2003) ‘Lattice reduction by random sampling and birthday methods’, Lecture Notes in Computer Science, pp. 145–156. doi:10.1007/3–540–36494–3_14.

49. Chen, Y. and Nguyen, P.Q. (2011) ‘BKZ 2.0: Better Lattice Security estimates’, Lecture Notes in Computer Science, pp. 1–20. doi:10.1007/978–3–642–25385–0_1.

50. Jianwei , L. and Nguyen , P.Q. (2020) A complete analysis of the BKZ lattice reduction algorithm. Available at: https://eprint.iacr.org/2020/1237.pdf (Accessed: 17 October 2024).

51. Albrecht, M. and Ducas, L. (2021) ‘Lattice attacks on NTRU and LWE: A history of refinements’, Computational Cryptography, pp. 15–40. doi:10.1017/9781108854207.004.

52. S. O. Kandii and I. D. Gorbenko, “Analysis of the Hermite factor in the BKZ algorithm on low–dimensional lattices,” CS&CS, no. 1(25), pp. 22–36, 2024. DOI: 10.26565/2519–2310–2024–1–02.

53. Bai, S., Stehlé, D. and Wen, W. (2018) ‘Measuring, simulating and exploiting the head concavity phenomenon in BKZ’, Lecture Notes in Computer Science, pp. 369–404. doi:10.1007/978–3–030–03326–2_13.

54. Zhao, Z. and Xu, G. (2023) ‘On the measurement and simulation of the BKZ behavior for q–ary lattices’, Lecture Notes in Computer Science, pp. 463–482. doi:10.1007/978–3–031–26553–2_25.

55. Kandii, S.O. and Gorbenko, I.D. (2024) ‘Assessing the influence of the algebraic structure of q–ary lattices on the complexity of cryptanalysis of problems on lattices’, Radiotekhnika, (217), pp. 79–99. doi:10.30837/rt.2024.2.217.07.

56. Albrecht, M., Bai, S. and Ducas, L. (2016) A subfield lattice attack on overstretched NTRU assumptions: Cryptanalysis of some FHE and graded encoding schemes, IACR Cryptology ePrint Archive. Available at: https://eprint.iacr.org/2016/127 (Accessed: 17 October 2024).

57. Kirchner, P. and Fouque, P.–A. (2017) ‘Revisiting lattice attacks on overstretched NTRU parameters’, Lecture Notes in Computer Science, pp. 3–26. doi:10.1007/978–3–319–56620–7_1.

58. Ducas, L. and Woerden, W. van (2021) NTRU fatigue: How stretched is overstretched?, IACR Cryptology ePrint Archive. Available at: https://eprint.iacr.org/2021/999 (Accessed: 17 October 2024).

59. van Vredendaal, C. (2016) ‘Reduced memory meet–in–the–middle attack against the NTRU Private Key’, LMS Journal of Computation and Mathematics, 19(A), pp. 43–57. doi:10.1112/s1461157016000206.

60. Guo, Q., Johansson, T. and Stankovski, P. (2015) ‘Coded–BKW: Solving LWE using lattice codes’, Lecture Notes in Computer Science, pp. 23–42. doi:10.1007/978–3–662–47989–6_2.

61. Arora, S. and Ge, R. (2011) ‘New algorithms for learning in presence of errors’, Lecture Notes in Computer Science, pp. 403–415. doi:10.1007/978–3–642–22006–7_34.

62. Bernstein , D. and Lange, T. (2021) Non–randomness of S–unit lattices – cryptology ePrint Archive. Available at: https://eprint.iacr.org/2021/1428.pdf (Accessed: 17 October 2024).

63. Lindner, R. and Peikert, C. (2011) ‘Better key sizes (and attacks) for LWE–based encryption’, Lecture Notes in Computer Science, pp. 319–339. doi:10.1007/978–3–642–19074–2_21.

64. Rückert, M. and Schneider, M. (2010) Estimating the security of lattice–based cryptosystems, IACR Cryptology ePrint Archive. Available at: https://eprint.iacr.org/2010/137 (Accessed: 02 November 2024).

65. Bi, L. et al. (2022) ‘Hybrid dual and meet–lwe attack’, Lecture Notes in Computer Science, pp. 168–188. doi:10.1007/978–3–031–22301–3_9.

66. Bai, S., Miller, S. and Wen, W. (2019) ‘A refined analysis of the cost for solving LWE via USVP’, Lecture Notes in Computer Science, pp. 181–205. doi:10.1007/978–3–030–23696–0_10.

67. Langlois, A. and Stehlé, D. (2014) ‘Worst–case to average–case reductions for module lattices’, Designs, Codes and Cryptography, 75(3), pp. 565–599. doi:10.1007/s10623–014–9938–4.

68. Albrecht, M.R. et al. (2017) ‘Revisiting the expected cost of solving USVP and applications to lwe’, Lecture Notes in Computer Science, pp. 297–322. doi:10.1007/978–3–319–70694–8_11.

69. Zhang, X., Zheng, Z. and Wang, X. (2021) ‘A detailed analysis of primal attack and its variants’, Science China Information Sciences, 65(3). doi:10.1007/s11432–020–2958–9.

70. Wunderer, T. (2018) ‘A detailed analysis of the hybrid lattice–reduction and meet–in–the–middle attack’, Journal of Mathematical Cryptology, 13(1), pp. 1–26. doi:10.1515/jmc–2016–0044.

71. S. O. Kandii and I. D. Gorbenko, “Refinement of security estimates of quantum–resistant standards of asymmetric encryption taking into account the structure of q–arry lattices,” Radiotekhnika, vol. 218, pp. ??, 2024. DOI: 10.30837/rt.2024.3.218.06.

72. Bindel, N. et al. (2019) ‘Tighter proofs of CCA security in the quantum random Oracle Model’, Lecture Notes in Computer Science, pp. 61–90. doi:10.1007/978–3–030–36033–7_3.

73. Hofheinz, D., Hövelmanns, K. and Kiltz, E. (2017) ‘A modular analysis of the Fujisaki–Okamoto Transformation’, Lecture Notes in Computer Science, pp. 341–371.doi:10.1007/978–3–319–70500–2_12.

74. Unruh, D. (2015) ‘Revocable quantum timed–release encryption’, Journal of the ACM, 62(6), pp. 1–76. doi:10.1145/2817206.

75. S. O. Kandii, “Analysis of the CPA–to–CCA transformation of DSTU 8961:2019 in the random oracle model,” Math. Comput. Modeling. Ser. Phys.–Math. Sci., no. 36, pp. 101–105, 2023. DOI: 10.15407/fmmit2023.36.101.

76. ДСТУ 8961:2019. Інформаційні технології. Криптографічний захист інформації. Алгоритми асиметричного шифрування та інкапсуляції ключів. Чинний від 21.12.2019. Вид. офіц. Київ: УкрНДНЦ, 2019. 72 с.

