Казахский национальный исследовательский технический университет имени К.И. Сатпаева

Институт автоматики и информационных технологий

УДК 51.74

На правах рукописи

ӘМІРХАНОВА ДАНА САЙРАНҒАЖЫҚЫЗЫ

Схема постквантового шифрования с открытом ключом на основе решетки с использованием принципов Эль-Гамаля

8D06301 – Системы информационной безопасности

Диссертация на соискание степени
доктора философии (PhD)

Научный консультант:
доктор PhD,
ассоциированный профессор
О.Ж. Мамырбаев
Зарубежные консультанты:
доктор PhD,
профессор
С. Гнатюк
(Украина)

доктор PhD,
профессор
М. Явич
(Грузия)

Республика Казахстан
Алматы, 2025

СОДЕРЖАНИЕ

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей диссертации использованы ссылки на следующие стандарты:

ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание. Общее требования и правила составления.

«Инструкция по оформлению диссертации и автореферата», МОН РК, Внешний аттестационный комитет. Алматы 2004, № 377-3ж.

ГОСО РК 5.04.034-2011. Государственный общеобязательный стандарт образования Республики Казахстан. Послевузовское образование. Докторантура. Основные положения: утв. приказом Министра образования и науки Республики Казахстан от 17 июня 2011 года, №261(с изменениями и дополнениями соответствии приказа Министра науки и высшего образования Республики Казахстан от 14 июня 2024 года № 294)

Инструкция по оформлению диссертации доктора философии PhD «КазНИТУ имени К.И.Сатпаева». И.029-04-03.2.01 – 2023.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

AES	– Advanced Encryption Standard/симметричный алгоритм блочного шифрования
AC	– classical Adversary / классический злоумышленник
AQQ	– a quantum adversary / квантовый противник
BLOCK CIPHERS	– блочные шифры
CCA	– chosen ciphertext attack / атака с использованием выбранного шифротекста
CFB	– cipher feedback/режим обратной связи
CVP	– ближайшая векторная задача (closest vector problem)
DES	– Data Encryption Standard/симметричный алгоритм шифрования
DHIES	– Diffie-Hellman integrated encryption scheme/интегрированная схема шифрования Диффи-Хеллмана
DIGITAL SIGNATURES	– цифровые подписи
DLP	– задача дискретного логарифма (discrete logarithm problem)
ECIES	– elliptic curve integrated encryption scheme /схема интегрированного шифрования с эллиптической кривой
FFT	– Fast Fourier Transform/преобразования сигнала из временной области в частотную
GGH	– Goldreich-Goldwasser-Halevi/ассиметричная криптосистема основанная на решетках
HASH FUNCTIONS	– хеш функции
HTTPS	– Hypertext transfer protocol secure/шифрования для безопасности
IND	– INDistinguishability / Неотличимость зашифрованные сообщения
LATTICE	– решетки
LWE	– обучение с ошибками(learning with errors)
McEILECE	– асимметричный алгоритм шифрования
NIST	– National Institute of Standards and Technology/национальный институт стандартов и технологий
NTRU	– шифрования открытым ключом теория чисел
NTRUSign	– National Institute of Standards and Technology/шифрования с открытым ключом цифровой подписи
Negl(n)	– negligible function / незначимая функция
PK	– public key/открытый ключ
PKC	– Public Key Cryptography/криптография с открытым ключом
PKE	– шифрование с открытым ключом (public key encryption)

PUBLIC KEY SYSTEMS	– системы открытых ключей
QKD	– Quantum Key Distribution/квантовое распределение ключей
QROM	– quantum random oracle model / модель квантового случайного оракула
RSA	– Rivest–Shamir–Adleman/открытый ключ
SIS	– короткое целочисленное решение (short integer solution)
SK	– secret key/секретный ключ
SSL	– Secure Sockets Layer/уровень защищенных сокетов
STREAM CIPHERS	– потоковые шифры
SVP	– задача о кратчайшем векторе (shortest vector problem)
SWIFFT	– алгоритм хэширования
TLS	– Transport Layer Security/безопасность на транспортном уровне
XOR	– eXclusive OR/логическая операция ИЛИ ()

ВВЕДЕНИЕ

Актуальность темы исследования. В настоящее время традиционная криптография столкнулась с множеством проблем в связи с быстрым развитием современных вычислительных технологий, таких как квантовые вычисления и технологии облачных вычислений. Квантовые вычисления, обладая потенциалом экспоненциального ускорения обработки данных, представляют серьёзную опасность для широко используемых криптосистем с открытым ключом, таких как Rivest–Shamir–Adleman (RSA), криптография на эллиптических кривых (ECC) и схема шифрования Эль-Гамаля. Это связано с возможностью применения квантовых алгоритмов, в частности алгоритма Шора, который эффективно факторизует большие числа и решает задачи дискретного логарифмирования. Облачные вычисления, с другой стороны, порождают новые проблемы безопасности, связанные с конфиденциальностью, целостностью и приватностью данных. Хотя они предоставляют удобный доступ к огромным вычислительным ресурсам и хранилищам, они также вызывают такие проблемы, как несанкционированный доступ к данным, утечки и внутренние угрозы. Обеспечение безопасности данных в облаке требует надёжных криптографических механизмов, защищённых протоколов связи и строгого контроля доступа. Также необходимо отметить, что на сегодняшний день недостаточно разработаны и исследованы модели и методы которые объединяют асимметричные и симметричные методы шифрования для безопасной связи.

Зарубежные ученые, как Miklos Ajtai , Cynthia Dwork, M. Jason Hinek, Regev Oded, Dan Boneh, Chris Peikert, Shafi Goldwasser, Vadim Lyubashevsky, Silvio Micali (США), Zhenfeng Zhang, Yu Chen, Shanxiang Lyu, Xiaoyun Wang, Jintai Ding (Китай), Akhil Gupta, Kartikeya Walia, Shweta Agrawal (Индия), Orhan Sonmez, Alptekin Küpçü (Турция), Sergiy Gnatyuk (Украина) и другие исследователи добились высоких результатов в совершенствовании традиционной криптографии. Кроме того, из Caucasus университета Maksim Iavich и его коллег (Грузия) исследует область постквантовой криптографии уже многие годы и достигли конкурентноспособных показателей в разработке и улучшении криптографических систем.

А также следует отметить, что отечественные ученые проводили исследования в области криптографических систем. В работах ученых из Satbayev University – Ахметова Б.С., Казахского национального университета им. Аль-Фараби – Мусиралиевой Ш.Ж., Международного университета информационных технологий – Синчева Б.К. и других были отражены разработки криптографических систем и их применение. Кроме того, исследователи Института информационных и вычислительных технологий – Бияшев Р.Г., Нысанбаева С.Е., Капалова Н.А., Алгазы К., Сакан К., Дюсенбаев Д.С. занимаются изучением постквантовой криптографии, основанной на хеш-функциях. Они разрабатывают новые алгоритмы и исследуют их эффективность, что позволило достичь значительных улучшений в криптографических системах. Однако, несмотря на прогресс в данном направлении, постквантовая

криптография, основанная на решетках, остается недостаточно разработанной и практически не исследованной. В условиях стремительного развития квантовых вычислений и усложнения вычислительных технологий возрастает необходимость изучения альтернативных криптографических механизмов. Одним из перспективных направлений является криптография на основе решеток, которая рассматривается как эффективное решение против угроз, создаваемых квантовыми компьютерами. Её надежность обусловлена сложностью решёточных задач, таких как поиск короткого целочисленного решения (SIS)**.** Развитие данного подхода может стать ключевым этапом в создании устойчивых к квантовым атакам криптографических систем, что делает его актуальным направлением для дальнейших исследований. В данной работе представлена новая схема шифрования с открытым ключом, устойчивая к квантовым атакам, основанная на решётках и использующая принципы Эль-Гамаля. Схема базируется на задаче поиска короткого целочисленного решения (SIS) и включает протокол обмена ключами, основанный на данной задаче, что обеспечивает безопасность как от квантовых, так и от классических атакующих. Конструкция схемы проста и предоставляет решение для безопасной передачи данных в современных криптографических средах.

Исходя из вышеизложенного, можно прийти к выводу, что в настоящий момент необходимость в эффективных моделях и алгоритмах для улучшения защиты от квантовых атак с использованием принципов шифрования Эль-Гамаля на основе решёток является актуальной.

Цель диссертационной работы. Разработка метода создания улучшенной схемы постквантовой криптографии с открытым ключом на основе решеток, использующей принципов шифрования Эль-Гамаля.

Задачи исследования. Для реализации поставленных целей исследования решаются следующие вопросы:

1) Анализ популярных методов и алгоритмов традиционной криптографии и их устойчивости в условиях постквантовой криптографии.
2) Исследование схем распределения ключей на основе решёток.
3) Разработка математической модели эффективной и безопасной постквантовой схемы обмена ключами на основе решёток с использованием принципов шифрования Эль-Гамаля.
4) Разработка алгоритма и программного прототипа постквантовой криптосистемы с открытым ключом на основе решёток, использующей принципов шифрования Эль-Гамаля.
5) Исследование и тестирование эффективности предложенной постквантовой криптосистемы

   Объект исследования. Процесс криптогафических защиты данных с использованием схем с открытым ключом от классических и квантовых атак.

   Предмет исследования. Алгоритмы, модели криптографической защиты данных в поствквантовый период.

Методы исследования. Классическая криптография, теория решеток, оценки эффективности алгоритмов, экспериментальное тестирование.

Новизна исследовательской работы.

1. Разработана математическая модель эффективной и безопасной постквантовой схемы обмена ключами на основе решеток с использованием принципов Эль-Гамаля, что позволяет создать эффективные постквантовые схемы с открытым ключом для криптографических протоколов, систем аутентификации, финансовых систем, блокчейн и IoT-технологий.

2. Разработан алгоритм и прототип постквантовой схемы с открытым ключом на основе решеток, использующей принципы шифрования Эль - Гамаля, что позволило повысить скорость генерации ключей шифрования в 240-583 раз (по сравнению с аналогами – LWE, Ring-LWE), и обеспечило стойкость к квантовым атакам (по сравнению с классической схемой Эль-Гамаля).

Теоретическое и практическое значение работы. Теоретическая значимость исследовательской работы заключается в том, что она способствует продвижению в разработке криптографических систем, устойчивых к квантовым атакам, и расширяет возможности применения математических методов в области информационной безопасности. Практическая значимость исследовательской работы заключается в применении разработанного алгоритма и программного обеспечения для дальнейшего использования в развитии других технологий, обеспечивающих защиту как от квантовых, так и от классических угроз. Разработанная схема отличается простотой и высокой скоростью работы, обеспечивая безопасную передачу данных в современных криптографических средах. Достигнутые результаты подчеркивают продолжающуюся эволюцию криптографии в решении задач, связанных с квантовыми вычислениями и технологиями облачных вычислений.

Основное положение, выносимое на защиту.

1. Разработана математическая модель эффективной и безопасной постквантовой схемы обмена ключами на основе решеток с использованием принципов Эль-Гамаля, что позволяет создать эффективные постквантовые схемы с открытым ключом для криптографических протоколов, систем аутентификации, финансовых систем, блокчейн и IoT-технологий.

2. Разработан алгоритм и прототип постквантовой схемы с открытым ключом на основе решеток, использующей принципы шифрования Эль - Гамаля, что позволило повысить скорость генерации ключей шифрования в 240-583 раз (по сравнению с аналогами – LWE, Ring-LWE), и обеспечило стойкость к квантовым атакам (по сравнению с классической схемой Эль-Гамаля).

Степень достоверности и апробация результатов. Исследования и результаты, относящиеся к теме диссертации, были представлены на основе следующих публикаций:

1. Dana Sairangazhykyzy Amirkhanova, Maksim Iavich and Orken Mamyrbayev.Cryptography 2024, *8*(3),31. https://doi.org/10.3390/cryptography8030031 (Scopus, процентиль 66). “Lattice- based Post-Quantum Public Key Encryption Scheme Using ElGamal’s Principles”.

2. Әмірханова Д.С., Мамырбаев О.Ж., Вестник КазНПУ им Абая. Серия: физико-математические науки Том 83 № 3(2023). “Cryptographic analysis of the scheme of polylinear cryptography”.

3. Әмірханова Д.С., Мамырбаев О.Ж., Вестник НАН РК: Серия: Physico-Mathematical Series ISSN 1991-346X Volume 3. № 351 (2024). “El-Gamal’s Cryptographic Algorithm: Mathematical Foundations, Applications and Analysis”.

4. Әмірханова Д.С., Мамырбаев О.Ж., Вестник ВКТУ: Серия: Информационно - коммуникационные технологии ISSN 1561-4212 № 1(2025) “Research And Development of a Cryptography Algorithm Based on Polylinear Algebra Using Blockchain Methodology”.

Личный вклад исследователя. Докторант самостоятельно выполнил и решил задачи диссертационной работы. Разработал математическую модель эффективной и безопасной постквантовой схемы обмена ключами на основе решеток с использованием принципов Эль-Гамаля что позволило создать эффективные постквантовые схемы с открытым ключом. Выполнил экспериментальное тестирование и оценку эффективности разработанного алгоритма и программного прототипа.

Связь темы диссертации с планами научно-исследовательской работы. Научно-исследовательские работы по диссертации были проведены в РГП на ПХВ 'Институт информационных и вычислительных технологий Комитета науки МОН РК и в Caucasus University.

Публикация основных результатов диссертационного исследования. По теме диссертационной работы опубликовано 3 статьи в журналах, рекомендованных Комитетом по контролю в сфере образования и науки МОН РК, 1 статья опубликована в изданиях имеющие ненулевой импакт-фактор, индексирумых базой Scopus и Web of Science:

1. Dana Sairangazhykyzy Amirkhanova, Maksim Iavich and Orken Mamyrbayev.*Cryptography* 2024, *8*(3),31. https://doi.org/10.3390/cryptography8030031 (Scopus, процентиль 66). “Lattice-Based Post-Quantum Public Key Encryption Scheme Using ElGamal’s Principles”.

2. Әмірханова Д.С., Мамырбаев О.Ж., Вестник КазНПУ им Абая. Серия: физико-математические науки Том 83 № 3(2023). “Cryptographic analysis of the scheme of polylinear cryptography”.

3. Әмірханова Д.С., Мамырбаев О.Ж., Вестник НАН РК: Серия: Physico-Mathematical Series ISSN 1991-346X Volume 3. № 351 (2024).“El-Gamal’s Cryptographic Algorithm: Mathematical Foundations, Applications And Analysis”.

4. Әмірханова Д.С., Мамырбаев О.Ж., Вестник ВКТУ: Серия: Информационно - коммуникационные технологии ISSN 1561-4212 № 1(2025) “Research And Development of a Cryptography Algorithm Based on Polylinear Algebra Using Blockchain Methodology”.

Структура и объем диссертационной работы. Диссертационная исследовательская работа состоит из введения, 3 разделов, заключения, списка литературы из 85 наименований и 1 приложения. Работа изложена на 96 страницах и содержит 39 рисунков, 1 таблицу.

1 КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ НА ОСНОВЕ РЕШЕТОК

1.1 Обзор современных технологий криптографических систем

На протяжении тысячелетий криптография, искусство тайного написания или решения кодов, была важной частью человеческого общения. С древних времен люди шифровали информацию для защиты своих секретов и обеспечения безопасной связи [1]. По всему миру ежедневно люди используют криптографию для защиты данных и информации, но большинство из них не знают, что они ее используют. Кроме того, криптография считается чрезвычайно полезной, потому что она очень хрупка, поскольку любая ошибка в программировании или спецификациях может привести к краху криптографических систем. Сьюзан и др. [2] отметили, что преподавание компьютерной безопасности является постоянной целью, поскольку сетевая и компьютерная безопасность – это новая и быстро развивающаяся технология в области информатики. Курсы по безопасности уделяют особое внимание алгоритмическим и математическим элементам, таким как методы хеширования и шифрования, поскольку взломщики находят новые способы взлома сетевых систем. Отман О. Халифа и др. [3] продемонстрировали основные концепции, характеристики и цели криптографии. Они обсудили, что в наш век, то есть век информации, коммуникация способствовала развитию технологий и, следовательно, играет важную роль, требующую защиты и конфиденциальности, гарантировано, когда данные передаются через средство связи. Нитин Джирван и др. [4] отметили, что передача данных зависит главным образом от передачи цифровых данных, при которой безопасность данных имеет высший приоритет при использовании алгоритмов шифрования, чтобы данные могли безопасно доходить до предполагаемых пользователей, не подвергаясь риску. Они также продемонстрировали различные криптографические методы, используемые в процессе передачи данных, такие как симметричные и асимметричные методы. В обзоре сетевой безопасности и криптографии Сандип Тайал и др. [5] отметили, что с появлением социальных сетей и коммерческих приложений организации по всему миру ежедневно производят огромные объемы данных. Это делает информационную безопасность огромной проблемой с точки зрения обеспечения гарантированной передачи данных через Интернет. Поскольку все больше пользователей подключаются к Интернету, эта проблема еще раз демонстрирует необходимость методов криптографии. В этом документе представлен обзор различных методов, используемых сетями для повышения безопасности, таких как криптография. Анджула Гупта и др. [6] продемонстрировали происхождение и значение криптографии, а также то, как информационная безопасность стала сложной проблемой в области компьютеров и коммуникаций. Помимо демонстрации криптографии как способа обеспечения идентификации, доступности, целостности, аутентификации и конфиденциальности пользователей и их данных путем обеспечения безопасности и конфиденциальности, в этом документе также

представлены различные асимметричные алгоритмы, которые дали нам возможность защищать и защищать данные. Исследование, проведенное Каллас Дж. [7], затрагивало такие темы, как криптография, технологии повышения конфиденциальности, правовые изменения, связанные с криптографией, надежность и технологии, используемые для повышения конфиденциальности. Он отметил, что именно то, как общество использует криптографию, определит будущее криптографии, которое зависит от правил, действующих законов и обычаев, а также от того, чего общество ожидает от нее. Он отметил, что в области криптографии существует множество пробелов, которые будущим исследователям предстоит заполнить. Кроме того, будущее криптографии зависит от системы управления, генерирующей надежные ключи, гарантирующей, что только нужные люди с правильными ключами смогут получить доступ, а другие, у которых нет ключей, не смогут. Продолжая тему целей криптографии, Джеймс Л. Мэсси [8] отметил, что есть две цели, которых криптография стремится достичь как таковые: подлинность и/или секретность. Что касается безопасности, которую она обеспечивает (может быть как практической, так и теоретической), он обсуждал как теорию теоретической секретности Шеннона, так и теорию теоретической аутентичности Симмона. Наконец, Шнайер [9] пришел к выводу, что секретность безопасности как хорошая вещь – это миф и что безопасности нехорошо быть секретной, поскольку безопасность, полностью полагающаяся на секретность, может быть хрупкой. Если бы эта тайна была утеряна, восстановить ее было бы невозможно. Шнайер далее заявил, что криптография, основанная на коротких секретных ключах, которые можно легко передавать и изменять, должна основываться на базовом принципе, заключается в том, что криптографические алгоритмы должны быть одновременно надежными и общедоступными, чтобы обеспечить хорошую безопасность. Единственный надежный способ добиться большего улучшения безопасности – это обеспечить общественный контроль. Дженнаро Р. [10] обсудил случайность в криптографии и объяснил, что случайный процесс – это процесс, последствия которого неизвестны, и упомянул, что именно поэтому случайность жизненно важна в криптографии, поскольку она обеспечивает способ создания информации, что злоумышленник не может изучить или предсказать это. Садхан С.Б. [11] указал на основные процессы и тенденции в областях криптографии со времен Юлия Цезаря до современной эпохи, а также упомянул нынешний статус арабских промышленных и академических усилий в этой области в прошлом, то есть связанные с существующими криптографическими методами и поиск новых методов оценки безопасности информации. Со временем криптография совершенствовалась и адаптировалась для решения социальных потребностей и изменяющихся технологий. Криптография стала важным инструментом для обеспечения безопасной связи в компьютерных сетях в современную эпоху. Развитие криптографии с открытым ключом в 1970-х годах произвело революцию в этой области, позволив безопасно обмениваться информацией без необходимости создания секретного ключа заранее. Сегодня криптография имеет решающее значение для многих

областей нашей жизни, от защиты транзакций онлайн-банкинга до защиты конфиденциальных сообщений правительства. Основная идея криптографической системы заключается в том, чтобы шифровать данные или информацию, чтобы сохранить конфиденциальность информации таким образом, чтобы никто, кроме вас, не смог ее понять [12]. Двумя наиболее распространенными способами использования криптографии являются либо передача данных через незащищенный канал, такой как Интернет, либо обеспечение того, чтобы люди, не имевшие доступа, не знали, на что они смотрят. В криптографии скрытая информация обычно называется «открытым текстом», а процесс маскировки открытого текста называется «шифрованием». Этот процесс достигается с помощью ряда правил, известных как «алгоритмы шифрования», а сам процесс шифрования начинается с «ключа шифрования», который затем передается алгоритму шифрования в качестве входных данных, а затем используется «алгоритм дешифрования» (рисунок 1).

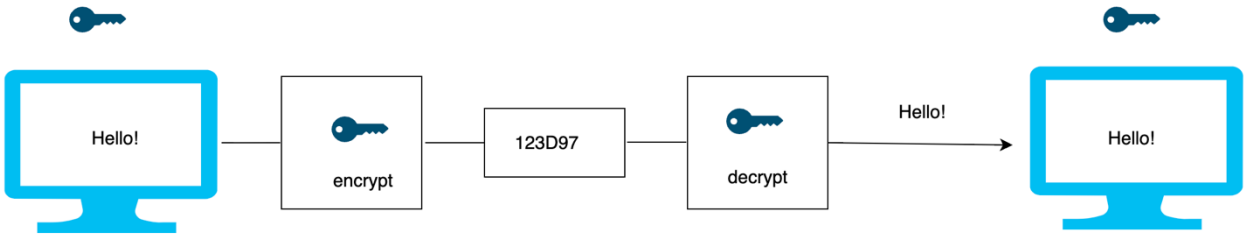


Рисунок 1 – Концепция криптографии

Современная криптография является основой компьютерной безопасности. Она базируется на различных математических концепциях, таких как теория чисел, теория сложности вычислений, теория вероятности. Виды современнных криптографических систем: симметричная и асимметричная.

Криптография с *симметричным ключом*, иногда называемая криптографией с секретным ключом, использует один и тот же ключ для дешифрования и шифрования данных [13]. Обычно для шифрования больших объемов данных используется этот метод, который работает быстрее, чем криптография с асимметричным ключом (рисунок 2). Симметричные алгоритмы делятся на два основных вида:

- Блочные шифры (Block Ciphers) - обрабатывают данные фиксированными блоками (например, 64 или 128 бит) и применяют различные математические преобразования для их защиты. Примеры криптографии с симметричным ключом включают расширенный стандарт шифрования (AES), стандарт шифрования данных (DES) и тройной DES (3DES), BlowFish, Twofish, Camellia.

- Потоковые шифры (Stream Ciphers) - работают с данными по одному биту или байту и генерируют ключевой поток, который комбинируется с исходными данными с помощью операции XOR. Примерами являются RC4, A5/1, A5/2, ChaCha20, Salsa20.

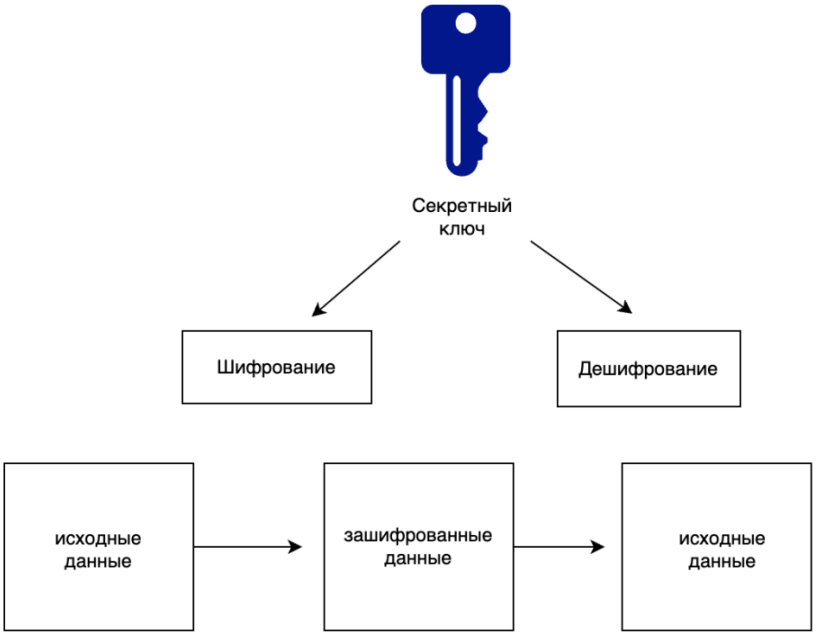


Рисунок 2 – Симметричное шифрование

В соответствии с рисунком 3, в начале 1970-х годов IBM разработала блочный шифр с симметричным ключом, известный как *стандарт шифрования данных* (DES), который впоследствии был утвержден Национальным бюро стандартов (NBS) в качестве федерального стандарта шифрования. Впоследствии DES был заменен более современными шифрами, такими как Advanced Encryption Standard (AES), но он все еще играет важную роль в истории современной криптографии [14, 15]. На каждом этапе процесса шифрования и дешифрования данных DES использует 56-битный ключ для шифрования и дешифрования данных. Процесс шифрования состоит из нескольких раундов замены и перестановки, которые преобразуют блок открытого текста в блок зашифрованного текста. Кроме того, ключ используется для создания серии дополнительных ключей, которые используются на каждом этапе процесса. Современные стандарты считают, что ключ DES слишком короткий, потому что его можно взломать с помощью грубой атаки с использованием современной вычислительной мощности. Тем не менее для повышения безопасности исходного алгоритма DES были разработаны несколько модификаций и расширений. Тройной DES (3DES) – модификация, которая использует алгоритм DES трижды с разными ключами. Эта модификация эффективно увеличивает длину ключа до 168 бит.

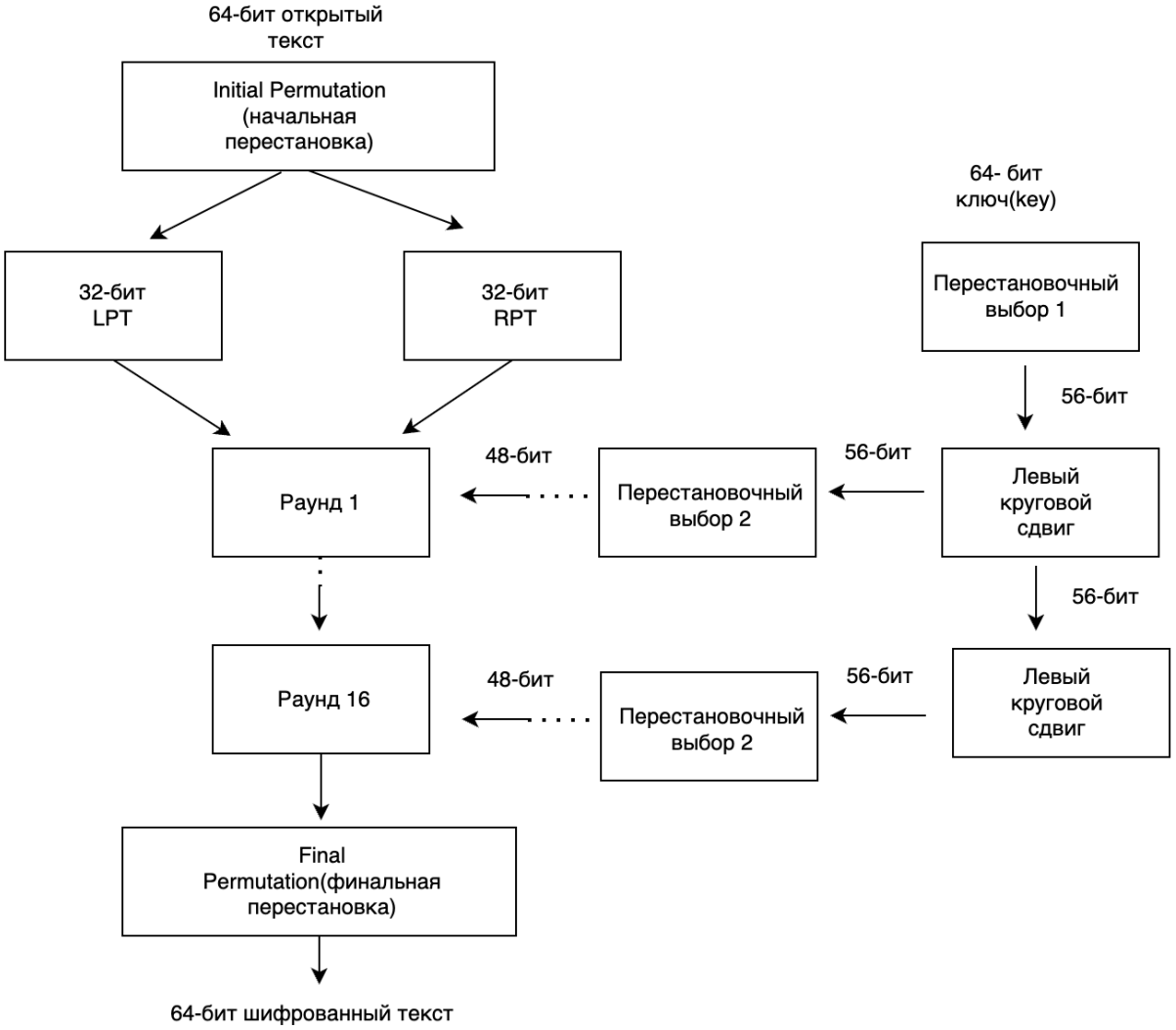


Рисунок 3 - Стандарт шифрования данных (DES)

С появлением квантовых вычислений безопасность DES еще больше снизилась из-за его малой длины ключа и уязвимости к квантовым атакам, в частности, атаке Гровера. Алгоритм Гровера делает возможным взлом DES за секунды при наличии квантового компьютера. DES полностью небезопасен даже перед классическими атаками и тем более перед квантовыми атаками**.**

Тройной DES (3DES) – это блочный шифр с симметричным ключом, модификация исходного шифра стандарта шифрования данных (DES). 3DES часто используется в устаревших системах, которые требуют надежного шифрования, но не могут использовать более современные шифры, такие как расширенный стандарт шифрования (AES).

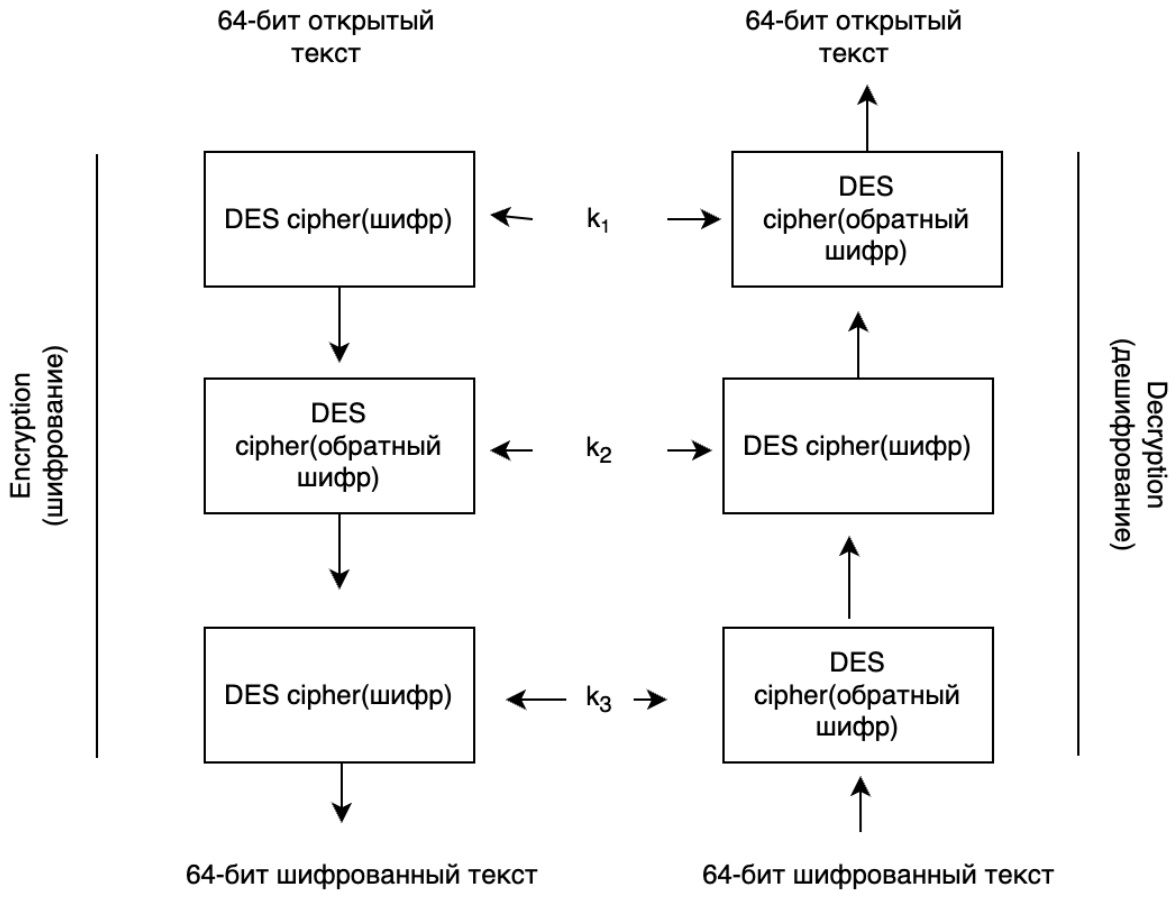


Рисунок 4 - Тройной стандарт шифрования данных (3DES)

В соответствии с рисунком 4, 3DES использует шифр DES трижды, используя два или три различных ключа для каждого блока данных. Независимые ключи могут использоваться для двух или всех трех шифрований. Этот метод известен как «варианты ввода ключей». Как и первый DES, 3DES поддерживает блоки данных с 64 битами. Тем не менее 3DES является гораздо более надежным ключом, чем DES, поскольку его длина составляет 168 бит. 3DES более безопасен, чем однократное шифрование DES, потому что он использует шифр DES трижды. 3DES может иметь недостаток в том, что он очень медленный по сравнению с современными шифрами, такими как AES. Тем не менее 3DES продолжает использоваться в устаревших системах и является надежным и безопасным шифром для многих приложений. Кроме того, его реализация не требует значительных обновлений оборудования и совместим с существующими реализациями DES [25]. Однако NIST запретил использование 3DES с 2023 года из-за низкой криптостойкости**.** Организации, должны перейти на AES-256 или постквантовые схемы.

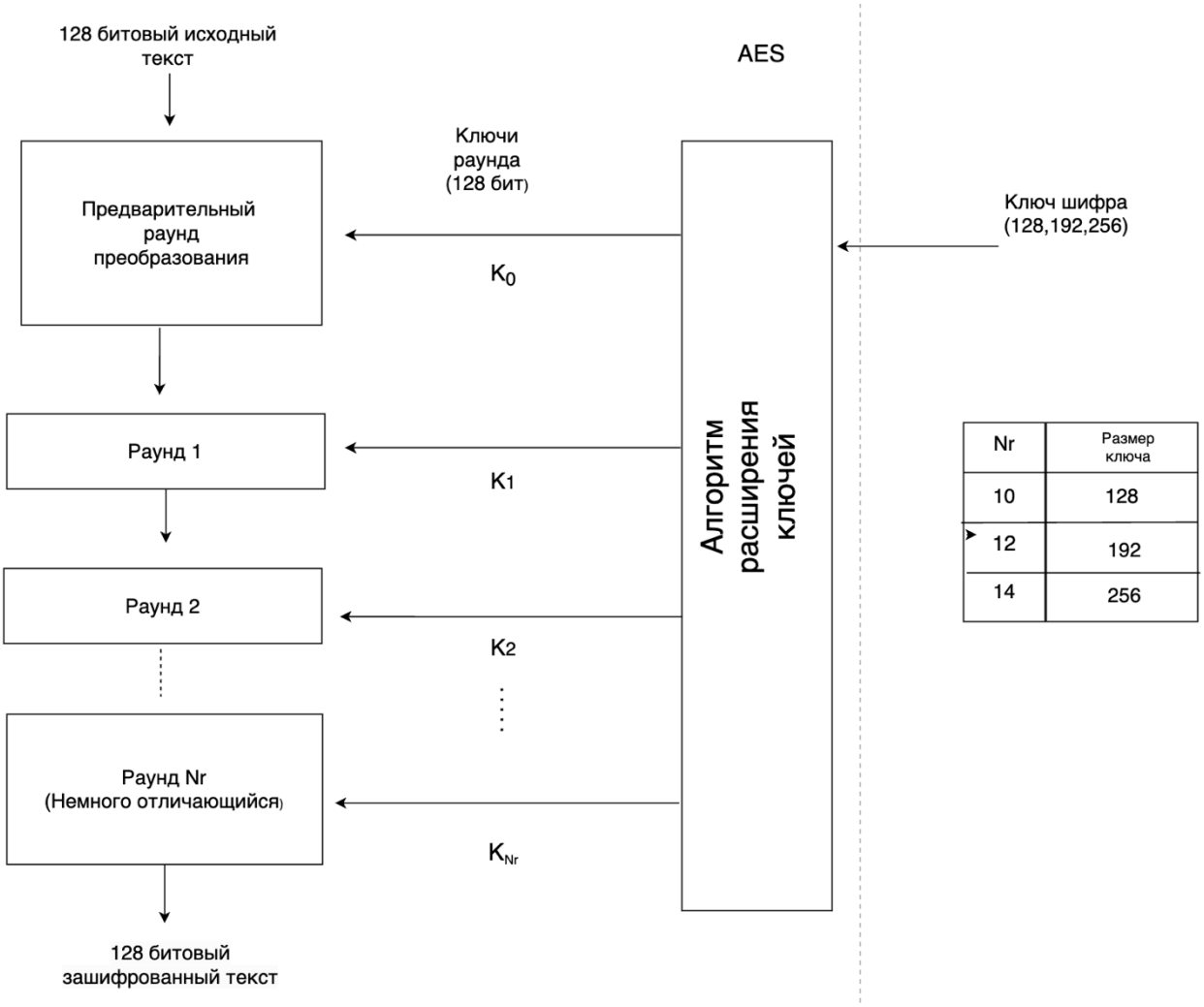


Рисунок 5 - Расширенный стандарт шифрования (AES)

В соответствии с рисунком 5, широко используется для шифрования данных блочный шифр с симметричным ключом, известный как *расширенный стандарт шифрования* (AES). Он был создан в конце 1990-х годов для замены устаревшего шифра стандарта шифрования данных (DES), который был подвержен атакам методом перебора из-за сравнительно короткой длины ключа. AES использует размер ключа 128, 192 или 256 бит и работает с блоками фиксированной длины по 128 бит. Чтобы превратить блок открытого текста в блок зашифрованного текста, он использует ряд операций перестановки и замены. Алгоритм «расписания ключей» используется AES для создания последовательности ключей, которые используются в каждом цикле процесса шифрования [17]. Для шифрования и дешифрования данных система AES использует определенные методы. Размер используемого ключа определяет количество раундов, которые он проходит. 10 раундов необходимы для 128-битных ключей, 12 раундов для 192-битных ключей и 14 раундов для 256-битных ключей. Либо исходный открытый текст, либо завершенный зашифрованный текст являются результатом. AES может обрабатывать данные длиной 128 бит, разделенные на четыре основных операционных блока. Эти блоки рассматриваются как массив байтов, называемый состоянием, и расположены в виде матрицы порядка 4 на 4. Этап AddRoundKey является начальным этапом

как шифрования, так и дешифрования. Выходные данные проходят девять основных раундов до финального раунда. Каждый из девяти раундов содержит четыре преобразования: суббайты, сдвиговые строки, микс-столбцы и добавление ключа раунда. В десятом и последнем раунде преобразование «Смешанный столбец» отсутствует. Алгоритм AES состоит из четырех разных преобразований, которые контролируют каждый раунд шифрования. Каждый 8-битный байт 128-битного блока данных преобразуется в новый блок с помощью 8-битного блока замены, известного как Rijndael Box, в первом преобразовании, называемом Substitute Byte. Поворотные строки – это второе преобразование, которое переупорядочивает байты в последних трёх строках состояния в соответствии с их расположением в строках. Умножение каждого столбца состояния на фиксированную матрицу – это третий тип преобразования. Наконец, добавочный раундовый ключ выполняет XOR 128-битного текущего состояния с помощью четвертого преобразования. Это преобразование обратимо и имеет обратный эффект. AES превосходит DES по значительной длине ключа, что значительно усложняет взлом с использованием грубой силы. Кроме того, AES быстрее и эффективнее DES, поэтому его можно использовать для многих задач, от защиты данных на жестких дисках до шифрования интернет-трафика. AES используется для многих целей, в том числе правительственных и военных. Многие веб-браузеры используют его для защиты HTTPS-соединений. Одной из потенциальных слабостей AES является то, что он может быть уязвим для атак по побочным каналам, использующих уязвимости в физической реализации шифрования [18]. Однако это относительно необычный вектор атаки, и его можно смягчить путем тщательного проектирования и реализации. В целом, AES – это эффективный и надежный шифр, который стал стандартом шифрования данных во многих приложениях. Однако основной угрозой для AES является атака Гровера, которая ускоряет полный перебор ключей. В результате AES-128 становится недостаточно безопасным в долгосрочной перспективе, тогда как AES-256 сохраняет устойчивость даже в условиях квантовых атак.

Blowfish - это симметричный блочный шифр, разработанный Брюсом Шнайером в 1993 году. Он был предложен как альтернатива устаревающему Data Encryption Standard (DES) и предназначен для быстрого и безопасного шифрования данных в программных системах. Blowfish отличается высокой скоростью работы и гибкостью, позволяя использовать ключи различной длины. Однако с развитием квантовых вычислений возникает необходимость переоценки его безопасности в новых условиях. Безопасность Blowfish в квантовой эре сомнительна из-за маленького размера блока и снижения стойкости ключей под атакой Гровера.

Twofish - блочный шифр, разработанный командой Брюса Шнайера в 1998 году как финалист конкурса AES [19]. Он является наследником Blowfish и предлагает улучшенные характеристики безопасности и производительности, которые представлены ниже.

Размер блока: 128 бит.

Длина ключа: 128, 192 или 256 бит.

Количество раундов: 16.

Основные технологии:

- сеть Фейстеля;
- предвычисленные ключевые таблицы (key-dependent S-boxes);
- линейное смешивание с помощью матрицы MDS.

Twofish поддерживает эффективную работу на малопроизводительных устройствах и защищен от различных типов криптоанализа, таких как дифференциальный и линейный анализ. Квантовый алгоритм Гровера снижает стойкость симметричных шифров, сокращая сложность атаки. Однако Twofish-256 остается безопасным перед квантовыми атаками.

Криптография с *асимметричным ключом*, также известная как криптография с открытым ключом, использует два отдельных ключа для шифрования и дешифрования данных (рисунок 6). В то время как один ключ является конфиденциальным и используется для расшифровки данных, другой является общедоступным и используется для шифрования данных. Хотя криптография с асимметричным ключом более безопасна, она медленнее, чем криптография с симметричным ключом. Примеры криптографии с асимметричным ключом включают RSA, Диффи-Хеллмана, ECC, схема шифрования Эль-Гамаля [20].

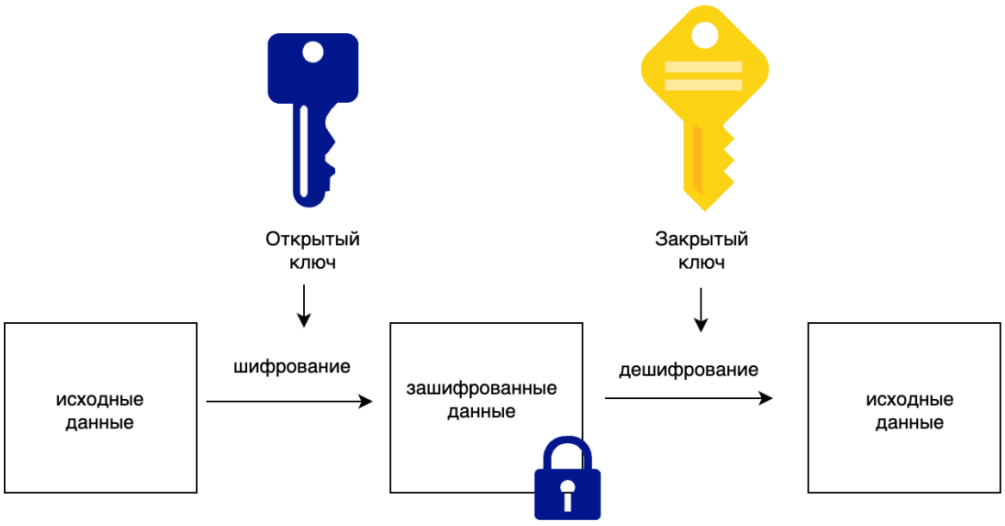


Рисунок 6 – Асимметричное шифрование

RSA – это алгоритм шифрования с открытым ключом, который был впервые представлен в 1977 году Роном Ривестом, Ади Шамиром и Леонардом Адлеманом. RSA был назван в честь трех изобретателей. Несмотря на то, что он широко используется для безопасной передачи данных, он является важным компонентом современной криптографии.

В основе шифрования RSA лежит математическая идея простых чисел. Алгоритм использует как закрытый, так и открытый ключ [21]. Закрытый ключ хранится в секрете, а открытый ключ можно передать кому угодно. Данные шифруются с помощью открытого ключа, а расшифровываются с помощью

закрытого ключа. Сложность факторизации больших простых чисел определяет безопасность RSA. Чтобы сгенерировать пару ключей RSA, два больших простых числа умножаются на большое составное число. Ключ RSA состоит из этого составного числа. Модуль работает в качестве открытого ключа, поскольку его коэффициенты хранятся в секрете. Протоколы защищенной связи, такие как Secure Socket Layer (SSL) и Transport Layer Security (TLS), используют RSA для безопасной передачи файлов, защиты электронной почты и безопасного просмотра веб-страниц. Кроме того, он используется для цифровых подписей: открытый ключ подписывающего лица используется для создания подписи, а закрытый ключ получателя используется для проверки подписи (рисунок 7).

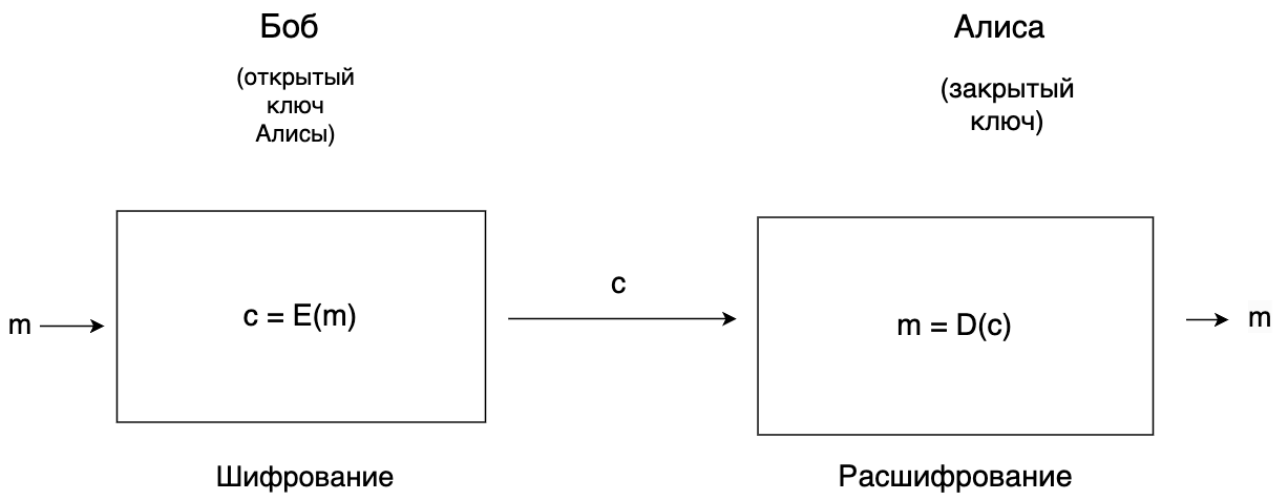


Рисунок 7 – Алгоритм RSA

Уязвимость RSA к атакам, основанным на квантовых вычислениях, является одним из его недостатков. Благодаря алгоритму Шора, факторизация больших чисел может быть выполнена за полиномиальное время. Алгоритм Шора экспоненциально ускоряет решение задачи факторизации по сравнению с классическими методами. Это делает RSA полностью уязвимым к атакам с использованием достаточно мощного квантового компьютера. [22].

Протокол обмена ключами Диффи-Хеллмана был разработан Уитфилдом Диффи и Мартином Хеллманом в 1976 году и широко используется в протоколах интернет-безопасности для установления общего секрета между двумя сторонами по незащищенному каналу. Протокол обмена ключами Диффи-Хеллмана позволяет двум людям, обычно называемым Алисой и Бобом, договориться об общем секрете, который они могут использовать для шифрования своего общения. Протокол работает с использованием модульной арифметики и математических свойств простых чисел. Алиса и Боб сходятся во мнении о большом простом числе и примитивном корне по модулю этого простого числа [23]. Затем Алиса и Боб создают секретное число и выполняют серию модульных возведений в степень, чтобы создать открытый ключ. Они

обмениваются этими открытыми ключами, что позволяет каждому из них вычислить общий секрет, используя свой секретный номер и открытый ключ другого, что позволяет использовать симметричное шифрование, которое работает лучше, когда используется открытый ключ для шифрования больших объемов информации. Протокол обмена ключами Диффи-Хеллмана широко используется в протоколах интернет-безопасности, таких, как Transport Layer Security (TLS), который защищает соединения HTTPS. Протокол обмена ключами Диффи-Хеллмана особенно полезен для безопасного общения через Интернет, поскольку позволяет двум сторонам установить общий секрет, даже если злоумышленник подслушивает их. Возможность атаки «человек посередине», когда злоумышленник перехватывает и изменяет открытые ключи, которыми Алиса и Боб обмениваются, является одной из потенциальных слабостей протокол обмена ключами Диффи-Хеллмана (рисунок 8). Для проверки подлинности открытых ключей протокол обмена ключами Диффи-Хеллмана обычно используется вместе с другими криптографическими протоколами, такими как цифровые подписи, чтобы снизить этот риск. В целом, протокол обмена ключами Диффи-Хеллмана - это популярный и важный протокол криптографии, который позволяет двум сторонам установить общий секрет по незащищенному каналу. Благодаря своей модульной конструкции, арифметике и простым числам протокол обмена ключами Диффи-Хеллмана является надежным и эффективным методом установления безопасной связи [24]. Однако алгоритм Шора, работающий на квантовом компьютере, способен решать задачу дискретного логарифма за полиномиальное время. Это означает, что квантовый компьютер может эффективно вычислять секретные ключи, используемые в протоколе Диффи - Хеллмана, на основе публичных параметров, передаваемых по открытому каналу связи.

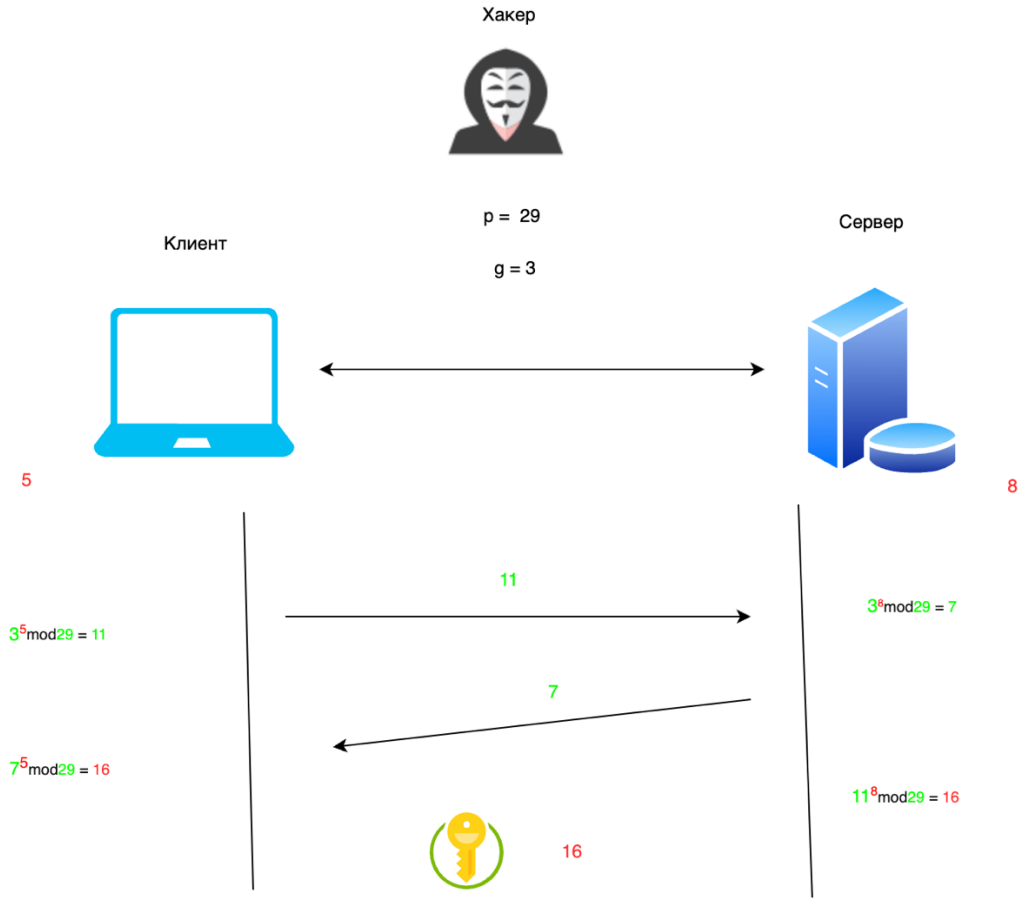


Рисунок 8 - Протокол обмен ключами Диффи-Хеллмана

В данном разделе приведен обзор и анализ существующих современных технологий криптографических систем, а также рассмотрены уязвимости от квантовых атак. Далее будет рассмотрена область применения криптографических методов.

1.1.1 Области применения современных криптографии

Криптография – наука, изучающая проблему обеспечения безопасности информационного сообщения с помощью тайной записи и дешифрования. Криптографы также заинтересованы в безопасности аутентификации и идентификации пользователей компьютерной системы, обменивающихся информацией. Для криптографической защиты данных используются различные средства и методы. Существует два принципа криптографической защиты: принцип сохранения конфиденциальности информации и принцип сохранения целостности данных. На них основана безопасность применяемых средств защиты [25,26]. Криптография тесно связана с дисциплинами криптологии и криптоанализа. Он включает в себя такие методы, как микроточки, слияние слов с изображениями и другие способы сокрытия информации при хранении или передаче. В современном компьютерно-ориентированном мире криптография чаще всего связана с шифрованием открытого текста (обычного текста, иногда

называемого открытым текстом) в зашифрованный текст (процесс, называемый шифрованием), а затем обратно (известный как дешифрование). Люди, которые практикуют эту область, известны как криптографы. Современная криптография придерживается следующих четыре цели:

- *Конфиденциальность.* Информация не может быть понята теми, для кого она была не предназначена.

- *Честность.* Информация не может быть изменена при хранении или передаче между отправителем и предполагаемым получателем без обнаружения изменения.

- *Неотречение.* Создатель/отправитель информации не может на более позднем этапе отрицать свои намерения по созданию или передаче информации.

- *Аутентификация.* Отправитель и получатель могут подтвердить личность друг друга и происхождение/назначение информации.

Криптосистемы – это протоколы и процедуры, которые соответствуют некоторым или всем вышеупомянутым требованиям. Часто считается, что криптосистемы включают только математические операции и компьютерные программы; однако они также контролируют поведение людей, например, выбирая трудно угадываемые пароли, выходя из неиспользуемых систем и избегая разговоров о конфиденциальных процедурах с людьми [27]. Современные технологии и гаджеты, которые уже давно стали неотъемлемой частью жизни человека, тесно связаны с криптографией. Другими словами, криптография всегда присутствует в цифровых и информационных технологиях. Работа в этой области становится все более привлекательной. Криптография играет жизненно важную роль в программировании и информационной безопасности. Это особенно верно для больших проектов. Современные программные продукты постоянно обмениваются данными, поэтому всегда есть вероятность утечки данных. Криптография предотвращает перехват данных. Мошенники не могут получить доступ к информации, потому что для каждой операции существуют отдельные протоколы безопасности. Это не единственные области, в которых используется криптография. Например, она позволяет пользователям разделять данные, и они могут получить к ним доступ, только объединившись (рисунок 9).

Рисунок 9 - Области применения современных криптографии

Этот метод часто используется во время голосования. Это лишь некоторые из наиболее распространенных применений криптографии:

Финансовые операции. Все транзакции, начиная от оплаты чашки кофе банковской картой и заканчивая переводами средств на счёт родственника, кодируются банками с помощью криптографических методов защиты информации.

Сохранность личных данных. Сайты, которые собирают личную информацию пользователей (имя, пол, возраст, контакты), используют шифрование данных. Это особенно важно для ресурсов, которые собирают паспортные данные и реквизиты банковских карт.

Конфиденциальность общения. Большинство популярных мессенджеров шифрует переписки пользователей, чтобы их могли прочитать только участники диалога. Так, телеграм изначально позиционировался как мессенджер с надёжной защитой конфиденциальности сообщений, а WhatsApp начал шифровать данные в 2021 году. Кроме того, сотовые операторы используют криптографию для кодирования данных телефонных переговоров[28].

Безопасность подключения. Без криптографии было бы невозможно пользоваться публичным Wi-Fi. При подключении к сети, например, в метро пользователь может быть уверен в том, что его данные не попадут к третьим лицам именно благодаря шифрованию.

Электронный документооборот. Криптография существенно упростила обмен документами благодаря тому, что бухгалтерская отчётность, электронные подписи и многие другие данные хранятся и передаются в зашифрованном виде, а значит, подлинны и надёжно защищены от третьих лиц.

Государственные службы и военные части. Любые заочные контакты государственных деятелей и глав стран, будь то телефонные переговоры или переписка, кодируются спецслужбами в целях национальной безопасности также с использованием криптографических методов.

1.1.2 Квантовые атаки на современную криптографию

Шифрование является жизненно важной технологией в нашем современном мире, поскольку экономика и общества процветают и зависят от распространения свободного и безопасного доступа к информации через Интернет. Шифрование является важным компонентом ряда компьютерных технологий, которые позволяют безопасно обмениваться частной информацией между общественными каналами. Кроме того, шифрование обеспечивает национальную безопасность, защищая конфиденциальную информацию и позволяя людям сохранять конфиденциальность и целостность своих данных, независимо от того, признается это или нет [29]. Тем не менее, шифрование полезно настолько, насколько оно безопасно, и многие ранее безопасные технологии шифрования могут вскоре стать уязвимыми для атак со стороны новых высокотехнологичных квантовых вычислительных систем. Поскольку они используют квантово-механическую структуру мелкомасштабной Вселенной для манипуляции и хранения данных, квантовые компьютеры могут выполнять вычисления, которые никогда не были возможными с любым другим типом компьютерной системы. Хотя эти способности, несомненно, приведут к значительным прорывам в отраслях обработки и анализа данных, развитие квантовых компьютеров имеет неприятный побочный эффект. Огромная вычислительная мощность и подходящий размер квантового компьютера сделают многие основные криптографические примитивы, которые делают возможным современное шифрование, небезопасными. Однако невозможно предположить, что квантовые компьютеры приведут к полному краху криптографии. Это связано с тем, что многие существующие криптосистемы защищены от опасностей, вызванных технологией квантовых вычислений. Изучение квантовых компьютеров, хотя они обычно не используются на практике, будет иметь решающее значение для разработки алгоритмов, необходимых для выживания криптографии в постквантовую эпоху [30]. Взлом современных криптографических систем с помощью квантовых вычислений представляет собой угрозу, которую представляют квантовые атаки. Квантовые атаки подразделяются на:

Алгоритм Шора: этот алгоритм эффективно решает задачи дискретного логарифма и факторизует большие числа. Это ставит под угрозу криптографические системы, такие как RSA и эллиптические кривые, поскольку они зависят от сложности этих процессов (рисунок 10).