77. Hoffstein, J., Pipher, J. and Silverman, J.H. (1998) ‘NTRU: A ring–based public key cryptosystem’, Lecture Notes in Computer Science, pp. 267–288. doi:10.1007/bfb0054868.

78. Kandii, S.O. and Gorbenko, I.D. (2023) ‘Analysis of DSTU 8961:2019 in the quantum random Oracle Model’, Radiotekhnika, (214), pp. 7–16. doi:10.30837/rt.2023.3.214.01.

79. Bos, J. et al. (2018) ‘Crystals – Kyber: A CCA–secure module–lattice–based KEM’, 2018 IEEE European Symposium on Security and Privacy (EuroS&P) [Preprint]. doi:10.1109/eurosp.2018.00032.

80. Gentry, C., Peikert, C. and Vaikuntanathan, V. (2008) ‘Trapdoors for hard lattices and new cryptographic constructions’, Proceedings of the fortieth annual ACM symposium on Theory of computing [Preprint]. doi:10.1145/1374376.1374407.

81. Kachko, O. et al. (2024) ‘Improving protection of falcon electronic signature software implementations against attacks based on Floating Point Noise’, Eastern–European Journal of Enterprise Technologies, 4(9 (130)), pp. 6–17. doi:10.15587/1729–4061.2024.310521.

82. Moody, D. (2023) Fast fourier sampling over NTRU lattices digital signature standard [Preprint]. doi:10.6028/nist.fips.206.ipd.

83. Kosuge, H. and Xagawa, K. (2024) ‘Probabilistic hash–and–sign with retry in the quantum random Oracle Model’, Lecture Notes in Computer Science, pp. 259–288. doi:10.1007/978–3–031–57718–5_9.

84. Gorbenko, I.D. et al. (2022) ‘Generation of general system parameters for Falcon cryptosystem for 256, 384, and 512 security bits’, Telecommunications and Radio Engineering, 81(2), pp. 49–59. doi:10.1615/telecomradeng.2022037071.

85. Potii, O. et al. (2024) ‘Determining the effect of a floating point on the Falcon Digital Signature Algorithm Security’, Eastern–European Journal of Enterprise Technologies, 1(9 (127)), pp. 52–59. doi:10.15587/1729–4061.2024.295160.

86. O. V. Potii, O. H. Kachko, S. O. Kandii, and Ye. Yu. Kaptiol, “Study of the impact of floating point on the security of the Falcon digital signature algorithm,” in Proc. II Int. Sci.–Pract. Conf. “Cyberfight: Intelligence, Defense, and Counteraction”, 2024, pp. 28–30.

87. Kiltz, E., Lyubashevsky, V. and Schaffner, C. (2018) ‘A concrete treatment of Fiat–Shamir signatures in the quantum random–oracle model’, Lecture Notes in Computer Science, pp. 552–586. doi:10.1007/978–3–319–78372–7_18.

88. S. O. Kandii and I. D. Gorbenko, “Provable security of DSTU 8961:2019 in the quantum random oracle model,” in Proc. III Int. Sci.–Tech. Conf. “Communication

Systems and Technologies, Informatization, and Cybersecurity: Current Issues and Development Trends", 2023.

89. Kandiy, S.O., Ostrianska, Ye.V., Gorbenko, I.D. and Yesina, M.V. (2022) 'Classification and analysis of vulnerabilities of modern information systems from classical and quantum attacks', Radiotekhnika, 211, pp. 7–21.

ДОДАТОК А
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукова публікація у зарубіжних виданнях, що входять до міжнародних наукометричних баз Scopus та Web of Scienc:

1. Gorbenko I.D., Yesina M.V., Kandy S.O., Ostryanska Ye. V. Generation of general system parameters for Falcon cryptosystem for 256, 384, and 512 security bits // Telecommunications and Radio Engineering. 2022. Vol. 81, Is. 2. P. 49–59.

DOI:10.1615/TelecomRadEng.2022037071.

URL:https://www.dl.begellhouse.com/journals/0632a9d54950b268,33bd45d917452b68,54c5c714496ce4ca.html

(Особистий внесок здобувача: розраховано параметри для 256, 384, 512 біт безпеки для електронного підпису Falcon. Особистий внесок Gorbenko I.D.: Обґрунтування вимог та постановка задачі щодо генерації загальносистемних параметрів електронного підпису Falcon. Особистий внесок Yesina M.V.: Пошук та формування сукупностей безумовних, умовних та прагматичних критеріїв для процесу оцінки та порівняння електронних підписів. Особистий внесок Ostryanska Ye. V.: Перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

2. Potii O.V., Kachko O.G., Kandii S.O., Kaptol Y.Y. Determining the effect of a floating point on the Falcon Digital Signature Algorithm Security // Eastern–European Journal of Enterprise Technologies. 2024. Vol. 1, Is. 9. P. 52–59.

DOI:10.15587/1729–4061.2024.295160.

URL: https://journals.uran.ua/eejet/article/view/295160

(Особистий внесок здобувача: запропоновано теоретичне обґрунтування атаки відновлення ключа, виконано моделювання атаки. Особистий внесок Potii O.V.: Обґрунтування вимог та постановка задачі щодо криптоаналізу фіналісту конкурсу NIST PQC, методу ЕП Falcon. Особистий внесок Kachko O.G.: Програмне моделювання процесів та елементів реалізованої атаки. Особистий внесок Kaptol Y.Y.: Верифікація та аналіз результатів здійснення атаки на алгоритм Falcon. Оцінка ймовірності реалізації атаки та впливу

виявленої вразливості методу та потенційних векторів атак на захищеність алгоритму із врахуванням релевантних моделей безпеки.)

3. Kachko O.G., Gorbenko Y.I., Kandii S.O., Kaptol Y.Y. Improving protection of falcon electronic signature software implementations against attacks based on Floating Point Noise // *Eastern–European Journal of Enterprise Technologies*. 2024. Vol. 4, Is. 9, P. 6–17.

DOI:10.15587/1729–4061.2024.310521.

URL: https://journals.uran.ua/eejet/article/view/310521

(Особистий внесок здобувача: знайдено оцінки необхідної кількості підписів для проведення атаки. Особистий внесок Kachko O.G.: Визначення необхідних змін до програмної реалізації для усунення можливості реалізації атаки. Особистий внесок Gorbenko Y.I.: обгрунтування вимог та постановка задачі щодо криптоаналізу та покращення фіналісту конкурсу NIST PQC, методу ЕП Falcon. Особистий внесок Kaptol Y.Y.: оцінено вплив на безпеку електронного підпису застосування фіксованої точки замість плаваючої точки на етапі генерації підпису.)

Наукові публікації у фахових виданнях України

4. Gorbenko Yu.I., Kandii S.O. Comparison of security arguments of promising key encapsulation mechanisms // *Radiotekhnika*. 2022. Vol. 210. P. 22–36.

DOI:10.30837/rt.2022.3.210.02.