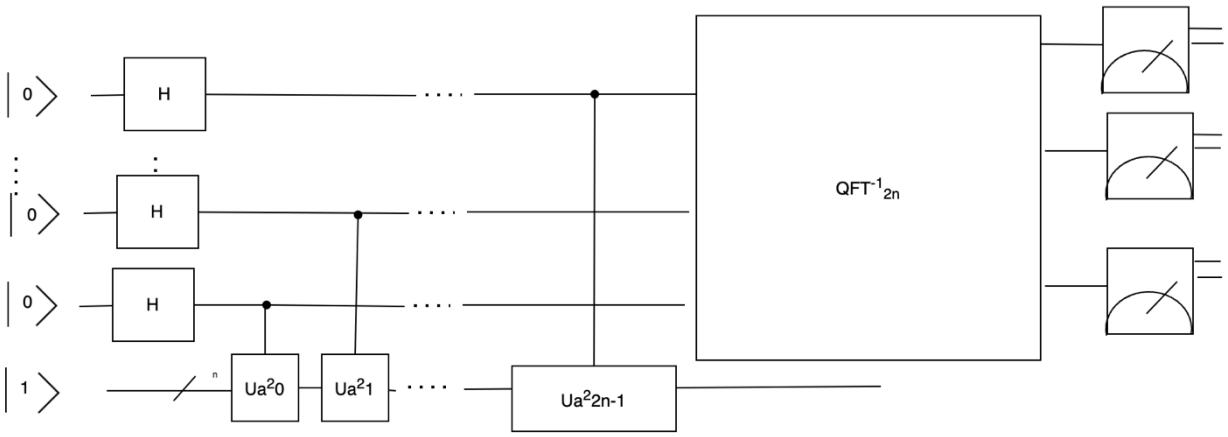


Рисунок 10 – Алгоритм Шора

Алгоритм Гровера: этот алгоритм может ускорить поиск в неструктурированных данных, что улучшает эффективность атак на симметричные криптосистемы (например, AES). Хотя время атаки можно сократить вдвое, сильные ключи, такие как 256-битные, все еще достаточно безопасны (рисунок 11).

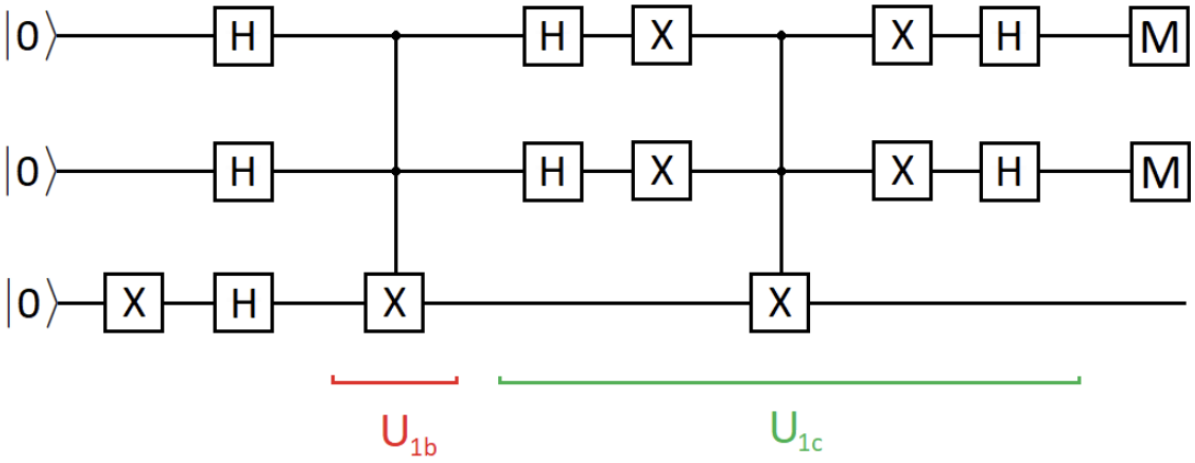


Рисунок 11 – Алгоритм Гровера

Следующее может произойти, если квантовые компьютеры станут достаточно мощными: Слом существующих систем:

1. Многие текущие системы шифрования, такие как HTTPS и электронная подпись, могут быть под угрозой.

2. Необходимость перехода на алгоритмы, работающие после квантов, привела к активной разработке новых криптографических алгоритмов, устойчивых к квантовым атакам.

Таким образом, появление квантовых атак обусловливает необходимость перехода к постквантовым криптографическим системам, устойчивым к квантовым вычислениям.

1.2 Квантовые распределение ключей

1.2.1 Основные принципы квантового распределения ключей

На протяжении большей части истории криптографии симметричные шифры были золотым стандартом безопасной связи. До появления более совершенных вычислительных технологий, начиная с конца 1930-х годов, ручные симметричные шифры были, по большей части, единственным типом шифровальных систем, которые были разработаны и широко использовались. Если посмотреть на эти шифры через призму проблемы распределения ключей, то, как и сами шифры были ручными, так и решения по распределению ключей были ручными. Они включали физическую транспортировку ключа - по почте, при личной встрече или через курьера - желаемому получателю будущего сообщения. Только после создания компьютерных алгоритмов шифрования (таких, как DES) возникла необходимость в решениях для распространения ключей по сетям, а не на физические расстояния. В связи с этим было разработано множество протоколов для распределения ключей в симметричных криптосистемах. Передача криптографического ключа с использованием принципов квантовой механики называется квантовым распределением ключей (Quantum Key Distribution, QKD) [**Ошибка! Источник ссылки не найден.**]. Первым решением проблемы распределения ключей для симметричных криптосистем является использование функции деривации ключа. Функция деривации ключа - это математическая функция, которая принимает фиксированный входной сигнал и выдает выходной и используется отправителем и получателем в качестве ключа шифрования/дешифрования[31]. Информация, передаваемая по незащищенному каналу, называется ключом деривации и состоит из случайной строки битов. Отправитель передает этот ключ получателю. Затем обе стороны используют ключ деривации в качестве входа в согласованную функцию деривации ключа, а результат будет использоваться в качестве ключа шифрования/дешифрования.

Для генерации ключевого материала используются математические операции, называемые псевдослучайными функциями. Эти функции разработаны таким образом, чтобы вести себя как можно ближе к случайному, поскольку успешно математически смоделировать случайное поведение удивительно сложно. Случайное поведение этих функций очень важно для подобной схемы, поскольку оно гарантирует, что создание ключа будет односторонним. То есть противнику будет крайне сложно создать правильный ключ после восстановления ключа деривации. Это связано с относительно случайным поведением сильных функций деривации ключа. Чем более случайным является поведение функции, тем сложнее противнику восстановить ключ из перехваченного материала. При рассмотрении безопасности функции деривации ключа также важно обратить внимание на длину используемого ключа деривации. Это зависит от того, какая псевдослучайная функция используется в конкретной функции деривации ключа. Две основные псевдослучайные функции, используемые в большинстве производных ключей,

- Hash Message Authenticated Code (HMAC) и Cipher Message Authenticated Code (CMAC) - требуют разной длины ключа для обеспечения безопасной производной ключа.

При использовании функции HMAC ключ деривации может быть любой длины. С другой стороны, при использовании CMAC в качестве псевдослучайной функции длина ключа деривации определяется длиной блоков, используемых в CMAC, который сам по себе является разновидностью блочного шифра. Безопасность функции деривации ключа также зависит от силы используемой псевдослучайной функции. В данном случае сила псевдослучайной функции определяется как количество усилий, которые потребуются противнику, чтобы правильно сопоставить случайную входную строку битов с правильной выходной строкой, иными словами, с ключом [**Ошибка! Источник ссылки не найден.**]. Таким образом, чем больше усилий потребуется противнику для восстановления правильного выходного материала, тем большей силой обладает псевдослучайная функция. Сила и, соответственно, требуемые усилия, скорее всего, зависят от того, как ведет себя используемая случайная функция. Чем больше функция стремится к действительно случайному поведению, тем больше усилий потребуется, чтобы обратить процесс вспять, поскольку гораздо сложнее обнаружить и использовать паттерны, когда поведение данных становится все более и более случайным. Последняя определяющая особенность функций деривации ключей - возможность их использования для создания иерархии ключей. В этом случае ключевой материал, полученный с помощью функции деривации ключей, возвращается в ту же функцию, в результате чего появляется еще больше ключей [33]. Этот процесс можно повторяться столько раз, сколько потребуется. Эта особенность функций деривации ключей важна при рассмотрении эффективности симметричной криптосистемы. Учитывая, что каждое сообщение должно быть зашифровано/расшифровано с помощью уникального общего ключа, возможность создать множество ключей на основе одного оригинала будет гораздо эффективнее, чем многократное повторение исходного процесса. Если рассматривать систему, в которой используется множество уникальных ключей, например модули, зашифрованные с помощью DES, то наличие сразу множества ключей, а не создание нового ключа каждый раз, когда он нужен, безусловно, более эффективно. Обертывание ключей - еще одна успешная и безопасная схема распространения симметричных криптографических ключей. Обертывание ключа - это практика шифрования самого ключевого материала, а затем передачи зашифрованного ключа по незащищенному каналу желаемому получателю [38].

Обертывание ключа отличается от использования функции деривации ключа тем, что по незащищенному каналу передается сам ключ, хотя и зашифрованный, в то время как при использовании функции деривации ключа передается тривиальная информация для последующего построения ключа. Безопасная схема обертывания ключей выполняется по методу, называемому «хэшировать - затем расшифровать» (Hash-then-Encrypt). В этом случае

ключевой материал сначала подается в хэш-функцию, а на выходе симметрично шифруется. Шифрование самого ключа обычно выполняется с помощью протокола AES. В таком случае исходный ключ хэшируется на блоки по 64 бита, и каждый блок из 64 бит шифруется в соответствии с протоколом AES [41]. Учитывая, что в процессе обертывания ключа обычно участвует AES, важно также рассмотреть, как свойства самого AES повлияют на безопасность и эффективность обертывания ключа. Первое свойство, которое следует рассмотреть, - это размер самого ключа, который подвергается обертыванию. В целом считается, что чем длиннее ключ, тем более безопасной будет обертка. Это объясняется просто: чем длиннее шифруемое сообщение (в данном случае шифруемое сообщение - это ключ), тем сложнее противнику его расшифровать. Это, опять же, становится проблемой усилий. По аналогии с функциями деривации ключей, где неотъемлемой частью безопасности было количество усилий, необходимых для решения проблемы, большая часть безопасности обертывания ключей основана на том же принципе. Ключ-обертка - это просто симметричный шифр, содержимое которого имеет критическое значение. Поэтому его безопасность ничем не отличается от безопасности любого другого симметричного шифра и в значительной степени зависит от длины и сложности шифруемого сообщения. Основная идея заключается в том, что при передаче информации через квантовые каналы любое вмешательство или измерение изменяет состояние квантовых частиц, что позволяет сторонам идентифицировать попытки вмешательства или перехвата информации.

Основные принципы:

- *Принцип неопределенности Гейзенберга*: В квантовой механике существует принцип, согласно которому точно измерить такие параметры частиц, как положение и импульс, одновременно невозможно. Это означает, что состояние квантовых частиц неизбежно нарушится, если кто-то попытается «слушать» передачу.
- *Квантовая суперпозиция*: Частицы, такие как фотоны, могут одновременно находироваться в нескольких состояниях. Эти состояния могут использоваться для кодирования информации, и их изменяет любое измерение, произведенное на них.
- *Принцип запутанности*: Когда две частицы находятся в квантовой запутанности, изменение состояния одной из них немедленно влияет на состояние другой частицы, независимо от того, насколько далеко они находятся друг от друга. Многие схемы квантового распределения ключей используют эту функцию (рисунок 12).

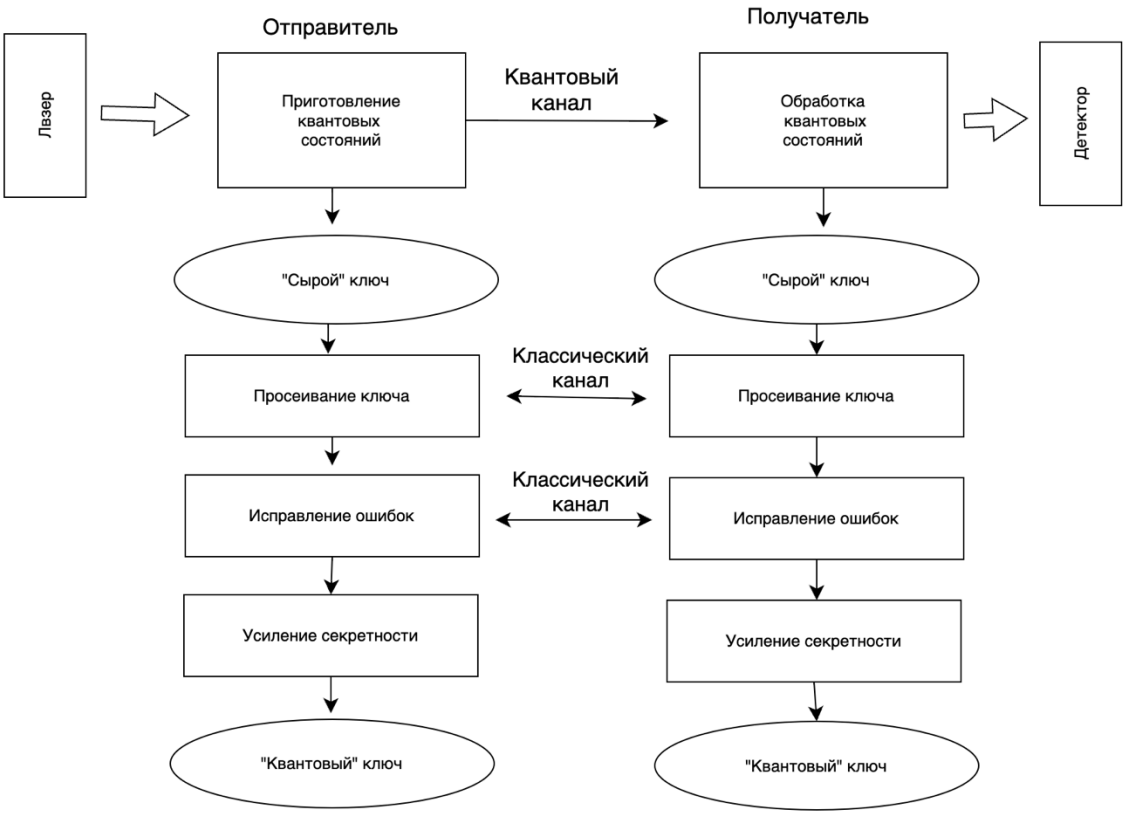


Рисунок 12 – Типовая схема квантового протокола распределения ключей

Основные виды можно классифицировать по используемым методам: на основе принципа неопределенности (например, протокол BB84) или на основе квантовой запутанности (например, протокол E91) [37]. Наиболее известные виды квантового распределения ключей является протокол BB84, B92 и E91. Протоколы QKD, такие как BB84, E91 и B92, опираются на фундаментальные принципы квантовой механики, включая принцип неопределённости Гейзенберга и невозможность клонирования квантовых состояний. Это позволяет не только гарантировать конфиденциальность передаваемой информации, но и обеспечить детектирование любых попыток перехвата данных на физическом уровне. Ниже приведен пример передачи 8 бит секретного ключа. После просеивания ключа остается всего 4 бита (рисунок 13).

Случайный бит Алисы	0	1	1	0	1	0	0	1
Случайный принцип отправки Алисы	X	+	+	+	X	+	X	X
Поляризация фотона Алиса посылает	D	V	V	H	A	H	D	A
Случайный принцип отправки Боба	X	X	+	X	+	+	+	X
Поляризация фотона Боб измеряет	D	D	V	A	V	H	V	A
публичное обсуждение основы								
общий секретный ключ	0		1			0		1

Рисунок 13 – Передача секретного ключа

Несмотря на высокий уровень безопасности, обеспечиваемый системами QKD, их широкое внедрение ограничивается рядом технологических факторов. Прежде всего, это высокая стоимость квантового оборудования, ограниченные дистанции передачи и необходимость создания специализированной инфраструктуры связи. Кроме того, полноценная реализация QKD требует сочетания квантовых и классических каналов передачи данных, что осложняет интеграцию этих систем в существующие криптографические протоколы. С учётом указанных ограничений, а также учитывая стремительное развитие квантовых вычислительных технологий, особую актуальность приобретает постквантовая криптография. В отличие от QKD, постквантовые криптосистемы разрабатываются для работы на классических вычислительных платформах и не требуют создания новой физической инфраструктуры. Они основаны на задачах, считающихся вычислительно сложными даже для квантовых компьютеров, таких как задачи на решётках, коды исправления ошибок и мультивариантные уравнения.

1.3 Постквантовая криптография

В отличие от квантовой криптографии, постквантовая криптография относится к схемам криптографии, которые, как предполагается, невозможно взломать даже с помощью квантового компьютера. В отличие от квантовой криптографии, эти алгоритмы могут использоваться на классическом оборудовании. Такие алгоритмы используются для решения сложных математических задач, в которых квантовые компьютеры не обладают вычислительным преимуществом [38,39]. Стойкость постквантового шифрования обеспечивается математическими доказательствами секретности каждого из алгоритмов, которые были проверены мировым научным математическим сообществом. В частности, это алгоритмы, основанные на теории решеток, хеш-функциях и линейном коде.

1.3.1 Основы постквантовой криптографии

Основные направления постквантовой криптографии включают криптографию на решетках, кодовую криптографию, многочленную криптографию, криптографию на основе хеш-функций и схемы на основе изогений эллиптических кривых. Каждое из этих направлений предлагает собственные подходы к построению безопасных схем шифрования, электронной подписи и обмена ключами.

Гипотеза, основанная на коде, заключается в том, что декодировать случайный линейный код очень сложно. Система McEliece, одна из первых систем с открытым ключом, создала первый такой алгоритм еще в 1978 году. В то время об атаках с использованием квантового компьютера не было и речи. Однако после появления алгоритма Шора, способного легко взломать повсеместно используемое шифрование, криптографы-исследователи обратили внимание на алгоритм McEliece [40]. Алгоритмы, основанные на теории решеток, являются дополнительным видом схем постквантовой криптографии. В частности, IBM использует такие схемы в своих приложениях безопасности, которые хорошо изучены и легко реализуются.

Хеш-функция - один из самых распространенных инструментов криптографии в постквантовом шифровании. Хеширование - это преобразование любого объема данных в набор определенной длины символов, который очень сложно расшифровать. Кроме того, постквантовые алгоритмы, использующие хеш-функцию, делают декодирование сообщения невозможным всеми известными методами. Основным элементом электронной подписи может быть хеш-функция. Этот процесс используется большинством алгоритмов хеширования:

Создайте сообщение. Пользователь определяет, что должно быть хешировано.

Выберите тип. Существуют десятки алгоритмов хеширования, и пользователь может решить, какой из них лучше всего подходит для данного сообщения.

Введите сообщение. Пользователь вводит сообщение в компьютер, на котором работает алгоритм.

Запустите хэш. Система преобразует сообщение, которое может иметь любую длину, в заранее определенный битовый размер. Обычно программы разбивают сообщение на ряд блоков одинакового размера, и каждый из них последовательно сжимается.

Сохраните или поделитесь. Пользователь отправляет хэш предполагаемому получателю, или хешированные данные сохраняются в этой форме.

Процесс сложный, но работает очень быстро. Через несколько секунд хеш будет завершен (рисунок 14).

Рисунок 14 – Алгоритм хеширования

Вы можете видеть изменения хеша, когда добавляете пробелы или другие символы. Однако, независимо от количества символов, которые вы добавите, длина хеша остается постоянной.

Криптографические системы, основанные на кодах, - это криптосистемы, в которых алгоритмический примитив (основная односторонняя функция) использует код, который исправляет ошибки C [40, с. 114-115]. Этот примитив может включать ошибку в слово C или вычислять синдром в отношении проверочной матрицы C. Схема шифрования с открытым ключом была предложена Робертом Дж. МакЭлисом в 1978 году. Открытый ключ генерирует случайную матрицу, которая генерирует случайно перестановочную версию этого кода, а закрытый ключ представляет собой случайный двоичный неприводимый код Гоппы. Зашифрованный текст - это кодовое слово, в котором есть ошибки, и только тот, кто владеет кодом Гоппы, может удалить эти ошибки. Три десятилетия спустя параметры были немного изменены, но ни одна атака, даже для квантового компьютера, не представляет серьезной опасности для системы. Подобные идеи использовались при разработке других криптосистем. Среди прочего отметим некоторые системы с открытым ключом, такие как схема

шифрования Нидеррайтера или схема подписи CFS, а также схемы идентификации, генераторы случайных чисел или криптографическая хеш-функция. Впоследствии почти все асимметричные криптографические схемы, основанные на теории кодирования, имеют один общий недостаток: они требуют большого количества памяти. Затем были разработаны несколько дополнительных схем, таких как хэш-функции, генераторы случайных чисел, схема идентификации Стерна и попытки создать схему подписи. Последние, однако, потерпели неудачу до того, как Куртуа, Финиас и Сендрие наконец сделали обещающее предложение в 2001 году. Но даже если последний не сломан, он не подходит для стандартных приложений из-за большой стоимости подписи для безопасных наборов параметров, а также размера открытого ключа. Разнообразие возможных криптографических приложений дает достаточную мотивацию для более тщательного изучения криптосистем, основанных на теории кодирования, как серьезной альтернативы устаревшим РКС, подобным темам, основанным на теории чисел. Даже если параметры безопасности криптоситемы МакЭлиса пришлось изменить примерно через пятнадцать лет после ее предложения, она остается неповрежденной в своей первоначальной версии. Секретный ключ McEliece РКС представляет собой код Гоппы в исходном описании, но его можно получить из любого подкласса класса альтернативных кодов [41, 42]. Тем не менее, такой выбор может не обеспечить необходимую безопасность. Криптосистема МакЭлиса требует знания того, какой эффективный алгоритм исправления ошибок для определенного класса кода, а также перестановки. Этот алгоритм доступен для каждого кода Гоппы. Ниже показано алгоритм McEliece РКС:

Параметр системы: n, t∈N, где t≤n.

Генератор ключей: учитывая параметры n, t, сгенерируйте следующие матрицы:

G : *k*x*n* матрица-генератор кода G над F размерности k и минимального расстояния d ≥ 2t + 1. (Двоичный неприводимый код Гоппы в исходном предложении).

S: *k*x*k* случайная двоичная неособая матрица.

P: *n*x*n* матрица случайных перестановок.

Затем вычислите матрицу размера k × n G^{pub} = SGP.

Public key (открытый ключ): (G^{pub}, t)

Private key (приватный ключ): (S,D_G,P), где D_G эффективный алгоритм декодирования для G.

Encryption (шифрование): ($E_(G^{pub},t_)$): чтобы зашифровать открытый текст m ∈ F^k, выберите вектор z ∈ F^n веса t случайным образом и вычислите зашифрованный текст с следующим образом:

$$c = mG^{pub} \oplus z \quad (1)$$

Decryption (дешифрование): ($D_{(S,Dg,P)}$): чтобы расшифровать зашифрованный текст, необходимо вычислить:

$$cP^{-1} = (mS)G \oplus zP^{-1} \quad (2)$$

сначала и примените к нему алгоритм декодирования D_G^{pub} для G. Поскольку cP^{-1} имеет расстояние Хэмминга t до G, мы получаем кодовое слово:

$$mSG = D_G(cP^{-1}) \quad (3)$$

Пусть $J \subseteq \{1, \cdots, n\}$ — множество, такое что G_J^{pub} обратимо, тогда мы можем вычислить открытый текст:

$$m = (mSG)_J(G_J)^{-1}S^{-1} \quad (4)$$

Когда выбираете параметры безопасности для McEliece PKC, вы должны учитывать известные атаки. К сожалению, закрытая формула не позволяет определить оптимальные параметры для определенного уровня безопасности. Это связано с размером открытого ключа.

Основная слабость McEliece PKC связана с гибкостью его зашифрованных текстов. Добавление кодовых слов, например строк G^{pub}, к зашифрованному тексту позволяет получить дополнительный действительный зашифрованный текст. В результате исходная криптосистема МакЭлиса не соответствует требованиям неподатливости. Злоумышленник может мгновенно совершить атаку CCA2, добавив второе сообщение m' к c, вычислив $c' = c \oplus m'G^{pub}$, которое можно расшифровать оракулом. Обратите внимание, что в случае Нидеррайтера пластичность не является проблемой. Это связано с тем, что мы не можем создавать новые декодируемые синдромы из старых с вероятностью, значительно большей, чем t/n. Противники схемы МакЭлиса могут легко использовать связь между двумя зашифрованными сообщениями, чтобы определить биты ошибок. Эта атака не может быть реализована криптосистемой Нидеррайтера. Пусть два сообщения m_1 и m_2 имеют известное отношение Λ, например, тексты $\Lambda(m_1, m_2) = m_1 \oplus m_2$ и c_1, c_2 являются соответствующими зашифрованными символами. Таким образом, вес $c_1 \oplus c_2 \oplus \Lambda(m_1, m_2)$ будет равен $\leq 2t \leq n - k$, а безошибочные позиции $m_1 \oplus m_2$ будут найдены не менее k. Это позволяет преступникам угадывать биты ошибки. При атаке с повторной отправкой сообщений злоумышленник может восстановить $z_1 \oplus z_2 = c_1 \oplus c_2$, что приводит к особому случаю связанных сообщений. Реактивная атака использует адаптивно выбранный зашифрованный текст для более слабой версии атаки. Злоумышленник не может получить доступ к оракулу полной расшифровки и может только наблюдать за реакцией получателя на возможные зашифрованные тексты. Злоумышленник может получить зашифрованные тексты, изменить некоторые биты, а затем следить за тем, как назначенный получатель реагирует на эти измененные зашифрованные тексты. Чтобы отправить изменения подлинного зашифрованного текста, необходимо добавить дополнительные биты ошибок. Если получатель не может декодировать (ответ: повторный

запрос), соответствующие биты изначально были правильными. Это позволяет преступнику восстановить набор безошибочной информации за k повторов. Поскольку для Niederreiter PKC не требуется пластичность, такая атака вполне возможна.

Криптосистемы с многомерным открытым ключом (MPKCs) обычно имеют общедоступную карту, состоящую из набора обычно квадратичных полиномов над конечным полем. Трудность задачи решения нелинейных уравнений над конечным полем подтверждает его основное предположение о безопасности. Это семейство PKC считается одним из наиболее важных, и может быть способно противостоять даже мощным квантовым компьютерам будущего. За последние два десятилетия многомерная криптография с открытым ключом развивалась быстро и интенсивно. Несмотря на то, что некоторые из первоначально заявленных конструкций оказались менее надежными, их продолжают использовать. По идеям Диффи и Хеллмана существование класса «потайных односторонних функций» необходимо для криптосистемы с открытым ключом (также известной как PKC). Все основные характеристики PKC будут определяться этим классом и математической структурой, лежащей в его основе [43]. Например, эллиптическая криптография характеризуется группой эллиптических кривых, тогда как NTRU характеризуется структурой целостной решетки. Многомерная криптография (с открытым ключом) - это тип исследования PKC, в котором односторонняя функция с люком имеет многомерный квадратный полиномиальный отображение над конечным полем. Иными словами, в большинстве случаев открытый ключ задается набором квадратных полиномов:

$$P = (p_1(w_1,...,w_n),..., p_m(w_1,...,w_n)),$$

где каждое число p_i представляет собой (обычно квадратичный) нелинейный многочлен от $w = (w_1,...,w_n)$:

$$z_k = p_k(w) : = \sum_i P_{ik} w_i + \sum_i Q_{ik} w_i^2 + \sum_{i>j} R_{ijk}\, w_i w_j \qquad (5)$$

со всеми переменными и коэффициентами в $K = F_q$, поле с q элементами. Проверка этих полиномов при любом их значении соответствует либо процедуре проверки, либо шифрования. Эти PKC называются многомерными криптосистемами с открытым ключом (также известными как MPKC). Решение набора квадратных уравнений над конечным полем или решение следующей задачи эквивалентно вращению многомерного квадратичного отображения:

Задача MQ: Решите систему $p_1(x) = p_2(x) =\cdot p_m(x) = 0$, где каждое число p_i является квадратичным относительно $x = (x_1,...,x_n)$. Все коэффициенты и переменные находятся в $K = F_q$, поле с q элементами. В целом MQ является NP-сложным. Такие задачи считаются сложными, если класс P не равен NP. Конечно, в MPKC случайный набор квадратных уравнений не будет использоваться, поскольку он не будет иметь лазейки. Идеалом, порожденным

этими полиномами, является соответствующая математическая структура системы полиномиальных уравнений; это не всегда является общим. Таким образом, с философской точки зрения многомерная криптография относится к математике, основанной на полиномиальных принципах, например, алгебраической геометрии. По крайней мере, алгебраическая геометрия, математика, которую использует MPKC, была разработана в 20-м веке [44]. Поскольку мы больше имеем дело с системами с определенными лазейками, а не с «случайными» или «универсальными», безопасность MPKC не гарантируется NP-трудностью MQ, и эффективные атаки могут происходить на любой лазейке, которую мы выбрали. Таким образом, история MPKC связана с расширением нашего понимания методов создания безопасных многомерных лазей (рисунок 16).

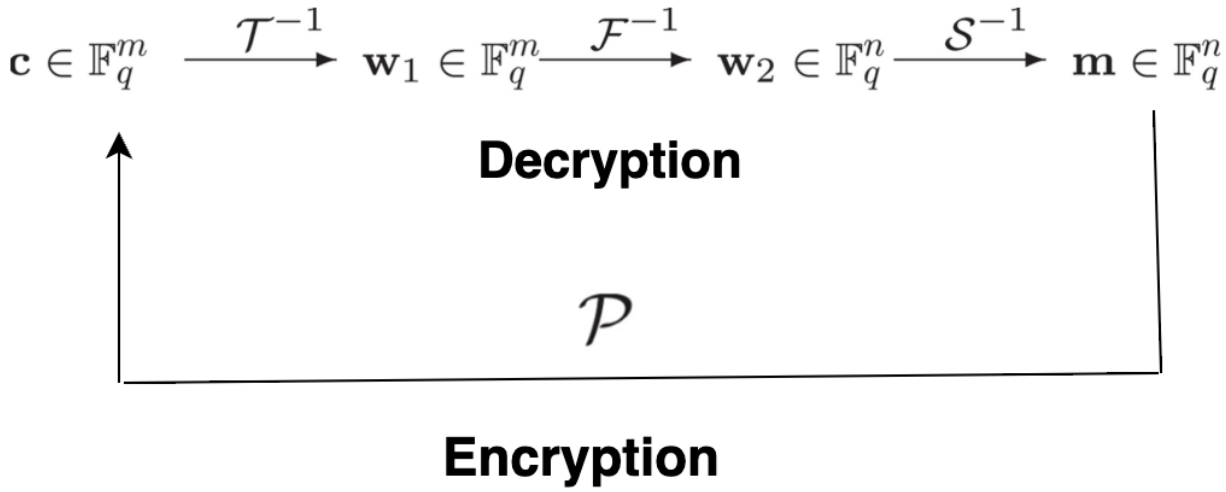


Рисунок 15 - Общий рабочий процесс многовариантных схем шифрования

Криптографические конструкции на основе решеток открывают большие перспективы для постквантовой криптографии, поскольку они имеют очень надежные доказательства безопасности, основанные на стойкости в наихудшем случае, относительно эффективных реализациях, а также большой простоте. Кроме того, считается, что криптография на основе решетки защищена от квантовых компьютеров [45].