URL: http://rt.nure.ua/article/view/268561/264140

(Особистий внесок здобувача: порівняння аргументів безпеки для перспективних механізмів інкапсуляції ключів. Особистий внесок Gorbenko Yu.I.: постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

5. Kandiy S.O. Analysis of DSTU 8961:2019 in random Oracle Model // *Radiotekhnika*. 2022. Vol. 211. P. 22–36.

DOI:10.30837/rt.2022.4.211.02.

URL: http://rt.nure.ua/article/view/278028

(Особистий внесок здобувача: аналіз безпеки стандарту ДСТУ 8961:2019 у моделі випадкового оракула)

6. Kandiy S.O., Ostrianska Ye.V., Gorbenko I.D., Yesina M.V. Classification and analysis of vulnerabilities of modern information systems from classical and quantum attacks. Radiotekhnika. 2022. Vol. 211. P. 7–21.

DOI: 10.30837/rt.2022.4.211.01

URL: http://rt.nure.ua/article/view/278027

(Особистий внесок здобувача: аналіз моделей на основі нерозрізнювальності для алгоритмів шифрування та інкапсуляції ключів. Особистий внесок Ostrianska Ye.V.: Аналіз атак на реалізацію криптографічних перетворень та моделей, що їх враховують. Особистий внесок Gorbenko I.D.: постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи. Особистий внесок Yesina M.V.: классифікація вразливостей сучасних інформаційних систем.)

7. Kandiy S.O., Gorbenko I.D. Security analysis of promising key encapsulation mechanisms in the core–SVP model // *Radiotekhnika*. 2023. Vol. 212. P. 66–84.

DOI:10.30837/rt.2023.1.212.06.

URL: http://rt.nure.ua/article/view/286564

(Особистий внесок здобувача: Отримання оцінок безпеки механізмів інкапсуляції ключів в моделі core–SVP. Особистий внесок Gorbenko I.D.: постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

8. Kandii S.O., Gorbenko I.D. Analysis of DSTU 8961:2019 in the quantum random Oracle Model // *Radiotekhnika*. 2023. Vol. 214. P. 7–16.

DOI:10.30837/rt.2023.3.214.01.

URL: http://rt.nure.ua/article/view/297798/290701

(Особистий внесок здобувача: аналіз ДСТУ 8961:2019 у моделі квантового випадкового оракула. Особистий внесок Gorbenko I.D.: постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

9. Kandii S.O., Gorbenko I.D. Refinement of security estimates of quantum–resistant standards of asymmetric encryption taking into account the structure of q–arry lattices // Radiotekhnika. 2024. Vol 218. P. 76–92

DOI:10.30837/rt.2024.3.218.06

URL: http://rt.nure.ua/article/view/318798/309118

(Особистий внесок здобувача: Уточнення оцінок захищеності квантово–стійких стандартів асиметричного шифрування та електронного підпису з врахуванням q–арної структури решіток. Особистий внесок Gorbenko I.D.: постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

10. Kandii S.O., Gorbenko I.D. Assessing the influence of the algebraic structure of q–ary lattices on the complexity of cryptanalysis of problems on lattices // *Radiotekhnika.* 2024. Vol. 217. P. 79–99.

DOI:10.30837/rt.2024.2.217.07.

URL: http://rt.nure.ua/article/view/310856

(Особистий внесок здобувача: оцінка впливу q–арної структури решіток на складність криптоаналізу складних проблем. Особистий внесок Gorbenko I.D.: постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

Наукові праці, які додатково відображають результати дисертації:

11. Кандій С.О., Горбенко І.Д. Аналіз фактору Ерміта алгоритму BKZ на решітках малої розмірності. Computer Science and Cybersecurity. 2024. Is. 1(25). P. 22–36

DOI: 10.26565/2519–2310–2024–1–02

URL:https://periodicals.karazin.ua/cscs/article/download/2519–2310–2024–1–02/22054/

(Особистий внесок здобувача: експерементальна оцінка фактора Ерміта та аналіз його впливу на криптоаналіз. Особистий внесок Gorbenko I.D.:

постановка проблеми дослідження, перевірка наукової достовірності отримуваних результатів, перевірка тексту роботи.)

12. Кандій С.О. Аналіз CPA–to–CCA перетворення ДСТУ 8961:2019 у моделі випадкового оракула. Мат. та комп'ютер. моделювання. Сер. Фіз.–мат. науки. 2023. Вип. 36. С. 101–105.

DOI: 10.15407/10.15407/fmmit2023.36.101

URL: http://www.fmmit.lviv.ua/index.php/fmmit/article/download/285/245/

(Особистий внесок здобувача: аналіз CPA–to–CCA перетворення ДСТУ 8961:2019 у моделі випадкового оракула.)

Наукові праці, які засвідчують апробацію матеріалів дисертації:

13. Кандій С.О. Острянська Є.В. Генерація загальносистемних параметрів для схеми електронного підпису Rainbow. І міжнародна науково–технічна коференція «Системи і технології зв’язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку». 2021. С. 226–227.

14. Кандій С.О. Порівняння аргументів безпеки перспективних механізмів інкапсуляції ключів. Праці 8–ої Міжнародній конференції «Комп’ютерне моделювання в наукоємних технологіях» (КМНТ–2022). 2022. С. 97–100.

15. Горбенко Ю.І., Острянська Є.В., Кандій С.О. Експериментальна оцінка точності фактора ерміта. IV Міжнародна науково-технічна конференція «Системи і технології зв’язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку». 2024. С. 48-49.

16. Потій О.В., Качко О.Г., Кандій С.О., Каптьол Є.Ю. Дослідження впливу плаваючої точки на безпеку алгоритму електронного підпису Falcon. II Міжнародна науково–практична конференція “Кіберборотьба: розвідка, захист та протидія”. 2024. С. 28–30.