Решетка – это набор точек в n-мерном пространстве с периодической структурой, такой как показанная на рисунке 16. Более официально, решетка, созданная n-линейно независимыми вектором $b_1,...,b_n \in R^n$, представляет собой набор независимых векторов

$$\mathcal{L}(b_1 \ldots b_n) = \{ \sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z} \}. \qquad (6)$$

Векторы $b_1, \ldots . b_n$ известны как базис решетки.

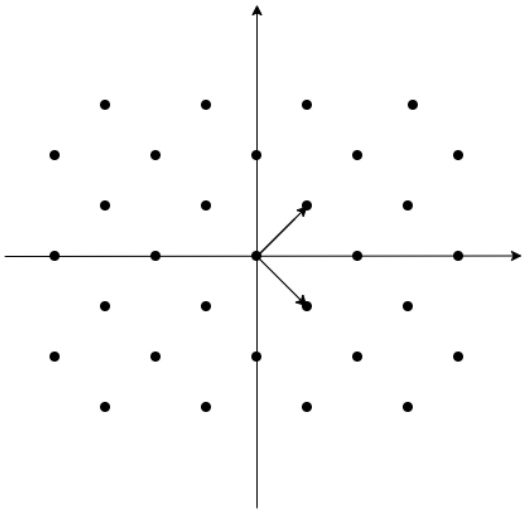


Рисунок 16 – Двухмерная решетка и два возможных основания
Решетка определяется как набор всех целочисленных комбинаций:

$$\mathcal{L}(b_1 \ldots b_n) = \{ \sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z} \text{ для } 1 \leq i \leq n \} \quad (7)$$

Базис можно представить матрицей $B = [b_1, \ldots, b_n] \in R^{nxn}$, имеющая базисные векторы в виде столбцов. Используя матричные обозначения, решетку, порожденную матрицей $B \in R^{nxn}$, можно определить как $\mathcal{L}(B) = \{Bx: x \in Z^n\}$, где Bx - обычное умножение матрицы на вектор. Нетрудно видеть, что если U - унимодулярная матрица (т.е. целочисленная квадратная матрица с определителем ±1), то базисы B и BU порождают одну и ту же решетку. (Фактически, $\mathcal{L}(B)=\mathcal{L}(B')$ тогда и только тогда, когда существует унимодулярная матрица. U такой, что $B' = BU$.) В частности, любая решетка допускает несколько базисов, и этот факт лежит в основе многих криптографических приложений [46].

Определителем решетки называется абсолютное значение определителя базисной матрицы $\det(\mathcal{L}(B)) = |\det(B)|$. Значение определителя не зависит от выбора базиса и геометрически соответствует обратной плотности точек решетки в R^n. Двойственная решетка $\mathcal{L}$ в R^n, обозначаемая L^*, - это решетка, заданная множеством всех векторов $y \in R^n$, удовлетворяющих условиям $\langle x, y \rangle \in Z$ для всех векторов $x \in \mathcal{L}$. Можно видеть, что для любого $B \in \mathcal{L}$ R^{nxn}, $\mathcal{L}(B)^* = \mathcal{L}((B^{-1})^T)$. Отсюда следует, что $\det(\mathcal{L}^*) = 1/\det(\mathcal{L})$.

В криптографии на основе решеток q-арные решетки очень важны. Это решетки $\mathcal{L}$, для которых $qZ^n \subseteq \mathcal{L} \subseteq Z^n$ для некоторого целого числа q, возможно, простого. Иными словами, x mod q определяет принадлежность вектора x к L. Линейные коды из Z_q^n находятся во взаимно однозначном соответствии с этими решетками. В среднем, q-ичные решетки используются в криптографических конструкциях на основе решеток. Заметим, что для некоторого q любая целочисленная решетка $\mathcal{L} \subseteq Z^n$ является q-арной решеткой. Это относится к случаям, когда q является целым кратным определителем $\det(\mathcal{L})$. Тем не менее, нас больше всего интересуют q-ичные решетки с q, меньшим, чем $\det(\mathcal{L})$.

Учитывая матрицу A ∈ Z_q^{nxm} для некоторых целых чисел q,m,n, мы можем определить два m-мерные q-ичные решетки

$$\Lambda_q(A) = \{y \in Z^{m.} : y = A^T s \bmod q \text{ для } s \in Z^n\} \quad (8)$$
$$\Lambda_q^{\perp}(A) = \{ y \in Z^{m.} : Ay = 0 \bmod q \}.$$

Строки A является источником первой q-ичной решетки [47]. Вторая решетка включает все векторы, ортогональные по модулю q строкам A. Иными словами, первая q-ичная решетка отвечает за код, созданный строками A, а вторая решетка отвечает за код, матрица проверки четности которого равна A. Эти решетки двойственны друг другу с точностью до нормировки, как показано из определения: $\Lambda_q^{\perp}(A) = q \cdot \Lambda_q(A)^*$ и $\Lambda_q(A) = q \cdot \Lambda_q^{\perp}(A)^*$.

Проблемы с решеткой:

Наиболее известными вычислительными задачами на решетках являются следующие:

1. Задача о кратчайшем векторе (SVP): учитывая базис решетки B, найдите кратчайший ненулевой вектор в $\mathcal{L}(B)$.
2. Задача ближайшего вектора (CVP): с учетом базиса решетки B и целевого вектора t (не обязательно в решетке), найдите ближайшую к t точку решетки $v \in \mathcal{L}$ (B).
3. Задача короткого целочисленного решения (SIS): найти вектор z который дает нулевой остаток, но при этом состоит из маленьких чисел. $||z|| \leq \beta$.

Революционная работа Аджтая раскрыла метод использования решеток в криптографии[48]. К настоящему времени его исследования превратились в целую область исследований, основное внимание которой уделяется расширению области применения решетчатой криптографии и созданию более практичных решетчатых криптосистем. По уровню безопасности криптографические конструкции на основе решеток разделяются на два: они открывают широкие возможности для постквантовой криптографии; многие из них очень эффективны, а некоторые даже превосходят более известные варианты; они обычно довольно просты в реализации; и, конечно же, считается, что все они защищены от квантовых компьютеров. В наихудшем сценарии безопасность имеет двоякое значение. Во-первых, это утверждает, что атаки на криптографическую конструкцию, скорее всего, не будут эффективны асимптотически, а только при небольшом выборе параметров. Иными словами, это подтверждает, что в нашей криптографической структуре нет существенных недостатков. Фактически, гарантия безопасности наихудшего случая может даже помочь нам при принятии проектных решений в некоторых ситуациях. Во-вторых, как будет показано далее, обеспечение безопасности в наихудшем случае способствует выбору определённых параметров криптосистемы, однако на практике это зачастую приводит к излишне консервативным оценкам.

1.4 Решеточные вычислительные задачи

С момента открытия Шором алгоритма квантового факторинга в середине 1990-х годов были предприняты усилия по использованию квантовых алгоритмов для решения решаемых задач. Однако до сих пор эти усилия не достигли значительного успеха, если вообще достигли. Основная проблема заключается в том, что метод нахождения периодичности, который используется в алгоритме факторизации Шора и его квантовых алгоритмах, похоже, не подходит для задач, связанных с решеткой. Таким образом, логично рассмотреть гипотезу 2, которая оправдывает использование решетчатой криптографии в постквантовой криптографии. Не существует квантового алгоритма с полиномиальным временем, который бы с точностью до полиномиальных множителей аппроксимировал решеточные задачи.

Наиболее значимыми и сложными задачами на решётках являются задачи CVP, SVP и SIS [49,50].В то время как CVP требует найти вектор, ближайший к конкретному вектору в решетке, SVP требует найти короткий ненулевой вектор, который является минимальной нормой Евклида. A SIS требует находить вектор, который состоит из маленьких целых чисел. ApprSVP и apprCVP являются расширениями SVP и CVP и представляют собой две задачи жесткой аппроксимации наихудшего случая на решетках. Как оказалось, они имеют удивительно много применений в криптографических конструкциях.

Определение 1. Проблема с кратчайшим вектором (SVP). Если в лабиринте $\mathcal{L}$ *(B)* есть, базис матрицы $B \in Z^{mxn}$, найдите самый короткий ненулевой вектор $b \in \mathcal{L}$ что $||b||$ равен D_{min} ($\mathcal{L}$).

Пусть дана решетка L в R^n , которая представляет собой подпространство, состоящее из всех линейных комбинаций базисных векторов $v_1, v_2, \ldots, v_m$ с целыми коэффициентами. Тогда задача о ближайшем векторе заключается в нахождении *кратчайшего ненулевого вектора* в этой решетке, то есть вектора $v \in L$ для которого:

$$|| v|| = min_{v \in L\{0\}} ||v|| \qquad (9)$$

где $\|\cdot\|$ – это, как правило, евклидова норма вектора. То есть, задача состоит в нахождении вектора, который имеет минимальную длину среди всех ненулевых векторов в решетке.

SVP является NP-трудной задачей, и в настоящее время нет эффективных алгоритмов для ее решения в целом. В частности, решение задачи SVP в многомерных пространствах требует экспоненциального времени и не требует полиномиальных алгоритмов. Вместо точного решения задачи SVP часто используется поиск приближенных решений. Например, с помощью алгоритмов *редукции базиса* (например, LLL-алгоритм) можно найти приближенное решение, которое будет не слишком большим, хотя оно и не обязательно будет минимальным [51].

Определение 2. Задача о ближайших векторах (CVP).

Для любой базисной матрицы B $\in Z^{mxn}$ решетки $\mathcal{L}(B)$ и вектора c, не входящего в $\mathcal{L}$, найдите вектор b $\in \mathcal{L}$, наиболее близкий к нему, т.е. найдите вектор b $\in \mathcal{L}$ такой, что $||c-b||=D_{min}(\mathcal{L})$.

Дано:

Решетка $\Lambda \subset R^n$, которая представляет собой множество всех линейных комбинаций базисных векторов решетки с целыми коэффициентами.

Точка t $\in R^n$, которая находится в пространстве, и нужно найти вектор решетки v$\in\Lambda$, который минимизирует расстояние до данной точки *t*.

Необходимо найти:

Вектор v$\in\Lambda$, который минимизирует расстояние $\|v-t\|$, где $\|\cdot\|$ - это обычно евклидова норма.

$$v = \arg\min_{v\in A}||v-t|| \qquad (10)$$

где Λ - это решетка, обычно заданная как $\Lambda=Z^nB$, где B - матрица, определяющая базис решетки.

CVP имеет важное значение в криптографии, особенно в контексте защиты с использованием решеток. Многие криптографические схемы, такие как схемы с функцией гомоморфного шифрования (например, схемы Gentry или LWE-based криптосистемы), основываются на сложности решения задачи CVP. Например, если можно эффективно решать задачу CVP, то это может привести к возможному разрушению безопасности этих схем. Для каждой точки t$\in R^n$ задача CVP заключается в нахождении наилучшего приближения этой точки в решетке. В общем случае это вычислительно сложная задача, которая не имеет полиномиального решения в стандартных моделях вычислений (например, в модели Тьюринга). Задача CVP является NP-трудной для произвольных решеток. Более того, она считается вычислительно сложной даже для приближенных решений. В некоторых случаях, например, для решеток с ограниченными характеристиками или для решеток с малыми размерами, можно применить более быстрые алгоритмы, но в общем случае нахождение ближайшего вектора является задаче с экспоненциальной сложностью. Сложность CVP также тесно связана с задачей поиска наименьшего вектора в решетке (SVP), которая является еще более сложной задачей, поскольку она требует нахождения наименьшего вектора, а не только ближайшего к точке. Задача CVP является одной из ключевых задач в теории решеток с важными приложениями в криптографии и других областях [52]. Хотя точное решение задачи может быть вычислительно сложным, существуют различные методы, направленные на приближенное решение этой задачи, и на основе этой сложности строятся многие криптографические протоколы.

Определение 3. Задача о коротком целочисленном решении (SIS).

(Однородная) задача $SIS_{n,q,m,\beta}$, заключается в следующем: для заданной равномерно случайной матрицы A $\in Z_q^{nxm}$ найти ненулевой интегральный вектор z $\in Z^m$такой, что

$$Az = 0 \ (\mathrm{mod} q) \text{ и } ||z|| \leq \beta \qquad (11)$$

Нормальная форма задачи заключается в поиске ненулевого интегрального вектора z = (z ∈ Z^m , e ∈ Z^n) такого, что

$$Az = e(\mathrm{mod} q) \text{ и } \|z\| \leq \beta \qquad (12)$$

Дано:

Матрица A - случайная матрица A ∈ Z_q^{nm} , где: n - количество строк, m - количество столбцов, q - модуль (простое большое число, используемое в вычислениях по модулю q).

Ограничение на норму вектора z - его длина должна быть не больше некоторого параметра β, $\|z\| \leq \beta$.

Необходимо найти:

Ненулевой целочисленный вектор z ∈ Z^m, который удовлетворяет уравнению (11).

Задача короткого целочисленного решения (SIS) обладает рядом важных свойств, которые делают её полезной в криптографии, особенно в постквантовых схемах.

Одним из ключевых свойств SIS является сложность в худшем случае (worst-case hardness). SIS является основой для постквантовой криптографии**,** так как пока не существует известных квантовых алгоритмов, которые могли бы её эффективно решать (в отличие от RSA и ECC, которые уязвимы перед алгоритмом Шора)[**Ошибка! Источник ссылки не найден.**]. Связанная с ней задача решения линейных уравнений по модулю с ограничением на длину вектора остаётся сложной даже для квантовых компьютеров. Доказано, что если можно эффективно решать SIS, то можно решить SVP (задача кратчайшего вектора) и GapSVP (различение решёток с короткими и длинными векторами) в худшем случае. Это означает, что надёжность криптографических схем, основанных на SIS, сводится к доказанной сложности задач на решётках, известных своей высокой вычислительной сложностью. Однако решётчатые задачи, такие как SIS**,** не подвержены полиномиальным квантовым атакам**.** К настоящему времени не существует квантовых алгоритмов, которые бы решали задачи SVP, CVP или SIS за полиномиальное время. Задача поиска коротких целых решений (SIS) демонстрирует устойчивость к квантовым атакам. На сегодняшний день отсутствуют квантовые алгоритмы, способные существенно ускорить решение этой задачи. Применение алгоритма Гровера лишь незначительно оптимизирует существующие вероятностные методы, снижая их сложность до порядка √N, что не оказывает критического влияния на общую экспоненциальную сложность задачи. Фундаментальная стойкость SIS обеспечивается её сводимостью к задачам наихудшего случая аппроксимации кратчайших векторов в решётке (SVP) и задачи ближайшего вектора (CVP), сложность которых сохраняется высокой даже при использовании квантовых вычислений. В результате, все известные как классические, так и квантовые алгоритмы решения задачи SIS требуют экспоненциального времени в зависимости от размерности решётки, что подтверждает её надёжность в условиях квантовых атак.

1.5 Полилинейная алгебра и ее связь с решетками

Полилинейная алгебра расширяет понятие линейной алгебры, рассматривая отображения, которые являются линейными по каждому аргументу независимо. Основным объектом изучения являются тензоры - многомерные массивы чисел, обобщающие матрицы. Полилинейное отображение определяется как:

$$\not{f} : V_1 \cdot V_2 \cdot \ldots \cdot V_k \longrightarrow W \qquad (13)$$

где V_i - векторные пространства, а отображение $\not{f}$ линейно по каждому аргументу. Тензоры находят применение в теории решёток, поскольку они позволяют описывать многомерные дискретные структуры и их трансформации[53]. В частности, тензорное произведение векторов и матриц часто используется для построения решёток, а полилинейные операции применяются в алгоритмических решениях задач, возникающих в криптографии. Некоторые важные вычислительные задачи на решётках, такие как задача кратчайшего вектора (SVP) и задача ближайшего вектора (CVP)**,** могут быть рассмотрены с использованием полилинейных методов. Например, приближенные алгоритмы решения SVP и CVP используют тензорные представления базисов решёток, что позволяет более эффективно оценивать структуру пространства. Полилинейные отображения можно применять для изучения различных свойств решёток, таких как их изоморфизм и симметрии. В частности, операции тензорного произведения позволяют строить новые решётки из существующих, комбинируя их свойства для создания криптографически устойчивых структур. Примером является использование полилинейных операций в гомоморфном шифровании, где данные зашифровываются таким образом, чтобы их можно было обрабатывать без расшифрования. Здесь решёточные схемы, такие как LWE (Learning With Errors)**,** могут быть обобщены с применением тензоров для работы с многомерными пространствами. Её сложность сводится к решению задачи малого целочисленного решения **(**SIS, Short Integer Solution), которая является фундаментальной проблемой решёточной криптографии. Пусть ошибка e_i моделируется гауссовым распределением χ. Тогда можно записать её через тензорное разложение:

$$e_i = \sum_{j=1}^{m} \lambda_j v_j \otimes w_j \qquad (14)$$

где λ_j - коэффициенты, а vv_j, w_j - базисные векторы пространства ошибок. Это представление позволяет лучше учитывать структуру ошибок и их влияние на решение задачи LWE. Задача малого целочисленного решения **(**Short Integer Solution, SIS**)** является одной из фундаментальных задач в постквантовой криптографии. Её сложность лежит в основе безопасности множества

современных криптографических схем, включая цифровые подписи, хеш-функции и системы с гомоморфным шифрованием. SIS тесно связана с задачами поиска коротких векторов в решётках, такими как SVP (Shortest Vector Problem) и CVP (Closest Vector Problem)**.**Тензорные методы являются мощным инструментом в решёточной криптографии, позволяя анализировать сложные многомерные структуры. Применение тензоров к задаче SIS позволяет эффективнее моделировать распределение ошибок, анализировать криптостойкость и оптимизировать алгоритмы решения. Несмотря на применение тензоров, SIS остаётся сложной задачей даже для квантовых компьютеров, что подтверждает её надёжность для построения защищённых криптографических систем. Используя формулу (11) можно тензорное представление задачи SIS. Для удобства анализа матрицу A можно представить в виде тензорного разложения :

$$\mathrm{A} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{r}} \lambda_{\boldsymbol{i}}\, \mathrm{A}_{\mathrm{i}}^{(1)} \otimes \mathrm{A}_{\mathrm{i}}^{(2)} \qquad (15)$$

где $\mathrm{A}_{\mathrm{i}}^{(1)}, \mathrm{A}_{\mathrm{i}}^{(2)}$ - компоненты матрицы, а $\lambda_{\boldsymbol{i}}$ - весовые коэффициенты. Это позволяет анализировать структуру решётки и выявлять её свойства. Рассмотрим решение x как сумму тензорных компонент:

$$x = \sum_{j=1}^{s} \beta_j x_j^{(1)} \otimes x_j^{(2)} \qquad (16)$$

где $x_j^{(1)}, x_j^{(2)}$ - базисные векторы малых решений, а β_j - коэффициенты.

Теперь уравнение Ax=0 mod q можно записать в тензорной форме, умножив (15) и (16) формулы:

$$\sum_{\boldsymbol{i=1}}^{\boldsymbol{r}} \sum_{j=1}^{s} \beta_j x_j^{(1)} \otimes x_j^{(2)}\ \lambda_{\boldsymbol{i}}\, \mathrm{A}_{\mathrm{i}}^{(1)} \otimes \mathrm{A}_{\mathrm{i}}^{(2)} = 0 \bmod \mathrm{q} \qquad (17)$$

Это представление позволяет использовать методы разложения ранга тензора для анализа сложности задачи SIS. Полилинейная алгебра играет ключевую роль в построении криптографических примитивов, особенно в контексте решёток и их применения в блокчейн-технологиях[55]. Использование тензоров и многомерных структур позволяет улучшить безопасность и эффективность блокчейнов, обеспечивая защиту от квантовых атак и оптимизацию вычислительных процессов.В будущем мы можем ожидать появления решёточных блокчейнов, где постквантовые алгоритмы будут интегрированы в механизмы консенсуса и цифровых подписей, а тензорные методы - в оптимизацию обработки данных.

1.6 Электронные подписи на основе решеток

Одним из самых важных криптографических примитивов являются схемы цифровой подписи. С теоретической точки зрения метод черного ящика может использовать односторонние функции для создания схем подписи. Таким образом, схемы подписи, основанные на наихудшей сложности решеточных задач, могут быть созданы с помощью односторонних функций. Тем не менее, конструкции черного ящика непрактичны и требуют больших накладных расходов. Мы рассмотрим несколько предложений для схем подписи, которые основаны на решеточных задачах и, как правило, гораздо более эффективны. Голдрейх и др. первоначально предложили схему подписи на основе решетки, основываясь на идеях, существовавших в их криптосистеме [56]. В 2003 году NTRU Cryptosystems представила NTRUSign, эффективную схему подписи. Эта схема подписи может быть рассмотрена как оптимизированная реализация схемы GGH на основе решеток NTRU. К сожалению, обе схемы могут быть взломаны в сильном смысле асимптотики в их базовой версии. Ни одна из схем не имела доказательств, подтверждающих безопасность. Мичианчо и Вадхан предложили первоначальную конструкцию эффективных схем подписи с поддерживающим доказательством безопасности (в модели случайного оракула). Они представили статистические системы доказательств с нулевым разглашением для различных проблем решетки и отметили, что такие системы доказательств могут быть преобразованы относительно эффективным образом сначала в безопасные схемы идентификации, а затем (с помощью эвристики Фиата-Шамира) в схемы подписи в моде Любашевский и Мичианчо, а также Джентри, Пейкерт и Вайкунтанатан недавно предложили более эффективные схемы. Интересно, что последний вариант может быть теоретически оправданным в качестве альтернативы схем подписи GGH и NTRUSign, поскольку он имеет наихудшие гарантии безопасности, основанные на общих решетках в модели случайного оракула. Схема Любашевского и Мичианчо предлагает наихудшие гарантии безопасности, основанные на идеальных решетках, аналогичных темам, рассматриваемым при построении хэш-функций, и это наиболее (асимптотически) эффективная конструкция, известная на сегодняшний день, дающая алгоритмы генерации и проверки подписей, которые работают почти за линейное время. Более того, безопасность не опирается на модель случайного оракула. Схемы цифровой подписи на основе решеток еще не достигли того же уровня зрелости, что и устойчивые к коллизиям хэш-функции и схемы шифрования с открытым ключом.

Теперь мы рассмотрим схему подписи GGH для NTRUSign. Схема шифрования GGH используется для выбора закрытых и открытых ключей. То есть базис решетки B, состоящий из коротких и достаточно ортогональных векторов, представляет собой закрытый ключ. Для той же решетки $\mathcal{L}(B)$ открытый ключ H является «плохим» базисом, состоящим из достаточно

длинных и удаленных от ортогональных векторов. Как и ранее, нормальная форма Эрмита для *B - H.*

Сначала мы отображаем сообщение в точку $m \in R^n$ с помощью некоторой хэш-функции, чтобы подписать его. Мы предполагаем, что хэш-функция работает как случайный оракул, поэтому m равномерно распределено в некоторой области пространства. Следующим шагом является использование секретного базиса для округления *m* до ближайшей точки решетки $s \in \mathcal{L}(B)$. Обычно для этого используется метод округления Бабая, который дает

$$s = B[B^{-1}m] \qquad (18)$$

Обратите внимание, что по определению это подразумевает, что

$$s - m \in P_{1/2}(B) = \{ Bx : x \in [-1/2, 1/2\]^n\} \quad (19)$$

Для проверки заданной пары сообщений-подписей (m, s) необходимо убедиться, что $s \in \mathcal{L}(H) = \mathcal{L}(B)$ и что расстояние $\|s - m\|$ мало (что должно быть мало, поскольку эта разность содержится в $P_{1/2}(B)$).

Джентри и Шидло дали некоторые ранние указания на то, что схемы подписи GGH и NTRUSign могут быть опасными, отметив, что каждая подпись допускает утечку некоторых секретных ключей [57]. Поскольку использование этой информации может быть вычислительно сложным, эта утечка информации не обязательно означает, что такие схемы небезопасны. Тем не менее, как показали Нгуен и Регев несколько лет спустя, эта утечка информации фактически приводит к атакам на схему. Точнее, они продемонстрировали, что можно восстановить закрытый ключ при наличии достаточного количества сообщений-подписок. Кроме того, их атаки были довольно успешными, и они были использованы для большинства разумных выборов параметров в GGH и NTRUSign. Это показало, что эти схемы подписи на самом деле небезопасны. Утечки данных и атаки на самом деле довольно просты. Основное наблюдение заключается в том, что разность $m - s$, полученная из пары сообщение-подпись (m,s), почти равномерно распределена в $P_{1/2}(B)$. Следовательно, при наличии достаточного количества таких пар мы приходим к следующей алгоритмической задаче, известной как задача скрытого параллелепипеда. Эта задача заключается в том, чтобы восстановить параллелепипед или его приближение, когда множество случайных точек равномерно распределено по неизвестному n-мерному параллелепипеду (рисунок 18). Атака, описанная выше, необходима для эффективного решения этой проблемы. В двумерном случае, сразу видно, что параллелепипед охватывает точки, и создание алгоритма, который может реализовать это, не представляет большого труда. Большие задачи часто очень сложны. Тем не менее, здесь задача оказывается простой. Алгоритм, основанный на четвертом моменте одномерных проекций, решает многомерную задачу оптимизации с помощью метода градиентного сглаживания.

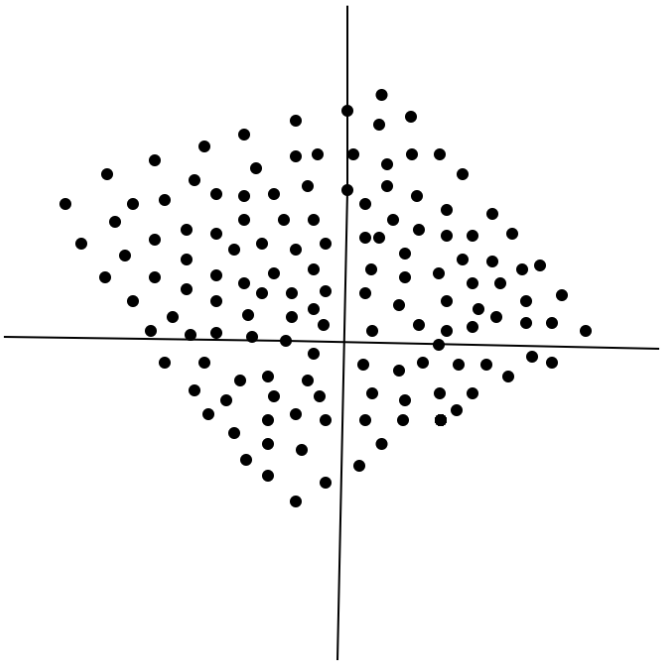


Рисунок 18 – Скрытая задача параллелепипеда в двух измерениях

Методы возмущения являются наиболее эффективными известными контрмерами против вышеупомянутых атак. Они меняют процесс генерации подписи, чтобы заменить скрытый параллелепипед более сложным телом. Похоже, это предотвращает атаки описанного выше типа. Возмущения увеличивают размер секретного ключа и замедляют производство подписей, что является их основным недостатком. Тем не менее, схема подписи NTRUSign все еще неплохо работает. Наконец, несмотря на возмущения, у NTRUSign нет доказательств безопасности. Схемы, основанные на хеш-функциях, устойчивых к коллизиям, а именно Любашевский и Мичианчо [58] предложили схему подписи, которая, по-видимому, оптимальна по всем направлениям, по крайней мере асимптотически: она допускает доказательство безопасности, основанное на предположениях о сложности в худшем случае, доказательство безопасности выполняется в стандартной вычислительной модели (нет необходимости в случайных оракулах), и схема асимптотически эффективна, поскольку размер ключа и время подписи/верификации почти линейны. В основе этой схемы лежит предположение решетки, которое гласит, что ни один алгоритм не может аппроксимировать SVP с точностью до полиномиальных множителей во всех идеальных решетках; более конкретно, это касается решеток, которые замкнуты относительно некоторого вида линейного преобразования F. Схема использует новую схему одноразовой подписи на основе хэша, т.е. схему подписи, которая позволяет безопасно подписывать одно сообщение. Такие схемы могут быть преобразованы в полноценные схемы подписи с использованием стандартных конструкций деревьев, но логарифмической потерей эффективности. Схема одноразовой подписи, в свою очередь, основана на устойчивой к коллизиям хэш-функции, основанной на идеальных решетках. Хэш-функция h может быть

выбрана в процессе генерации ключа или быть фиксированным глобальным параметром. Предполагается, что поиск коллизий в h является вычислительно сложным. Входные данные h можно интерпретировать как последовательность векторов $y_1, \dots . y_{\frac{m}{n}} \in \mathbb{Z}_q^n$ с малыми координатами. Секретный ключ хэш-функции - это пара случайно выбранных входных данных $x_1, \dots . x_{\frac{m}{n}} \in \mathbb{Z}_q^n$ и $y_1, \dots . y_{\frac{m}{n}} \in \mathbb{Z}_q^n$ каждый из которых выбирается в соответствии с соответствующим распределением, которое генерирует короткие векторы с высокой вероятностью

Открытый ключ задается изображениями этих двух входов под хэш-функцией X = h ($x_1, \dots . x_{\frac{m}{n}}$), Y = h ($y_1, \dots . y_{\frac{m}{n}}$). Сообщения, которые необходимо подписать, представлены короткими векторами $\mathbf{m} \in \mathbb{Z}_q^n$. Подпись сообщения m просто вычисляется как:

$$\sigma = \left(\sigma_1, \dots . \sigma_{\frac{m}{n}} \right) = (|F * m| x_1 + y_1, \dots, |F * m \left| x_{\frac{m}{n}} + y_{\frac{m}{n}} \right) mod\ q \quad (20)$$

Подпись проверяется путем проверки того, что σ представляет собой последовательность коротких векторов, хэш-функция которых равна $[F*m]$X + Y mod q.

Безопасность схемы основана на том факте, что даже после просмотра подписи точное значение секретного ключа по-прежнему остается информацией, теоретически скрытой от злоумышленника. Поэтому, если злоумышленнику удастся придумать поддельную подпись, она, скорее всего, будет отличаться от той, которую законный подписчик может вычислить с помощью секретного ключа. Поскольку поддельная подпись и законная подпись хэшируются к одному и тому же значению, они создают коллизию в хэш-функции.

С учетом растущих угроз, связанных с развитием квантовых вычислений, электронные подписи на основе решеток являются одним из наиболее перспективных направлений в современной криптографии. Эти подписи основаны на решении сложных математических задач, таких как проблемы с решётками, которые, как известно, остаются вычислительно устойчивыми даже для квантовых компьютеров, как показали последние данные. Это делает их неотъемлемой частью разработки постквантовых криптографических систем. Основным преимуществом таких подписей является их высокая устойчивость к квантовым атакам, что позволяет их использовать для защиты важных данных. Схемы, такие как Falcon и CRYSTALS-Dilithium, демонстрируют высокую производительность и подходят для широкого спектра приложений, включая защищенную коммуникацию, блокчейн-технологии и Интернет вещей.

В рамках конкурса NIST для создания постквантовых стандартов был разработан современный криптографический алгоритм CRYSTALS-Dilithium [59]. Он разработан с целью реализации схем цифровой подписи, устойчивых к атакам квантовых компьютеров. Проблемы поиска кратчайшего вектора (SVP) и проблема нахождения решеточного вектора, близкого к целевому (LWE), являются одними из математических задач, связанных с решетками, на которых

основано дилимиум. Даже квантовые компьютеры, обеспечивающие высокий уровень безопасности, считают эти задачи вычислительно сложными.

Основные характеристики CRYSTALS-Dilithium:

1. Криптографическая основа

Dilithium использует структуру модульных решеток, что делает его высокоэффективным в вычислительном плане. Он основан на алгоритме Fiat-Shamir с преобразованием без перехеширования, что позволяет минимизировать сложность и повысить производительность.

2. Параметры и безопасность

Схема обеспечивает три уровня безопасности, соответствующие классическим и постквантовым стандартам:

Уровень 2 (аналогично RSA-2048).

Уровень 3 (аналогично AES-192).

Уровень 5 (аналогично AES-256).

3. Размеры ключей и подписей

Открытый ключ: от 1 до 2 килобайт в зависимости от уровня безопасности.

Подпись: от 2,4 до 4,5 килобайт.

4. Производительность

– быстрая генерация ключей и создание подписи;

– высокая скорость проверки подписи;

– эффективная реализация на устройствах с ограниченными ресурсами.

Преимущества CRYSTALS-Dilithium:

Устойчивость к квантовым атакам. Используемые решеточные задачи остаются сложными даже для квантовых компьютеров, что гарантирует долгосрочную безопасность.

Производительность:

– алгоритм хорошо оптимизирован для работы как на классических устройствах, так и в распределённых системах;

– эффективно реализуется на встраиваемых устройствах и в облачных средах.

Стандартизация. Dilithium выбран NIST в качестве основного кандидата для постквантового стандарта цифровых подписей, что подчеркивает его надежность и готовность к практическому использованию. Недостатки и вызовы.

Размеры ключей и подписей. В сравнении с классическими схемами (например, RSA или ECDSA), размеры ключей и подписей Dilithium значительно больше, что может стать проблемой для систем с ограниченной пропускной способностью или памятью.

Атаки на побочные каналы. Как и любой криптографический алгоритм, Dilithium уязвим к атакам, использующим побочные каналы (например, анализ потребления энергии или времени выполнения), что требует дополнительных мер защиты.

Falcon – это один из постквантовых криптографических алгоритмов цифровой подписи, который был разработан для обеспечения устойчивости к атакам квантовых компьютеров. Он основан на решеточной криптографии и использует сложные математические задачи, связанные с решетками, в частности, NTRU (Nth Degree Truncated Polynomial Ring). Falcon входит в число алгоритмов, выбранных NIST в качестве финалистов для стандартизации постквантовых цифровых подписей. Основные характеристики Falcon:

1. *Криптографическая основа*. Falcon использует структуру решеток и методы коррекции ошибок, такие как Fast Fourier Sampling (FFS). Этот подход обеспечивает высокую эффективность и компактные размеры ключей и подписей.

2. *Параметры и безопасность*. Falcon предоставляет различные уровни безопасности, аналогичные стандартам AES:

Уровень 1: обеспечивает защиту, эквивалентную AES-128.

Уровень 5: обеспечивает защиту, эквивалентную AES-256.

3. Размеры ключей и подписей

Открытый ключ: около 897 байт (для уровня 1) и 1,793 байта (для уровня 5).

Подпись: около 666 байт (для уровня 1) и 1,280 байт (для уровня 5). Это делает Falcon одним из самых компактных алгоритмов цифровой подписи среди постквантовых схем.

4. Производительность

– высокая скорость создания подписи;

– быстрая проверка подписи благодаря использованию быстрого преобразования Фурье (FFT).

Преимущества Falcon:

1. *Компактные размеры ключей и подписей.* Falcon имеет одни из самых маленьких размеров подписей и ключей среди постквантовых алгоритмов, что делает его подходящим для систем с ограниченными ресурсами, таких как Интернет вещей (IoT).

2. *Высокая производительность.* Алгоритм оптимизирован для работы как на классических процессорах, так и на встраиваемых устройствах. Быстрота генерации и проверки подписей делает Falcon эффективным для реальных приложений.

3. *Устойчивость к квантовым атакам***.** Основание на задачах решеточной криптографии обеспечивает защиту от атак с использованием квантовых компьютеров.

4. *Стандартизация***.** Falcon был выбран NIST в числе лидеров для стандартизации, что подчеркивает его надежность и пригодность для практического использования.

Недостатки Falcon:

1. *Сложность реализации.* Реализация Falcon требует точных и сложных вычислений с плавающей точкой, что делает алгоритм чувствительным к

ошибкам при программировании и увеличивает сложность его внедрения в аппаратные и программные системы.

2. *Уязвимость к атакам на побочные каналы.* Как и другие решеточные алгоритмы, Falcon подвержен атакам на побочные каналы (например, анализу времени выполнения или энергопотребления). Для его защиты требуется дополнительное усиление безопасности.

3. *Ограниченная производительность на маломощных устройствах.* Хотя Falcon подходит для систем с ограниченными ресурсами, его зависимость от вычислений с плавающей точкой может быть проблемой для некоторых устройств с низкой производительностью.

4. *Сложность анализа безопасности.* Из-за специфики используемых решеточных задач и методов, алгоритм Falcon требует тщательного математического анализа для исключения потенциальных уязвимостей.

Ключевые проблемы, такие как большие размеры подписей и ключей, все еще существуют. Это вызывает дополнительные исследования, направленные на улучшение таких схем и адаптацию их к устройствам с ограниченными ресурсами. Это подтверждает важность электронных подписей на основе решеток, которые уже включены в стандарты постквантовой криптографии. Они обеспечивают надежную основу для безопасности данных в долгосрочной перспективе в условиях развития квантовых технологий. Внедрение и развитие электронных подписей на основе решеток будет важным шагом в обеспечении безопасности цифровых систем, адаптированных к вызовам постквантовой эпохи. Эти технологии не только обеспечивают защиту от квантовых атак, но и предоставляют новые возможности для использования в современных протоколах криптографической криптографии [60]. Например, они могут быть интегрированы в системы электронного правительства, финансовых транзакций и инфраструктуры Интернета вещей, где требуется высокая степень защиты и надежности. Благодаря своей математической основе такие подписи обладают уникальными свойствами, которые делают их подходящими для масштабирования в глобальных системах. С дальнейшим совершенствованием алгоритмов и увеличением их эффективности электронные подписи на основе решеток могут стать стандартом де-факто в индустрии. Это подчеркнет важность криптографических решений, устойчивых как к квантовым, так и к классическим атакам, для долгосрочной защиты данных. Таким образом, исследования в этой области не только решают актуальные проблемы, но и прокладывают путь для развития безопасных технологий будущего.

1.7 Постквантовые стандарты NIST

Квантовые вычисления создают огромные экономические и научные возможности, учитывая их способность значительно повышать вычислительную мощность. Однако квантовые вычисления, которые используют квантовую механику для решения некоторых сложных вычислительных задач, также могут сделать некоторые из текущих алгоритмов шифрования устаревшими, что

создает серьезные риски для кибербезопасности. Квантовые технологии в целом все еще находятся на раннем этапе развития. Однако краткосрочные и долгосрочные прогнозы показывают, что технология имеет большой потенциал для создания новых возможностей в области кибербезопасности. Хотя квантовые компьютеры все еще находятся в процессе разработки, эксперты прогнозируют, что в течение следующих десяти лет квантовые компьютеры смогут взломать шифрование, что поставит под угрозу «безопасность и конфиденциальность отдельных лиц, организаций и целых стран». На протяжении многих лет открытие и использование квантовых явлений проложило путь для многих технологических инноваций, таких как лазеры, полупроводники и системы медицинской визуализации. В настоящее время обычно говорят о квантовом в трех основных областях: варианты использования квантовых вычислений; квантовые технологии безопасности, такие как квантовое распределение ключей (QKD) и квантовая генерация случайных чисел (QRNG); и постквантовая криптография (PQC).

Разработчики и пользователи средств криптографической защиты информации беспокоятся об угрозе квантового компьютера после публикации алгоритма Шора для факторизации и дискретного логарифмирования в 1994-1995 годах. Это связано с тем, что наиболее распространенные асимметричные схемы шифрования и цифровой подписи, такие как RSA, DSA могут стать уязвимыми в случае появления квантового компьютера с достаточной производительностью [61]. Начиная с 2017 года усилия Национального института стандартов и технологий США (NIST) придали новый импульс исследованиям в области синтеза и анализа квантово-устойчивых криптографических схем. NIST организовал открытую международную площадку (конкурс) для подачи предложений по стандартизации и обсуждения постквантовых криптографических схем. Алгоритмы-кандидаты, представленные для участия в конкурсе NIST, выполняют следующие механизмы: цифровую подпись и инкапсуляцию ключей. Эти механизмы подвергаются определенным требованиям по устойчивости, которые являются как теоретическими (официально подтвержденными), так и практическими (установлены определенные уровни устойчивости относительно всех известных классических и квантовых атак). Практическая применимость предлагаемых алгоритмов также является важным критерием. Алгоритмы проходят как внутреннюю проверку сотрудниками NIST, так и открытую проверку криптографами со всего мира. На сайте NIST и в публичном списке рассылки обсуждаются результаты экспертизы. Новые стандарты предназначены для двух основных задач, для которых обычно используется шифрование: общее шифрование, используемое для защиты информации, передаваемой через общедоступную сеть и цифровые подписи, используемые для аутентификации личности. Национальный институт стандартов и технологий США (NIST) объявил победителей конкурса криптоалгоритмов, стойких к подбору на квантовом компьютере. Он стартовал шесть лет назад и был нацелен на выбор алгоритмов постквантовой криптографии, пригодной для выдвижения в качестве

стандартов. Выбранные алгоритмы шифрования станут частью постквантового криптографического стандарта NIST, который, как ожидается, будет завершён примерно через два года. Победителем среди универсальных алгоритмов, которые можно использовать для защиты передачи информации в компьютерных сетях, выбрали CRYSTALS-Kyber, сильные стороны которого - относительно небольшой размер ключей и высокая скорость работы.

Kyber вышел из метода, опубликованного Одедом Регевым в 2005 году. Этот метод был разработан разработчиками из Европы и Северной Америки, которые работали в частных компаниях, государственных университетах или научно-исследовательских институтах с финансированием Европейской комиссии, Швейцарии, Нидерландов и Германии. В качестве дополнительного элемента их «Криптографического набора алгебраических решеток» (CRYSTALS) они также разработали схему подписи Dilithium. Kyber широко использует хеширование внутри компании, как и другие методы PQC-KEM. В Kyber используются варианты Keccak (SHA-3/SHAKE) для генерации псевдослучайных чисел. Национальный институт стандартов и технологий США (NIST) использовал этот метод для своего публичного процесса отбора для Первого стандарта для квантово-безопасных криптографических примитивов (NISTPQC) в 2017 году. Это единственный механизм инкапсуляции ключей, выбранный для стандартизации в конце третьего раунда процесса стандартизации NIST. В отчете, объявляющем о решении, говорится, что это связано с выполнением нескольких патентных соглашений, в которых NTRU является запасным вариантом. Целью стандартизации дополнительного KEM в настоящее время является четвертый раунд процесса стандартизации. На втором этапе процесса выбора были изменены несколько параметров алгоритма, а также было исключено сжатие открытых ключей. В NIST сосредоточился на расходах с точки зрения времени выполнения и реализаций, маскирующих время выполнения, чтобы предотвратить потенциальные атаки по сторонним каналам (SCA). На четвертом раунде процесса постквантовой стандартизации NIST выбрал Falcon в качестве постквантовой схемы подписи. Томас Прест, Пьер-Ален Фуке, Джеффри Хоффштейн, Пол Киршнер, Вадим Любашев, Томас Порнин, Томас Рикоссет, Грегор Сейлер, Уильям Уайт и Чжэньфэй Чжан были разработчиками. Он основан на методе хэш-и-подписи над решетками NTRU для фреймворка Джентри, Пейкерта и Вайкунтанатана[62,63]. Название Falcon является аббревиатурой от компактных подписей, основанных на Fast Fourier grille over NTRU. Поскольку квантовые вычисления развиваются так быстро, выпуск стандартов NIST представляет собой чрезвычайно важную инициативу, которая помогает организациям перейти на квантово-безопасные технологии. Из алгоритмов, нацеленных на работу с цифровыми подписями, выделили CRYSTALS-Dilithium, FALCON и SPHINCS+.

Алгоритмы CRYSTALS-Dilithium и FALCON отличаются высокой эффективностью. В качестве первичного алгоритма для цифровых подписей рекомендовали CRYSTALS-Dilithium, FALCON ориентирован на решения, в которых требуется минимальный размер подписи. SPHINCS+ отстаёт от первых

двух алгоритмов по размеру подписей и скорости работы, но его оставили в числе финалистов в качестве запасного варианта.

PQC использует новые математические алгоритмы криптографии с открытым ключом, которые разработаны так, чтобы быть неуязвимыми для атак алгоритма Шора. Эти стандарты идеально подходят для обновления текущих криптографических алгоритмов, чтобы быть защищенными от (известных в настоящее время) квантовых атак, и могут быть реализованы в программных решениях в рамках существующей инфраструктуры. Это позволяет осуществлять почти прямое развертывание в существующей инфраструктуре с четкими, глобальными, проверенными и признанными стандартами. Хотя они идеально подходят для миграции криптографических алгоритмов в квантово-безопасную версию, они также могут иметь определенные текущие недостатки производительности и могут быть оспорены потенциальными разработками в области классических и квантовых атак (криптоанализа), которые могут повлиять на безопасность этих схем в будущем. Многочисленные усилия направлены на снижение угрозы квантов [64]. Хотя эти технологии не представляют собой единое целое, они могут быть использованы для определенных целей и сценариев. Помимо PQC, есть и другие технологии, которые также привлекают внимание и могут помочь смягчить риск, создаваемый квантовой криптографией с открытым ключом: QKD и QRNG.

QKD разрабатывает квантовые методы на основе физики для создания защищенных каналов связи, которые могут использоваться для распространения ключей шифрования. Считается, что протокол невосприимчив к атакам методом подбора, даже при бесконечной вычислительной мощности, и использует принцип «суперпозиции», чтобы гарантировать, что подслушивающий не сможет незаметно прослушать сообщение. Этот протокол предназначен для обмена секретными ключами, которые впоследствии используются для шифрования сообщения с использованием квантово-безопасных алгоритмов. Таким образом, QKD может помочь снизить квантовый риск и может дополнить PQC и другие криптографические алгоритмы, предоставляя безопасный метод распространения ключей. Хотя преимущества безопасности могут быть значительными, использование QKD требует значительных финансовых вложений в специализированное оборудование, имеет ограничения по расстоянию и требует отдельного канала аутентификации[65].

Для создания случайных чисел с высокой энтропией QRNG использует основные квантовые свойства. Случайность является ключевой частью криптографии. QRNG потенциально может производить более проверенные энтропийные источники, чем обычные процессы, что может повысить безопасность при определенных условиях. Генерация случайных чисел играет решающую роль в криптографии, как для генерации криптографических ключей, так и в некоторых алгоритмах. В то время как классические генераторы случайных чисел (ГСЧ) выводятся из некоторого источника энтропии (например, теплового шума), QRNG по своей сути случайны. Таким образом, QRNG могут повысить безопасность криптографических систем в целом, хотя они специально

не смягчают квантовую угрозу. Кроме того, некоторые приложения требуют повторяемости, что невозможно для QRNG. При наличии множества решений, в зависимости от вариантов их использования, организации, внедряющие квантово-устойчивую безопасность, могут использовать гибридные решения, которые интегрируют как классические, так и квантово-готовые подходы. Гибридные решения (системы как с классическими криптографическими, так и с квантово-ориентированными компонентами шифрования) также требуют от организаций повышения их крипто-гибкости для создания постоянных возможностей для развития криптографических стандартов и решений. Этот крипто-гибкий подход требует свежего взгляда на криптографическое управление и изучения новых способов развертывания крипто-гибких программных фреймворков и архитектур. Крайне важно, чтобы организации начали свой путь к квантовой кибер-готовности сегодня, разработав стратегию и дорожную карту сегодня. Несмотря на достигнутые успехи, разработка постквантовых алгоритмов продолжает оставаться проблемой. Это требует тщательного подхода к оптимизации их реализации, уменьшению количества подписей и ключей, а также созданию средств предотвращения атак на побочные каналы.

Постквантовые стандарты NIST положили основу для будущей криптографической инфраструктуры, которая обеспечит безопасность данных в условиях быстрой технической революции. Они должны быть успешно реализованы, чтобы обеспечить устойчивость цифровых систем в ближайшие десятилетия.

Выводы по первому разделу.

Данный раздел посвящен полному обзору и анализу современной криптографии, который представлен ниже.

1. Приведена обзор современной криптографии, которая представляет собой сложную и многоуровневую систему защиты информации, которая развивается в ответ на новые вызовы безопасности в цифровом мире, описаны области применения современной криптографии, где важным направлением является защита государственных и корпоративных систем от кибератак, а также обеспечение безопасности на уровне блокчейн-технологий и криптовалют.
2. Описана постквантовая криптография, которая разрабатывается с целью обеспечения безопасности данных в условиях появления квантовых компьютеров, способных нарушить устойчивость традиционных криптографических алгоритмов. Для этого используются перспективные подходы, такие как криптография на основе решеток, кодов, хеширования и многомерных вычислений.
3. Рассмотрены решеточные вычислительные задачи, которые являются важной частью постквантовой криптографии, включая задачи о ближайшем, кратчайшем и коротком целочисленном решении векторах.
4. Описаны полилинейные алгебра и ее связь с решетками. Рассмотрены тензорные разложения в SIS и LWE.

5. Рассмотрены электронные подписи на основе решеток, а именно схема подписи GGH в NTRUSign. Описана схема шифрования GGH, которая использует решеточные задачи для создания эффективного и безопасного метода шифрования.

6. Приведен обзор постквантовых стандартов NIST, а именно алгоритмов CRYSTALS-Kyber, CRYSTALS-Dilithium, SPHINCS+ и FALCON, которые являются одними из самых популярных. Эти стандарты используются для решения двух основных задач, для которых обычно применяется шифрование: общее шифрование, обеспечивающее защиту информации, передаваемой через общедоступные сети, и цифровые подписи, используемые для аутентификации личности.

2 ОБМЕН КРИПТОГРАФИЧЕСКИМИ КЛЮЧАМИ

Обмен криптографическим ключом, особенно в симметричной криптографии, имеет решающее значение для безопасности многих криптографических систем. Тем не менее, существует ряд проблем, связанных с этим процессом. К ним относятся необходимость безопасной передачи ключа через незащищенные каналы, управление большим количеством ключей в масштабируемых системах и защита от атак, таких как атаки посредника. Протоколы обмена ключами можно разделить на две категории: протоколы передачи ключей и протоколы согласования ключей. В протоколах передачи ключей сеансовый ключ сначала создается одним из участников взаимодействия, а затем передается другому. С другой стороны, протокол согласования ключей опирается на некоторую информацию от обеих сторон для получения сеансового ключа. Один из первых появившихся протоколов обмена ключами называется Diffie-Hellman key exchange[66]. Целью протокола Diffe-Hellman является предоставление возможности двум сторонам безопасно обмениваться сеансовым ключом, который затем может использоваться для следующего симметричного шифрования сообщений. Идея протокола Diffe-Hellman заключается в вычислении сеансового ключа взаимодействующими субъектами на основе открытых параметров, которые являются общими на начальном этапе. Этот тип протокола называется протоколом согласования ключей. Эффективность Diffe-Hellman обусловлена сложностью вычисления дискретных логарифмов. Однако протокол можно использовать только для обмена секретными данными без аутентификации двух сторон. Вот почему Diffe-Hellman небезопасен для атак типа «человек посередине». Решением этой уязвимости является использование цифровой подписи. С момента его появления предлагаются варианты протоколов Diffe-Hellman для преодоления различных проблем и уязвимостей, включая упомянутую. Мы можем рассматривать протоколы обмена ключами или установления с двух точек зрения: стоимость, эффективность и безопасность. Стоимость включает как затраты на обработку, так и затраты на связь. Чтобы получить низкую стоимость обработки, исследователям следует избегать использования схем шифрования с открытым ключом, таких как RSA, ECC и ElGamal. Безопасность означает устойчивость протокола к известным атакам, таким как атака с подменой ключа (KCI), атака с подменой эфемерного ключа (ECI), атака по словарю и т.д.Методы обмена ключами, такие как протоколы Диффи-Хеллмана и методы на основе асимметричной криптографии, позволяют безопасно обмениваться ключами, не передавая их напрямую. Важно также постоянно использовать системы аутентификации и обновлять ключи, чтобы предотвратить несанкционированный доступ. В области криптографии и защиты данных совершенствование методов обмена ключами и повышение их безопасности остаются важными задачами.

2.1 Проблемы обмена ключами в симметричной криптографии

Симметричная криптография использует один и тот же секретный ключ, который должен быть известен обеим сторонам для шифрования и дешифрования данных. Однако в процессе обмена этим ключом есть несколько проблем, которые могут повлиять на безопасность системы, и мы рассмотренные ниже.

1. Безопасность канала связи.

Для безопасного обмена симметричными ключами между сторонами необходимо наличие защищенного канала связи. Злоумышленник может перехватить или изменить ключ этого канала, если он небезопасен, например, если он используется в Интернете без защиты [67,68]. Ключ может быть перехвачен на стадии обмена, что делает криптосистему уязвимой для атак. Использование безопасных протоколов, таких как TLS, или предварительная договоренность сторон через физически защищенные каналы связи, может помочь решить эту проблему.

2. Уязвимость к атаке «человек посередине» (Man-in-the-Middle).

Атака «человек посередине» возможна при обмене симметричным ключом без предварительной аутентификации сторон. Злоумышленник может вмешаться в процесс обмена ключами, подменив ключ и получить доступ к зашифрованной информации. Чтобы предотвратить эту атаку, необходимо использовать механизм аутентификации сторон, что значительно усложняет сам процесс обмена.

3. Масштабируемость.

Чтобы обеспечить безопасность обмена ключами между участниками системы, каждый участник должен создать собственный секретный ключ, который может использоваться для связи с каждым другим участником. В случае n пользователей каждая пара участников потребует передачи ключей друг другу, что приводит к экспоненциальному росту числа ключей. Такая система требует создания нескольких ключей для каждого нового участника, что может быть неэффективным и сложным для масштабируемых сетей.

4. Управление ключами.

Ключи симметричной криптографии должны быть надежно защищены и управляться. Это позволяет избежать утечки ключей. Чтобы предотвратить несанкционированный доступ, строгие правила безопасности должны соблюдаться при сохранении и передаче ключей. Проблемы с ключами включают потерю, просачивание или неправильную замену. Кроме того, необходимо регулярно обновлять ключи, что делает управление криптографической системой еще более сложным.

5. Проблемы с распределением ключей в больших системах.

Большие распределенные системы требуют безопасного и эффективного распределения ключей. В такой системе ручное распределение ключей может быть неэффективным, а использование централизованной системы для обмена ключами может создать уязвимость, поскольку компрометация этого центра поставит под угрозу всю систему. Это особенно важно для Интернета вещей

(IoT), поскольку каждый объект нуждается в отдельном ключе для обмена данными.

6. Отсутствие аутентификации.

Симметричная криптография не всегда использует встроенную аутентификацию в процессе обмена ключами; это делает систему уязвимой, если злоумышленник может выдать себя за доверенную сторону и взять ключ. Это отличается от асимметричной криптографии, где цифровые подписи могут использоваться для аутентификации сторон.

7. Обновление ключей.

Однако частая замена симметричных ключей требует дополнительных операций и может усложнить процесс, особенно в больших распределенных системах, поэтому они должны обновляться регулярно.

Для обеспечения безопасности связи необходимо, чтобы обе стороны обменивались известной информацией. Эта информация, называемая криптографическим ключом, позволяет отправляющей стороне успешно зашифровать, а принимающей - успешно расшифровать передаваемую информацию. Однако, поскольку для успешной передачи информации обе стороны должны обладать одним и тем же ключом, криптографы на протяжении всей истории человечества пытались найти способ передать этот ключ от отправляющей стороны к принимающей, не перехватывая его по пути. Какое-то время, вплоть до появления электронной криптографии, ключи приходилось передавать лично, по почте или с курьером, поскольку это были единственные относительно безопасные способы передачи информации от одного человека к другому. Однако позже, в компьютерную эру, ученые-компьютерщики и криптографы стали разрабатывать схемы успешной и безопасной передачи ключевого материала по незащищенным каналам. В результате сегодня существуют решения проблемы распределения ключей как для симметричных, так и для криптосистем с открытым ключом. С симметричной стороны наиболее перспективными являются два решения: использование функции деривации ключа или схемы обертывания ключа.

Функция деривации ключа или схема обертывания ключа [69]. Оба эти метода обеспечивают безопасную передачу симметричного криптографического ключа от отправителя к получателю без угрозы внешнего вторжения. Для систем с открытым ключом жизнеспособными и эффективными решениями являются как головоломки Меркла, так и обмен ключами Диффи-Хеллмана. Хотя эти методы различаются по типу криптосистемы, в которой они применяются, существуют и различия в методах их действия. Оба метода, используемые в симметричных системах, - функции деривации ключа и обертывание ключа - представляют собой протоколы, которые напрямую передают ключевой материал. Другими словами, сам ключ (хотя и успешно экранированный) передается по незащищенному каналу. И наоборот, оба протокола, используемые в системах с открытым ключом, являются протоколами согласования ключей. В этих схемах по незащищенному каналу передается тривиальная информация, которая используется обеими сторонами для создания

одного и того же общего ключа, управляемого заранее определенным набором правил. С помощью всех четырех протоколов криптографические ключи могут успешно и безопасно передаваться от одной стороны к другой. Однако при рассмотрении будущего вычислительной техники и ее влияния на криптографический ландшафт следует учитывать проблемы представлены ниже.

Квантовые вычисления - это технология будущего, которая уже сегодня демонстрирует впечатляющие возможности. В 2025 году компании, такие как Google, IBM и другие, продолжают активно разрабатывать и тестировать квантовые процессоры с увеличенным числом кубитов и улучшенной коррекцией ошибок. Современные квантовые компьютеры уже показывают вычислительное превосходство над классическими системами в решении узких задач, таких как моделирование квантовых систем и оптимизационные задачи.

С развитием полноценных квантовых вычислений существующие криптографические протоколы, включая алгоритмы шифрования (RSA, ECC), дешифрования и распределения ключей, становятся уязвимыми. Это обусловлено тем, что квантовые компьютеры, используя алгоритм Шора, смогут эффективно разложить большие числа на множители, а алгоритм Гровера ускоряет атаку на симметричные шифры. В результате традиционные криптографические алгоритмы оказываются под угрозой взлома, что требует разработки и стандартизации постквантовой криптографии (PQC, Post-Quantum Cryptography)**.** Национальные и международные организации, такие как NIST, ETSI и ISO, уже ведут активную работу по утверждению новых криптографических стандартов, устойчивых к квантовым атакам[70]. В 2022 году NIST выбрал несколько финальных кандидатов, а в 2024 году опубликованы первые стандартизированные постквантовые алгоритмы, включая CRYSTALS-Kyber, CRYSTALS-Dilithium, Falcon и SPHINCS+. Проблема распределения ключей также переносится в квантовое пространство, что привело к разработке квантово-безопасных методов, таких как протокол BB84 и другие схемы квантового распределения ключей (QKD). В 2025 году несколько стран и крупных корпораций уже начали внедрение квантово-защищённых коммуникационных сетей, а спутниковые системы QKD демонстрируют практическую реализуемость безопасной связи на глобальном уровне. Стандартизация и внедрение постквантовой криптографии в государственные, финансовые и корпоративные системы продолжается, что позволит гарантировать безопасность данных и конфиденциальность пользователей в условиях эры квантовых вычислений.

В соответствии с рисунком 19, одним из самых важных проблем для безопасности является проблема обмена ключами в симметричной криптографии.

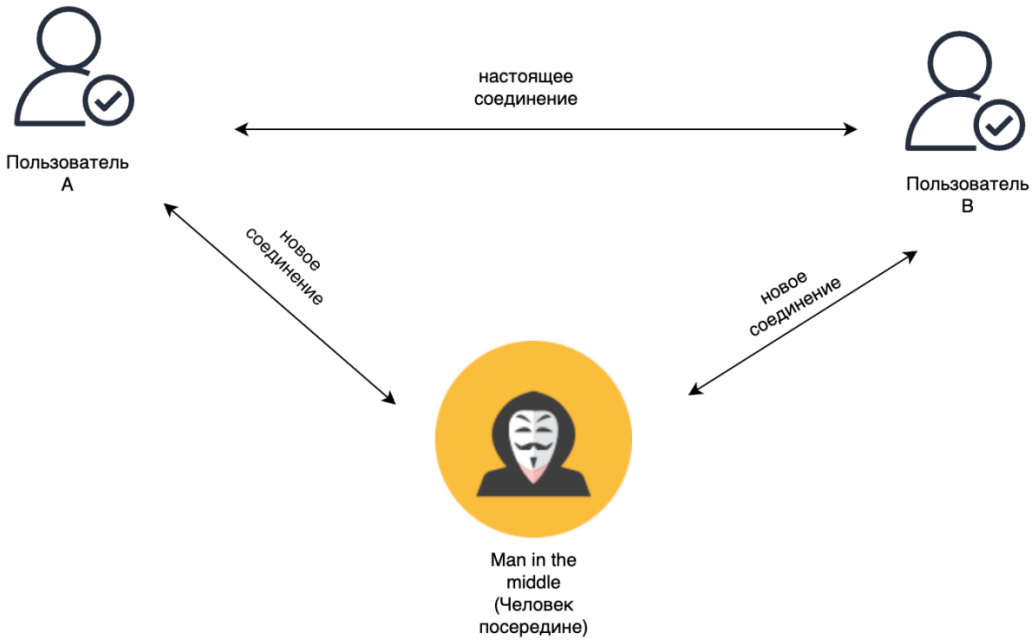


Рисунок 19 - Проблема обмена ключами в симметричной криптографии (Man in the middle)

В настоящее время широко используются гибридные системы, которые объединяют симметричное и асимметричное шифрование. Для безопасного обмена симметричными ключами можно использовать асимметричные подходы, такие как использование публичных и приватных ключей [71]. Эти подходы значительно улучшают безопасность и упрощает управление ключами. Кроме того, создание протоколов криптографического обмена ключами, таких как протоколы Diffie-Hellman и протоколы с использованием цифровых подписей, помогает снизить риски, связанные с обменом ключами.