ДОДАТОК Б

ІСХОДНІ КОДИ МОДЕЛЕЙ РЕДУКЦІЇ РЕШІТОК

В.1 Модель GSA

```
std::vector<double> ZGSA(double q, int d, int m, int block_size, int max_loops) {
    if (d <= 0 || m < 0 || m > d || block_size < 0) {
        throw std::invalid_argument("Invalid input parameters");
    }

    int nq = m;
    int n1 = d – nq;

    if (block_size == 0) {
        std::vector<double> L(d);
        std::fill_n(L.begin(), nq, std::log2(q));
        std::fill_n(L.begin() + nq, n1, 0.0);
        return L;
    }

    double slope = –2 * std::log2(HermiteFactor(block_size));
    double lq = std::log2(q);
    int B = static_cast<int>(std::floor(std::log2(q) / –slope));

    size_t required_size = static_cast<size_t>(nq) + B + 1 + n1;
    if (required_size > std::vector<double>().max_size()) {
        throw std::runtime_error("Required vector size too large");
    }

    std::vector<double> L;
    L.reserve(required_size);

    L.insert(L.end(), nq, lq);
    for (int i = 1; i <= B + 1; ++i) {
        L.push_back(lq + i * slope);
    }
    L.insert(L.end(), n1, 0.0);

    int x = 0;
    double lv = std::accumulate(L.begin(), L.begin() + d, 0.0);
    double glv = nq * lq;

    while (lv > glv && x <= B && (x + d) < L.size()) {
        lv –= L[x];
        lv += L[x + d];
        ++x;
    }

    if (x > B) {
        throw std::runtime_error("Window slid too far");
    }
```

```
    std::vector<double> final_L(L.begin() + x, L.begin() + x + d);
    int a = std::max(0, nq – x);
    B = std::min(B, d – a);

    double diff = glv – lv;
    if (std::abs(diff) >= lq) {
        throw std::runtime_error("Volume difference too large");
    }

    if (B > 0) {
        double adjustment = diff / B;
        for (int i = a; i < a + B; ++i) {
            final_L[i] += adjustment;
        }
    }

    double final_lv = std::accumulate(final_L.begin(), final_L.end(), 0.0);
    if (std::abs(final_lv/glv – 1) >= 1e–6) {
        throw std::runtime_error("Final volume check failed");
    }

    return final_L;
}
```

В.2 Симулятор Чена–Нгуєна

```
std::vector<double> _NguenChenSimulator(const std::vector<double>& r, int block_size, int
max_loops) {
    int d = r.size();
    std::vector<double> r1 = r;
    std::vector<double> r2 = r;

    std::vector<double> c(45);
    for (int i = 1; i <= 45; ++i) {
        c[i – 1] = rk[rk.size() – i] – std::accumulate(rk.rbegin(), rk.rbegin() + i, 0.0) / i;
    }
    for (int beta = 46; beta <= block_size; ++beta) {
        c.push_back((std::lgamma(beta / 2.0 + 1) * (1.0 / beta) – std::log(std::sqrt(M_PI))) /
std::log(2.0));
    }

    int N = max_loops ? max_loops : d;

    for (int i = 0; i < N; ++i) {
        bool phi = true;
        for (int k = 0; k < d – std::min(45, block_size); ++k) {
            int beta = std::min(block_size, d – k);
            int f = k + beta;
            double logV = std::accumulate(r1.begin(), r1.begin() + f, 0.0) – std::accumulate(r2.begin(),
r2.begin() + k, 0.0);
            double lma = logV / beta + c[beta – 1];

            if (phi) {
```

```
            if (lma < r1[k]) {
               r2[k] = lma;
               phi = false;
            }
         } else {
            r2[k] = lma;
         }
      }

      if (phi || r1 == r2) {
         break;
      } else {
         int beta = std::min(45, block_size);
         double logV = std::accumulate(r1.begin(), r1.end(), 0.0) – std::accumulate(r2.begin(),
r2.end() – beta, 0.0);

         std::vector<double> rk1;
         if (block_size < 45) {
            double tmp = std::accumulate(rk.end() – block_size, rk.end(), 0.0) / block_size;
            std::transform(rk.end() – block_size, rk.end(), std::back_inserter(rk1), [tmp](double r_) {
return r_ – tmp; });
         } else {
            rk1 = rk;
         }

         for (int k = d – beta, i = 0; k < d; ++k, ++i) {
            r2[k] = logV / beta + rk1[i];
         }
         r1 = r2;
      }
   }
   return r1;
}
```

B.3 Рандомізований симулятор Чена–Нгуєна

```
std::vector<double> normalize_GSO_unitary(const std::vector<double>& l) {
   double log_det = std::accumulate(l.begin(), l.end(), 0.0);
   int n = l.size();
   std::vector<double> nor_log_det(n);
   for (int i = 0; i < n; ++i) {
      nor_log_det[i] = l[i] – log_det / n;
   }
   return nor_log_det;
}


std::pair<std::vector<double>, int> simulate_prob(
   std::vector<double> r, int block_size, int max_loops, unsigned int prng_seed = 0xdeadbeef
) {
   std::mt19937 gen(prng_seed);
   std::exponential_distribution<> dist(0.5);
```

```
    int d = r.size();
    std::vector<double> r1 = r;
    std::vector<double> r2 = r;
    std::vector<bool> t0(d, true), t1(d);

    std::vector<double> c(45);
    for (int j = 1; j <= 45; ++j) {
        c[j – 1] = rk[rk.size() – j] – std::accumulate(rk.rbegin(), rk.rbegin() + j, 0.0) / j;
    }
    for (int beta = 46; beta <= block_size; ++beta) {
        c.push_back((std::lgamma(beta / 2.0 + 1) * (1.0 / beta) – std::log(std::sqrt(M_PI))) /
std::log(2.0));
    }

    int N = max_loops ? max_loops : d;
    for (int i = 0; i < N; ++i) {
        std::fill(t1.begin(), t1.end(), false);
        for (int k = 0; k < d – std::min(45, block_size); ++k) {
            int beta = std::min(block_size, d – k);
            int f = k + beta;
            bool phi = std::any_of(t0.begin() + k, t0.begin() + f, [](bool val) { return val; });
            double logV = std::accumulate(r1.begin(), r1.begin() + f, 0.0) – std::accumulate(r2.begin(),
r2.begin() + k, 0.0);

            if (phi) {
                double X = dist(gen);
                double lma = (std::log2(X) + logV) / beta + c[beta – 1];
                if (lma < r1[k]) {
                    r2[k] = lma;
                    r2[k + 1] = r1[k] + std::log2(std::sqrt(1 – 1.0 / beta));
                    double dec = (r1[k] – lma) + (r1[k + 1] – r2[k + 1]);
                    for (int j = k + 2; j < f; ++j) {
                        r2[j] = r1[j] + dec / (beta – 2);
                        t1[j] = true;
                    }
                    phi = false;
                }
            }

            for (int j = k; j < f; ++j) {
                r1[j] = r2[j];
            }
        }

        if (std::none_of(t1.begin(), t1.end(), [](bool val) { return val; })) {
            break;
        }

        int beta = std::min(45, block_size);
        double logV = std::accumulate(r1.begin(), r1.end(), 0.0) – std::accumulate(r2.begin(), r2.end()
– beta, 0.0);
```

```
        std::vector<double> rk1;

        if (block_size < 45) {
            rk1 = normalize_GSO_unitary(std::vector<double>(rk.end() – beta, rk.end()));
        } else {
            rk1 = rk;
        }

        for (int k = d – beta, i = 0; k < d; ++k, ++i) {
            r2[k] = logV / beta + rk1[i];
            t1[k] = true;
        }

        if (r1 == r2) {
            break;
        }
        r1 = r2;
        t0 = t1;
    }
    return {r1, N};
}

std::vector<double> _RandomNguenChenSimulator(std::vector<double> L, int block_size, int
max_loops) {
    int tries = 1;

    std::vector<double> i(L.size());
    int j = 0;

    for (int t = 0; t < tries; ++t) {
        auto [x, y] = simulate_prob(L, block_size, max_loops, t + 1);
        if (t == 0) {
            i = x;
            j = y;
        } else {
            std::transform(i.begin(), i.end(), x.begin(), i.begin(), std::plus<>());
            j += y;
        }
    }

    for (double& val : i) {
        val /= tries;
    }
    return i;
}

std::vector<double> RandomNguenChenSimulator(double q, int d, int m, int block_size, int
max_loops)
{
    std::vector<double> r = gso(d, m, q);
    return _RandomNguenChenSimulator(r, block_size, max_loops);
}
```

В.4 Симулятор Альбрехта–Лі

```
std::vector<double> ah_constant(int beta, double scale, int prec = 53) {
  std::vector<double> h = {
    1.00000000000000, 1.00000000000000, 1.02492695474985, 1.05161980747554,
    1.10445074764426, 1.12738094965280, 1.18113914825808, 1.21037399094975,
    1.25380229130682, 1.31275912013118, 1.32737148909975, 1.36036038598451,
    1.43966294868932, 1.46160335381056, 1.55544109937333, 1.51556419271870,
    1.62261393856106, 1.62261542586149, 1.69386128187285, 1.74294139717984,
    1.76511645842542, 1.85062593391314, 1.92910191727097, 1.95924301565493,
    2.01827070103134, 2.09555655837831, 2.10779553489076, 2.16072906694966,
    2.17866707693959, 2.24637113675736, 2.30374935080714, 2.32703241781875,
    2.38719370287110, 2.44070021621789, 2.47587251254528, 2.57636556347475,
    2.59580776592391, 2.70737106887894, 2.71293172456551, 2.80972868887803,
    2.85337256682321, 2.96643112072014, 2.92901889083372, 2.99441283605227,
    3.03884800785631, 3.09367293382578, 3.15593321975939, 3.21699662499604,
    3.26306485591489, 3.32320705128515, 3.37871256503968, 3.49314144958915,
    3.52315959119194, 3.60596719185516, 3.62302348800297, 3.67106448779994,
    3.75288918941554, 3.80729231066355, 3.83017943995872, 3.87680907526471,
    3.99304695483873, 4.04539449313441, 4.12835783024051, 4.12989317768338,
    4.25784341671535, 4.23117443161792, 4.34900834929189, 4.37617075735239,
    4.44178322064643, 4.53471153665029, 4.54352770842019, 4.65495019943735,
    4.63508961564190, 4.73512141747094, 4.76413486921241, 4.87996877805749,
    4.94652066938356, 4.98495063065338, 5.07692160777643, 5.04745948303442,
    5.12518339690435
  };

  std::vector<double> ah(beta + 1, 0.0);
  std::vector<double> c(beta + 1, 0.0);

  for (int j = 0; j <= std::min(beta, 80); ++j) {
    ah[j] = h[j];
  }

  if (beta > 49) {
    double lga = std::lgamma(beta / 2.0 + 1) * (2.0 / beta) – std::log(PI);
    ah[beta] = std::exp(lga);

    int z = std::min(7 + static_cast<int>(std::log2(scale)), beta – 4);
    int x = std::min(beta – z, 80);

    ah[x – 1] = (ah[beta] – h[x – 2]) / (beta – x + 2) / 2 + h[x – 2] / 2 + h[x – 1] / 2;

    for (int j = x; j < beta; ++j) {
      double u = (ah[beta] – h[x – 2]) * (j – x + 2) / (beta – x + 2) + h[x – 2];
      double v = (ah[beta] – h[x – 1]) * (j – x + 1) / (beta – x + 1) + h[x – 1];
      double w = (ah[beta] – h[x]) * (j – x) / (beta – x) + h[x];
      ah[j] = (u + v + w) / 3.0;
    }
  }
```

```
   for (int j = 2; j <= beta; ++j) {
      c[j] = 0.5 * std::log2(ah[j]);
   }

   return c;
}

std::vector<double> one_tour(std::vector<double>& r, int block_size, const std::vector<double>&
c) {
   int d = r.size();

   for (int i = 0; i < d – 1; ++i) {
      int p = std::min(block_size, d – i);
      int j = std::min(i + block_size, d);
      double g = std::accumulate(r.begin() + i, r.begin() + j, 0.0);

      if (r[i] > c[p] + g / p) {
         r[i] = c[p] + g / p;

         for (int s = i + 1; s < j; ++s) {
            int t = j – s;
            double y = std::accumulate(r.begin() + i, r.begin() + s, 0.0);
            r[s] = c[t] + (g – y) / t;
         }
      }
   }

   return r;
}

std::vector<double> _AlbrechtLiSimulator(std::vector<double> r, int block_size, int tours) {
   double scale = 1.0;
   int prec = 53;
   std::vector<double> c = ah_constant(block_size, scale, prec);

   for (int i = 0; i < tours; ++i) {
      r = one_tour(r, block_size, c);
   }

   return r;
}

std::vector<double> AlbrechtLiSimulator(double q, int d, int m, int block_size, int max_loops)
{
   std::vector<double> r = gso(d, m, q);
   return _AlbrechtLiSimulator(r, block_size, max_loops);
}
```

B.5 Рандомізований симулятор Альбрехта–Лі

```
std::vector<double> zsimulate_prob(std::vector<double> r, int block_size, int max_loops, const
std::vector<double>& c, int prng_seed = 88) {
   if (block_size <= 2) {
      throw std::invalid_argument("The simulator requires block size >= 3.");
   }

   std::mt19937 gen(prng_seed);
   int d = r.size();
   std::vector<double> r1 = r;
   std::vector<double> r2 = r;
   int N = max_loops ? max_loops : d;

   std::vector<bool> t0(d, true);

   for (int i = 0; i < N; ++i) {
      std::vector<bool> t1(d, false);

      for (int k = 0; k < d – std::min(50, block_size); ++k) {
         int beta = std::min(block_size, d – k);
         int f = k + beta;
         bool phi = false;

         for (int kp = k; kp < f; ++kp) {
            phi |= t0[kp];
         }

         double logV = std::accumulate(r1.begin(), r1.begin() + f, 0.0) – std::accumulate(r2.begin(),
r2.begin() + k, 0.0);

         if (phi) {
            double X = expovariate(0.5, gen);
            double lma = (std::log2(X) + logV) / beta + c[beta];

            if (lma < r1[k]) {
               r2[k] = lma;
               r2[k + 1] = r1[k] + std::log2(std::sqrt(1 – 1.0 / beta));
               double dec = (r1[k] – lma) + (r1[k + 1] – r2[k + 1]);

               for (int j = k + 2; j < f; ++j) {
                  r2[j] = r1[j] + dec / (beta – 2);
                  t1[j] = true;
               }
               phi = false;
            }
         }

         for (int j = k; j < f; ++j) {
            r1[j] = r2[j];
         }
```

```
      }

      if (std::none_of(t1.begin(), t1.end(), [](bool v) { return v; })) {
         break;
      }

      int beta = std::min(50, block_size);
      double logV = std::accumulate(r1.begin(), r1.end(), 0.0) – std::accumulate(r2.begin(),
r2.begin() + d – beta, 0.0);

      for (int k = d – beta; k < d; ++k) {
         r2[k] = logV / (d – k) + c[d – k];
         logV –= r2[k];
         t1[k] = true;
      }

      if (r1 == r2) {
         break;
      }

      r1 = r2;
      t0 = t1;
   }

   return r1;
}

std::vector<double> _RandomAlbrechtLiSimulator(std::vector<double> r, int block_size, int tours)
{
   double scale = 1.0;
   int prec = 53;
   std::vector<double> c = ah_constant(block_size, scale, prec);
   return zsimulate_prob(r, block_size, tours, c);
}

std::vector<double> RandomAlbrechtLiSimulator(double q, int d, int m, int block_size, int
max_loops)
{
   std::vector<double> r = gso(d, m, q);
   return _RandomAlbrechtLiSimulator(r, block_size, max_loops);
}
```