2.2 Схема шифрования открытого ключа Эль-Гамаля

Алгоритм шифрования с открытым ключом, разработанный Тахером Эль-Гамалем [72,73] в 1985 году и основанный на обмене ключами Диффи-Хеллмана, известен как система шифрования Эль-Гамаля. Предположим, что A и B хотят поделиться секретным K_{AB} , где у A есть секрет x_A, а у B есть секрет x_B. Пусть p будет большим простым числом, а α будет примитивным элементом mod p, оба известны. A вычисляет $y_A \equiv \alpha^{X_A} \bmod p$ и отправляет y_A. Также B вычисляет $y_B \equiv \alpha^{X_B} \bmod p$ и отправляет y_B.Тогда секретный K_{AB} вычисляется как

$$K_{AB} \equiv \alpha^{X_A X_B} \bmod p \equiv y_A^{X_B} \bmod p \equiv y_B^{X_A} \bmod p \quad (21)$$

Следовательно, A и B способны вычислить K_{AB}. Но для злоумышленника вычисление K_{AB} кажется сложным. Это еще не доказательство того, что взлом системы эквивалентен вычислению дискретных логарифмов.

В любой из криптографических систем на основе дискретных логарифмов p должно быть выбрано таким образом, чтобы p-1 имело хотя бы один большой

простой множитель. Если p-1 имеет только маленькие простые множители, то вычисление дискретных логарифмов выполняется легко.

Теперь предположим, что A хочет отправить B сообщение m, где $0 \leq m \leq p-1$. Сначала A выбирает число k равномерно между 0 и p-1. Обратите внимание, что k будет служить секретом x_A в схеме распределения ключей. Затем A вычисляет «ключ»

$$K \equiv y_B^k \bmod p \tag{22}$$

где $y_B \equiv \alpha^{X_B} \bmod p$ либо находится в общедоступном файле, либо отправлено B. Зашифрованное сообщение представляет собой пару (c_1, c_2), где

$$c_1 \equiv \alpha^k \bmod p \quad c_2 \equiv Km \bmod p \tag{23}$$

и K рассчитывается в (22).

Обратите внимание, что размер шифртекста в два раза больше размера сообщения. Кроме того, операция умножения в (23) может быть заменена любой другой обратимой операцией, например суммированием по модулю p. Операция расшифровки делится на две части. Первый этап - восстановление K, что легко для B, поскольку $K \equiv (\alpha^k)^{X_B} \equiv c_1^{X_B} \bmod p$, а x_B известно только B. Второй этап - деление c_2 на K и восстановление сообщения m.

Публичный файл включает одну запись для каждого пользователя, а именно y_i для пользователя i (так как α и p указаны для всех пользователей). Вполне вероятно, что каждый пользователь выбирает свои собственные α и p, что более подходит с точки зрения безопасности, несмотря на то, что это утроит размер публичного файла [74]. Нежелательно использовать одно и то же значение k для шифрования более чем одного блока сообщения, поскольку при многократном использовании k знание одного блока m_1 сообщения позволяет злоумышленнику вычислить другие блоки следующим образом.

Пусть $c_{1,1} \equiv \alpha^k \bmod p$, $c_{2,1} \equiv m_1 K \bmod p$, $c_{1,2} \equiv \alpha^k \bmod p$, $c_{2,2} \equiv m_2 K \bmod p$.

После этого $m_1/m_2 \equiv c_{2,1}/c_{2,2} \bmod p$, и m_2 легко вычисляется, если известно m_1 (рисунок 20). В соответствии с рисунком 20, взлом системы идентичен разрушению схемы распределения Диффи-Хеллмана. Во-первых, если *m* также может быть получено из c_1, c_2 и *y*, то *K* также может быть получено из y, c_1 и c_2 (что кажется случайным числом, поскольку k и *m* неизвестны). Этого достаточно, чтобы подорвать схему распределения. Во-вторых, (даже предположив, что *m* известно) вычисление k или x из c_1 и c_2 и y равно вычислению дискретных логарифмов. Причина в том, что и x, и k появляются в показателе степени в y и c_1. Схема шифрования с открытым ключом Эль-Гамаля не устойчива к квантовым атакам. Она основана на сложности задачи дискретного логарифма, которая может быть эффективно решена квантовыми алгоритмами, такими как алгоритм Шора.

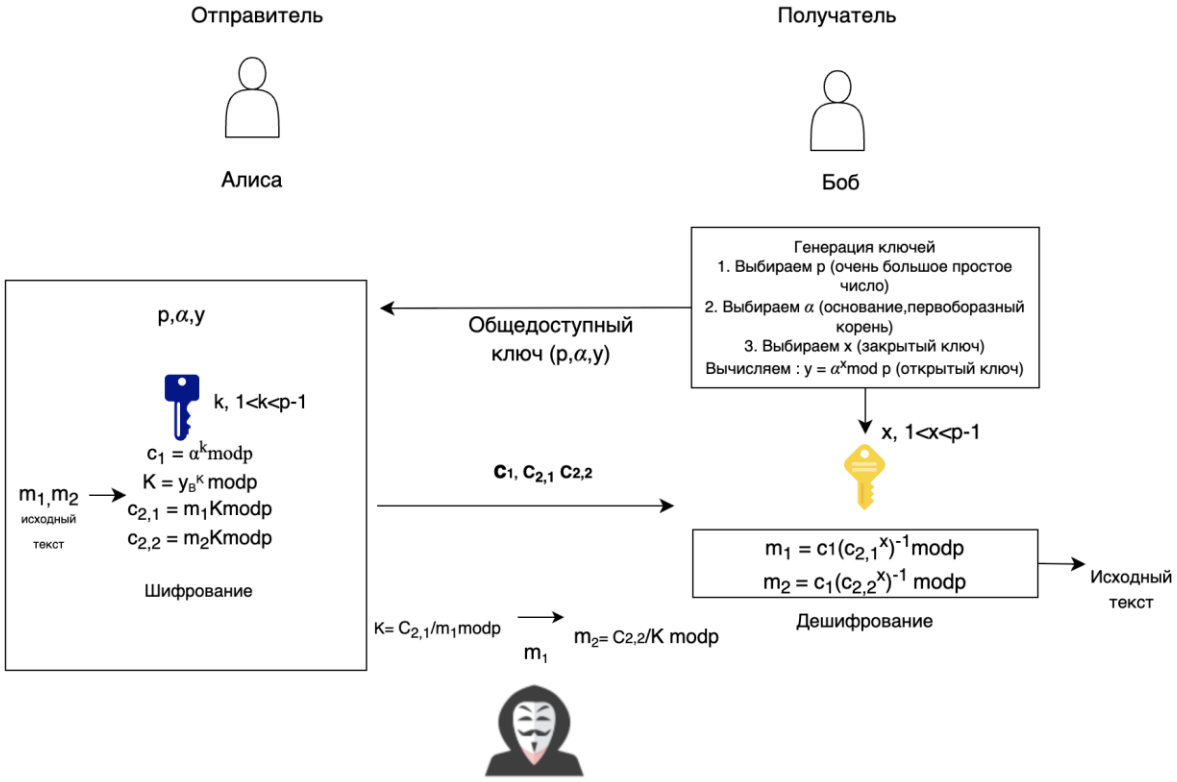


Рисунок 20 – Уязвимость схема шифрования с открытым ключом Эль-Гамаля

2.3 Построение протокола обмена ключами на основе задачи SIS

Одной из фундаментальных задач криптографии является обеспечение надёжного и безопасного обмена секретными ключами между двумя сторонами, взаимодействующими по открытому, потенциально прослушиваемому каналу. Такие протоколы лежат в основе защищённого общения в сети Интернет, мобильных коммуникаций, банковских транзакций и других критически важных цифровых сервисов. На протяжении десятилетий основой криптографической безопасности служили задачи, эффективность решения которых невозможна при использовании классических вычислительных методов. К таким задачам относятся факторизация больших целых чисел, дискретный логарифм и дискретный логарифм по эллиптической кривой. Однако с развитием квантовых вычислительных технологий, особенно в связи с появлением алгоритма Шора, оказалось, что эти задачи могут быть решены за полиномиальное время на квантовом компьютере. Это делает значительную часть современных криптографических протоколов уязвимыми в постквантовой парадигме. В ответ на этот вызов научное сообщество активно исследует так называемую постквантовую криптографию - направление, ориентированное на разработку криптографических схем, устойчивых к атакам с использованием квантовых вычислений. Одним из наиболее перспективных направлений в этой области является криптография, основанная на задачах теории решёток**,** отличающихся высокой вычислительной сложностью даже для квантовых алгоритмов. Одной из центральных и наиболее изученных задач в

этом направлении является задача нахождения коротких целых решений (SIS - Short Integer Solution)**.** Эта задача была предложена Аджтаем (Miklós Ajtai)[75] в 1996 году как часть фундаментальной работы по построению однонаправленных функций, надёжность которых можно строго обосновать на основе средней сложности соответствующей вычислительной задачи. Впервые в истории криптографии удалось связать безопасность криптографической конструкции с наихудшим случаем сложной математической задачи, что открыло новые горизонты для создания безопасных схем. Задача SIS формулируется следующим образом: дано случайным образом сгенерированное множество векторов (например, в виде матрицы над кольцом вычетов по модулю большого простого числа), требуется найти не нулевой короткий целочисленный вектор, который является линейной комбинацией этих векторов и даёт нулевой вектор по модулю q по формуле (11). Несмотря на то, что задача кажется простой по формулировке, она является вычислительно сложной даже при наличии мощных вычислительных ресурсов, а существующие алгоритмы для её решения - экспоненциальные по времени. Благодаря своим уникальным свойствам, SIS используется в широком спектре криптографических схем: от построения хеш-функций и схем цифровой подписи до шифрования и протоколов обмена ключами. Особенно важно то, что безопасность этих схем может быть строго сведена к сложности задачи SIS в худшем случае, что делает их особенно привлекательными для применения в условиях угроз квантовых атак [76 с.6]. Несмотря на различные попытки разработать протоколы обмена ключами на основе проблемы решения коротких целых чисел, фундаментальная архитектура остается без изменений и представлен ниже.

1. Предположим, что Алиса и Боб соглашаются на обмен ключами. Система генерирует случайную матрицу R$\in Z_q^{nxm}$, где q - заранее определённое большое простое число, а n и m - параметры размерности. Данная матрица является общей для обеих сторон и передаётся по открытому каналу.

2. Алиса выбирает секретный ключ $s_A \in Z_q^m$ с нормой $\|s_A\| \leq \beta$. Она вычисляет публичное значение

$$P_A = Rs_A \qquad (24)$$

и отправляет P_A Бобу по открытому каналу.

3. Боб выбирает секретный ключ $s_B \in Z_q^n$ с нормой $\|s_B\| \leq \beta$. Он вычисляет

$$P_B = s_B^T R \qquad (25)$$

и отправляет P_B Алисе.

4. Теперь обе стороны могут независимо вычислить общий секретный ключ.

5. Получив P_B , Алиса вычисляет

$$K_A = s_A^T P_B^T = s_A^T R^T s_B \qquad (26)$$

6. Получив P_A, Боб вычисляет

$$K_B = P_A^T s_B = s_A^T R^T s_B \qquad (27)$$

7. $K_A = K_B = K$.

Для обеспечения надёжной скрытности секретного вектора Алисы s_A, необходимо, чтобы множество возможных решений уравнения $P_A = Rs_A$ было достаточно большим. Это возможно, если число переменных $n \leq m$. В этом случае задача восстановления s_A сводится к задаче поиска короткого вектора в решётке высокой размерности, что является вычислительно трудной задачей SIS. С другой стороны, чтобы обеспечить аналогичную защиту секретного вектора Боба s_B, необходимо, чтобы при известной матрице R и векторе $P_B = s_B^T R$ существовало большое множество коротких решений s_B, затрудняющее его восстановление. Для этого требуется $m \leq n$.

В результате обеим сторонам необходимо получить больше числовых переменных, чем уравнений, что делает непрактичным обмен ключами в задаче решения коротких целых чисел.

Выводы по второму разделу

Во втором разделе приведено описание и применение обмен криптографического ключа. В ней:

1. Рассмотрена проблема обмена ключами в симметричной криптографии, заключающаяся в необходимости безопасной передачи секретного ключа. Эта проблема становится особенно актуальной при использовании открытых каналов связи, так как существует риск перехвата ключа злоумышленниками.

2. Описана схема шифрования открытого ключа Эль-Гамаля и устойчивость к квантовым атакам. Схема шифрования Эль-Гамаля представляет собой мощный и безопасный метод, основанный на трудности вычисления дискретного логарифма, и является основой для многих современных криптографических приложений, но уязвим от квантовых атак и быстро решается алгоритмом Шора.

3. Рассмотрен протокол обмена ключами на основе задачи SIS, а также приведено подробное описание алгоритма, который демонстрирует непрактичность задачи SIS.

3 НОВАЯ КОНСТРУКЦИЯ ПОСТКВАНТОВОЙ КРИПТОСИСТЕМЫ ЭЛЬ-ГАМАЛЯ

3.1 Общее описание и реализация постквантовой криптосистемы Эль-Гамаля

На основе описанного выше обмена ключами мы предлагаем криптосистему с открытым ключом, которая работает как классическая криптосистема Эль- Гамаля[76, с.7].

1. Для генерации ключей нам необходимо реализовать следующие шаги:
   – входные данные: параметр безопасности 1^{nxm};
   – выходные данные: открытый ключ (q, n, m, M, $P_A = Ms_A$), а закрытый ключ – $s_A \in Z_q^m$ с нормой $\| s_A \| \leq \beta$;
   – шаги:
   1) запустить алгоритм генерации ключей $Gen(1^{nxm})$ для получения параметров q, n, m, где q – модуль, n – количество строк, а m – количество столбцов;
   2) сгенерировать случайную матрицу M размером n×m с элементами, выбранными равномерно из кольца Z_q;
   3) сгенерировать случайный секретный вектор s_A размером m с элементами, выбранными равномерно из кольца Z_q , такими, что норма $\|s_A\|$ меньше или равна пороговому значению β;
   4) вычислить открытый ключ (q, n, m, M, P_A), где $P_A = Ms_A$.
2. Чтобы зашифровать сообщение, нам нужно реализовать следующие шаги:
   – вход: Открытый ключ pk= (q, n, m, M, $P_A = Ms_A$) и сообщение $m \in Z_q^{nxm}$;
   – выход: Зашифрованный текст (c_1 ,c_2);
   – шаги:
   1) выбрать равномерный вектор $s_B \in Z_q^n$ n с элементами, выбранными равномерно из кольца Z_q, такими, что норма $\| s_B\|$ меньше или равна пороговому значению β;
   2) вычислить $c_1 = s_B^T R$, где R – случайно сгенерированная матрица;
   3) вычислить $c_2 = P_A^T s_B m$;
   4) вывести зашифрованный текст (c_1 ,c_2).
3. Чтобы расшифровать сообщение, нам необходимо выполнить следующие шаги
   – вход: Зашифрованный текст и закрытый ключ s_A ;
   – выход: Расшифрованное сообщение;
   – шаги:
   1) вычислить: $c_1 s_A^T$;
   2) вычислить поэлементное деление c_2 на $c_1 s_A^T$;
   3) вывести расшифрованное сообщение.

Алгоритм генерации ключей позволяет нам генерировать пару закрытого и открытого ключей, используя параметры, полученные из алгоритма генерации

ключей. Эти ключи обеспечивают безопасные процессы шифрования и дешифрования.

Алгоритм шифрования принимает открытый ключ и сообщение в качестве входных данных и создает зашифрованный текст. Он использует случайные векторы, чтобы скрыть сообщение и обеспечить конфиденциальность.

Алгоритм дешифрования с помощью закрытого ключа позволяет получателю восстановить исходное сообщение из зашифрованного текста. Он использует математические операции для обратного процесса шифрования, сохраняя при этом целостность сообщения.

Вместе эти шаги обеспечивают безопасный метод передачи конфиденциальной информации по незащищенным каналам, гарантируя, что только уполномоченные стороны могут получить доступ к исходному сообщению.

Процесс проиллюстрирован на рисунке 31.

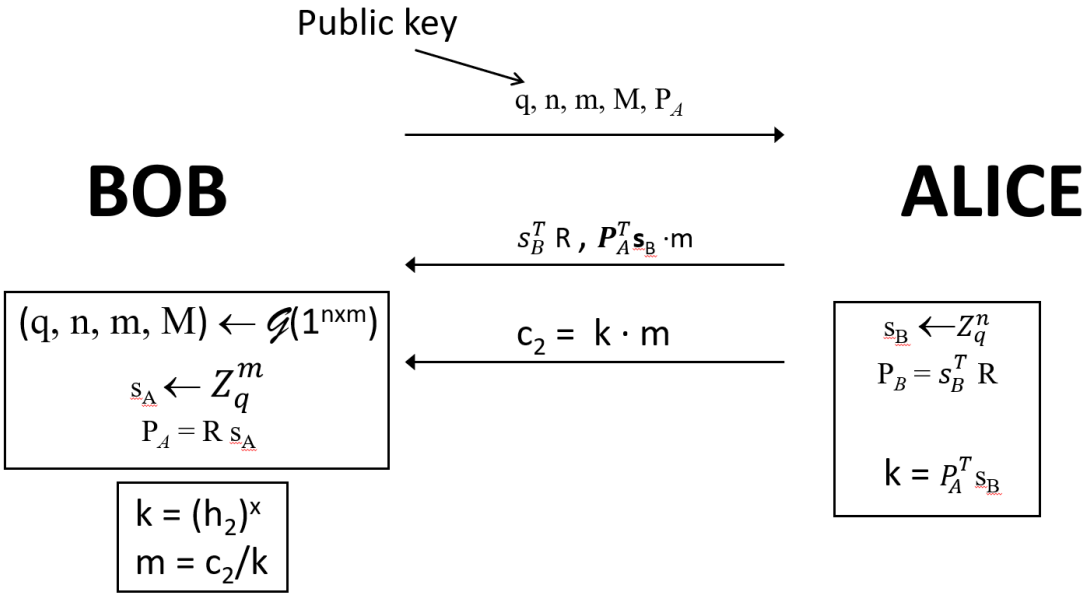


Рисунок 31 - Постквантовый Эль-Гамаль

Предлагаемые схемы состоят из следующих трех алгоритмов:

1. Gen(1^{nxm}):

Запуск G(1^n) для получения q, n и m. На основе этих параметров алгоритм генерирует случайную матрицу M∈ Z_q^{nxm} .

Открытый ключ - (q, n, m, M, PA = Ms_A), а закрытый ключ - $s_A \in Z_q^m$ с нормой $\| s_A \| \leq \beta$

2. Enc_{pk}(m), где pk = (q, n, m, M, $P_A = Ms_A$) и m∈$\mathbb{Z}_q^{nxm}$:

Выберите равномерное $s_B \in Z_q^n$ с нормой $\| s_B \| \leq \beta$. Шифртекст (c_1, c_2), где $c_1 = s_B^T$ R and $c_2 = P_A^T s_B m$

3. $Dec_{sk}(c_1, c_2)$, Выход $c_2/c_1 s_A^T$:

Ниже представлен псевдокод схемы:

function KeyGeneration():

q, n, m = RunG(1^n) // Запустить алгоритм генерации ключей G для получения параметров.

M = GenerateRandomMatrix(n, m, q) // Генерация случайной матрицы M.

s_A = GenerateRandomVector(m, q, β) // Генерация случайной матрицы s_A.

P_A = M * s_A // Вычислить открытый ключ.

return (q, n, m, M, PA), sA // Вернуть открытый ключ и закрытый ключ

function Encrypt(pk, m):

q, n, m, M, PA = pk // Распаковать открытый ключ.

sB = GenerateRandomVector(n, q, β) // Генерация случайного вектора s

c_1 = DotProduct(s_B, R) // Вычислить $c_1 = s_B^T$ R.

c_2 = DotProduct(P_A ,s_B) m // Вычислить $c_2 = P_A^T s_B m$.

k = Hash(P_A^T * s_B) // Вычислить ключ шифрования k с помощью хэш-функции.

return (c_1, c_2, k) // Вернуть зашифрованный текст и ключ шифрования.

function Decrypt(s_k , c_1, c_2, k):

s_A = s_k // Получить закрытый ключ.

m = c_2 / (c_1 * s_A^T) // Вычислить расшифрованное сообщение.

return m // Вернуть расшифрованное сообщение.

3.2 Анализ безопасности предлагаемой схемы

Безопасность предлагаемой криптосистемы Эль-Гамаль на основе решетки основана на стойкости проблемы решения коротких целых чисел (SIS) и безопасности атаки выбранного шифротекста (CCA) базовой структуры шифрованиях[77]. Схема разработана для противостояния как классическим, так и квантовым злоумышленникам, обеспечивая постквантовую безопасность.

Цель безопасности изложено ниже.

Схема достигает неразличимости при атаке выбранного шифротекста (IND-CCA) в модели квантового случайного оракула (QROM), предполагая, что проблема SIS сложна для заданных параметров.

Возможности состязательности:

Квантовый злоумышленник (AQQ): может запрашивать хеш-функции в суперпозиции (QROM), использовать квантовые ускорения (например, алгоритм Гровера) и выполнять квантовые вычисления за полиномиальное время.

Классический злоумышленник (AC): ограничен классическими вычислениями за полиномиальное время.

Теорема 1. (безопасность IND-CCA)

Если проблема SIS сложна для параметров (n,m,q,β), а хэш-функция H устойчива к коллизиям в QROM, предлагаемая схема является безопасной IND-CCA.

Доказательства представлено ниже.

1. Экземпляр SIS:

Пусть B будет претендентом, получающим экземпляр SIS A$\in Z_q^{nxm}$ и которому поручено найти ненулевой вектор z $\in Z_q^{nxm}$такой, что:

По формуле (11) :

$$A\ z \equiv 0\ \mathrm{mod} q\ "\ и\ "\ \|\ z\ \| \leq \beta \qquad (11)$$

2. Противник A:

- Нарушает безопасность IND-CCA с преимуществом ϵ.
- Выполняет q_H запросов к хэш-функции и q_D запросов на расшифровку.

3. Гибридная модель.

Противник A функционирует в модели квантово-доступного случайного оракула (QROM), где запросы к хэш-функции являются квантово-доступными.

Построение симмулятора B.

1. Генерация ключа:

- B устанавливает открытый ключ pk = (q, n, m, $P_A = M \cdot s_A$), где M получено из экземпляра задачи SIS.
- Закрытый ключ $s_k = s_A$ хранится в секрете.

2. Хэш-запросы:

- Для каждого запроса H(x_i), B возвращает равномерно случайное значение $h_i \in \{0,1\}^{\lambda}$.
- Поддерживает список $L_H = \{(x_i, h_i)\}$.

3. Запросы шифрования:

- Для открытого текста m, B вычисляет:
  $c_1 = s_B^T R, c_2 = P_A^T s_B \cdot m, k = H(P_A^T s_B)$ (28)
- Возвращает зашифрованный текст (c_1 ,c_2)

4. Запросы на дешифровку:

- Для зашифрованного текста (c_1 ,c_2) B проверяет, соответствуют ли с c_1 ,c_2 с L_H.
- Если верно, вычисляет $m = c_2/(c_1 \cdot s_A^T)$

Фальсификация и редукция.

1. Подделка противника.

A выводит зашифрованный текст (c_1^*, c_2^*), который расшифровывается как $m^* \neq m_i$ для любого запрошенного m_i.

2. Извлечение решения задачи SIS:

- Если (c_1^*, c_2^*) является допустимым, но не полученным из какого-либо запроса шифрования, B извлекает:
  $$M \cdot (s_A^* - s_A) \equiv 0\ \mathrm{mod} q \qquad (29)$$

где s_A^* - поддельный секретный вектор.

- По свойству связывания SIS, z = $s_A^* - s_A$ является ненулевым решением экземпляра SIS.

3. Обнаружение коллизий.

- Если $H(P_A^T s_B^*) = H(P_A^T s_B)$ для $s_B^* \neq s_B$, B выводит s_B^*, s_B как пару столкновений для H.

Анализ вероятности успеха представлен ниже.
Преимущество B в решении SIS заключается в следующем:

$$\epsilon' \geq \epsilon - \Pr[\text{ Hash Collision }] - \Pr[\text{ Random Oracle Guess }], \quad (30)$$

где:

- $\Pr[\text{ Hash Collision }] \leq \frac{q_H^2}{2^\lambda}$ (birthday bound) (Вероятность коллизии хэш-функции)
- $\Pr[\text{ Random Oracle Guess }] \leq \frac{q_D}{2^\lambda}$. (Вероятность угадывания случайного оракула)

Итоговая граница безопасности:

$$\epsilon' \geq \epsilon - \frac{(q_H+q_D)^2}{2^\lambda} - \text{negl}(n) \quad (31)$$

Заключение.

Если противник A достигает успеха с не пренебрежимо малым преимуществом ϵ, то выполняется одно из следующих условий:

1. B решает задачу SIS, что противоречит предполагаемой сложности её решения.
2. A находит коллизию хэш-функции, тем самым нарушая её устойчивость к коллизиям H.

Таким образом, предлагаемая схема является безопасной по стандарту IND-CCA в соответствии с предположениями SIS и устойчивости к коллизиям, обеспечивая надежную защиту как от классических, так и от квантовых злоумышленников.

3.3 Программное и аппаратное обеспечения для реализации криптосистемы

Программа была реализована на языке Python в среде colab.google, который применяется главным образом для машинного обучения, обработки данных и образовательных проектов [76, с.13]. Для проведения вычислений были применены библиотеки программного обеспечения с открытым исходным кодом import numpy,import matplotlib.pyplot,import.matplotlib.animation.

Numpy – это библиотека Python, которую применяют для математических вычислений: начиная с базовых функций и заканчивая линейной алгеброй. Полное название библиотеки - Numerical Python extensions, или «Числовые расширения Python». Эта библиотека стала популярным инструментом из-за нескольких важных качеств. Во-первых, NumPy называют открытым модулем Python, поскольку его исходный код доступен для всех на GitHub. Для начала поговорим об устройстве массивов, обрабатываемом NumPy. Принимая во внимание однородный двумерный массив, внешне он похож на обычную таблицу с двумя осями значений и ячейками, которые называются элементами

массива. Массив будет трехмерным в случае добавления третьей оси. Важное условие: каждый элемент должен иметь один тип данных, например, целые числа (рисунок 32).

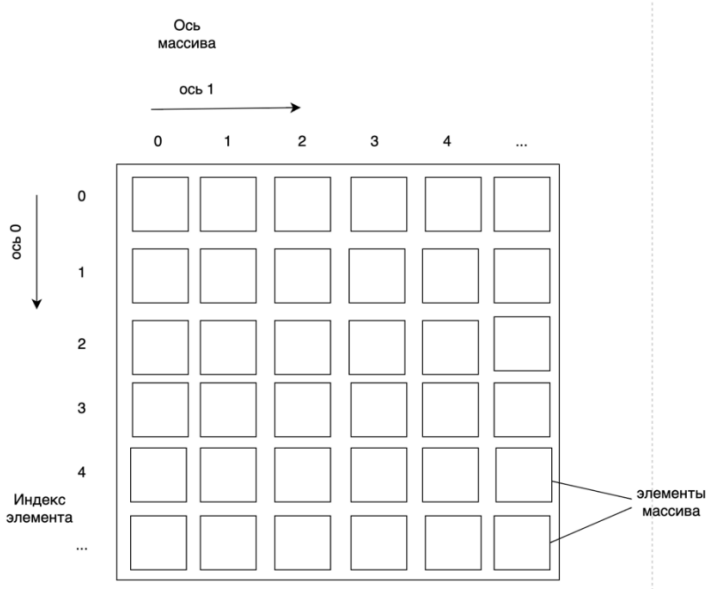


Рисунок 32 – Пример визуализации двумерного массива

Конечно, библиотека NumPy обрабатывает не только двумерные массивы, но и массивы с различным количеством осей. Как любая переменная в математической задаче, число N используется для обозначения этой вариативности. Таким образом, NumPy обычно используется для работы с N-мерными массивами данных. Для загрузки NumPy просто введите в командной строке: import numpy as np. Модуль NumPy входит в состав множества пакетов и других библиотек. А также, можно написать просто import numpy, ошибки в этом не будет. Но np - это общепринятое название, которое стало правилом и упростило процесс написания кода, поэтому, один раз прописав import numpy as np, в последующих строках можно использовать np вместо numpy. В нашем случае мы использовали библиотеку Numpy для работы с матрицами,чтобы декодировать и шифровать сообщения. Ниже предоставлен код с использованием библиотеки Numpy:

```
import numpy as np
# Вспомогательная функция для генерации случайных элементов из конечного поля Z_q
def generate_random_matrix(n, m, q):
return np.random.randint(0, q, (n, m))
# Алгоритм генерации ключей
def key_generation(security_param):
q, n, m, beta = security_param
M = generate_random_matrix(n, m, q)
sA = generate_random_matrix(m, 1, q)
while np.linalg.norm(sA) > beta:
sA = generate_random_matrix(m, 1, q)
```

Алгоритм шифрования (работает с матрицами)

Matplotlib – одна из самых известных библиотек для визуализации данных, доступных в Python. Благодаря этому можно создавать различные графики и диаграммы, которые улучшают понимание и интерпретацию данных. Поскольку она богата документацией и интуитивно понятна, эта библиотека особенно полезна. В анализе данных визуализация данных имеет решающее значение, поскольку она позволяет быстро определить тенденции [78,79], аномалии и взаимосвязи, которые могут быть неочевидны при просмотре сырых данных. Matplotlib является мощным инструментом для анализа данных, когда он используется в сочетании с другими библиотеками, такими как NumPy и Pandas. Мы использовали в нашей работе функцию pyplot и animation.

Pyplot – это модуль, который предоставляет интерфейс для работы с графиками, похожий на функциональность MATLAB. Он широко используется для создания статических, интерактивных и публикационно-готовых графиков. Основные возможности pyplot:

Построение графиков: Функция plot() позволяет создавать простые линейные графики. Например, график изменения данных во времени.

Поддержка разных типов диаграмм: Точечные диаграммы (scatter()). *столбчатые диаграммы* (bar()), *гистограммы* (hist()), *круговые диаграммы* (pie().