ДОДАТОК В

ІСХОДНІ КОДИ МЕОДІВ ОЦІНКИ LWE, NTRU, SIS

Г.1 Атака вкладення

```
bool IsPrimalAttackWork(
   LatticeParams*         latticeParams,
   EmbeddingAttackParams*  attackParams)
{
   int          dimention;
   double         lhs, rhs;
   int          lowDimention;
   double         lowRhs;
   int          highDimention;
   double         highRhs;
   double            hermiteFactor;

   hermiteFactor = HermiteFactor(attackParams–>blockSize);

   lowDimention = std::max(attackParams–>blockSize, latticeParams–>n+1);
   highDimention = latticeParams–>n * 2;
   //std::cout<<highDimention<<" "<<lowDimention<<std::endl;

   if (latticeParams–>isTernary)
      latticeParams–>dim = highDimention;
   lhs = attackParams–>targetNorm(latticeParams, attackParams–>blockSize);
   //std::cout<<"lhs="<<" "<<lhs<<std::endl;
   highRhs = GetNormAt(
      attackParams–>simulator,
      latticeParams–>q,
      highDimention,
      highDimention – latticeParams–>n, // q count
      attackParams–>blockSize,
      attackParams–>maxLoops,
      highDimention – attackParams–>blockSize);
   //std::cout<<"highRhs="<<" "<<highRhs<<std::endl;
   if (lhs > highRhs)
   {
      //std::cout<<"attack not work!\n";
      return false;
   }

   if (latticeParams–>isTernary)
      latticeParams–>dim =lowDimention;
   lhs = attackParams–>targetNorm(latticeParams, attackParams–>blockSize);
   //std::cout<<"lhs="<<" "<<lhs<<std::endl;
   lowRhs = GetNormAt(
      attackParams–>simulator,
      latticeParams–>q,
      lowDimention,
      lowDimention – latticeParams–>n, // q count
      attackParams–>blockSize,
```

```
        attackParams–>maxLoops,
        lowDimention – attackParams–>blockSize);
    //std::cout<<"lowRhs="<<" "<<highRhs<<std::endl;
    if (lowRhs <= rhs)
        return true;

    while(true)
    {
        dimention = lowDimention+1;
        if (latticeParams–>isTernary)
            latticeParams–>dim =dimention;
        lhs = attackParams–>targetNorm(latticeParams, attackParams–>blockSize);
        //std::cout<<"lhs: "<<lhs<<std::endl;
        lowRhs = GetNormAt(
            attackParams–>simulator,
            latticeParams–>q,
            dimention,
            dimention – latticeParams–>n, // q count
            attackParams–>blockSize,
            attackParams–>maxLoops,
            dimention – attackParams–>blockSize);
        //std::cout<<"lowRhs: "<<lowRhs<<std::endl;
        //std::cout<<highDimention<<" "<<lowDimention<<std::endl;
        if (lhs > lowRhs)
        {
            lowDimention = (ceil((lowDimention + highDimention)/2));
            continue;
        }
        //printf("test %f\n", lowDimention);

        for (dimention = lowDimention; dimention >= std::max(latticeParams–>n+1, attackParams–
>blockSize); dimention—)
        {
            //printf("test %d\n", dimention);
            //if (dimention < attackParams–>blockSize)
            //  return true;
            attackParams–>dimention = dimention;
            latticeParams–>dim =dimention;
            if (latticeParams–>isTernary)
                latticeParams–>dim =dimention;
            lhs = attackParams–>targetNorm(latticeParams, attackParams–>blockSize);
            rhs = GetNormAt(
                attackParams–>simulator,
                latticeParams–>q,
                dimention,
                dimention – latticeParams–>n, // q count
                attackParams–>blockSize,
                attackParams–>maxLoops,
                dimention – attackParams–>blockSize);
            if (lhs > rhs)
            {
                //printf("test2\n");
```

```
            attackParams–>dimention = dimention+1;
            return true;
         }
      }
      if (dimention < std::max(latticeParams–>n+1, attackParams–>blockSize))
      {
         attackParams–>dimention = lowDimention;
         return true;
      }
   }


   return false;
}

bool EmbedingAttack(
   LatticeParams*  latticeParams,
   EmbeddingAttackParams* attackParams)
{
   int minBlockSize = 0xFFFFFF;
   int minDimension = 0xFFFFFF;
   double minCost = 999999999.9;
   int maxDimension = latticeParams–>n * 2;
   int currentDimension;
   double cost;

   for (int blockSize = 100; blockSize < maxDimension; blockSize++)
   {
      //std::cout<<"Block size"<<blockSize<<std::endl;
      attackParams–>blockSize = blockSize;
      if (IsPrimalAttackWork(
         latticeParams,
         attackParams))
      {
         cost = 0.292 * blockSize;
         //std::cout<<"Cost "<<cost<<std::endl;
         if (cost < minCost)
         {
            minCost = cost;
            minBlockSize = blockSize;
            minDimension = attackParams–>dimention;
            break;
         }
      }
   }

   if (minCost == 999999999.9)
      return false;

   attackParams–>dimention = minDimension;
   attackParams–>blockSize = minBlockSize;
   attackParams–>cost = minCost;
```

```
    return true;
}
```