Оформление графиков: pyplot предоставляет функции для добавления заголовков, подписей осей, легенд и сеток: plt.title("Название графика") plt.xlabel("Ось X") plt.ylabel("Ось Y") plt.legend() plt.grid().

Таким образом, *pyplot* - это универсальный инструмент для создания информативных и эстетичных графиков, который часто применяется в научных исследованиях, машинном обучении и обработке данных.

Animation – это модуль Matplotlib, предназначенный для создания анимаций, которые позволяют динамически отображать изменение данных.

Модуль animation в Matplotlib предоставляет гибкие инструменты для создания динамических визуализаций. С его помощью можно демонстрировать изменения данных во времени, иллюстрировать сложные процессы и алгоритмы, а также создавать анимации, подходящие для образовательных и аналитических целей. Основной класс *FuncAnimation* позволяет легко обновлять графики в реальном времени, а результат можно экспортировать в форматы, такие как GIF или видео [80]. Этот модуль является мощным дополнением к статическим визуализациям, делая анализ данных более интерактивным и наглядным.Ниже предоставлен код с использованием библиотеки Matplotlib с pyplot и animation:

```
# Функция визуализации для отображения матриц в виде тепловых карт.
def plot_heatmap(matrix, title="Matrix", ax=None):
if ax is None:
fig, ax = plt.subplots(figsize=(6, 6))  # Adjust the size to fit matrix
sns.heatmap(matrix, annot=True, fmt="d", cmap="YlGnBu", cbar=True, ax=ax,
square=True, xticklabels=False, yticklabels=False)
ax.set_title(title)
```

```
ax.set_xlabel('Columns')
ax.set_ylabel('Rows')
plt.tight_layout()
```

Эти библиотеки помогли не только построить графики для визуализации данных, но и реализовать процесс шифрования и дешифрования сообщений с открытым ключом на основе решеток, используя принципы Эль-Гамаля. Благодаря использованию таких инструментов, как NumPy, Matplotlib и другие, удалось обеспечить наглядность работы криптосистемы и продемонстрировать её устойчивость к потенциальным квантовым угрозам.

3.4 Анализ и тестирование эффективности предложенной постквантовой криптосистемы

В настоящее время существует очень мало криптосистем, которые обеспечивают защиту от квантовых атак. Мы использовали шифрование на основе решеток с применением принципов Эль-Гамаля [76, с.14-15]. Для демонстрации эффективности и надежности предложенной криптосистемы был проведен экспериментальный анализ. Реализация схемы на основе SIS была выполнена с использованием Python 3.8 и нескольких библиотек, таких как NumPy, matplotlib.pyplot и matplotlib.animation, которые использовались для криптографических функций. Для тестирования проводились эксперименты на настольном компьютере с Windows 10 Pro.

Криптографические операции обрабатывались скриптами Python, а для сбора информации о производительности использовались инструменты профилирования памяти и модуль времени Python.

Конфигурация машины, используемой для экспериментов:

1. Процессор: Intel Core i7-8565U (4 ядра, 8 потоков, с базовой частотой 1,8 ГГц).
2. Память: 16 ГБ DDR4 RAM (работает на частоте 3200 МГц).
3. Накопитель: 1 ТБ NVMe SSD.
4. Операционная система: Windows 10 Pro (64-разрядная версия).

Мы измерили ключевые аспекты производительности схемы, включая время генерации ключа, время выполнения операций, размеры ключей и потребление памяти.

Реализованный подход использует стратегию хеширования, при которой данные сначала хешируются в дайджест фиксированного размера. Эксперименты повторялись несколько раз для обеспечения точности и минимизации влияния факторов, таких как загрузка системы и фоновые процессы.

Время генерации ключа:

– 100×100: 0,0000664 секунды;
– 200×200: 0,0001529 секунды;
– 300×300: 0,0002469 секунды.

Время выполнения операций для различных размеров данных:

– 1 КБ: 0,20 миллисекунд;

– 10 КБ: 1,15 миллисекунд;
– 100 КБ: 9,30 миллисекунд.

Размеры ключей:
– открытый ключ: 4 КБ;
– закрытый ключ: 2 КБ;
– размер выходных данных (все размеры сообщений): 1,7 КБ.

Потребление памяти:
– генерация ключа: 50 МБ;
– выполнение операций: 12 МБ.

Использование стратегии хеширования гарантирует, что выходные данные остаются неизменными для разных размеров входных данных, что снижает влияние на эффективность хранения и передачи. Однако результаты эксперимента подчеркивают компромисс между безопасностью и эффективностью. Увеличение размеров матрицы и ключей повышает безопасность, но приводит к более высоким вычислительным затратам и увеличению использования памяти. Хотя более крупные ключи повышают устойчивость к атакам, они также увеличивают потребление памяти и время обработки. Этот баланс между более высокой безопасностью и большими требованиями к ресурсам подчеркивает необходимость тщательного выбора параметров для оптимизации производительности в зависимости от конкретных требований приложения.

Направления будущих исследований:

1. Безопасные коммуникации IoT: тестирование схемы на устройствах с ограниченными ресурсами (например, Raspberry Pi, Arduino) для оценки производительности в средах IoT.

2. Интеграция с блокчейном: применение схемы для аутентификации транзакций в блокчейн-сетях и сравнение с существующими постквантовыми решениями.

3. Оптимизация в протоколе TLS: исследование возможности интеграции с TLS для оценки задержки соединения, уровня безопасности и производительности по сравнению с традиционными криптографическими схемами, такими как RSA и ECDSA.

Важно отметить, что производительность схемы на основе SIS не зависит от размеров входных данных, поскольку сначала выполняется хеширование, после чего дальнейшие криптографические операции работают с дайджестом фиксированного размера. Такой подход обеспечивает постоянное время выполнения операций, независимо от исходного объема данных, что делает систему более предсказуемой и эффективной. Тем не менее, мы признаем необходимость дальнейшей оценки производительности при различных нагрузках, что станет предметом будущих исследований [76, с. 10-12].

Также проведено сравнение предложенной схемы на основе SIS с алгоритмами LWE, Ring-LWE и схемой Эль-Гамаля. Результаты эксперимента показали, что время генерации ключей существенно возрастает с увеличением их размера: для ключей размером 100×100 наблюдается увеличение в 240 раз,

для 200×200 – в 418 раз, а для 300×300 – в 583 раза. Это свидетельствует о том, что увеличение размеров ключей приводит к экспоненциальному росту времени их генерации по сравнению с алгоритмами LWE и Ring-LWE. Схема Эль-Гамаля не была включена в детальное сравнение, так как, несмотря на высокую скорость работы, она не обеспечивает защиту от квантовых атак. Напротив, предлагаемая схема на основе SIS демонстрирует устойчивость к квантовым атакам, таким как алгоритм Шора, алгоритм Гровера и другие. Кроме того, проведён анализ устойчивости схемы к ошибкам, что подтверждает её надёжность и простоту по сравнению с аналогами LWE и Ring-LWE, а также высокую стойкость к квантовым атакам по сравнению со схемой Эль-Гамаля." (таблице 1).

Таблица 1 – Сравнение схемы предложенной схемы с LWE, Ring-LWE, Эль-Гамаля

Критерий	Предложенная схема на основе SIS	LWE (Learning With Errors)	Ring-LWE	Эль-Гамаль
Математическая сложность	O(nlogn)	$O(n^2 logq)$	$O(n^2 logq)$	O(logp)
Безопасность (постквантовая)	128-256 бит устойчива к квантовым атакам	256+бит, но требуется больше параметров для защиты	128-256 бит, но более уязвима в случае ошибок в полиномиальной арифметике	Не защищен от квантовых атак
Размер ключа	2048-4096 бит	8196+бит, большие осо бенно при высо ком уровне безопасности	1024-2048 бит, меньшие ключи благодаря структуре кольца	2048-3072 бит, так как включают простые числа
Эффективность	$O(n^2)$, требует больших вычислительных ресурсов	$O(n^3)$ менее эффективна из-за сложных операций матричной алгебры	O(nlogn) высокая, использует полиномиальную арифметику для оптимизации	O(logp) высокая, относительно быстрая генерация ключей и шифрование
Простота реализации	O(n) легче для реализации и анализа	$O(n^2)$ сложнее в реализации из-за структуры ошибок и распределений	O(nlogn) более сложная в реализации из-за дополнительных вычислительных шагов	O(n) простая в реализации, хорошо документирована
Устойчивость к ошибкам	Высокая, так как не использует сложные вероятностные модели	Зависит от правильности настройки гауссовых распределений	Зависит от точности полиномиальной арифметики и ошибок	Уязвима к ошибкам в расчетах с большими простыми числами

Время генерации ключей (сек) для 100x100	0,0000664 сек	0,0160000 сек	0,0160000 сек	0,0000016 сек
Время генерации ключей (сек) для 200x200	0,0001529 сек	0,0640000 сек	0,0640000 сек	0,0000016 сек
Время генерации ключей (сек) для 300x300	0,0002469 сек	0,1440000 сек	0,1440000 сек	0,0000016 сек

Предложенная схема на основе задачи о кратчайшем целочисленном решении (Short Integer Solution, SIS) представляет собой криптографический подход, основанный на сложности задач в решетках. Ниже приведено подробное описание каждого критерия и обоснование значений, указанных в таблице 2. Сложность SIS связана с нахождением короткого вектора в решетке, заданной матрицей A. Алгоритмы для решения SIS (например, LLL-алгоритм) имеют сложность (O(nlog n)), где (n) - размерность решетки[81]. Это значение взято из теоретических исследований и анализа алгоритмов работы с решетками. SIS считается устойчивой к квантовым атакам, так как задача нахождения короткого вектора в решетке остается сложной даже для квантовых компьютеров. Уровень безопасности зависит от выбора параметров (например, размерности решетки n и модуля q. Значения 128 - 256 бит основаны на современных рекомендациях (например, NIST Post-Quantum Cryptography Standardization)[82]. Размер ключа определяется размерностью решетки n и модулем q. Для обеспечения безопасности 128 - 256 бит требуется размер ключа 2048 - 4096 бит. Это значение взято из анализа параметров, предложенных в исследованиях по криптографии на решетках. Операции в SIS включают умножение матриц и векторов, что имеет сложность $O(n^2)$. Это значение основано на теоретическом анализе вычислительной сложности операций с решетками. Предложенная схема на основе SIS легче реализовать по сравнению с другими схемами на решетках (например, LWE), так как она не требует сложных вероятностных моделей. Это значение основано на опыте реализации криптографических схем на решетках. Предложенная схема на основе SIS не использует сложные вероятностные модели, что делает её более устойчивой к ошибкам по сравнению с LWE и Ring-LWE. Это значение основано на теоретическом анализе и экспериментальных результатах. Время генерации ключей измерено экспериментально на настольном компьютере с использованием Python 3.8 и библиотеки NumPy. Для точного измерения времени выполнения операций была использована функция time из стандартной библиотеки Python. Время генерации ключей увеличивается с ростом размерности матриц, что подтверждает теоретическую сложность операций в криптографии на решетках.

Для демонстрации работоспособности и эффективности схемы был разработан прототип программы, реализующий основные этапы

криптосистемы.Преимущество такого подхода заключается в его устойчивости к алгоритмам квантового вычисления, включая известный алгоритм Шора, который способен взламывать традиционные асимметричные криптосистемы. Использование решеточных структур и методов шифрования, таких как Эль-Гамаль, обеспечивает высокий уровень безопасности даже при наличии квантовых угроз[83]. Более того, наша криптосистема сочетает в себе эффективность вычислений и минимальные затраты ресурсов, что делает её практически применимой для современных задач информационной безопасности:

1. В первую очередь выбираем необходимые данные и запускаем нашу криптосистему. Для начала мы установили значения q=17, n=5, m=5, а также ключ b=5. После этого, нажимая кнопку 'Запуск', активируем криптосистему (рисунок 33).

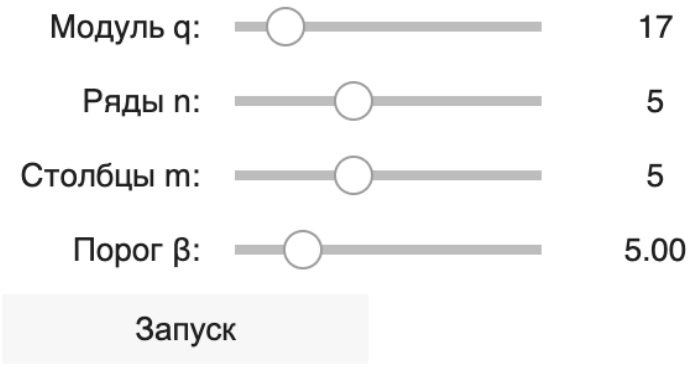


Рисунок 33 – Запуск криптосистемы

2. После запуска криптосистемы будет выводиться сообщение: 'Запуск постквантовой криптосистемы Эль-Гамаля...' с данными о исходном сообщении и зашифрованном сообщении, созданном с использованием ключей c1 и c2 (рисунки 34, 35).

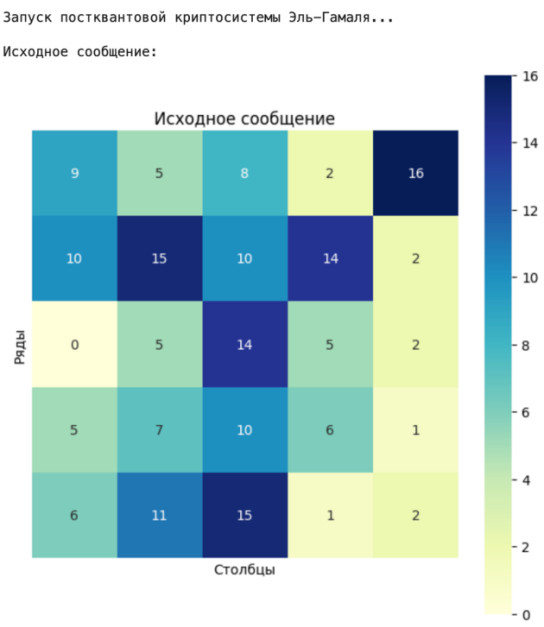
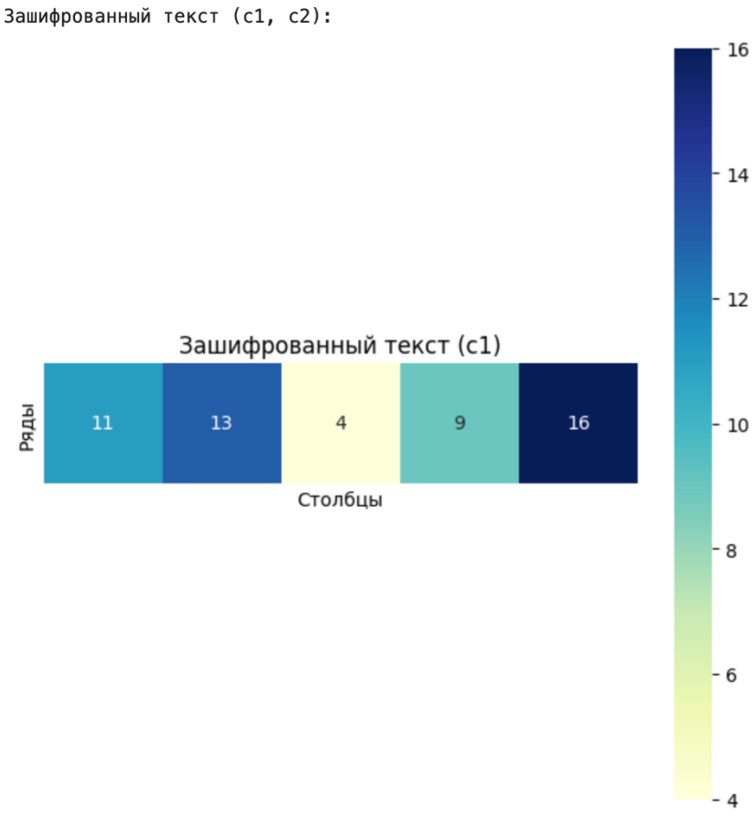
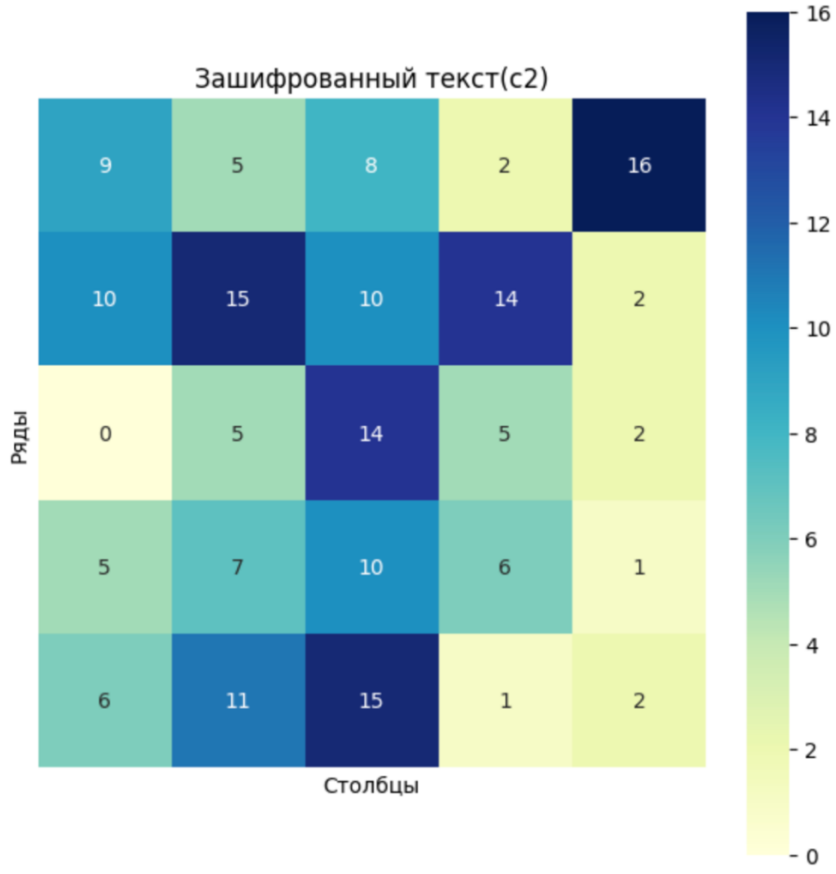


Рисунок 34 – Исходные сообщение



а



б

Рисунок 35 – Зашифрованные данные с ключами (c1,c2)

3. Далее выводятся расшифрованные данные (рисунок 36).

```
Расшифрованное сообщение:
```

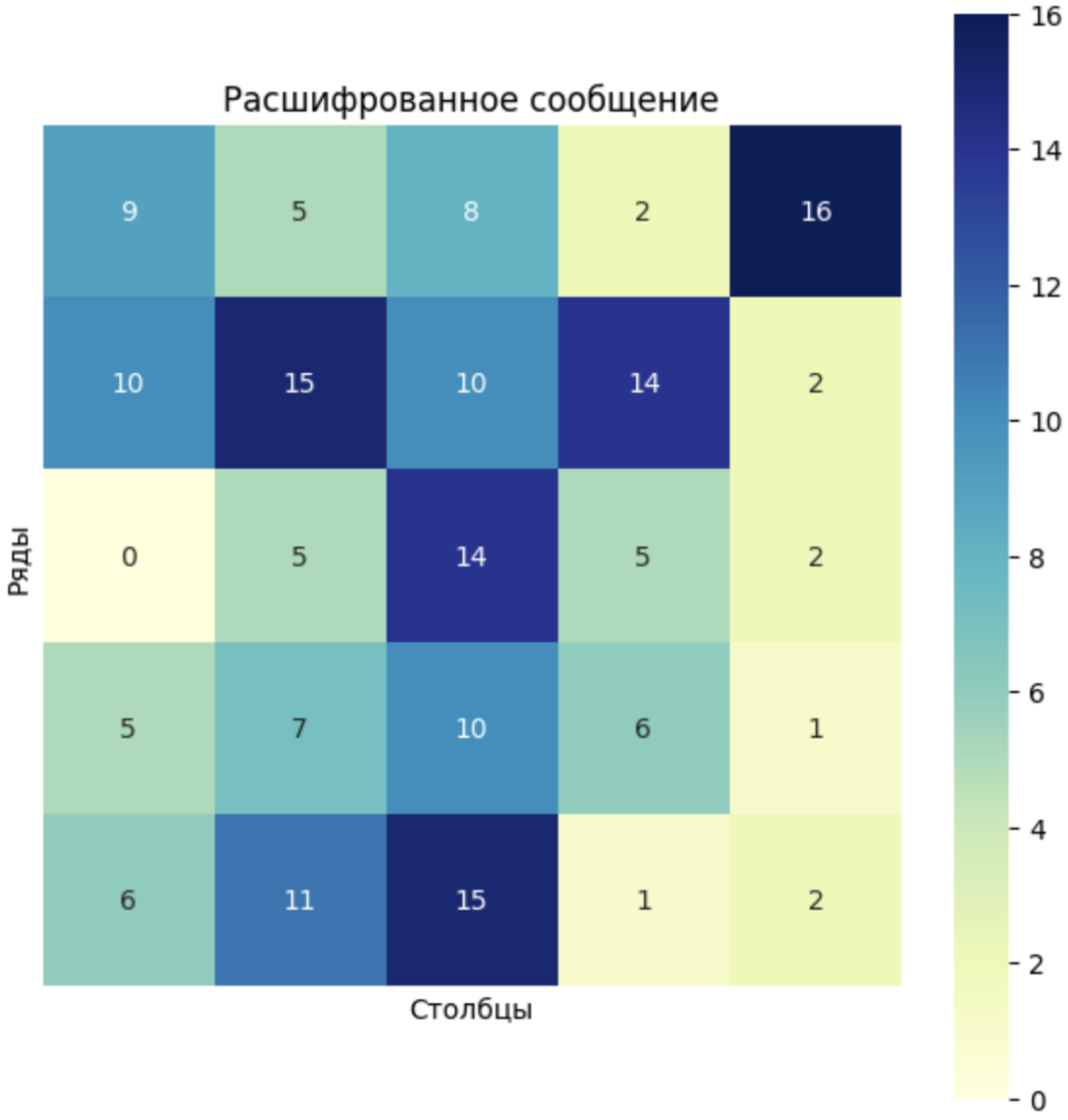
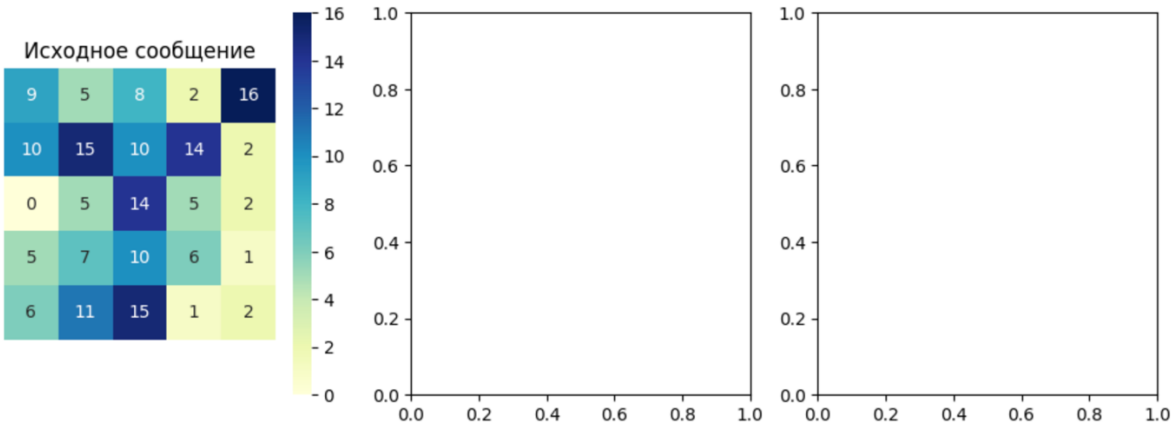
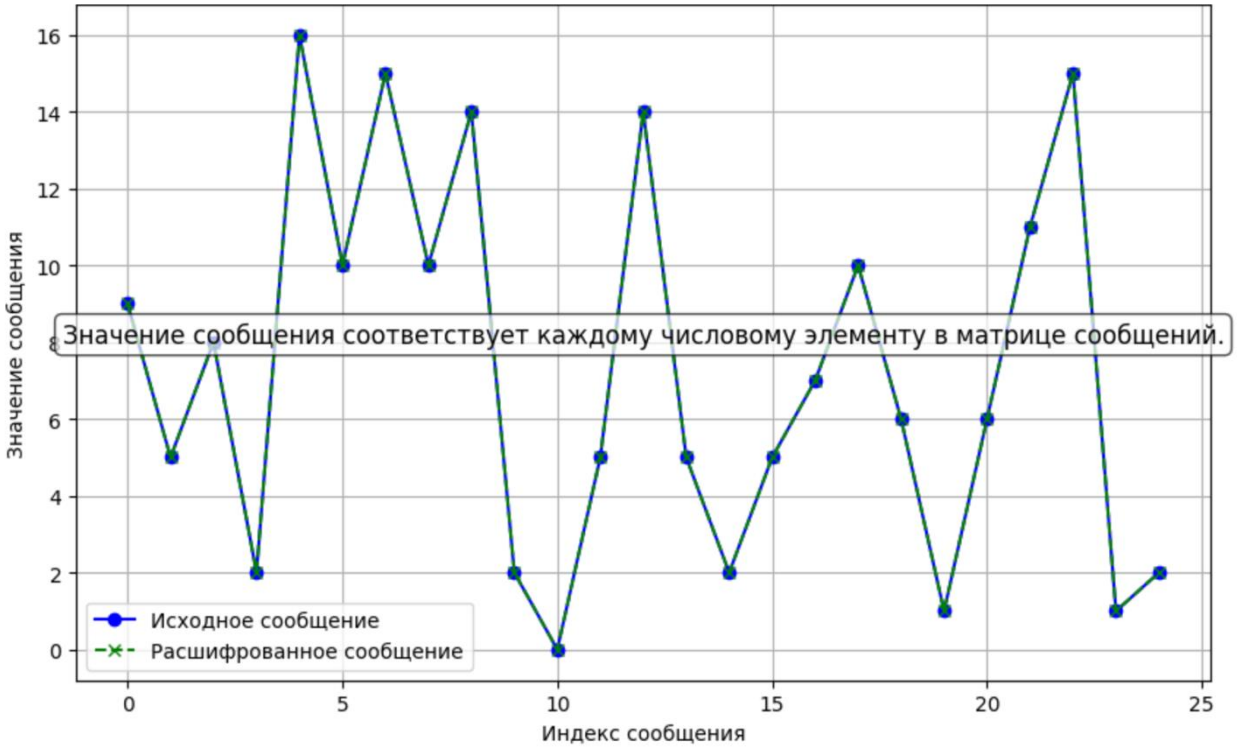


Рисунок 36 – Расшифрованные данные

4. После расшифрования сообщения можно увидеть, что оно точно совпадает с исходным. Это подтверждается графиком, представленным ниже (рисунок 37).



а



б

Рисунок 37 – Исходное и Расшифрованное сообщение

Также можно заметить, что исходное сообщение и зашифрованное сообщение могут иметь разную длину, но в результате расшифрования восстанавливаются исходные данные (рисунок 38).

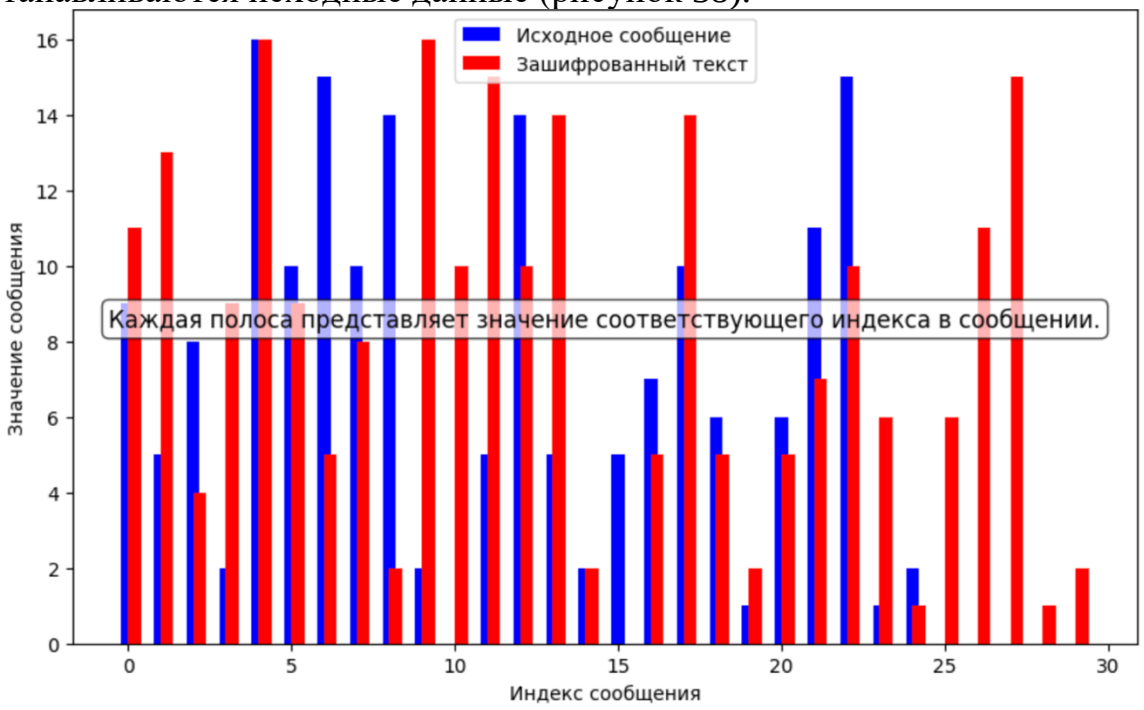


Рисунок 38 – Исходное сообщение и Зашифрованный текст

В результате мы демонстрируем, что решетки, использующие ключ Эль-Гамаля, надежно защищены от квантовых атак. Зашифрованные тексты точно совпадают с исходными сообщениями без изменений. Это можно увидеть на графике, представленном ниже (рисунок 39). Данная устойчивость связана с использованием алгоритмических основ, сложных для решения на квантовых компьютерах. Таким образом, предложенная схема обеспечивает высокий уровень безопасности, что делает её перспективной для постквантовой криптографии.

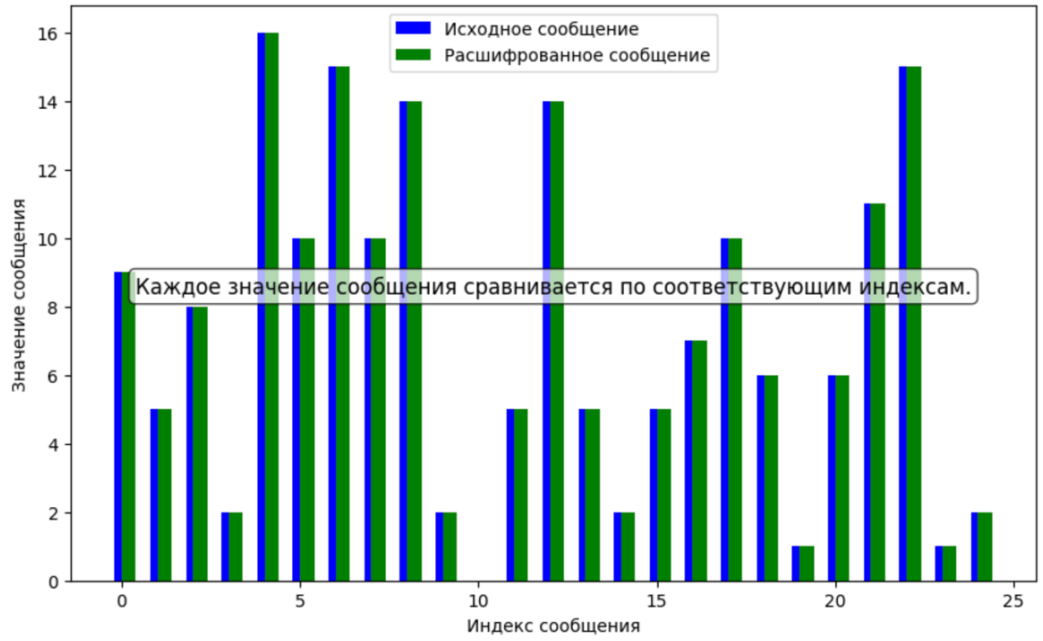


Рисунок 39 – Исходное и Расшифрованное сообщение

Традиционные криптографические системы, основанные на сложных вычислительных задачах, таких как факторизация или вычисление дискретного логарифма, подвергаются серьезной опасности в результате квантовых атак. Тем не менее, схемы постквантового шифрования на основе решеток, которые используют принципы Эль-Гамаля, показывают, что они очень устойчивы к этим атакам. Квантовые алгоритмы, такие как алгоритм Шора, очень сложно анализировать решеточные структуры, что делает их одним из наиболее перспективных направлений в области постквантовой криптографии[84,85]. Подход Эль-Гамаля в сочетании с решетками повышает надежность и гибкость, позволяя эффективно шифровать данные и защищать их даже от самых сложных квантовых угроз. Таким образом, для обеспечения безопасности данных в долгосрочной перспективе в условиях развития квантовых технологий предпочтительнее использовать постквантовую схему шифрования решеток на основе принципов Эль-Гамаля.

Выводы по третьему разделу

Последний раздел диссертационной работы посвящен анализу эффективности криптосистемы, а также разработке прототипа и математической модели, проверяющей данные на защиту от квантовых атак.

Рассмотрена и приведена общее описание и реализация постквантовой криптосистемы Эль-Гамаля. На основе описанного выше обмена ключами мы предлагаем криптосистему с открытым ключом, которая функционирует аналогично классической криптосистеме Эль-Гамаля, адаптированной к постквантовым условиям. В нашей реализации учтены новейшие криптографические подходы, такие как использование решеток и алгоритмов устойчивых к квантовым угрозам, что обеспечивает более высокий уровень безопасности по сравнению с традиционными методами. Основные этапы включают генерацию ключей, шифрование сообщения и его расшифровку, адаптированные к квантовой устойчивости.

Описан процесс анализа безопасности предлагаемой схемы, который связан с защищенной схемой шифрования с закрытым ключом CCA.

Описано программное и аппаратное обеспечение для реализации криптосистемы, которое было разработано на языке Python в среде Colab Google, используемой главным образом для машинного обучения, обработки данных и образовательных проектов. Для проведения вычислений были применены библиотеки с открытым исходным кодом, такие как numpy, matplotlib.pyplot и matplotlib.animation.

Разработан и приведен анализ эффективности постквантовой криптосистемы прописан каждый шаг алгоритма и результата что решетки с использованием приниципов Эль-Гамаля эффективно и надежно (Приложение А, Б).

ЗАКЛЮЧЕНИЕ

В ходе выполнения диссертационной работы по исследованию и разработке математической модели постквантового метода распределения ключей и разработки прототипа, который проверяет данные на защиту от квантовых атак, были получены следующие научные и практические результаты представленные ниже.

1. Осуществлен анализ популярных методов и алгоритмов традиционной криптографии, который показал, что они не обладают устойчивостью к квантовым атакам, а квантовые алгоритмы способны эффективно взламывать такие системы.
2. Выполнен анализ схем распределения ключей на основе решеток, который показал их высокую устойчивость к квантовым атакам благодаря сложности задач, лежащих в их основе. Эти схемы считаются перспективным направлением в постквантовой криптографии и могут обеспечить надежную защиту информации в условиях квантовых вычислений.
3. Разработана математическая модель эффективной и безопасной постквантовой схемы обмена ключами на основе решеток с использованием принципов Эль-Гамаля, что позволяет создать эффективные постквантовые схемы с открытым ключом для криптографических протоколов, систем аутентификации, финансовых систем, блокчейн и IoT-технологий.
4. Разработан алгоритм и прототип постквантовой схемы с открытым ключом на основе решеток, использующей принципов Эль-Гамаля на основе SIS, что позволила повысить скорость генерации ключей шифрования в 240-583 раз (по сравнению с аналогами – LWE, Ring-LWE), а также устойчивость к ошибкам и стойкость к квантовым атакам (по сравнению с классической схемой Эль-Гамаля).
5. Исследование и тестирование эффективности предложенной постквантовой криптосистемы Эль-Гамаля на основе SIS показали ее устойчивость к квантовым атакам. Кроме того, данная система продемонстрировала высокую скорость работы, что делает ее перспективной для практического применения.

Поставленные задачи диссертационной работы полностью решены, и полученные результаты могут быть использованы для дальнейшего улучшения схемы и разработки системы на основе постквантовой криптографии, соответствующей требованиям NIST. В будущем целесообразно сосредоточиться на повышении эффективности и адаптации предлагаемого подхода к новым угрозам и вызовам.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Sharma N. et al. A Review of Information Security using Cryptography Technique // International Journal of Advanced Research in Computer Science. – 2017. – Vol. 8. – P. 323-326.
2. Lincke J., Hollan A. Network Security: Focus on Security, Skills, and Stability // Procced. 37th ASEE/IEEE Frontiers in Education conf. – Milwaukee, 2007. – P. 1063-1075.
3. Khalifa O.O., Islam M.R., Khan S. et al. Communications cryptography // Procced. RF and Microwave conf. – Selangor, 2004. – P. 220-223.
4. Jirwan N., Singh A., Vijay S. Review and Analysis of Cryptography Techniques // International Journal of Scientific & Engineering Research. – 2013. – Vol. 3, Issue 4. – P. 1-6.
5. Tayal S., Gupta N., Gupta P. et al. A Review paper on Network Security and Cryptography // Advances in Computational Sciences and Technology. – 2017. – Vol. 10, Issue 5. – P. 763-770.
6. Gupta A., Walia N.K. Cryptography Algorithms: A Review // International journal of engineering development and research. – 2014. – Vol. 2, Issue 2. – P. 1667-1672.
7. Callas J. The Future of Cryptography // Information Systems Security. – 2007. – Vol. 16, Issue 1. – P. 15-22.
8. Massey J.L. Cryptography – A selective survey // Digital Communications. – 1986. – Vol. 85. – P. 3-25.
9. Schneier B. The Non-Security of Secrecy // Communications of the ACM. – 2004. – Vol. 47, Issue 10. – P. 120.
10. Gennaro R. IEEE Security & Privacy // IEEE Security & Privacy. – 2006. – Vol. 4, Issue 2. – P. 64-67.
11. Sadkhan S.B. Cryptography: current status and future trends // Procced. internat. conf. on Information and Communication Technologies: From Theory to Applications. – Damascus, 2004. – P. 417-418.
12. Dooley J.F. A Brief History of Cryptology and Cryptographic Algorithms. – NY.: Springer, 2013. – 99 p.
13. Preneel B. Understanding Cryptography: A Textbook for Students and Practitioners. – London: Springer, 2010. – 372 p.
14. Бабаш А.В., Шанкин П. Криптография. – М., 2007. – 512 с.
15. Blaum M., McEliece R.J. Coding protection for magnetic tapes: A generalization of the Patel - Hong code // IEEE Transactions on Information Theory. – 1985. – Vol. 31, Issue 5. – P. 690-693.
16. Lee H., Lee K., Shin Y. AES Implementation and Performance Evaluation on 8-bit Microcontrollers // International Journal of Computer Science and Information Security. – 2009. – Vol. 6, Issue 1. – P. 70-74.
17. Feldhofer M., Wolkerstorfer J., Rijmen V. AES implementation on a grain of sand // IEE Proc. Inf. Security. – 2005. – Vol. 2005. – P. 13-20.

18. Hellman W.D.A.M.E. New directions in cryptography // IEEE Transactions on Information Theory. – 1976. – Vol. IT-22, Issue 6. – P. 644-654.
19. Грибунин В.Г., Мартынов А.П., Николаев Д.Б. и др. Криптография и безопасность цифровых систем: учеб. пос. – Саров, 2006. – 256 с.
20. Goldwasser S., Bellare M. Lecture Notes on Cryptography. – Cambridge, 2008. – 289 p.
21. Stinson D.R. Cryptography: theory and practice. – Boca Raton, 1995. – 434 p.
22. Трунова А.А. Исследование криптосистем с открытым ключом на основе анализа алгоритма RSA // Молодой учен. – 2015. – №13(93). – С. 39-44.
23. Menezes A.J., van Oorschot P.C., Vanstone S.A. Handbook of Applied Cryptography. – London, 1997. – 810 p.
24. Mao W. Modern Cryptography: Theory and Practice. – New Jersey, 2003. – 707 p.
25. Katz J., Lindell Y. lntroduction to Modern Cryptography. – London, 2008. – 598 p.
26. Масленников М. Практическая криптография: монография. – М.: БХВ-Петербург, 2003. – 466 с.
27. Криптографический протокол // https://gb.ru/blog/kriptografiya. 10.10.2024.
28. Смарт Н. Криптография / пер. с англ. – М.: Техносфера, 2003. – 528 с.
29. Gauravram P. Cryptographic Hash Functions: Cryptanalysis, design and applications. – Brisbane, 2003. – 298 p.
30. Бутакова Н.Г., Семененко В.А., Федоров Н.В. Криптографическая защита информации: учеб. пос. – М.: МГИУ, 2011. – 316 с.
31. Holton W. Quantum computer // https://www.britannica.com. 10.10.2024.
32. Jian L., Yang Y.G., Chen X.B. et al. Practical quantum private database queries based on passive round-robin differential phase-shift quantum key distribution // Sci. Rep. – 2016. – Vol. 6. – P. 31738.
33. Bouwmeester D., Ekert A., Zeilinger A. The Physics of Quantum Information: Quantum Cryptography, Quantum Teleportation, Quantum Computation. – Berlin, 2000. – 315 p.
34. Sönmez O., Paar I.C. Symmetric key management, key derivation and key wrap. Ruhr-universität Bochum, Germany // https://www.emsec.rub.de. 10.10.2024.
35. Bennett C.H., Brassard G. Quantum cryptography: Public key distribution and coin tossing // Theor. Comput. Sci. – 2014. – Vol. 560. – P. 7-11.
36. Inoue K. Quantum key distribution technologies // IEEE Journal of Selected Topics in Quantum Electronics. – 2006. – Vol. 12. – P. 888-896.
37. Priyadharshini S.P., Kalaivani J. A Study on Quantum Cryptography // International Journal of Pure and Applied Mathematics. – 2018. – Vol. 119, Issue 15. – P. 3185-3191
38. Lenstra A.K., Lenstra H.W. et al. The Development of the Number Field Sieve. – Berlin, 1993. – 140 p.
39. Buchmann J., Dahmen E., Klintsevich E. et al. Merkle signatures with virtually unlimited signature capacity // Applied Cryptography and Network Security: procced. 5th internat. conf. (ACNS 2007). – Berlin, 2007. – P. 31-45.

40. McElice R.J. A public-key cryptosystem based on algebraic coding theory // DSN Progress Report. – 1978. – Vol. 42-44. – P. 114-116.
41. Berson T. Failure of the McEliece public-key cryptosystem under message-resend and related-message attack // Proceed. Advances in Cryptology - CRYPTO '97: 17th Annual internat. Cryptology conf. – Berlin, 1997. – P. 213-220.
42. Yang B.Y., Cheng D.C.M., Chen B.R. et al. Implementing minimized multivariate public-key cryptosystems on low-resource embedded systems // Lecture Notes in Computer Science. – 2006. – Vol. 3934. – P. 73-88.
43. Stinson D.R. Cryptography: theory and practice. – Boca Raton, 1995. – 434 p.
44. Barg A. Complexity issues in coding theory // Handbook of Coding theory. – 1998. – Vol. 1, ch. 7. – P. 649-754.
45. Hinek M.J. Lattice Attacks in Cryptography: A Partial Overview. – Ontario: University of Waterloo, 2004. – 84 p.
46. Wang S., Zhu Y., Ma D. et al. Lattice-based key exchange on small integer solution problem // Sci China Inf Sci. – 2014. – Vol. 57, Issue 11. – P. 1-12.
47. Iavich M. et al. Lattice based merkle .IVUS,2019 – p.112-115.
48. Ajtai M. Generating hard instances of lattice problems // Proceed. of the 28th Annual ACM sympos. on Theory of Computing. – NY., 1996. – P. 99-108.
49. Becker A. et al. Solving shortest and closest vector problems: The decomposition approach // https://eprint.iacr.org/2013/685. 10.10.2024.
50. Micciancio D. et al. Closest vector problem // In book: Complexity of Lattice Problems: A Cryptographic Perspective. – Boston, 2002. – P. 45-68.
51. Regev O. New Lattice-Based Cryptographic Constructions // Journal of the ACM. – 2004. – Vol. 51, Issue 6. – P. 899-942.
52. Bandara H. et al. On advances of lattice-based cryptographic schemes and their implementations // Cryptography. – 2022. – Vol. 6, Issue 4. – P. 56-1-56-22.
53. Lyu S. et al. Better lattice quantizers constructed from complex integers // IEEE Transactions on Communications. – 2022. – Vol. 70, Issue 12. – P. 7932-7940.
54. Amirkhanova D.S., Mamyrbayev O. Cryptographic analysis of the scheme of polylinear cryptography. // Bulletin of ABAI KAZNPU. Series: Physical and mathematical sciences volume 83 № 3(2023).
55. Amirkhanova D.S., Mamyrbayev O. Research and development of a cryptography algorithm based on polylinear algebra using blockchain methodology. // Bulletin EKTU: Vol: Information and communication technologies ISSN 1561-4212 № 1(2025)
56. Gentry C., Szydlo M. Cryptanalysis of the Revised NTRU Signature Scheme // Advances in Cryptology – EUROCRYPT 2002: procced. internat. conf. on the Theory and Applications of Cryptographic Techniques. – Amsterdam, 2002. – P. 299-320.
57. Nguyen P.Q., Regev O. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures // Procced. the 25th internat. Cryptology conf. (EUROCRYPT). – SPb., 2006. – P. 271-288.

58. Lyubashevsky V., Micciancio D., Peikert C. et al. SWIFFT: a modest proposal for FFT hashing // Fast Software Encryption: procced. 15th internat. Workshop (FSE). – Berlin, 2008. – P. 54-72.
59. NIST pqc Round 3 submissions, 2020 // https://csrc.nist.gov. 10.10.2024.
60. Kaur R., Kaur A.K. Digital signature // Proceed. internat. conf. on Computing Sciences (ICCS). – Phagwara, 2012. – P. 295-301.
61. Hounkpevi A.C. et al. Eaglesign: A new post-quantum Elamal-like signature over lattices // Submission to the NIST's post-quantum cryptography standardization process. – 2023. – Vol. 2023. – P. 1-40.
62. CRYSTALS: Cryptographic Suite for Algebraic Lattices, 2020 // https://pq-crystals.org/dilithium/. 10.10.2024.
63. NIST PQC Round 3 submissions, 2020 // https://csrc.nist.gov. 02.09.2022.
64. Wright M.A. The Impact of Quantum Computing on Cryptography // Network Security. – 2000. – Vol. 2000, Issue 9. – P. 13-15.
65. Chalkias K., Baldimtsi F., Hristu-Varsakelis D. et al. Two Types of Key-Compromise Impersonation Attacks against One-Pass Key Establishment Protocols // E-business and Telecommunications: procced. 4th internat. conf. (ICETE 2007). – Barcelona, 2007. – P. 227-238.
66. Diffie W., Hellman M. New Directions in Cryptography // IEEE Transactions. – 1976. – Vol. 22, Issue 6. – P. 644-654.
67. S.M.Bellovin and M.Merritt, Encrypted key exchange: password- based protocols secure against dictionary attacks, in Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, May 1992, pp. 72–84
68. K.H.Rozen, Elementary Number Theory and its Applications, Addison-edn.,1993,p.714
69. Chen L. Recommendation for key derivation using pseudorandom functions // https://www.nist.gov/publications/recommendation-key-derivation. 10.10.2024.
70. Gennaro R., Halevi S. More on Key Wrapping // In book: Selected Areas in Cryptography. – Berlin, 2009. – P. 53-70.
71. Stallings W. Diffie-Hellman Key Exchange // In book: Cryptography and Network Security Principles and Practice. – London, 2013. – P. 287-291.
72. Hounkpevi A.C. et al. Eaglesign: A new post-quantum elgamal-like signature over lattices // Submission to the NIST's post-quantum cryptography standardization process. – 2023. – Vol. 2023. – P. 1-40.
73. Amirkhanova D.S., Mamyrbayev O. El-Gamal's cryptographic algorithm: Mathematical foundations, applications and analysis.// NAS RK: Vol: Physico-Mathematical Series ISSN 1991-346X Volume 3. № 351 (2024)
74. Tsiounis Y., Yung M. On the security of ElGamal based encryption // Public Key Cryptography: procced. 1st internat. Workshop on Practice and Theory in Public Key Cryptography (PKC'98). – Berlin, 1998. – P. 117-134.

75. Nguyen P., Stern J. Cryptanalysis of the Ajtai-Dwork Cryptosystem // Procced. 18th annual internat. Cryptology conf. “Advances in cryptology (CRYPTO’98)”. – Santa Barbara, 1998. – P. 223-242
76. Amirkhanova D.S., Mamyrbayev O. El-Gamal’s cryptographic algorithm: Mathematical foundations, applications and analysis.// NAS RK: Vol: Physico-Mathematical Series ISSN 1991-346X Volume 3. № 351 (2024)
77. Diffie W., Hellman M. New Directions in Cryptography // IEEE Transactions. – 1976. – Vol. 22, Issue 6. – P. 644-654.
78. Günther C.G. An Identity-Based Key-Exchange Protocol // Procced. Advances in Cryptology - EUROCRYPT 89: Workshop on the Theory and Application of Cryptographic Techniques. – Berlin, 1990. – P. 29-37.
79. Bacon D., Childs A.M., van Dam W. From optimal measurement to efficient quantum algorithms for the hidden subgroup problem over semidirect product groups // Procced. 46th Annual IEEE symposy
80. Goldreich O., Goldwasser S., Halevi S. Collision-free hashing from lattice problems // Stud Complex Cryptogr. – 2011. – Vol. 6650. – P. 30-39.
81. Gennaro R., Halevi S. More on Key Wrapping // In book: Selected Areas in Cryptography. – Berlin, 2009. – P. 53-70.
82. Chen L. Recommendation for key derivation using pseudorandom functions // https://www.nist.gov/publications/recommendation-key-derivation. 10.10.2024.
83. Lenstra A.K., Lenstra H.W. et al. The Development of the Number Field Sieve. – Berlin, 1993. – 140 p.
84. ElGamal T. (1984 a , 469). A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31(4): 469-472.
85. Mike Rosulek (2008-12-13). "Elgamal encryption scheme". University of Illinois at Urbana-Champaign. Archived from the original on 2016-07-22.

ПРИЛОЖЕНИЕ А

Фрагмент кода программы для тестирования системы

```
import numpy as np
import ipywidgets as widgets
from IPython.display import display, clear_output
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.animation import FuncAnimation

# Helper function to generate random elements from a finite field Z_q
def generate_random_matrix(n, m, q):
    return np.random.randint(0, q, (n, m))

# Key generation algorithm
def key_generation(security_param):
    q, n, m, beta = security_param
    M = generate_random_matrix(n, m, q)
    sA = generate_random_matrix(m, 1, q)

    while np.linalg.norm(sA) > beta:
        sA = generate_random_matrix(m, 1, q)

    PA = np.dot(M, sA) % q
    public_key = (q, n, m, M, PA)
    private_key = sA
    return public_key, private_key

# Encryption algorithm (works with matrices)
def encrypt(public_key, message):
    q, n, m, M, PA = public_key
    sB = generate_random_matrix(n, 1, q)
    while np.linalg.norm(sB) > np.linalg.norm(sB):
        sB = generate_random_matrix(n, 1, q)

    c1 = np.dot(sB.T, M) % q
    c2 = (np.dot(PA.T, sB) + message) % q
    return c1, c2

# Decryption algorithm (works with matrices)
def decrypt(public_key, private_key, ciphertext):
    q, n, m, M, PA = public_key
    sA = private_key
```

```
        c1, c2 = ciphertext
        c1_sA = np.dot(c1, sA) % q
        decrypted_message = (c2 - c1_sA) % q
        return decrypted_message

    # Visualization function to display matrices as heatmaps
    def plot_heatmap(matrix, title="Matrix", ax=None):
        if ax is None:
            fig, ax = plt.subplots(figsize=(6, 6))  # Adjust the size to fit matrix
        sns.heatmap(matrix, annot=True, fmt="d", cmap="YlGnBu", cbar=True, ax=ax,
                    square=True, xticklabels=False, yticklabels=False)
        ax.set_title(title)
        ax.set_xlabel('Столбцы')
        ax.set_ylabel('Ряды')
        plt.tight_layout()
        plt.show()

    # Function to plot analysis graphs comparing original, ciphertext, and decrypted message
    def plot_analysis_graphs(original_message, decrypted_message, ciphertext):
        # Flatten the matrices to 1D arrays for easier plotting
        original_flat = original_message.flatten()
        decrypted_flat = decrypted_message.flatten()
        ciphertext_flat = np.concatenate(ciphertext).flatten()

        # исходное и расшифрованное сообщение
        fig, ax = plt.subplots(figsize=(10, 6))
        ax.plot(original_flat, label='Исходное сообщение', marker='o', linestyle='-', color='blue')
        ax.plot(decrypted_flat, label='Расшифрованное сообщение', marker='x', linestyle='--', color='green')
        ax.set_title('Исходное vs Расшифрованное сообщение')
        ax.set_xlabel('Индекс сообщения')
        ax.set_ylabel('Значение сообщения')
        ax.legend()
        ax.annotate("Значение сообщения соответствует каждому числовому элементу в матрице сообщений.",
                    xy=(0.5, 0.5), xycoords='axes fraction', ha='center', fontsize=12, color='black',
                    bbox=dict(facecolor='white', alpha=0.7, boxstyle="round,pad=0.3"))
        plt.grid(True)
        plt.show()
```

```
        # Сравнение гистограммы зашифрованного текста и исходного сообщения
        fig, ax = plt.subplots(figsize=(10, 6))
        ax.bar(np.arange(len(original_flat)), original_flat, width=0.4, label='Исходное сообщение', color='blue', align='center')
        ax.bar(np.arange(len(ciphertext_flat)), ciphertext_flat, width=0.4, label='Зашифрованный текст', color='red', align='edge')
        ax.set_title('Исходное сообщение vs Зашифрованный текст')
        ax.set_xlabel('Индекс сообщения')
        ax.set_ylabel('Значение сообщения')
        ax.legend()
        ax.annotate("Каждая полоса представляет значение соответствующего индекса в сообщении.",
                    xy=(0.5, 0.5), xycoords='axes fraction', ha='center', fontsize=12, color='black',
                    bbox=dict(facecolor='white', alpha=0.7, boxstyle="round,pad=0.3"))
        plt.show()

        # Bar plot comparison of Original Message, Ciphertext, and Decrypted Message
        fig, ax = plt.subplots(figsize=(10, 6))
        ax.bar(np.arange(len(original_flat)), original_flat, width=0.4, label='Исходное сообщение', color='blue', align='center')
        ax.bar(np.arange(len(decrypted_flat)), decrypted_flat, width=0.4, label='Расшифрованное сообщение', color='green', align='edge')
        ax.set_title('Исходное vs Расшифрованное сообщение')
        ax.set_xlabel('Индекс сообщения')
        ax.set_ylabel('Значение сообщения')
        ax.legend()
        ax.annotate("Каждое значение сообщения сравнивается по соответствующим индексам.",
                    xy=(0.5, 0.5), xycoords='axes fraction', ha='center', fontsize=12, color='black',
                    bbox=dict(facecolor='white', alpha=0.7, boxstyle="round,pad=0.3"))
        plt.show()

    # Function to animate the encryption and decryption process
    def animate_encryption_decryption(original, encrypted, decrypted):
        fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(12, 4))

        # Prepare the heatmaps for the original, encrypted, and decrypted matrices
        sns.heatmap(original, annot=True, fmt="d", cmap="YlGnBu", cbar=True, ax=ax1,
                    square=True, xticklabels=False, yticklabels=False)
```

```
        ax1.set_title("Исходное сообщение")

        # Animation function
        def update(frame):
            ax2.clear()
            ax3.clear()

            # Show the encryption steps (show how original matrix turns into encrypted)
            if frame < len(encrypted[0]):
                ax2.set_title("Ciphertext (c1)")
                sns.heatmap(encrypted[0][:frame+1], annot=True, fmt="d", cmap="YlGnBu", cbar=True, ax=ax2,
                            square=True, xticklabels=False, yticklabels=False)
                ax3.set_title("Ciphertext (c2)")
                sns.heatmap(encrypted[1][:frame+1], annot=True, fmt="d", cmap="YlGnBu", cbar=True, ax=ax3,
                            square=True, xticklabels=False, yticklabels=False)
            else:
                # Once encryption steps are complete, show decrypted message and stop the animation
                ax2.set_title("Ciphertext (c1)")
                sns.heatmap(encrypted[0], annot=True, fmt="d", cmap="YlGnBu", cbar=True, ax=ax2,
                            square=True, xticklabels=False, yticklabels=False)
                ax3.set_title("Ciphertext (c2)")
                sns.heatmap(encrypted[1], annot=True, fmt="d", cmap="YlGnBu", cbar=True, ax=ax3,
                            square=True, xticklabels=False, yticklabels=False)

                ax3.set_title("Decrypted Message")
                sns.heatmap(decrypted, annot=True, fmt="d", cmap="YlGnBu", cbar=True, ax=ax3,
                            square=True, xticklabels=False, yticklabels=False)

                # Stop the animation after showing the decryption
                ani.event_source.stop()

        ani = FuncAnimation(fig, update, frames=range(0, len(encrypted[0])+1), interval=1000)
        plt.show()

    # Function for interactive encryption-decryption
    def run_crypto_system(q, n, m, beta, message=None):
```

```
        security_param = (q, n, m, beta)
        public_key, private_key = key_generation(security_param)

        # If no message is provided, generate a random matrix message
        if message is None:
            message = np.random.randint(0, q, (n, m))  # Random message

        print("\nИсходное сообщение:")
        plot_heatmap(message, "Исходное сообщение")

        # Encrypt the message
        ciphertext = encrypt(public_key, message)
        print("\nЗашифрованный текст (c1, c2):")
        plot_heatmap(ciphertext[0], "Зашифрованный текст (c1)")
        plot_heatmap(ciphertext[1], "Зашифрованный текст(c2)")

        # Decrypt the message
        decrypted_message = decrypt(public_key, private_key, ciphertext)
        print("\nРасшифрованное сообщение:")
        plot_heatmap(decrypted_message, "Расшифрованное сообщение")

        # Visualize the encryption and decryption process
        animate_encryption_decryption(message, ciphertext, decrypted_message)

        # Plot message analysis graphs (including message values comparison)
        plot_analysis_graphs(message, decrypted_message, ciphertext)

        return ciphertext, decrypted_message

    # Create interactive widgets for the user
    q_slider = widgets.IntSlider(value=17, min=2, max=100, step=1, description="Модуль q:")
    n_slider = widgets.IntSlider(value=5, min=2, max=10, step=1, description="Ряды n:")
    m_slider = widgets.IntSlider(value=5, min=2, max=10, step=1, description="Столбцы m:")
    beta_slider = widgets.FloatSlider(value=5.0, min=1.0, max=20.0, step=0.5, description="Порог β:")

    # Button to run the encryption-decryption process
    run_button = widgets.Button(description="Запуск")

    # Function to execute when the button is clicked
    def on_button_click(b):
```

```
    q = q_slider.value
    n = n_slider.value
    m = m_slider.value
    beta = beta_slider.value

    print("\nЗапуск постквантовой криптосистемы Эль-Гамаля...")
    run_crypto_system(q, n, m, beta)

# Link the button with the function
run_button.on_click(on_button_click)

# Display the widgets
display(q_slider, n_slider, m_slider, beta_slider, run_button)
```

ПРИЛОЖЕНИЕ Б

Свидетельства государственной регистрации прав на объект авторского права

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

СВИДЕТЕЛЬСТВО

О ВНЕСЕНИИ СВЕДЕНИЙ В ГОСУДАРСТВЕННЫЙ РЕЕСТР ПРАВ НА ОБЪЕКТЫ, ОХРАНЯЕМЫЕ АВТОРСКИМ ПРАВОМ

№ 55885 от «18» марта 2025 года

Фамилия, имя, отчество, (если оно указано в документе, удостоверяющем личность) автора (ов):
ӘМІРХАНОВА ДАНА САЙРАНҒАЖЫҚЫЗЫ, Мамырбаев Оркен Жумажанович

Вид объекта авторского права: **программа для ЭВМ**

Название объекта: **Схема постквантового шифрования с открытом ключом на основе решетки с использованием принципов Эль-Гамаля**

Дата создания объекта: **10.02.2025**

Құжат түпнұсқалығын http://www.kazpatent.kz/ru сайтының "Авторлық құқық" бөлімінде тексеруге болады https://copyright.kazpatent.kz

Подлинность документа возможно проверить на сайте kazpatent.kz в разделе «Авторское право» https://copyright.kazpatent.kz

Подписано ЭЦП С. Ахметов