Г.2 Атака декодування

```
double GetBddProbability_(
    const std::vector<double>& profile,
    const std::vector<int>& d,
    double sigma)
{
    size_t profileLength = profile.size();
    double sqrtPI = std::sqrt(M_PI);
    double alpha = sqrtPI / (2 * sigma);

    mpfr_t probability, term, mp_alpha;
    mpfr_init2(probability, 256);
    mpfr_init2(term, 256);
    mpfr_init_set_d(probability, 1.0, MPFR_RNDD);
    mpfr_init_set_d(mp_alpha, alpha, MPFR_RNDD);

    for (size_t i = 0; i < profileLength; i++) {

        mpfr_set_d(term, d[i] * profile[i]*sqrt(2.0)/sigma, MPFR_RNDD);

        mpfr_erf(term, term, MPFR_RNDD);
        mpfr_mul(probability, probability, term, MPFR_RNDD);
    }

    double result = mpfr_get_d(probability, MPFR_RNDD);


    mpfr_clear(probability);
    mpfr_clear(term);
    mpfr_clear(mp_alpha);

    return result;
}

double GetFullCost(
    std::vector<double>     profile,
    std::vector<int>        d,
    double                  sigma)
{
    return GetBddCost(d);
}

std::vector<int> OptimizeBddParams(
    std::vector<double>     profile,
    double                  sigma,
    double                  minProb)
{
    std::vector<int>        d;
    size_t                  profileLength;
    double                  targetProb;
```

```
    double              alpha;
    double              beta;

    profileLength = profile.size();
    targetProb = minProb;//exp(log(minProb)/profileLength);
    alpha = sqrtPI / (2 * sigma);
    beta = erfinv(targetProb);

    for (int i = 0; i < profileLength; i++)
    {
        d.push_back(
            ceil(beta / (alpha * profile[i])));// round? floor?
        if (d[i]==0)
            d[i] = 1;
    }
    return d;
}

std::vector<int> OptimizeBddProb(
    std::vector<double>     profile,
    double                  sigma)
{
    double minProb = 0.92;
    auto d = OptimizeBddParams(profile, sigma, minProb);
    //double minCost = GetFullCost(profile, d, sigma);
    for (int i = 0; i <d.size(); i++)
    {
        if (d[i] == 0)
            d[i] = 1;
    }
    return d;

}

inline double GetOptimalDimention(
    double n,
    double q,
    double logDelta)
{
    return ceil(sqrt(n * log(q) / logDelta));
}
double calculate_T(int beta, double sigma, const std::vector<double>& R, const std::vector<int>&
d) {
    if (beta <= 0 || sigma <= 0.0) {
        throw std::invalid_argument("beta and sigma must be positive.");
    }
    if (R.empty()) {
        throw std::invalid_argument("R cannot be empty.");
    }

    // Initialize MPFR variables
    mpfr_t log2_result, term, prod, temp, erf_input, erf_result, pi_sqrt_2, prod2;
```

```
mpfr_inits2(256, log2_result, term, prod, temp, erf_input, erf_result, pi_sqrt_2 , prod2, NULL);

// Set constants
mpfr_const_pi(pi_sqrt_2, MPFR_RNDN); // pi
mpfr_sqrt_ui(pi_sqrt_2, 2, MPFR_RNDN); // sqrt(2)
mpfr_mul_ui(pi_sqrt_2, pi_sqrt_2, 2, MPFR_RNDN); // 2 * sqrt(2)



// Calculate product p = ∏ erf(R[i] / (2 * sqrt(2) * sigma))
mpfr_set_ui(prod, 1, MPFR_RNDN); // Initialize product to 1
mpfr_set_ui(prod2, 1, MPFR_RNDN);
for (int i = 0; i < R.size(); i++) {
    // erf_input = R[i] / (2 * sqrt(2) * sigma)s
    mpfr_set_d(erf_input, R[i] * d[i], MPFR_RNDN);
    mpfr_mul_d(prod2, prod2, d[i], MPFR_RNDN);

    //mpfr_div_d(erf_input, erf_input, sqrt(2) / sigma, MPFR_RNDN);
    mpfr_div_d(erf_input, erf_input,  2 * sqrt(2) * sigma, MPFR_RNDN);

    // erf_result = erf(erf_input)
    mpfr_erf(erf_result, erf_input, MPFR_RNDN);



    // temp = prod * erf_result
    mpfr_mul(temp, prod, erf_result, MPFR_RNDN);

    // Update product
    mpfr_set(prod, temp, MPFR_RNDN);

    //mpfr_set_d(prod, 1.0, MPFR_RNDN);
}

// Calculate log2_result = log2(2^(0.292 * beta) / p)
// term = 2^(0.292 * beta)
mpfr_set_d(term, 0.292 * beta, MPFR_RNDN);
mpfr_exp2(term, term, MPFR_RNDN);
mpfr_add(term, term, prod2, MPFR_RNDN);

// log2_result = log2(term / prod)
mpfr_div(temp, term, prod, MPFR_RNDN); // term / prod
mpfr_log2(log2_result, temp, MPFR_RNDN); // log2(term / prod)

// Get the result as a double
double T = mpfr_get_d(log2_result, MPFR_RNDN);
double prob = mpfr_get_d(prod, MPFR_RNDN);
//mpfr_printf("boost:%Re\n", prod);

// Free MPFR variables
mpfr_clears(log2_result, term, prod, temp, erf_input, erf_result, pi_sqrt_2,prod2, NULL);
```

```
    return T;
}


#include <cmath>
void DecodingAttack(
    LatticeParams *latticeParams,
    DecodingAttackParams *attackParams)
{
    //double sigma = latticeParams–>sigma;
    attackParams–>totalCost = 100000000;
    int frozenLoops = 0;

    int maxDimention = latticeParams–>n * 2;
    int minDimention = std::max(attackParams–>blockSize, latticeParams–>n + 1);
    //ceil(latticeParams–>n * 1.0);//std::max(attackParams–>blockSize, latticeParams–>n + 1);
    //ceil(latticeParams–>n * 1.9);//std::max(attackParams–>blockSize, latticeParams–>n + 1);
    int dimentionRange = maxDimention – minDimention + 1;

    for (int blockSize = 500; blockSize < latticeParams–>n; blockSize+=5) {

        double hermiteFactor = HermiteFactor(blockSize);
        minDimention = std::max(attackParams–>blockSize, latticeParams–>n + 1);
        //
        //minDimention = maxDimention–16;//
        bool isUpdated = false;
        printf("blockSize=%d lastCost=%f\n", blockSize, attackParams–>totalCost);

        std::vector<std::future<void>> futures;
        int numThreads = 16;
        int chunkSize = dimentionRange / numThreads;

        for (int t = 0; t < numThreads; ++t) {

            int startDimention = maxDimention – t * chunkSize;
            int endDimention = (t == numThreads – 1) ? minDimention : startDimention – chunkSize;

            futures.emplace_back(std::async(std::launch::async, [&, startDimention, endDimention]() {
                mpfr_t localBddCost, localBkzCost;
                mpfr_init(localBddCost);
                mpfr_init(localBkzCost);
                double totalCost = 100000000;
                DecodingAttackParams localAttackParams;
                LatticeParams localLattice = *latticeParams;
                //int dimention = std::max((int)GetOptimalDimention(latticeParams–>n, latticeParams–
>q, log(hermiteFactor)), blockSize);
                //dimention = std::min(dimention, (int)latticeParams–>n * 2);
                for (int dimention = startDimention; dimention >= endDimention; —dimention)
                /*int dimention = GetOptimalDimention(latticeParams–>n, latticeParams–>q,
log(hermiteFactor));
                if (dimention < blockSize)
                    dimention = blockSize;
```

```
            if (dimention < (latticeParams–>n+1))
               dimention = latticeParams–>n+1;
            if (dimention > latticeParams–>n * 2)
               dimention = latticeParams–>n * 2;*/

            //printf("dimention = %d max_dim=%d\n", dimention, latticeParams–>n * 2);
            {
               std::vector<double> profile = attackParams–>simulator(
                  latticeParams–>q,
                  dimention,
                  dimention – latticeParams–>n,
                  blockSize,
                  attackParams–>maxLoops);

               for (int i =0; i < profile.size(); i++) {
                  profile[i] = std::pow(2.0, profile[i]);
               }
               localLattice.dim = dimention;
               double vecLen = sqrt((double)latticeParams–>df + (double)(dimention–latticeParams–
>n)*latticeParams–>dg/latticeParams–>n);
               //std::pow(2.0, GetTargetShortestVectorTernary(&localLattice, blockSize));

               double sigma = vecLen / sqrt(dimention);
               //sigma /= sqrt(2*M_PI);
               //sigma = sigma * sigma;
               //sigma = sqrt(sigma);
               //printf("sigma = %f\n", sigma);

               //sigma = sigma * 6;

               //sigma = ((double)latticeParams–>df/latticeParams–>n + (double)latticeParams–
>dg/latticeParams–>n * ((double)dimention–latticeParams–>n)/ latticeParams–>n);
               //sigma = sigma / (2*std::sqrt(2));
               //sqrt(((double)latticeParams–>df + ((double)dimention–latticeParams–
>n)/((double)latticeParams–>n*latticeParams–>dg))/dimention);
               //sigma = sigma;//s *sqrt(2*M_PI);
               //vecLen / sqrt(dimention)*sqrt(2*M_PI);
               //printf("sigma=%f df=%d dg=%d\n", sigma, latticeParams–>df,latticeParams–>dg);

               auto d = OptimizeBddProb(profile, sigma);

               double logBddCost = 0;//GetBddCost(d);
               double logBkzCost = 0.292 * blockSize;
               double logTotalCost = calculate_T(blockSize, sigma, profile, d);

               if (logTotalCost <= totalCost) {
                  totalCost = logTotalCost;
                  localAttackParams.blockSize = blockSize;
                  localAttackParams.totalCost = logTotalCost;
                  localAttackParams.probability = 1;
                  localAttackParams.bddCost = logBddCost;
                  localAttackParams.bkzCost = logBkzCost;
```

```
                    localAttackParams.profile = profile;
                    isUpdated = true;
                } else {
                    //break;
                }
            }

            mtx.lock();
            if (totalCost < attackParams–>totalCost)
            {
                attackParams–>blockSize = localAttackParams.blockSize;
                attackParams–>totalCost = localAttackParams.totalCost;
                attackParams–>probability = localAttackParams.probability;
                attackParams–>bddCost = localAttackParams.bddCost;
                attackParams–>bkzCost = localAttackParams.bkzCost;
                attackParams–>profile = localAttackParams.profile;
            }
            mtx.unlock();

            mpfr_clear(localBddCost);
            mpfr_clear(localBkzCost);
        }));
    }

    for (auto &fut : futures) {
        fut.get();
    }

    if (!isUpdated)
        frozenLoops++;
    else
        frozenLoops = 0;
  }

  //mpfr_clear(bddCost);
  //mpfr_clear(bkzCost);
}
```

Г.3 Атака на SIS

```
double calculate_log2_expression(double B, double v_norm, int n, double log2_N, double log2_T)
{
    if (v_norm > B * sqrt(n))
    return 0;
  mpfr_set_default_prec(256);

  mpfr_t B_mp, v_norm_mp, n_mp, N_mp, p_mp, exponent_mp, denominator_mp, result_mp;

  mpfr_inits(B_mp, v_norm_mp, n_mp, N_mp, p_mp, exponent_mp, denominator_mp,
result_mp, nullptr);

  mpfr_set_d(B_mp, B, MPFR_RNDN);
```

```
    mpfr_set_d(v_norm_mp, v_norm, MPFR_RNDN);
    mpfr_set_si(n_mp, n, MPFR_RNDN);

    mpfr_set_d(N_mp, log2_N, MPFR_RNDN);
    mpfr_exp2(N_mp, N_mp, MPFR_RNDN);

    mpfr_sqrt(n_mp, n_mp, MPFR_RNDN);           // sqrt(n)
    mpfr_mul(B_mp, B_mp, n_mp, MPFR_RNDN);      // B * sqrt(n)
    mpfr_div(exponent_mp, B_mp, v_norm_mp, MPFR_RNDN);  // B * sqrt(n) / ||v||_2

    // Вычисляем p = 2 * exp(– (B * sqrt(n) / ||v||_2)^2 / 2)
    mpfr_pow_ui(exponent_mp, exponent_mp, 2, MPFR_RNDN); // (B * sqrt(n) / ||v||_2)^2
    mpfr_div_ui(exponent_mp, exponent_mp, 2, MPFR_RNDN); // / 2
    mpfr_neg(exponent_mp, exponent_mp, MPFR_RNDN);      // отрицательное значение для
экспоненты
    mpfr_exp(exponent_mp, exponent_mp, MPFR_RNDN);      // exp(– ...)
    mpfr_mul_ui(p_mp, exponent_mp, 2, MPFR_RNDN);       // p = 2 * exp(...)
    mpfr_sub_d(p_mp, p_mp, 1.0, MPFR_RNDN);
    mpfr_neg(p_mp, p_mp, MPFR_RNDN);

    // Вычисляем (1 – p)^N
    mpfr_ui_sub(exponent_mp, 1, p_mp, MPFR_RNDN);       // 1 – p
    mpfr_pow(denominator_mp, exponent_mp, N_mp, MPFR_RNDN); // (1 – p)^N

    // Вычисляем знаменатель: 1 – (1 – p)^N
    mpfr_ui_sub(denominator_mp, 1, denominator_mp, MPFR_RNDN);

    // Полное выражение: 2 * T / (1 – (1 – p)^N)
    mpfr_set_d(result_mp, log2_T, MPFR_RNDN);           // log2(T)
    mpfr_add_ui(result_mp, result_mp, 1, MPFR_RNDN);    // log2(2 * T) = log2(T) + 1
    mpfr_sub(result_mp, result_mp, denominator_mp, MPFR_RNDN); // log2((2 * T) / (1 – (1 –
p)^N))

    // Преобразуем результат в double
    double final_result = mpfr_get_d(result_mp, MPFR_RNDN);

    // Освобождаем память
    mpfr_clears(B_mp, v_norm_mp, n_mp, N_mp, p_mp, exponent_mp, denominator_mp,
result_mp, nullptr);

    // Возвращаем результат как double
    return final_result;
}
```